# Logica ad Informatica

Andrea Asperti e Agata Ciabattoni

2 dicembre 2008

# Indice

# Prefazione

## La Razionalità e le Macchine

di Giuseppe Longo

L'Informatica è nata dalla Logica, come Minerva dalla testa di Giove. La genesi è stata lunga ed è avvenuta parallelamente alla nascita della nozione moderna, formale e "linguistica", di rigore matematico.

Forse si può iniziare a raccontare la vicenda di questo percorso verso la Logica (Matematica) moderna, e le sue conseguenze, a partire dalla riflessione di Cartesio. Descartes non ama la logica aristotelica: la trova insufficiente al ragionamento e, soprattutto, alla deduzione matematica. Nelle "Regulae ad Regulationem ingenii", un vero capolavoro di "rigore cartesiano", conduce un'analisi molto profonda della deduzione, cuore della razionalità umana. Essa va ben al di là delle poverissime regole di instanziazione, al centro della tradizione aristotelica (gli uomini sono mortali, Socrate è un uomo, dunque Socrate è mortale): frantuma il ragionamento nei suoi elementi cruciali, lo dispiega nel suo procedere passo dopo passo da "verità evidenti", intuizioni dello spazio e del mondo. Malgrado la netta distinzione fra anima e i meccanismi del corpo fisico (il dualismo di Descartes), nelle regole per l'ingegno che egli enuncia si intravedono i passi certi, scanditi meccanicamente, dei meravigliosi orologi dei suoi tempi o delle macchine a venire.

Leibniz, con le sue intuizioni premonitrici (l'esperto pensi anche alle idee sul continuo, *non-standard* diremmo oggi), andrà oltre. Egli propone di trattare il ragionamento (matematico) con una "Lingua Caracteristica", un *linguaggio formale*, nella terminologia moderna. Con Leibniz cioè, insieme alla crescente esigenza di rigore matematico, inizia ad emergere l'idea che le deduzioni della Matematica, ma forse di più, il ragionamento umano, possano essere trattati come sistema di regole linguistiche esplicite, come "calcolo dei segni". Hobbes arriverà a dire

> con *ragionamento intendo calcolo* ... ogni ragionamento si basa su queste due operazioni dello spirito, la somma e la sottrazione [Computation sive Logica *in* De Corpore, 1655].

Non è stato tuttavia facile arrivare a concepire ed a realizzare formalmente tali progetti, solo in nuce in Leibniz o in Hobbes: essi presuppongono il trasferimento dell'intera prassi matematica in un linguaggio formale. La Matematica infatti

è ricca di riferimenti allo spazio, colti dall'intuizione di Descartes; la geometria domina l'interazione ricchissima fra Matematica e Fisica. Il movimento dei corpi ha luogo nello spazio di Newton: l'analisi infinitesimale, la nozione di continuità attingono la loro certezza nell'intuizione pura dello spazio euclideo, che è assoluto e perfettamente certo, come confermano i filosofi. In effetti, problematiche e risultati eminentemente geometrici dominano le scuole francesi e tedesche. Fra fine '700 ed inizi '800, Monge, Lagrange, Laplace, Cauchy, Poncelet, ma anche Gauss, pensano la Matematica, in particolare l'Analisi Matematica, nello spazio; interpretano persino il Teorema fondamentale dell'Algebra ed i numeri complessi, sul piano cartesiano (l'interpretazione di Argand-Gauss).

Sono gli algebristi inglesi dell'inizio dell'ottocento a precisare un'altra visione della Matematica, riprendendo, loro modo, le idee di Leibniz e di Hobbes. Woodhouse, in un celebre articolo del 1801, propone un netto distacco fra geometria ed algebra: quest'ultima si basa sulla manipolazione "meccanica" di simboli e deve trovare nella correttezza logica del ragionamento formale il suo fondamento, non già nel riferimento allo spazio. Dopo di lui, Peacok, Babbage, Boole, fra il 1820 e il 1850, confermeranno quest'ottica: i calcoli algebrico-deduttivi hanno una loro autonomia, indipendente dal significato dei simboli manipolati. Ed ecco, il "Calculus of Deductive Reasoning" e le celeberrime "Laws of Thought" di Boole, ed infine, e soprattutto, la macchina analitico-deduttiva di Babbage. Vero calcolatore, ruote dentate che codificano elementi del calcolo differenziale, "ragionamento algebrico" mosso da macchine a vapore.

Più o meno negli stessi anni, tuttavia, anche sul continente la fiducia nella geometria viene messa a dura prova. Dopo le intuizioni di Gauss dei primissimi del secolo, fra il '35 ed il '50, Lobacevskij, Bolyai e Riemann dimostreranno che si può descrivere il mondo o che sono perlomeno compatibili con esso, geometrie radicalmente diverse, in cui nessuna od infinite rette parallele passano per un punto del piano esterno ad una retta. Ma allora, dove va a finire la certezza dello sguardo sul foglio, dell'intuizione pura dello spazio assoluto? La matematica ha bisogno del linguaggio. Anzi solo nel linguaggio, spiegherà Frege, può trovare il suo fondamento. In alcuni libri di grande interesse, innazitutto nell'"Ideografia" (1879) e nei "Fondamenti dell'Aritmetica" (1884), Frege propone il paradigma moderno della Logica Matematica. Il calcolo è deduzione, la deduzione è calcolo: un conto aritmetico è una dimostrazione, una dimostrazione procede come un conto. I calcoli sono deduzioni logiche proprio come è logico-linguistico il fondamento dell'Aritmetica: l'Aritmetica è un sistema di segni che coincide con le sue stesse prove. In effetti, l'ideografia di Frege, o calcolo formale dei segni, sviluppa una analisi concettuale rigorosissima delle variabili e della quantificazione ("Per ogni ... Esiste ...") e ne fa un linguaggio formalmente perfetto, proprio come il lettore lo troverà descritto nel testo che segue. Il balzo in avanti rispetto alla logica Booleana, semplice calcolo proposizionale, è enorme: Frege formalizza l'uso matematico delle variabili e della loro quantificazione ed individua nel sistema logico, che ne regola l'impiego, l'origine ed il fondamento della stessa Aritmetica, ovvero della Teoria Formale dei Numeri. Ora, cosa vi è di più importante del numero e della sua teoria, in Matematica, nonché dell'uso delle variabili? L'analisi di Frege centra gli aspetti della deduzione matematica

che saranno al cuore dell'investigazione logica per tutto il XX secolo.

Sarà Hilbert tuttavia a render pienamente matematica la proposta di Frege. Con lui nasce la Metamatematica, ovvero l'analisi matematica della stessa deduzione matematica, nasce la Teoria della Dimostrazione. Il linguaggio formale, l'ideografia di Frege, è esteso ed arricchito fino a conglobare, grazie ad opportuni assiomi formali, puramente linguistici, la geometria. E con ciò, il linguaggio si distacca definitivamente da quest'ultima, perchè la certezza è raggiunta solo nella manipolazione di stringhe finite di simboli, indipendente dal significato assunto da tali simboli su eventuali strutture geometriche. Lo sguardo nello spazio, l'intuizione geometrica, che Poincarè cerca invano di difendere, e le ambiguità semantiche sono all'origine delle incertezze nella deduzione, di quei mostri orrendi, i paradossi, che tanto avevano inquietato i matematici a cavallo fra i due secoli. Strumento chiave dell' analisi di Hilbert è la distinzione fra il linguaggio in cui formalizzare la Matematica, oggetto di studio della Teoria della Dimostrazione, e la Metamatematica, descritta in un metalinguaggio e che include la Teoria della Dimostrazione stessa. Linguaggio oggetto e metalinguaggio dunque, da cui distinguere un terzo livello, quello del significato (la semantica come raccontata nel testo che segue): una semantica eventualmente geometrica. Ma ora, dopo Leibniz, Boole, Frege (e Peano, non bisogna dimenticare), il linguaggio oggetto è finalmente abbastanza ricco da poter rappresentare la matematica, con le variabili, la quantificazione ed ogni altro costrutto linguistico di base; ora, si può precisare che cosa si intende per completezza di questa rappresentazione, per decidibilità (potenziale) di ogni asserto della Aritmetica e dell'Analisi. Bisogna solo rendere rigorosa l'intuizione di Boole e di Frege e definire formalmente che cosa sia un calcolo logico-deduttivo, di come esso si correli ai calcoli aritmetici. Proprio a Parigi, nel 1900, regno di Poincarè, Hilbert fa la lista dei problemi aperti e delle congetture in Matematica ed in Logica, che forgeranno, fra l'altro, la Logica Matematica di questo secolo: fra queste, la completezza, la decidibilità, la dimostrabilità della coerenza (non contraddittorietà) dell'Aritmetica e dell'Analisi. Puri calcoli di segni finitari devono poter dare la certezza fondazionale, grazie alla coerenza, asserire la completezza dei sistemi formali e la decidibilità di ogni asserto ben formato della Matematica (o del suo nucleo logico: l'Aritmetica).

Gli anni trenta danno la risposta. In primo luogo il Teorema di completezza, nel senso che il lettore troverà più oltre. Quindi l'osservazione che ogni deduzione matematica, nei termini hilbertiani, è una specifica funzione aritmetica, dai numeri nei numeri. È una "funzione calcolabile", spiegheranno Gödel, Church, Kleene, Turing ed altri con i loro formalismi. Turing, ispirato forse dalla distinzione fra metateorie e teorie e fra teoria e semantica, distinguerà, in una macchina formale, pura astrazione matematica, fra ... hardware e software. E, sorpresa straordinaria, tutti otterranno, con tecniche e strumenti matematici diversi, esattamente la stessa classe di funzioni; ovvero, i diversi formalismi, di Herbrand, Gödel, Church, Kleene, le Macchine di Turing definiscono tutti la stessa classe di funzioni. Abbiamo un assoluto, dirà Gödel nel '36: la classe delle deduzioni effettive, tutte equivalentemente rappresentate nei numerosi e diversi calcoli logici proposti in quegli anni. Nasce la Teoria della Calcolabilità,

ben prima dei calcolatori moderni, la teoria dell'unica classe delle deduzioni, come calcoli, e delle funzioni effettivamente eseguibili in modo meccanico, in quanto elaborazioni su stringhe finite di segni, guidate da regole finitarie. La distinzione, poi, fra hardware e software, proposta da Turing per i soli scopi dell'analisi logica della deduzione, sarà all'origine del salto qualitativo rappresentato dai calcolatori moderni rispetto alle macchine del secolo precedente. Lo stesso Turing e VonNeuman disegneranno, infatti, le prime architetture di calcolo, nell'immediato dopoguerra: non più ingranaggi in cui è inscritta per sempre l'operazione da fare, come nei marchingegni meccanici di una volta, ma materiale inerme (hardware) attivato da linguaggi formali o di programmazione (software), più o meno umanizzati: regole linguistico-algebriche da eseguire, cartesianamente, passo dopo passo. Un salto qualitativo, logico, enorme, rispetto alle macchine di Babbage. Per di più, l'idea di Turing della "Macchina Universale" permetterà di concepire anche i compilatori, veri intermediari fra la macchina ed i programmi.

Questa rapidissima e parziale cavalcata attraverso i secoli ha saltato cento precursori e mille passaggi. Già Pascal, infatti, aveva sottolineato il rigore quasi meccanico necessario alla deduzione matematica e realizzato una macchina per fare le quattro operazioni, ma non aveva cercato un nesso fra aritmetica o calcolo e logica, tantomeno un nesso con la geometria. Descartes invece rende rigorosa la matematica dello spazio grazie alla algebrizzazione di quest'ultimo: la geometria può essere in principio ridotta a calcoli algebrici in sistemi di "coordinate cartesiane". La sua geometria analitica, infatti, trasformando in equazioni le figure nello spazio, dà alla loro elaborazione la certezza della manipolazione algebrica: passo dopo passo, con regole quasi meccaniche, si trasforma un'equazione in un'altra; le trasformazioni e le intersezioni di figure diventano calcoli. Così Dedekind e Peano formalizzano l'aritmetica, ma Frege ed Hilbert ne vedono il rilievo logico, la pongono al centro dell'investigazione metamatematica, rendendo matematico (e rigoroso) lo studio delle prove. E, come si diceva, questa tensione verso il rigore formale raggiunge il massimo della sua astrattezza e potenza quando distacca definitivamente i simboli dei linguaggi formali, luogo del rigore, dal loro significato, al punto ... che questi possano essere manipolati da macchine. Allora, appunto, nasce l'Informatica, nei fantastici anni '30 della Logica. Di nuovo, Power, alla fine dell'800, sapeva usare schede perforate di cartone e queste permettevano di arricchire le potenzialità incise negli ingranaggi della sua macchina, ma solo con la Logica e con Turing si arriva a concepire un linguaggio formale, poi di programmazione, perfettamente espressivo o completo o che permetta di calcolare tutte le funzioni calcolabili. In effetti, i moderni linguaggi di programmazione imperativi sono l'evoluzione diretta delle idee di Turing, i linguaggi funzionali di quelle di Church e risultati di Herbrand sono alla base dei linguaggi della programmazione logica (e dei "metodi di risoluzione", v. cap.6).

Ma che ne è stato del progetto originario di Leibniz e Frege, almeno così come presentato dalle grandi congetture di Hilbert del 1900 (la completezza, la decidibilità, la dimostrabile coerenza dell'Aritmetica)? Grazie agli strumenti inventati per analizzare la deduzione come calcolo, gli splendidi anni '30 della

Logica ci hanno dato due grandi "risultati": da un lato la nascita dell' Informatica (le macchine per dedurre, manipolando simboli linguistici), dall'altro hanno spiegato che il percorso secolare verso la totale formalizzazione linguistica della matematica si urta ad un limite invalicabile. Tale limite è anche dovuto, come cercava di dire Poincaré, al ruolo dello spazio e della Geometria nella prova: la riduzione linguistica, che ci ha permesso di concepire le moderne "macchine per calcoli formali" è essenzialmente incompleta. I risultati di incompletezza di Gödel degli anni '30 (l'Aritmetica formale è incompleta, indecidibile e nessun sistema logico finitario ne può provare la coerenza, come verrà spiegato nel testo che segue) assicurano infatti che il progetto globalizzante esplicitato da Hilbert nel '900 è fallito. Nella dimostrazione matematica, quale praticata dagli uomini, si fa un uso essenziale di argomenti infinitari, dello spazio in quanto tale: i formalismi finitari ed il linguaggi per la calcolabilità non consentono di dedurre tutti gli asserti che noi sappiamo dimostrare veri sulle specifiche strutture della matematica, in particolare la coerenza del suo cuore logico-formale, l'Aritmetica. Dunque, quell'assoluto, la classe delle funzioni calcolabili e delle deduzioni "effettive", rimane tale per le macchine, almeno come definite negli anni '30 e realizzate in seguito, ma la pratica della matematica usa anche altri strumenti, perfettamente chiari, basati sullo sguardo nello spazio e la geometria, su alberi planari infiniti, ad esempio, per dimostrare la coerenza dell'Aritmetica. Non vi è nessun miracolo ontologico nella dimostrazione matematica, essa è solo ricca di significato, geometrico in primo luogo, essenziale al ragionamento, ed è inoltre basata su una pratica antichissima dello spazio e dell'infinito, inadeguata ad elaboratori meccanici di stringhe di simboli. Il lettore, incuriosito (ed istruito) da questo libro, potrà poi andare a leggersi qualcuna delle prove della coerenza dell'Aritmetica, assolutamente rigorose e convincenti, ma non meccanizzabili (gli alberi infiniti nel Teorema di Kruskal o nella "determinacy for $\delta_0$ trees"). Potrà continuare con lo studio di un altro luogo della "rivincita della geometria" sulla riduzione linguistica: la Logica Lineare di J.Y. Girard. In tale sistema, la geometria entra nella prova stessa, tramite l'uso di "reti di prova": la rappresentazione geometrica di deduzioni, con connessioni, percorsi che si incrociano e che, incrociandosi, danno informazioni essenziali, viene con esse a far parte della Teoria della Dimostrazione.

Il testo chiaro e completo che segue, porterà il lettore attraverso i metodi ed i sistemi deduttivi fondamentali della logica post-fregeana. La decidibilità, la completezza, il gioco fra sintassi e semantica gli diverranno chiari e riuscirà a percepire come essi siano stati all'origine dell'elaborazione meccanica e siano ancora oggi cruciali per capire (e sviluppare!) linguaggi e sistemi di programmazione. Potrà apprezzare il fascino dell' "argomento diagonale" nel Teorema di incompletezza di Gödel. Verrà portato alle soglie delle tecniche più attuali studiando la "risoluzione", ad esempio, strumento importante in dimostrazione automatica ed in molte altre applicazioni. Il lettore curioso di Logica troverà quindi un panorama più che esaustivo degli elementi della Logica Matematica moderna, in cui le motivazioni informatiche, concentrate soprattutto in esempi e note, non faranno che arricchire la lettura. Lo studente di Informatica, il ricercatore, potrà acquisire le basi logico-matematiche necessarie a capire e svi-

luppare, poi, i linguaggi di programmazione funzionali, logici ecc. ... ma anche ad affrontare i nuovi problemi che l'Informatica pone alla Logica. Difatti, in questi ultimi anni, la Fisica ha fatto prepotentemente il suo ingresso in scena: il costo minimo dei chips, la velocità d'elaborazione e le enormi reti di comunicazione attuali ci hanno dato strumenti nuovi di calcolo. I sistemi distribuiti, asincroni, paralleli e concorrenti richiedono invenzione matematica per essere costruiti, descritti con rigore e gestiti con efficacia. Lo studio dei fondamenti logici del calcolo e della deduzione formali costituisce il passo preliminare per chi voglia andare oltre le macchine che Descartes e Leibniz, Boole e Babbage, Gödel e Turing hanno saputo darci.

Giuseppe Longo
CNRS e Ecole Normale Supérieure,
Paris, 17 Giugno 1997.

# Introduzione

## Logica a Informatica

Il presente volume è stato espressamente concepito come testo di supporto didattico per il corso fondamentale di "Logica Matematica" del secondo anno del Corso di Studi in Informatica. Il titolo stesso del libro, nella sua accezione più immediata, intende rispecchiare le finalità, modeste ma specifiche, del presente lavoro.

Esiste tuttavia una seconda possibile interpretazione del titolo, più sottile e assai più ambiziosa, laddove non si intenda la preposizione "a" come una semplice specificazione di luogo, ma come un complemento di termine o di destinazione: l'espressione di un punto di arrivo naturale e necessario.

La vicinanza culturale e metodologica tra Logica ed Informatica si è evidenziata notevolmente negli ultimi decenni. L'inserimento stesso di un corso fondamentale di Logica Matematica nel nuovo ordinamento del corso di Studi in Informatica non fa che testimoniare la rilevanza recentemente assunta da questa disciplina per lo studio, la progettazione e la verifica di sistemi informatici. Le applicazioni attuali della Logica sono innumerevoli e completamente trasversali a tutte le aree dell'Informatica. Esse vanno dalla progettazione di circuiti digitali, alla modellizzazione algebrica di macchine astratte, dalla progettazione di linguaggi logici e linguaggi di interrogazione, alla verifica della correttezza dei programmi, dalla implementazione di protocolli di comunicazione, ai problemi di tipizzazione, polimorfismo ed ereditarietà nei linguaggi di programmazione. È inoltre prevedibile che l'influenza e le applicazioni della Logica Matematica nei confronti dell'Informatica vadano ancora accentuandosi nei prossimi anni.

Il numero e la rilevanza di tali applicazioni non è certo casuale, ma è una conseguenza diretta delle profonde affinità metodologiche e culturali tra le due discipline. Come è stato ampiamente spiegato nella prefazione, un intero settore dell'Informatica nasce in modo diretto dalla Logica. D'altra parte, la Logica deve all'Informatica il suo improvviso risveglio a partire dai primi anni settanta.

Per usare una metafora, la Logica potrebbe e dovrebbe assumere, nei confronti dell'Informatica, lo stesso ruolo svolto dalla Matematica, e in particolare dall'Analisi, nei confronti della Fisica. Se dunque l'Informatica trova o dovrebbe trovare nella Logica i suoi principali strumenti di indagine e metodologie

di sviluppo, la Logica trova nell'Informatica le sue stesse motivazioni e nuovi, continui stimoli alla ricerca.

La profonda relazione che lega la Logica all'Informatica è tutto sommato assai meno sorprendente di quanto possa apparire a prima vista, considerando che entrambe le discipline si occupano, sebbene da punti di vista differenti e con tradizioni filosofiche distinte, di problemi di *formalizzazione*, *elaborazione* e *comunicazione* della conoscenza. L'enfasi sulla natura umana o meccanica dell'agente di calcolo è in realtà del tutto marginale: l'ovvio punto di incontro è costituito dalla necessità, per entrambe le discipline, di utilizzare un *linguaggio formale*.

La Logica pone in essere dei sistemi di prova: calcolo dei predicati, aritmetica, risoluzione, calcoli logici di ordine superiore, ecc., cioè insiemi strutturati di regole elementari che permettono di costruire dei "procedimenti" di calcolo complessi (e dunque, almeno implicitamente, dei processi); dove, tuttavia, la effettiva interpretazione *processuale*, se esiste, deve essere esplicitata.

D'altro canto, l'Informatica si occupa principalmente di meccanismi di calcolo: iterazione, ricorsione, assegnamenti, eccezioni, metodi, polimorfismo, ecc., cioè, anche in questo caso, insiemi di regole elementari che permettono la costruzione di processi computazionali complessi. Solo che in questo caso è la *logica* del processo che deve essere esplicitata.

Si evince, dunque, facilmente la stretta relazione tra Logica ed Informatica e più in generale la necessità di sviluppare una vera e propria Scienza dell'Informazione. Tutti sappiamo cosa sia un programma, visto come mera composizione sintattica delle sue istruzioni elementari. Tuttavia, la *logica* del processo complessivo, il suo significato, e dunque il suo comportamento ci è ancora largamente sconosciuto.

## Logica Matematica

Nella sua accezione filosofica, estremamente ambiziosa e anche piuttosto obsoleta, la Logica è lo studio dei meccanismi tipici del *ragionamento*, cioè della capacità di trarre *conseguenze* da un certo insieme di *premesse*. Fin dall'antichità si è osservato che determinati schemi di ragionamento sono in una certa misura indipendenti dal *senso* che si attribuisce alle singole parti del discorso. Consideriamo il celebre sillogismo "Ogni uomo è mortale, Socrate è un uomo, dunque Socrate è mortale". La correttezza di questa inferenza logica non dipende né dal significato della proprietà di essere mortale, né dal particolare individuo in oggetto (in questo caso, Socrate). In generale, se una proprietà $P$ è soddisfatta da tutti gli elementi di un determinato insieme, allora, preso un qualunque elemento $a$ di quell'insieme, la proprietà $P$ è necessariamente soddisfatta da $a$.

La necessità di astrarre dal contenuto informativo (il senso) delle sentenze, considerando solo la loro forma o struttura, porta inevitabilmente ad utilizzare un linguaggio di natura algebrica e formale per descrivere sia le sentenze che sono oggetto della logica, che le sue regole di inferenza. In questo senso più

ristretto deve essere intesa la *Logica Matematica* o *Logica Simbolica*, oggetto del presente volume.

Molto più semplicemente, potremmo osservare che ogni scienza formale ha bisogno di utilizzare un linguaggio per poter rappresentare e dunque comunicare informazioni e soprattutto risultati (cioè, elaborazioni delle informazioni). Tale linguaggio deve avere una natura precisa e formale, in modo da poter essere interpretato senza ambiguità dalle persone con cui si vuole comunicare. La Logica intende studiare i principi fondazionali su cui basare tale rappresentazione. Si noti, per inciso, la profonda analogia con le problematiche tipiche dell'Informatica, dove l'elaborazione dell'informazione è codificata in *linguaggi di programmazione* atti ad essere compresi da un agente finale, la macchina, del tutto privo di capacità intellettive autonome (e a maggior ragione interpretative).

Date queste premesse, è facile capire che non esiste un "linguaggio logico" universale. Ambiti differenti richiedono spesso formalismi logici particolari per poter rappresentare in modo conveniente problematiche specifiche. Questa è la ragione per la miriade di "logiche" proposte in letteratura: classica, intuizionista, affine, lineare, modale, temporale, multi-valore, fuzzy, non-monotona, ecc., per citare solo alcune delle più celebri.

In questo libro tratteremo unicamente di logica classica, la madre di tutte le logiche, e faremo solo alcuni brevi accenni alle problematiche tipiche del costruttivi-smo e della logica intuizionista.

La logica classica stessa si divide, poi, tradizionalmente in vari livelli di espressività crescente: *logica proposizionale*, *logica dei predicati* (o logica del primo ordine), *logiche di ordine superiore*. Solo i primi due di questi livelli saranno affrontati nel testo.

La logica proposizionale è essenzialmente rivolta allo studio dei *connettivi* logici, cioè di quelle particelle (e, o, se ... allora ... ) che permettono la costruzione di proposizioni complesse a partire da proposizioni elementari o atomiche. Le proposizioni hanno dunque una loro *sintassi* (cioè la loro struttura logica in quanto combinazione, mediante connettivi, di proposizioni atomiche), e una semantica (classica e tarskiana) che è la loro denotazione: esse possono essere vere oppure false. I connettivi logici hanno una ovvia interpretazione *intesa*: ad esempio $A$ *e* $B$ è vera se e solo se $A$ è vera e $B$ è vera[1]. E, tuttavia, anche questa semantica "intesa" e assolutamente priva di ambizioni porta alle prime fondamentali nozioni della Logica (*validità, soddisfacibilità, conseguenza, equivalenza*) e ai primi interessanti corollari di queste (leggi di De Morgan, forme normali congiuntive e disgiuntive, completezza funzionale, dualità).

In particolare, una formula è *valida* (diremo anche che è una *tautologia*) se è vera indipendentemente dai valori di verità dei suoi componenti (ad esempio, $A$ o $\neg A$ è vera indipendentemente dal fatto di sapere se $A$ sia vera o falsa).

Tuttavia, se una formula è valida, ci si aspetta anche di poterla *dimostrare*, cioè di poterla inferire a partire da un semplice insieme di verità e regole logiche di "indubbia evidenza". Questo porta a considerare i sistemi deduttivi: veri e propri calcoli logici per formalizzare la nozione intuitiva di *ragionamen-*

[1]Semantica che verrà formalizzata utilizzando *tabelle di verità*.

*to*. In particolare, in questo libro, verranno introdotti tre di questi sistemi: la Deduzione Naturale, i Sistemi Assiomatici ed il Calcolo dei Sequenti. Un sistema deduttivo è *corretto* se ogni formula logica dimostrabile è una tautologia; è *completo* se permette di dimostrare *tutte* le tautologie.

L'enfasi dei primi tre capitoli del libro è dunque interamente sulla natura dei linguaggi formali nei loro tre aspetti fondamentali (sintassi, calcolo e semantica), nonchè sulle relazioni tra questi, fornita dai teoremi di correttezza e completezza.

Dopo aver discusso la distinzione e le relazioni tra sintassi, calcolo e semantica a livello proposizionale, lo studente non dovrebbe incontrare grosse difficoltà (quantomeno concettuali, se non tecniche) ad affrontare le stesse problematiche per la *logica dei predicati*.

La logica dei predicati (o logica del primo ordine) arricchisce il linguaggio proposizionale introducendo le nozioni di *termine*, *relazione* e *quantificazione*. Consideriamo la semplice sentenza

(1) 3 è un numero pari oppure 3 è un numero dispari

Ciò che si osserva a livello proposizionale sono due sentenze assolutamente distinte, unite da un connettivo di disgiunzione. Il fatto che le due sentenze trattino di uno *stesso* oggetto (in questo caso il numero intero 3) affermando che esso gode o della proprietà di essere pari o di essere dispari è del tutto trasparente all'analisi proposizionale.
È necessario dunque arricchire il linguaggio logico con una sintassi che permetta di esprimere *termini* (gli oggetti del dominio inteso del discorso), e relazioni (o, nel caso unario, predicati) su questi termini. Ad esempio, se indichiamo con $P(x)$ la proprietà "$x$ è pari" e con $D(x)$ la proprietà "$x$ è dispari" potremo formalizzare l'asserzione precedente nella formula

$$(2) \qquad P(3) \; o \; D(3)$$

Tuttavia, vogliamo anche essere in grado di poter esprimere forme di *quantificazione* esistenziale (*esiste* un termine che gode di una certa proprietà) e universale (*ogni* termine gode di una certa proprietà). Per esempio, vorremo poter dire che

(3) Ogni numero intero è pari oppure dispari

Per poter parlare di termini "generici" abbiamo bisogno della nozione algebrica di *variabile*. Queste variabili possono essere quantificate esistenzialmente (si userà il simbolo $\exists$) o universalmente ($\forall$). Nella notazione utilizzata per (2), potremo allora formalizzare la sentenza (3) nel modo seguente:

$$(4) \qquad \forall x(P(x) \; o \; D(x))$$

La stratificazione tra *termini* e *relazioni*, la nozione di variabile, i complessi problemi inerenti ai quantificatori, ai loro legami ed al campo d'azione di questi sono al centro di tutta la trattazione della logica del primo ordine. Si cercherà quindi di superare la comprensibile perplessità degli studenti di fronte alla piatta semantica tarskiana ($\forall x A(x)$ è vero se per ogni elemento $a$ del dominio inteso

del discorso $A(a)$ è vero), sia presentando alcuni dei principali e notevolissimi risultati della teoria dei modelli, sia spiegando l'uso della semantica in alcune delle sue applicazioni più rilevanti: verifiche di consistenza di teorie logiche o dimostrazioni di non-validità di formule logiche attraverso la definizione di controesempi.

Il passo successivo, che conduce alle logiche di ordine superiore, consiste, ovviamente, nel permettere di quantificare non solo su termini, ma anche su relazioni. Si pensi, ad esempio, al principio di induzione numerica, che afferma che *per ogni predicato* $P(x)$, se vale $P(0)$ e se, per ogni $n$, $P(n)$ implica $P(n+1)$, allora $P$ vale per tutti i numeri interi[2]. Tuttavia la trattazione assai delicata di queste logiche esula completamente dagli intenti del presente volume.

Nel capitolo conclusivo vengono affrontati in dettaglio i notevoli risultati semantici di Herbrand (ricordiamo il ruolo essenziale svolto dal "modello dei termini" in tutta la trattazione algebrica dei linguaggi di programmazione) ed il *metodo di risoluzione* di Robinson che è alla base della programmazione logica e di svariati algoritmi per la dimostrazione automatica o semi-automatica di teoremi.

## Note per il Docente

Il presente volume nasce direttamente da un insieme di note didattiche preparate per l'esame di "Logica Matematica" del Corso di Studi in Informatica. Data la natura semestrale del corso ed essendo previsto al secondo anno di studi dell'ordinamento di Informatica, si è volutamente cercato di mantenere un carattere agile e snello al testo, pur toccando tutti gli argomenti di maggior rilevanza per un'introduzione all'argomento. Nonostante la trattazione della Logica Matematica fatta nel presente volume sia mirata al discorso informatico, le sue innumerevoli applicazioni attuali sono solo accennate brevemente ed in modo del tutto discorsivo. Il compito prioritario del Corso di Logica Matematica non è quello di studiare questa o quella particolare applicazione, ma piuttosto quello di fornire le basi minimali, sia teoriche che tecniche, indispensabili alla comprensione della materia. In questo senso il testo risponde in modo positivo alle esigenze professionali della moderna figura del laureato in Informatica, contrastando e non subendo una certa primitiva avversione degli studenti rispetto alla matematica "tradizionale". L'affermazione della vocazione professionale dell'informatico, perseguita dal nuovo Ordinamento degli Studi, trova in questo caso un immediato riscontro nella trattazione tecnica, formale e dunque operativa degli argomenti.

Nonostante gli sforzi di coincisione e la sofferta omissione di interi argomenti (pensiamo in particolare a tutta la trattazione delle strutture di semantica algebrica, o alla tecniche di prova basate su tableaux), il presente volume contiene ancora una quantità di nozioni eccessiva rispetto a quelle che possono essere effettivamente trattate nell'ambito di un corso semestrale. Questo è dovuto sia

[2]Come vedremo, questo caso particolare di quantificazione può essere trattato in modo *implicito* attraverso la nozione di *schema*.

a ragioni di completezza espositiva, sia per consentire al Docente di personalizzare il corso seguendo un percorso didattico individuale. Esistono, a nostro modo di vedere, almeno due possibilità, di pari interesse e coerenza concettuale, che essenzialmente differiscono tra loro sulla scelta del sistema deduttivo che si intende privilegiare: Deduzione Naturale o Calcolo dei Sequenti[3]. Tale scelta si ripercuote, ovviamente, sui relativi teoremi di completezza. In particolare, se a livello proposizionale si possono ancora discutere a lezione più dimostrazioni differenti, per ragioni di tempo questo non è più possibile al primo ordine. Se la preferenza del Docente ricade sulla Deduzione Naturale, è possibile, dopo avere dimostrato la completezza con il metodo di Henkin, entrare in qualche dettaglio dei principali risultati di Teoria dei Modelli. Viceversa, se la preferenza ricade sul Calcolo dei Sequenti, vale la pena di enfatizzare la parte di ricerca automatica (a scapito della Teoria dei Modelli, i cui risultati potranno solo essere menzionati).

Il libro non richiede particolari prerequisiti. Ricordiamo tuttavia che il Corso è previsto per il Secondo Anno dell'ordinamento del Piano di Studi di Informatica. Dunque, si suppone che lo Studente abbia già seguito corsi quali Analisi Matematica I e Matematica Discreta, acquisendo una buona padronanza del linguaggio matematico e dei suoi metodi formali.

Per facilitare la lettura del testo e la scelta degli argomenti, le sezioni "avanzate" sono state contrassegnate con un asterisco. Queste possono essere omesse ad una prima lettura senza compromettere la coerenza del testo.

Lo stile del libro è volutamente molto scarno, soprattutto nelle sue parti espositive. Come ogni testo universitario, il libro deve essere inteso come strumento di *riferimento e consultazione*: non può (e non deve) sostituirsi al corso, dove il Docente avrà cura di introdurre e discutere gli argomenti secondo la propria visione personale.

Viceversa, si è dedicata una certa cura alla parte di trattazione storica, concludendo la maggior parte dei capitoli con dei cenni sullo sviluppo del pensiero logico moderno. Analogamente, si è tentato di arricchire il più possibile il testo con puntatori e riferimenti ad argomenti che, per la loro rilevanza sia logica che informatica, potrebbero suscitare la curiosità dello studente, benchè esulino inevitabilmente dal programma del corso. Per ovvie ragioni, la terminologia utilizzata in questi casi è del tutto informale. Speriamo dunque che il lettore non si formalizzi se, ad esempio, accenneremo in una nota ai termini del lambda-calcolo senza avere ancora discusso la nozione generale di "termine". Il peccato ci appare del tutto veniale se confrontato, ad esempio, al fatto (ahimè, assai più rilevante e sorprendentemente passato sotto silenzio in ogni libro di Logica, quasi per una tacita convenzione) che in tutta la trattazione della logica proposizionale si usa implicitamente un (meta)linguaggio del primo ordine!

Come ogni disciplina di natura fondazionale, la Logica è, soprattutto nei suoi aspetti semantici, inevitabilmente mal fondata. Riteniamo dunque superfluo (e anche filosoficamente datato) cominciare il corso con delle lunghe disquisizioni

[3] I sistemi assiomatici sono ugualmente trattati nel volume. Benchè per ragioni sia storiche che concettuali sia opportuno presentarli allo studente, se ne sconsiglia l'adozione come formalismo di riferimento, data la loro complessità pratica.

filosofiche sul significato delle nozioni di “nome”, “termine”, “variabile”, “funzione” ecc., lasciando che esse si spieghino da sole nell’unico modo possibile, ovvero, dall’uso linguistico che se ne fa in un determinato contesto.

## Ringraziamenti

Ringraziamo innanzi tutto il Prof. Giuseppe Longo sia per aver cortesemente accettato di scrivere la prefazione al libro, sia per i suoi innumerevoli consigli sulla stesura del testo. Andrea Masini e Marco Benini sono stati attenti e puntigliosi lettori delle versioni preliminari. Le loro correzioni ed i loro suggerimenti sono stati per noi di grandissimo aiuto. Molti altri amici e colleghi ci hanno aiutato a migliorare ulteriormente il testo con le loro critiche e le loro osservazioni. Ringraziamo in modo particolare Simone Martini, Daniele Mundici, Alessandro Berarducci, Giuseppe Rosolini, Maria I. Sessa, Stefano Aguzzoli e Mauro Ferrari.

# Capitolo 1

# Logica proposizionale

## 1.1 Senso e denotazione

L'unità di espressione del linguaggio naturale[1] è la *sentenza*, cioè un aggregato di parole di senso compiuto che racchiude un pensiero completo. Quando il significato della sentenza è quello di una *asserzione*, la sentenza è detta *dichiarativa*. Nel seguito, useremo il termine sentenza intendendo una sentenza dichiarativa. Esempi di sentenze dichiarative sono le frasi

quattro è un numero pari
oggi piove
la radice quadrata di due è un numero razionale

Dal punto di vista logico, seguendo una linea di pensiero che risale a Frege, le sentenze sono considerate come dei *nomi* in senso astratto. Un nome è una qualunque espressione linguistica che determina o *denota* in modo univoco una qualche entità. Ad esempio, le espressioni

quattro
il quadrato di due
il predecessore di cinque
tre più uno

sono modi differenti per indicare lo stesso oggetto che, in questo caso, è il numero quattro. Dal punto di vista logico, diremo che sono tutti *nomi* che *denotano* il numero quattro.
Ovviamente, benché le frasi precedenti denotino tutte lo stesso oggetto, il loro *senso* è differente; parleremo in questo caso di *valore connotazionale* del nome, associato a quanto questo effettivamente esprime (essere il quadrato di due, il predecessore di cinque, la somma di tre e di uno, ecc.).
La differenza tra senso e denotazione, può essere meglio spiegata dal seguente

[1] Per linguaggio naturale si intende il linguaggio che si usa quotidianamente per esprimere e comunicare.

esempio tratto dall'informatica. Consideriamo due programmi di ordinamento come MergeSort e QuickSort. Entrambi calcolano la stessa funzione che, preso in input un array di elementi disordinati, restituisce un array degli stessi elementi ordinati secondo una qualche relazione fissata. Da questo punto di vista, la *denotazione* dei due programmi è identica. Tuttavia, il loro senso, che in tal caso coincide con il particolare algoritmo di ordinamento implementato dai due programmi, è differente.
Nel linguaggio naturale, il *senso* è spesso più importante o altrettanto importante della denotazione. Tuttavia, esso è troppo complesso e dipendente dal contesto per essere oggetto di una rigorosa analisi matematica. In particolare, le denotazioni godono di una essenziale proprietà di *invarianza per sostituzione* che è il vero fondamento della loro trattazione formale, e che, al contrario, non è soddisfatta dalle connotazioni. Tale proprietà asserisce che se in un nome complesso (cioè costituito da altri nomi) si sostituisce un nome componente con un nome denotazionalmente equivalente ad esso, la denotazione del nome complesso non cambia. Consideriamo ad esempio la frase "il quadrato di due". Il nome "due" è denotazionalmente equivalente a "uno più uno". Sostituendo questo nome al posto di "due" nella frase originaria si ottiene la frase "il quadrato di (uno più uno)" che ovviamente denota sempre il numero quattro. Analogamente, si potrebbe sostituire l'espressione "il quadrato di 2" con quella denotazionalmente equivalente "il prodotto di 2 per se stesso", e ottenere la frase complessa "il prodotto di (uno più uno) per se stesso", che continua pur sempre a denotare il numero quattro (si osservi, al contrario, che le connotazioni di queste espressioni sono sensibilmente differenti).

Dunque la logica simbolica assimila le asserzioni, o sentenze dichiarative, a particolari nomi. Come per tutti i nomi, dal punto di vista logico non si è interessati a ciò che la sentenza asserisce, vale a dire al suo senso o valore connotazionale, ma unicamente a ciò che essa *denota* ed al modo in cui la denotazione di una sentenza composta, ottenuta da altre più semplici mediante l'uso di connettivi, dipende dalle denotazioni delle componenti. In altre parole, la Logica intende trattare le sentenze in base alla loro struttura e non al loro contenuto.
Resta ancora da capire che cosa effettivamente denoti una sentenza. A questo proposito, ricordiamo che la denotazione deve godere della proprietà di invarianza per sostituzione. Consideriamo allora l'asserzione:

quattro è il predecessore di cinque

Poichè "il predecessore di cinque" è denotazionalmente equivalente a "quattro", si ottiene per sostituzione la frase

quattro è quattro

Osserviamo come il *senso* della frase sia stato stravolto! Tuttavia, per poter assimilare le sentenze dichiarative a dei nomi, si vuole che la denotazione delle due frasi rimanga immutata. Ora, che cosa hanno in comune le due frasi suddette? Unicamente il fatto che sono entrambe vere!

Per questa ragione, dal punto di vista classico, la denotazione associata ad una sentenza non può che essere un valore di verità ("vero" oppure "falso").
Ricapitolando, ogni sentenza denota un valore di verità ed esprime un senso (che tuttavia non è oggetto diretto dell'indagine logica). Il senso determina la denotazione, o diremo anche che è un *concetto*[2] della denotazione.
Ogni concetto di un valore di verità è detto *proposizione*.

L'oggetto di indagine della logica proposizionale è rappresentato, appunto, dalle proposizioni. Queste possono essere di due tipi: *atomiche* e *composte*. Le prime, come si può evincere dal nome, sono le proposizioni più semplici che non possono essere scomposte in altre, mentre le seconde sono costruite, a partire da quelle atomiche, mediante l'utilizzo di *connettivi*.
Esempi di connettivi, nel linguaggio naturale, sono i seguenti : "e", "non", "se.. allora", "ma", "poiché", "o", "come",.. ed altri ancora. Per le ragioni già enunciate in precedenza, in logica simbolica vengono considerati solo i connettivi *estensionali*, cosiddetti in quanto la denotazione (il valore di verità) di una proposizione composta costruita mediante questi dipende unicamente dalle denotazioni (dai valori di verità) delle proposizioni che la compongono.
Spesso, tale condizione non è soddisfatta dai connettivi del linguaggio naturale; il motivo di questo fatto è che esso, al contrario dei linguaggi formali, consente un uso *obliquo* dei nomi. Un nome ha un uso obliquo quando la sua denotazione è il senso che il nome esprimerebbe nel suo uso corrente.
Un esempio di connettivo che induce ad un uso obliquo dei nomi è il connettivo "poiché". Consideriamo ad esempio le seguenti sentenze:

2 è un numero primo poiché 2 è divisibile solo per 1 e per se stesso
2 è un numero primo poiché 2 è pari

Pur essendo entrambe costruite a partire da proposizioni vere per mezzo del connettivo "poiché", la prima risulta vera, mentre la seconda risulta falsa. La ragione di questo fatto è che il nome (la sentenza) "2 è divisibile solo per 1 e per se stesso" è usato obliquamente nella prima frase: la sua denotazione non è semplicemente un valore di verità, ma il senso che essa effettivamente esprime.
Si noti che l'effetto del connettivo "poiché" nelle frasi precedenti è quello di spostare la frase da un livello linguistico (2 è un numero primo) ad un livello meta-linguistico[3], in cui si spiegano le ragioni per cui la prima frase deve essere considerata vera (che è appunto un ragionamento sul linguaggio stesso). L'uso obliquo dei nomi nel linguaggio naturale è generato dalla confusione tra i due livelli: in altre parole, il linguaggio è sovente utilizzato come meta-linguaggio di se stesso (cosa che deve essere evitata nei linguaggi formali).

[2] A differenza della *sentenza*, un *concetto* non ha necessariamente una natura linguistica. Si può anche ipotizzare l'esistenza di concetti non esprimibili in nessun linguaggio di uso corrente, ovvero di concetti che non coincidono con il senso attribuito a nessuno dei nomi (sentenze) di tali linguaggi.

[3] Un linguaggio utilizzato per descrivere o esprimere proprietà di un altro linguaggio è detto meta-linguaggio (di quest ultimo).

D'ora in poi, lavoreremo dunque con proposizioni e connettivi estensionali su queste.

## 1.2 Connettivi

La logica proposizionale, per motivi di economicità e chiarezza (che verranno precisati nella sezione 1.4.5), considera come primitivi un sottoinsieme proprio dei connettivi estensionali, ricavando gli altri in funzione di quelli scelti; in questa trattazione il nostro punto di partenza sarà l'insieme composto da: "non", "e", "o" "se.. allora". Come vedremo, la scelta di questi connettivi è ampiamente (anche se non completamente) arbitraria.
Mostriamo ora dei semplici esempi per ognuno di essi.

- L'operazione espressa dal connettivo "non" è chiamata *negazione* e la proposizione ottenuta dalla sua applicazione è detta *proposizione negata* (della proposizione di partenza). Ad esempio, data la proposizione atomica

  2 è un numero primo

  la sua negata è

  2 *non* è un numero primo

- L'operazione espressa dal connettivo "e" viene detta *congiunzione*. Ad esempio

  3 è un numero dispari *ed* è primo

  è una proposizione composta costruita congiungendo le due proposizioni atomiche:

  3 è un numero dispari
  3 è un numero primo

- L'operazione espressa dal connettivo "o" è detta *disgiunzione*. Un esempio di proposizione costruita con esso è la seguente:

  2 è un numero dispari *o* è primo

- Infine, l'operazione espressa dal connettivo "se.. allora" viene detta *implicazione* e la proposizione composta ottenuta dalla sua applicazione è chiamata proposizione *condizionale*. Un esempio di proposizione condizionale è la seguente:

  *se* 3 è dispari *allora* è primo

la proposizione "3 è dispari" è detta *premessa* (dell'implicazione) mentre "3 è primo" è la sua *conseguenza* o *conclusione*.

Dopo aver introdotto, in maniera intuitiva, le proposizioni ed i connettivi, procediamo ad una *definizione formale* della logica proposizionale. A tal fine dovremo:

1. introdurre un linguaggio formale specificandone la *sintassi*, mediante una grammatica (senza attribuire alcun significato ai simboli). La specifica grammaticale consentirà di stabilire le frasi sintatticamente corrette del linguaggio;

2. definire una *semantica* per il linguaggio introdotto, che ne interpreta le frasi sintatticamente corrette, cioè assegna loro un significato.

Passeremo quindi, nei capitoli successivi, al problema di definire dei *calcoli* logici, cioè dei sistemi formali per esprimere ragionamenti su formule del linguaggio proposizionale.

## 1.3 Sintassi

In questa sezione definiremo il linguaggio formale della logica proposizionale considerandone gli elementi come stringhe, senza associare loro alcun significato. Per fare ciò è necessario, anzitutto, specificarne l'alfabeto (vale a dire i simboli che lo compongono), quindi le regole che, a partire da esso, consentono di costruire le frasi sintatticamente corrette.
In logica, è conveniente utilizzare dei simboli per rappresentare le proposizioni ed i connettivi. In questo capitolo, per motivi di chiarezza espositiva, le proposizioni atomiche verranno indicate con lettere iniziali maiuscole dell'alfabeto: "A","B",... (eventualmente indiciate). Si utilizzeranno, invece, le lettere maiuscole "P","Q",... (eventualmente indiciate) come *meta-simboli* per proposizioni composte.
I connettivi sono rappresentati dai seguenti simboli:

| | |
|---|---|
| $\neg$ | "non" |
| $\wedge$ | "e" |
| $\vee$ | "o" |
| $\rightarrow$ | "se.. allora" |

Utilizzeremo inoltre il simbolo $\bot$ per rappresenta la falsità, l'assurdo.

**Definizione 1.1** *L'* alfabeto *del linguaggio della logica proposizionale consta di:*

1. *simboli* atomici *di proposizione: "A","B",....*

2. *connettivi:* $\neg, \wedge, \vee, \rightarrow, \bot$

3. *simboli ausiliari: "(",")"*

Presentiamo ora le regole che consentono di costruire le frasi sintatticamente corrette del linguaggio, dette *formule ben formate* (fbf), o semplicemente proposizioni, che corrispondono alle proposizioni (atomiche e composte).

**Definizione 1.2** *L'insieme di* fbf (FBF) *è il minimo insieme $X$ tale che*

1. $A, B, \ldots \in X$ *per ogni simbolo atomico di proposizione;*
2. $\bot \in X$.
3. *Se* $P \in X$, $(\neg P) \in X$.
4. *Se* $P, Q \in X$, $(P \wedge Q), (P \vee Q), (P \to Q) \in X$.

$\neg$ $\wedge$, $\vee$ *e* $\to$ *sono, rispettivamente, i connettivi principali di* $\neg P$, $(P \wedge Q)$, $(P \vee Q)$ *e* $(P \to Q)$.

Si noti l'uso metalinguistico delle lettere $P$ e $Q$ nella definizione precedente: esse sono variabili che rappresentano arbitrarie fbf.

Esempi di formule ben formate sono: $(A \vee B) \to (\bot \wedge C)$, $(\bot \vee (A \to B))$, $((\neg A) \to B)$.
Osserviamo che è facile provare l'appartenenza di una data formula all'insieme delle fbf; meno semplice invece risulta dimostrarne la non appartenenza.

**Esempio 1.1** $((\to, \neg\vee, \to \neg\bot \notin X$.
Mostriamo, a scopo esemplificativo, che $\to \neg\bot \notin X$. Supponiamo per assurdo che $\to \neg\bot \in X$ e che $X$ soddisfi le condizioni 1-4 della Definizione 1.2. Sia $Y = X - \{\to \neg\bot\}$. Anche $Y$ soddisfa le medesime condizioni, in quanto $A, B, \ldots, \bot \in Y$, se $P \in Y, (\neg P) \in Y$ e se $P, Q \in Y$ allora $(P \wedge Q), (P \vee Q), (P \to Q) \in Y$ essendo diverse da $\to \neg\bot$. Dunque $X$ non è il minimo insieme che soddisfa le condizioni della Definizione 1.2, ma questo è assurdo, quindi $\to \neg\bot$ non può appartenere ad $X$.

Data una fbf $P$, ogni fbf $Q$ che appare come componente di $P$ è detta *sottoformula* di $P$. La definizione formale è la seguente (osserviamo che si ammette anche $P$ come sottoformula di $P$):

**Definizione 1.3** *Sia* $P$ *una fbf*

1. *Se* $P$ *è* $\bot$ *oppure una proposizione atomica, allora* $P$ *stessa è la sua sola sottoformula.*
2. *Se* $P$ *è* $\neg P_1$*, allora le sottoformule di* $P$ *sono* $P$ *stessa e quelle di* $P_1$.
3. *Se* $P$ *è* $(P_1 \vee P_2), (P_1 \wedge P_2), (P_1 \to P_2)$*, allora le sue sottoformule sono* $P$ *stessa e quelle di* $P_1$ *ed* $P_2$.

**Esempio 1.2** Sia $P$ la fbf $\neg((A \vee B) \wedge (\neg C))$. Le sue sottoformule sono: $P, ((A \vee B) \wedge (\neg C)), (A \vee B), A, B, (\neg C), C$.

Il linguaggio che abbiamo definito fa un largo uso di parentesi per evitare ambiguità nell'interpretazione delle formule. Ad esempio, se scrivessimo semplicemente $A \wedge B \rightarrow C$, potremmo interpretare questa formula sia come $((A \wedge B) \rightarrow C)$ che come $(A \wedge (B \rightarrow C))$.
Per non appesantire la struttura delle fbf ed aumentarne la leggibilità è bene introdurre delle regole di precedenza tra gli operatori al fine di eliminare, quando è possibile, le parentesi. Le precedenze sono definite nel modo seguente[4]: $\neg > \wedge > \vee > \rightarrow$. Questo significa che data una sentenza possibilmente ambigua, essa deve essere interpretata privilegiando le sottoformule i cui connettivi principali hanno precedenza più alta (ometteremo anche abitualmente la parentesi più esterna).

**Esempio 1.3** In base alle convenzioni appena discusse, la formula
$A \wedge \neg B \rightarrow C$ deve essere interpretata come $((A \wedge (\neg B)) \rightarrow C)$, in quanto il connettivo di negazione ha precedenza su quello di congiunzione che a sua volta ha precedenza maggiore rispetto a quello di implicazione. Analogamente, al posto di $((((\neg A) \wedge B) \vee C) \rightarrow (\neg D))$, si scriverà semplicemente $\neg A \wedge B \vee C \rightarrow \neg D$. D'altro canto, non è possibile eliminare le parentesi in $(A \rightarrow B) \vee C$ (altrimenti, sarebbe scorrettamente interpretata come $A \rightarrow (B \vee C)$).

### 1.3.1 Induzione

Numerosi risultati sulle fbf vengono dimostrati per induzione. È possibile utilizzare l'induzione "ordinaria" (sulla lunghezza della formula), ma è spesso più conveniente servirsi del seguente teorema, detto *Principio di Induzione Strutturale* per formule ben formate.

**Teorema 1.4** *Sia $\mathcal{A}$ una proprietà, allora $\mathcal{A}(P)$[5] per tutte le fbf $P$ se:*

1. *$\mathcal{A}$ è verificata per tutte le proposizioni atomiche*
2. *$\mathcal{A}(\bot)$*
3. *Se $\forall\, P \in FBF$ risulta $\mathcal{A}(P) \Rightarrow \mathcal{A}(\neg P)$*
4. *Se $\forall\ P, Q \in FBF$ risulta $\mathcal{A}(P)$ e $\mathcal{A}(Q) \Rightarrow \mathcal{A}(P \vee Q), \mathcal{A}(P \wedge Q)$ e $\mathcal{A}(P \rightarrow Q)$*

*Dimostrazione.* Sia $Y = \{F \in \text{FBF} \mid \mathcal{A}(F)\}$, allora $Y$ soddisfa le condizioni 1-4 della Definizione 1.2. Quindi FBF $\subseteq Y$, ma $Y \subseteq$ FBF, quindi $Y =$ FBF. □

Vedremo nel seguito vari utilizzi di tale principio.

[4] Per semplicità, indicheremo con il simbolo ">" l'espressione "$\cdots$ ha precedenza maggiore di $\cdots$"

[5] $\mathcal{A}$ è verificata.

## 1.4 Semantica

In questa sezione presenteremo la semantica del linguaggio precedentemente introdotto attribuendo un valore di verità ad ogni frase sintatticamente corretta (fbf).
Poiché le fbf sono costruite a partire dalle proposizioni atomiche mediante l'utilizzo dei connettivi $\neg, \wedge, \vee, \rightarrow$, l'idea è quella di assegnare alle proposizioni atomiche dei valori di verità ed estenderli a quelle composte in base al significato dei connettivi logici, che verrà descritto nel seguito.
Denoteremo con 1 il valore di verità "vero" e con 0 il valore di verità "falso". Sia $v$ una funzione dalle fbf all'insieme $\{0, 1\}$, tale che se $A$ è una proposizione atomica vera $v(A) = 1$, altrimenti $v(A) = 0$. Poiché $\perp$ rappresenta intuitivamente il falso, dovremo supporre sempre $v(\perp) = 0$. Per poter estendere $v$ all'insieme di tutte le proposizioni basta definire il valore di verità di una formula composta del tipo $\neg P$, $P \wedge Q$, $P \vee Q$, $P \rightarrow Q$ in funzione del valore di verità $v(P)$ e $v(Q)$ delle componenti $P$ e $Q$. Ovviamente, questo valore sarà diverso per i vari connettivi e in accordo con la loro semantica intesa.
Analizziamo dunque i vari connettivi:

- *Negazione*
  $\neg P$ è vera quando $P$ è falsa, e viceversa. Dunque, $v(\neg P) = 1 \Longleftrightarrow v(P) = 0$ o anche $v(\neg P) = 1 - v(P)$. Un modo conveniente per rappresentare il valore di verità di una formula composta è quello di utilizzare una tabella (o tavola) di verità, di ovvia interpretazione:

| $P$ | $\neg P$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

- *Congiunzione*
  $P \wedge Q$ è vera se e solo se $P$ è vera e $Q$ è vera. Avremo quindi $v(P \wedge Q) = 1 \Longleftrightarrow v(P) = v(Q) = 1$ o anche, in modo più compatto, $v(P \wedge Q) = min(v(P), v(Q))$. La tavola di verità di $P \wedge Q$ è la seguente:

| $P$ | $Q$ | $P \wedge Q$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

- *Disgiunzione*
  $P \vee Q$ risulta vera se *almeno*[6] una tra $P$ e $Q$ è vera.
  Dunque $v(P \vee Q) = 1 \Longleftrightarrow v(P) = 1$ oppure $v(Q) = 1$, o, più brevemente, $v(P \vee Q) = max(v(P), v(Q))$. La tavola di verità di $P \vee Q$ è la seguente:

[6] Il connettivo di disgiunzione va inteso in senso inclusivo (il "vel" latino) e non in senso esclusivo (l'"aut" latino).

| $P$ | $Q$ | $P \vee Q$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

- *Implicazione*
  Il connettivo di implicazione è meno ovvio dei precedenti. Consideriamo la formula $P \to Q$. Se $P$ e $Q$ sono entrambe vere o entrambe false, è facile convincersi che l'implicazione deve essere vera: $P$ e $Q$ denotano in questo caso la stessa informazione e dunque si può dire che una implica l'altra; in altri termini, si vuole che $A \to A$ sia sempre vera, indipendentemente dal valore di verità di $A$[7]. Inoltre, se $Q$ è vera, $P \to Q$ è ancora vera: questo essenzialmente dice che la nozione di verità deve essere stabile rispetto all'aggiunta di nuove ipotesi (se $Q$ è vera, allora resta vera anche se vale $P$). Rimane il caso in cui $P$ è vera e $Q$ è falsa. Ovviamente, in tale situazione l'implicazione è falsa.
  Dunque, $P \to Q$ risulta falsa soltanto quando $P$ è vera e $Q$ è falsa. Avremo quindi $v(P \to Q) = 0 \iff v(P) = 1$ e $v(Q) = 0$, o anche $v(P \to Q) = 1 \iff v(P) \leq v(Q)$. La tavola di verità di $P \to Q$ è la seguente:

| $P$ | $Q$ | $P \to Q$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

- Il simbolo $\perp$ viene interpretato come falso, quindi $v(\perp) = 0$.

Vediamo ora la definizione formale di interpretazione $v$.

**Definizione 1.5** *Una funzione* $v : FBF \to \{0, 1\}$ *è un'* interpretazione *se:*

- $v(\neg P) = 1 - v(P)$
- $v(\perp) = 0$
- $v(P \wedge Q) = min(v(P), v(Q))$
- $v(P \vee Q) = max(v(P), v(Q))$
- $v(P \to Q) = 0 \iff v(P) = 1$ *e* $v(Q) = 0$

[7] Russel [Rus03] considera l'enunciato $A \to A$ come del tutto equivalente all'asserzione "A è una proposizione", in quanto "qualsiasi proposizione implica se stessa, e qualsiasi cosa che non sia un proposizione non implica nulla". Dunque $A$ è una proposizione se e solo se implica se stessa.

Dalla Definizione 1.5, il valore $v(P)$ di una fbf $P$ risulta definito da una funzione di interpretazione che associa un significato a tutte le proposizioni, comprese quelle che non compaiono in $P$; tuttavia, è facile dimostrare che $v(P)$ dipende solo dagli assegnamenti dei valori di verità delle formule atomiche che compongono $P$; infatti:

**Lemma 1.6** *Sia $P$ una fbf, e $v, v'$ due interpretazioni. Se $v(A_i) = v'(A_i)$ per tutte le proposizioni atomiche $A_i$ che occorrono in $P$, allora $v(P) = v'(P)$.*

*Dimostrazione.* Per induzione sulla struttura di $P$.
(*caso base*) Se $P$ è una proposizione atomica, l'asserto segue banalmente dalle ipotesi. Veniamo al caso induttivo. Supponiamo che $P$ abbia la forma $P_1 \wedge P_2$. Per ipotesi induttiva, $v(P_1) = v'(P_1)$ e $v(P_2) = v'(P_2)$. Dunque $v(P) = v(P_1 \wedge P_2) = min(v(P_1), v(P_2)) = min(v'(P_1), v'(P_2)) = v'(P_1 \wedge P_2) = v'(P)$. I rimanenti casi sono analoghi e vengono lasciati al lettore come esercizio. □

**Esempio 1.4** Sia $P = \neg(A \rightarrow A \wedge B)$ e $v$ un'interpretazione la cui restrizione a $\{A, B\}$ è la seguente: $v(A) = 1$ e $v(B) = 0$.
Dalla Definizione 1.5 segue che $v(\neg(A \rightarrow A \wedge B)) = 1 - v(A \rightarrow A \wedge B)$, $v(A \wedge B) = min(v(A), v(B)) = min(0, 1) = 0$ quindi $v(\neg(A \rightarrow A \wedge B)) = 1$.
La computazione appena descritta può essere rappresentata, in modo più compatto, da una tavola di verità, come segue:

| $A$ | $B$ | $A \wedge B$ | $A \rightarrow A \wedge B$ | $\neg(A \rightarrow A \wedge B)$ |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 |

**Definizione 1.7** *Sia $P$ una fbf e $v$ una qualche interpretazione. Se $v(P) = 1$, allora si dice che $P$ è* soddisfatta *nell'interpretazione $v$, oppure, analogamente, che $v$ è un* modello *per $P$; in tale situazione scriveremo $v \models P$.*

**Definizione 1.8** *Una fbf $P$ è* soddisfacibile *se ha almeno un modello, cioè se esiste almeno una interpretazione che la soddisfa; in caso contrario $P$ è* insoddisfacibile *o* contraddittoria*.*

**Definizione 1.9** *Una fbf $P$ è una* tautologia *(è* valida*) se ogni interpretazione $v$ è un modello per $P$ (cioè se $P$ risulta vera in ogni interpretazione). In tale situazione scriveremo $\models P$.*

Dal Lemma 1.6 segue che per determinare se una data formula $P$ è contraddittoria, soddisfacibile o tautologica, è sufficiente controllare un numero finito di assegnamenti di valori in $\{0, 1\}$ per le proposizioni atomiche distinte che vi compaiono. Tale controllo si può effettuare mediante una tavola di verità. Siano $P$ la fbf da esaminare, $A_i$ con $i = 1, \ldots, n$ le proposizioni atomiche distinte che la compongono e $v_j$ con $j = 1, \ldots, 2^n$ le possibili interpretazioni.

| $A_1$ | $A_2$ | $\cdots$ | $A_n$ | $P$ |
|---|---|---|---|---|
| $v_1(A_1)$ | $v_1(A_2)$ | $\cdots$ | $v_1(A_n)$ | $v_1(P)$ |
| $v_2(A_1)$ | $v_2(A_2)$ | $\cdots$ | $v_2(A_n)$ | $v_2(P)$ |
| $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ |
| $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ |
| $v_{2^n}(A_1)$ | $v_{2^n}(A_2)$ | $\cdots$ | $v_{2^n}(A_n)$ | $v_{2^n}(P)$ |

$P$ è soddisfacibile se e solo se la sequenza di valori di verità ottenuti per $P$ $(v_1(P), \ldots, v_{2^n}(P))$ contiene almeno un 1, è una tautologia se tale sequenza è composta soltanto da 1, è contraddittoria se è composta unicamente da 0.

**Esempio 1.5** 1. Sia $P$ la fbf $A \to \neg A$

| $A$ | $\neg A$ | $A \to \neg A$ |
|---|---|---|
| 0 | 1 | 1 |
| 1 | 0 | 0 |

$P$ è soddisfacibile, e l'interpretazione che la soddisfa è quella che assegna ad $A$ il valore 0.

2. Un semplice esempio di formula contraddittoria è $A \wedge \neg A$. Infatti:

| $A$ | $\neg A$ | $A \wedge \neg A$ |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 0 | 0 |

3. Sia $Q$ la fbf $A \to A \vee B$.

| $A$ | $B$ | $A \vee B$ | $A \to A \vee B$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 |

Q è una tautologia.

**Teorema 1.10** *Una fbf $P$ è una tautologia se e solo se $\neg P$ è insoddisfacibile.*

*Dimostrazione.*

P è una tautologia $\iff$ ogni interpretazione è un modello per $P$
$\iff$ ogni interpretazione non è un modello per $\neg P$
$\iff$ $\neg P$ non ha modelli
$\iff$ $\neg P$ è insoddisfacibile.

□

Le nozioni di soddisfacibilità ed insoddisfacibilità si estendono in modo ovvio a insiemi (eventualmente infiniti) di formule. Infatti

**Definizione 1.11** *Sia* $\Gamma$ *un insieme di fbf,* $\Gamma$ *è* soddisfacibile *se esiste un'interpretazione* $v$ *tale che* $\forall P_i \in \Gamma, v(P_i) = 1$. $\Gamma$ *è* insoddisfacibile *se per ogni interpretazione* $v$, $\exists P_i \in \Gamma$ *tale che* $v(P_i) = 0$.

**Definizione 1.12** *Sia* $\Gamma$ *un insieme di fbf. Diremo che* $Q$ *è* conseguenza (semantica) *di* $\Gamma$, *e scriveremo* $\Gamma \models Q$, *se e solo se* $\forall v \quad ((\forall P_i \in \Gamma, v(P_i) = 1) \Rightarrow v(Q) = 1)$. *In caso contrario scriveremo* $\Gamma \not\models Q$.

In altri termini $\Gamma \models Q$ se e solo se $Q$ è vera per tutte le interpretazioni che sono dei modelli per $\Gamma$.
Vediamo ora come la nozione di conseguenza semantica sia correlata a quelle di soddisfacibilità e tautologia.

**Lemma 1.13** $\Gamma \models P$ *se e solo se* $\Gamma \cup \{\neg P\}$ *è insoddisfacibile.*

*Dimostrazione.* Per definizione, $\Gamma \models P$ se e solo se per ogni interpretazione $v$ $((\forall P_i \in \Gamma, v(P_i) = 1) \Rightarrow v(P) = 1)$. Questo equivale a dire che per ogni $v$ o esiste un qualche $P_i \in \Gamma$ tale che $v(P_i) = 0$ oppure $v(P) = 1$ e dunque, per ogni $v$ esiste almeno una formula $Q \in \Gamma \cup \{\neg P\}$ per cui $v(Q) = 0$. □

**Lemma 1.14** $P \models Q$ *se e solo se* $\models P \rightarrow Q$.

*Dimostrazione.* ($\Rightarrow$) Supponiamo che $P \models Q$, allora $\forall v, v(P) = 1 \Rightarrow v(Q) = 1$ e quindi $v(P \rightarrow Q) = 1$ $\forall v$, vale a dire che $\models P \rightarrow Q$.
($\Leftarrow$) Supponiamo che $\models P \rightarrow Q$, allora $\forall v, v(P \rightarrow Q) = 1$; ma poiché $v(P \rightarrow Q) = 0 \iff v(P) = 1$ e $v(Q) = 0$, vuol dire che $\forall v, v(P) = 1 \Rightarrow v(Q) = 1$, ossia $P \models Q$. □

**Teorema 1.15 (Deduzione semantica)**
$P_1, \cdots, P_n \models Q$ *se e solo se* $P_1, \cdots, P_{n-1} \models P_n \rightarrow Q \quad \forall n \geq 1$.

*Dimostrazione.* Si effettua, per induzione su $n$, utilizzando il lemma precedente. □

**Esempio 1.6** Pur nella sua povertà espressiva, la logica proposizionale è sufficiente per formalizzare ragionamenti di una certa complessità. Consideriamo ad esempio le seguenti ipotesi:
(a) Se Carlo è americano e Giovanni non è francese, allora Elena è tedesca.
(b) Se Elena è tedesca, allora Lucia è spagnola oppure Giovanni è francese.
(c) Se Lucia non è spagnola allora Carlo è americano.
(d) Giovanni non è francese.
Vogliamo dimostrare che queste implicano che Lucia è spagnola.
Il primo passo consiste nel formalizzare il problema in logica proposizionale. A tale scopo, si individuano innanzitutto le sentenze elementari e si associano ad esse, per semplicità, dei nomi simbolici. Ad esempio, si può porre

A = Carlo è americano
B = Giovanni è francese
C = Elena è tedesca
D = Lucia è spagnola

Le ipotesi iniziali si riscrivono quindi nel modo seguente:

(a) = $A \wedge \neg B \rightarrow C$
(b) = $C \rightarrow D \vee B$
(c) = $\neg D \rightarrow A$
(d) = $\neg B$

ed il problema consiste nel dimostrare che

$$(A \wedge \neg B \rightarrow C) \wedge (C \rightarrow D \vee B) \wedge (\neg D \rightarrow A) \wedge (\neg B) \models D$$

o anche, per il teorema di deduzione semantica,

$$\models (A \wedge \neg B \rightarrow C) \wedge (C \rightarrow D \vee B) \wedge (\neg D \rightarrow A) \wedge (\neg B) \rightarrow D$$

La natura tautologica della formula

$$P = (A \wedge \neg B \rightarrow C) \wedge (C \rightarrow D \vee B) \wedge (\neg D \rightarrow A) \wedge (\neg B) \rightarrow D$$

può ora essere facilmente verificata mediante la costruzione della sua tabella di verità:

| $A$ | $B$ | $C$ | $D$ | $(a)$ | $(b)$ | $(c)$ | $(d)$ | $(a) \wedge (b) \wedge (c) \wedge (d)$ | $P$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

### 1.4.1 Decidibilità della logica proposizionale ⋆

Nella sezione precedente abbiamo visto che è possibile *decidere* se una fbf $P$ è soddisfacibile costruendone la tabella di verità e verificando se la colonna corrispondente alla formula $P$ contiene almeno un 1. Se tale formula contiene $n$

proposizioni atomiche distinte, la sua tabella di verità sarà costituita da $2^n$ righe, corrispondenti a tutte le differenti interpretazioni di $P$. Dunque, nel caso peggiore, il suddetto algoritmo di verifica può richiedere un tempo esponenziale rispetto alla dimensione della formula.
È naturale chiedersi se non sia possibile fare meglio, ed in particolare se il problema della soddisfacibilità per formule della logica proposizionale (che indicheremo brevemente con SAT) non sia risolubile in tempo polinomiale. Fino ad ora nessuno è stato in grado di provare che SAT sia *intrinsecamente* esponenziale. Tuttavia, esiste una certa evidenza di questo fatto, in quanto è possibile dimostrare che se SAT fosse risolubile in tempo polinomiale allora tutta una serie di importanti problemi matematici, fisici ed informatici di cui, per anni, si è cercato invano una soluzione algoritmicamente efficiente sarebbero risolubili in modo sostanzialmente migliore di quanto non sia attualmente noto (in particolare, sarebbero anch'essi risolubili in tempo polinomiale).
Vediamo di essere un pò più formali. Sia $\mathcal{P}$ la classe dei problemi polinomiali, o meglio la classe dei linguaggi per i quali esiste un algoritmo di riconoscimento, cioè un algoritmo in grado di decidere se una data espressione appartiene o meno al linguaggio in esame in tempo polinomiale.
Sia $\mathcal{NP}$ la classe dei linguaggi per cui esiste un algoritmo *non deterministico* di riconoscimento che opera in tempo polinomiale. Ricordiamo che un algoritmo non deterministico è un algoritmo che, dovendo effettuare delle "decisioni", ha la capacità di scegliere sempre quella più conveniente per la risoluzione del problema. Equivalentemente, si può immaginare che l'algoritmo, posto di fronte a più alternative possibili, sia in grado di duplicare se stesso, procedendo in parallelo su tutte le computazioni fino a che una di esse non termina con successo.
Ovviamente, SAT è un problema in $\mathcal{NP}$, in quanto per determinare se una proposizione $Q$ è soddisfacibile basta "indovinare" non deterministicamente l'interpretazione corretta e verificare che essa soddisfa effettivamente $Q$ (questo corrisponde al calcolo di una riga della tabella di verità, e richiede un tempo lineare nella dimensione di $Q$). In modo equivalente, si possono considerare in parallelo le varie interpretazioni possibili, o meglio, duplicare l'algoritmo di verifica ogni volta che si incontra una nuova proposizione atomica $A$ in $Q$ in base alle due possibili alternative $A = 0$ o $A = 1$.
Naturalmente, risulta $\mathcal{P} \subseteq \mathcal{NP}$ in quanto gli algoritmi deterministici possono essere visti come casi particolari di quelli non deterministici. Non è noto, tuttavia, se tale inclusione sia stretta, cioè se $\mathcal{P} \neq \mathcal{NP}$, oppure se le due classi coincidono. Questo resta tuttoggi il più famoso dei problemi aperti nell'ambito dell'informatica teorica.

**Definizione 1.16** *Dati due linguaggi $L_1$ e $L_2$ diremo che $L_1$ è* riducibile in tempo polinomiale *a $L_2$ se esiste un algoritmo che, in tempo polinomiale, trasforma ogni stringa $x$ in una stringa $g(x)$ tale che $x \in L_1$ se e solo se $g(x) \in L_2$.*

Un linguaggio si dice $\mathcal{NP}$-*completo*, se appartiene ad $\mathcal{NP}$ ed *ogni* altro linguaggio in $\mathcal{NP}$ è riducibile ad esso in tempo polinomiale.
A prima vista, il fatto che esistano linguaggi $\mathcal{NP}$-*completi* può risultare sorprendente. In realtà, molti importanti problemi pratici sono $\mathcal{NP}$-*completi*. Per

nessuno di questi è nota una soluzione polinomiale (che implicherebbe $\mathcal{P} = \mathcal{NP}$), il che induce alla congettura $\mathcal{P} \neq \mathcal{NP}$.

**Teorema 1.17 (di Cook)**
*SAT è $\mathcal{NP}$-completo.*

La dimostrazione di tale teorema richiederebbe una formalizzazione più precisa delle classi $\mathcal{P}$ e $\mathcal{NP}$ con riferimento a particolari agenti di calcolo (di solito, a tale scopo, si utilizzano le macchine di Turing[8]) che esula dagli obiettivi del presente volume. Tuttavia, al fine soprattutto di mostrare le sorprendenti potenzialità espressive del calcolo proposizionale presentiamo un'intuizione della dimostrazione.

*Idea della dimostrazione*
Sia $L$ un linguaggio in $\mathcal{NP}$. Allora esiste un algoritmo (o programma) non deterministico in grado di decidere in un tempo polinomiale $p(n)$ se una data stringa $x$ di lunghezza $n$ appartiene o meno ad $L$. Vogliamo costruire una formula $Q_x$ che sia soddisfacibile se e solo se $x \in L$. Inoltre, per rispettare la *riducibilità polinomiale*, $Q_x$ deve poter essere generata in tempo polinomiale rispetto ad $x$. Consideriamo gli stati successivi $s_0, s_1, \ldots s_{p(n)}$ della computazione del programma. Ogni stato è essenzialmente definito dalla particolare configurazione della memoria al tempo $t$, con $1 \leq t \leq p(n)$. Ora, un risultato elementare, ma essenziale, della teoria della complessità afferma che la complessità in spazio è sempre inferiore o uguale alla complessità in tempo. Infatti, poiché ogni operazione elementare è finita, può al massimo scrivere in un numero finito (supponiamo unitario) di nuove locazioni di memoria. Dunque, la computazione del programma può essere circoscritta ad un'area di memoria di dimensione $p(n)$.
Come conseguenza, l'intera sequenza degli stati $s_0, s_1, \ldots s_{p(n)}$ può essere rappresentata da una matrice $C$ di dimensione $p(n)^2$. La prima riga descriverà la memoria al tempo 1, la seconda riga la memoria al tempo 2, e così via fino all'istante $p(n)$. Ogni cella $C[i, j]$ della matrice definisce dunque il contenuto $X$ della locazione di memoria $j$ all'istante $i$. Poiché ogni locazione di memoria ha dimensione finita, i possibili valori di $X$ sono finiti.
Definiamo quindi una proposizione atomica $C_{i,j,X}$ per ogni possibile valore di $i, j, X$. L'idea è che $C_{i,j,X} = 1$ se e solo se $C[i, j] = X$. Alcune semplici formule logiche basteranno per garantire che, per ogni $i$ e $j$, esattamente una sola proposizione atomica $C_{i,j,X}$ risulti vera. Questo è sufficiente per rappresentare la matrice in logica proposizionale. Dobbiamo quindi descrivere, mediante formule proposizionali, quali sono le configurazioni ammissibili dello stato all'istante $i + 1$ (riga $i + 1$ della matrice) rispetto allo stato all'istante precedente (ovvero alla riga i). Data la natura finita dell'agente di calcolo (il microprocessore), i cambiamenti ammissibili della configurazione sono finiti, ed hanno una natura "locale" ed "uniforme" che in generale ne consente una semplice codifica. Per meglio dire è possibile definire una funzione $f(W, X)$ tale che $X$ può apparire nello stato $i + 1$ in una qualche posizione $j$, se e solo se lo stato all'istante $i$

[8] Per una semplice introduzione alle Macchine di Turing si vedano [Man78] e [AAAM79].

in un opportuno "intorno" finito di $j$ (che in generale dipende dal particolare agente di calcolo) è W. A questo punto restano solo da aggiungere delle formule che specificano lo stato iniziale e finale della computazione.
La formula $Q_x$ risultante dalla congiunzione di tutte le formule precedenti è soddisfacibile se e solo se il programma era in grado di riconoscere $x$. Inoltre $Q_x$ è sufficientemente semplice da essere generata in tempo polinomiale a partire da $x$. □

Abbiamo discusso fino ad ora il problema della soddisfacibilità di formule proposizionali. Che possiamo dire riguardo alla loro tautologicità? Poiché una formula è una tautologia se e solo se la sua negata è insoddisfacibile (Teorema 1.10), tale problema è il complementare di SAT (verrà nel seguito indicato con $\overline{SAT}$). Intuitivamente, decidere se una formula $P$, contenente $n$ proposizioni atomiche distinte, è insoddisfacibile appare più complesso che non decidere la sua soddisfacibilità, in quanto, nel primo caso, si devono *comunque* controllare *tutte* le $2^n$ possibili interpretazioni. Anche avendo a disposizione un algoritmo non deterministico, è comunque necessario attendere il risultato di *tutte* le computazioni per ogni interpretazione e verificare che nessuna di queste soddisfi la formula.
In generale, non è noto se la classe dei linguaggi $\mathcal{NP}$ sia chiusa per complementazione. Se questo non fosse vero, risulterebbe $\mathcal{P} \neq \mathcal{NP}$, in quanto $\mathcal{P}$ è banalmente chiusa per complementazione. In particolare, non esiste nessun problema $\mathcal{NP}$-completo il cui complemento sia noto essere in $\mathcal{NP}$.
La classe dei linguaggi il cui complemento è in $\mathcal{NP}$ si dice co-$\mathcal{NP}$. Ovviamente, $\mathcal{NP} \bigcap co-\mathcal{NP} \neq \emptyset$ in quanto $\mathcal{P} \in \mathcal{NP} \bigcap co-\mathcal{NP}$. Esistono inoltre dei problemi (il più famoso è quello di decidere se un numero intero $n$ non è primo, cioè se esistono due fattori $p$ e $q$ tali che $n = pq$), che sono in $\mathcal{NP}$, hanno complementare in $\mathcal{NP}$ e per i quali non è noto un algoritmo polinomiale. Questo suggerisce la congettura che possano esistere linguaggi nell'intersezione di $\mathcal{NP}$ e co-$\mathcal{NP}$ che non sono in $\mathcal{P}$.

### 1.4.2 Teorema di compattezza

In questa sezione dimostreremo un risultato semantico di particolare importanza: il teorema di compattezza.

**Teorema 1.18 (Compattezza[9])**
*Un insieme* $\Gamma$ *di fbf è soddisfacibile se e solo se ogni sottoinsieme finito* $\Delta$ *di* $\Gamma$ *lo è.*

*Dimostrazione.* ($\Rightarrow$) Immediata.
($\Leftarrow$) Per ipotesi ogni sottoinsieme finito di $\Gamma$ è soddisfacibile. Siano $A_1, A_2, A_3 \cdots$ le formule atomiche distinte che compaiono in $\Gamma$. Supponiamo di aver definito un assegnamento $v$ di valori di verità ad $A_1, \cdots, A_n$ tale che ogni sottoinsieme finito di $\Gamma$ ha un modello nel quale $A_1, \cdots, A_n$ assumono i valori $v(A_1), \cdots, v(A_n)$

[9] Il nome "compattezza" deriva dal fatto che tale teorema implica la compattezza (in senso topologico) di un opportuno spazio topologico detto *spazio di Stone* (cfr.[BM77] p.182).

e mostriamo che è sempre possibile estendere $v$ ad $A_{n+1}$ in modo che tale proprietà continui a valere.
Supponiamo, per fissare le idee, che ponendo $v(A_{n+1}) = 0$ non sia verificata la summenzionata condizione; questo significa che esiste $\Delta' \subseteq \Gamma$, $\Delta'$ finito, che non ha un modello nel quale $A_1 \cdots A_n, A_{n+1}$ assumono i valori $v(A_1), \cdots, v(A_n)$ e $v(A_{n+1}) = 0$. Proviamo allora che ogni sottoinsieme finito di $\Gamma$ ha un modello nel quale $A_1, \cdots, A_n, A_{n+1}$ assumono i valori $v(A_1), \cdots, v(A_n)$, e $v(A_{n+1}) = 1$. Infatti, sia $\Delta$ un qualunque sottoinsieme finito di $\Gamma$; allora $\Delta \cup \Delta'$ è ancora un sottoinsieme finito di $\Gamma$ e quindi, per ipotesi induttiva ha un modello nel quale $A_1, \cdots, A_n$ assumono i valori $v(A_1), \cdots, v(A_n)$, e, per come si è scelto $\Delta'$, $v(A_{n+1})$ dovrà assumere il valore 1.
Il limite $v$ di questa costruzione è un modello per $\Gamma$. Infatti, $\forall P \in \Gamma$, scelto $n$ sufficientemente grande, tutte le proposizioni atomiche in $P$ occorrono in $\{A_1, \cdots, A_n\}$. Essendo $\{P\}$ un sottoinsieme finito di $\Gamma$ e poichè per ogni $n$ tutti i sottoinsiemi finiti di $\Gamma$ hanno un modello nel quale $A_1, \cdots, A_n$ assumono i valori $v(A_1), \cdots, v(A_n)$ risulta $v(P) = 1$. □

***Osservazione*** Tale dimostrazione non è *costruttiva* in quanto garantisce l'esistenza di un modello per $\Gamma$, ma non lo esibisce.

Il teorema di compattezza può essere formulato, in modo equivalente, come segue:

**Teorema 1.19** *Un insieme $\Gamma$ di fbf è insoddisfacibile se e solo se esiste un sottoinsieme finito $\Delta$ di $\Gamma$ che lo è.*

Nel capitolo dedicato al *Metodo di risoluzione* verrà utilizzato in questa seconda forma. Come semplice corollario della formulazione precedente del teorema di compattezza si ottiene:

**Teorema 1.20** *Sia $\Gamma$ un insieme di fbf e $P$ una proposizione. $\Gamma \models P$ se e solo se esiste un sottoinsieme finito $\Delta$ di $\Gamma$ tale che $\Delta \models P$.*

*Dimostrazione.* ($\Leftarrow$) Immediata.
($\Rightarrow$) Per il Lemma 1.13, $\Gamma \models P$ se e solo $\Gamma \cup \{\neg P\}$ è insoddisfacibile. Dal Teorema 1.19 esiste allora un sottoinsieme finito $\Delta$ di $\Gamma$ tale che $\Delta \cup \{\neg P\}$ è insoddisfacibile. Da cui segue, ancora per il Lemma 1.13 che $\Delta \models P$. □

Applicazioni di tale teorema alla logica verranno mostrate nei capitoli 5 e 6; come esempio di utilizzo dello stesso nella matematica si veda l'Esercizio 1.21.

### 1.4.3 Nota sul connettivo di implicazione ⋆

La semantica fornita in precedenza per il connettivo di implicazione non rispecchia del tutto fedelmente il significato intuitivo che si dà alle espressioni condizionali nel linguaggio naturale. Osserviamo in particolare che in base a tale semantica si ritiene vera una formula implicativa del tipo $A \rightarrow B$ anche nel caso in cui $A$ sia falsa e $B$ sia vera. Sebbene questa sia l'interpretazione data

al connettivo di implicazione fin dall'antichità (Filone il Megarico), e riassunta nel motto latino *ex falso sequitur quodlibet*, essa può portare a conseguenze apparentemente paradossali.
Consideriamo ad esempio la frase *se oggi è martedi domani piove oppure se domani piove oggi è martedi*. Questa è apparentemente priva di senso. Tuttavia, se formalizzata in logica proposizionale, essa assume la forma $(A \to B) \vee (B \to A)$, che è in effetti una *tautologia*, infatti:

| $A$ | $B$ | $A \to B$ | $B \to A$ | $(A \to B) \vee (B \to A)$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Le obiezioni relative all'interpretazione del connettivo di implicazione sono essenzialmente riconducibili a due ordini differenti di problemi:

1. relazione tra il piano intensionale e quello estensionale;
2. relazione tra il piano linguistico e quello metalinguistico.

Per quanto riguarda il primo aspetto, ricordiamo che la logica proposizionale non intende trattare le intensioni delle sentenze (il senso), vale a dire ciò che le sentenze effettivamente esprimono, ma unicamente le relazioni tra le loro denotazioni, ovvero tra i loro valori di verità. Dunque si deve sempre fare riferimento alla forma astratta delle sentenze composte, vedendo i costituenti elementari come dei semplici nomi. Nel caso precedente, la formula da considerare non è dunque *se oggi è martedi domani piove oppure se domani piove oggi è martedi*, ma la sua astrazione $(A \to B) \vee (B \to A)$. Questo risolve parte dell'ambiguità. Tuttavia, ancora, su di un piano strettamente logico, non si vede alcuna ragione per cui date due formule completamente arbitrarie $A$ e $B$ si debba necessariamente avere che $A$ implichi $B$ oppure $B$ implichi $A$. Il problema in questo caso è più sottile, e legato al fatto che intuitivamente si è portati a leggere l'implicazione $A \to B$ come una asserzione di inferibilità, cioè il fatto che da $A$ siamo in grado di concludere $B$ *mediante un qualche ragionamento logico*. Ovviamente, questa relazione di inferibilità, che verrà trattata formalmente nel prossimo capitolo, appartiene ad un livello metalinguistico. Dunque, il problema si riflette a questo livello. Indichiamo informalmente con $A \vdash B$ il fatto che siamo in grado di "concludere" $B$ dalla ipotesi $A$. Ovviamente, per $A$ e $B$ arbitrari, non sarà mai vero che $A \vdash B$ oppure $B \vdash A$ (dunque non abbiamo una corrispondenza *immediata* tra il livello linguistico e quello metalinguistico). Tuttavia, se si suppone $A$, ci si aspetta di poter concludere $B \to A$, e a maggior ragione anche $(A \to B) \vee (B \to A)$. Allo stesso modo, se si suppone $\neg A$, ci si aspetta di poter ancora concludere $A \to B$, e quindi ancora $(A \to B) \vee (B \to A)$. Dunque, sia nel caso in cui valga $A$ sia nel caso in cui valga $\neg A$ siamo in grado di concludere logicamente $(A \to B) \vee (B \to A)$. Possiamo ora legittimamente concludere che *in ogni caso* vale $(A \to B) \vee (B \to A)$? Si, se sul piano deduttivo (metalinguistico) si assume che $A \vee \neg A$ sia comunque dimostrabile, no altrimenti. Come

vedremo, l'approccio classico, mirato ad avere una completa corrispondenza tra la semantica a valori di verità e la relazione di derivabilità del calcolo logico, ammetterà tale assioma (o una qualche forma ad esso equivalente). Altri sistemi logici, tra i quali assume una particolare rilevanza il *sistema intuizionista*, a cui accenneremo brevemente nel prossimo capitolo, rifiutano invece questa regola (per cui una formula del tipo $(A \to B) \vee (B \to A)$ non sarà derivabile). In generale, sistemi siffatti sono detti *non classici*.

### 1.4.4 Equivalenza semantica

Quando i valori di verità di due formule coincidono per ogni interpretazione, queste si dicono semanticamente equivalenti. Formalmente:

**Definizione 1.21** *Due fbf $P$ e $Q$ sono* (semanticamente) equivalenti *se per ogni interpretazione $v$, $v(P) = v(Q)$. Simbolicamente, denoteremo ciò con $P \equiv Q$.*

In altri termini, due fbf sono equivalenti se hanno la stessa tabella di verità.

Intuitivamente, sostituendo una proposizione $P$ con una proposizione $Q$ ad essa equivalente all'interno di una proposizione $R$, ci si aspetta di ottenere una proposizione equivalente ad $R$. Per formalizzare questa intuizione è necessario introdurre la nozione di *sostituzione*.

**Definizione 1.22** *Siano $R$ e $P$ due fbf, e sia $A$ una fbf atomica (non necessariamente contenuta in $R$). $R[^P/_A]$ è la fbf ottenuta rimpiazzando tutte le occorrenze di $A$ in $R$ con $P$. La* sostituzione *di $P$ al posto di $A$ definisce una funzione dalle formule ben formate alle formule ben formate che può essere formalizzata, per induzione, nel seguente modo:*

$$R[^P/_A] = \begin{cases} R & \text{se } R \text{ è una proposizione atomica diversa da } A \\ P & \text{se } R = A \end{cases}$$

$$\begin{aligned}
(\neg R)[^P/_A] &= \neg R[^P/_A] \\
(R_1 \vee R_2)[^P/_A] &= R_1[^P/_A] \vee R_2[^P/_A] \\
(R_1 \wedge R_2)[^P/_A] &= R_1[^P/_A] \wedge R_2[^P/_A] \\
(R_1 \to R_2)[^P/_A] &= R_1[^P/_A] \to R_2[^P/_A]
\end{aligned}$$

**Teorema 1.23** *Sia $v$ una qualche interpretazione. Se $v(P) = v(Q)$, allora, per ogni $R$, $v(R[^P/_A]) = v(R[^Q/_A])$.*

*Dimostrazione.* Per induzione sulla struttura di $R$.
(*caso base*) Se $R$ è una proposizione atomica diversa da $A$, allora l'asserto è banalmente verificato, in quanto in tale situazione $R[^P/_A] = R = R[^Q/_A]$. Se invece $R = A$, allora risulta $v(R[^P/_A]) = v(P) = v(Q) = v(R[^Q/_A])$.
Veniamo al caso induttivo.

- Se $R$ è $\neg R_1$. Per ipotesi induttiva $v(R_1[^P/_A]) = v(R_1[^Q/_A])$. Allora, $v((\neg R_1)[^P/_A]) = 1 - v(R_1[^P/_A]) = 1 - v(R_1[^Q/_A]) = v((\neg R_1)[^Q/_A])$.

- Se $R$ è $(R_1 \wedge R_2)$. Per ipotesi induttiva $v(R_i[^P/_A]) = v(R_i[^Q/_A])$ per $i = 1, 2$. Risulta:

$$\begin{array}{rcl} v((R_1 \wedge R_2)[^P/_A]) & = & v(R_1[^P/_A] \wedge R_2[^P/_A]) \\ & = & min\{v(R_1[^P/_A], v(R_2[^P/_A])\} \\ & = & min\{v(R_1[^Q/_A], v(R_2[^Q/_A])\} \\ & = & v(R_1[^Q/_A] \wedge R_2[^Q/_A]) \\ & = & v((R_1 \wedge R_2)[^Q/_A]). \end{array}$$

I rimanenti casi sono lasciati al lettore come esercizio. □

Vediamo una semplice applicazione del teorema precedente.

**Esempio 1.7** Sia $\top = \neg\bot$. Ovviamente, in ogni interpretazione risulta $v(\top) = 1$. Vogliamo dimostrare che per ogni formula $P$ ed ogni proposizione atomica $A$ la formula

$$Q = P \rightarrow (A \rightarrow P[^\top/_A])$$

è una tautologia. Sia dunque $v$ una generica interpretazione. Se $v(P) = 0$, $Q$ è banalmente vera. Supponiamo dunque che $v(P) = 1$ e distinguiamo due casi a seconda che $v(A) = 0$ o $v(A) = 1$. Nel primo caso, $v(A \rightarrow P[^\top/_A]) = 1$ e dunque anche $v(Q) = 1$. Nel secondo caso, $v(A) = v(\top)$ e per il Teorema 1.23 deve essere $v(P[^\top/_A]) = v(P[^A/_A]) = v(P) = 1$. Come conseguenza, $v(A \rightarrow P[^\top/_A]) = 1$ e dunque anche $v(Q) = 1$.

Dal Teorema 1.23 segue che in ogni fbf sostituendo parti tra loro equivalenti si ottengono fbf equivalenti; questo risultato è espresso dal seguente teorema, detto teorema di sostituzione (TS):

**Teorema 1.24** *Se $P \equiv Q$, allora $R[^P/_A] \equiv R[^Q/_A]$.*

*Dimostrazione.* Sia $v$ una generica interpretazione. Poichè $P \equiv Q$, $v(P) = v(Q)$. Dunque, per il Teorema 1.23, $v(R[^P/_A]) = v(R[^Q/_A])$. □

Vediamo qui di seguito delle utili equivalenze di fbf.

**Teorema 1.25**

| | |
|---|---|
| *idempotenza* | $P \vee P \equiv P$<br>$P \wedge P \equiv P$ |
| *commutatività* | $P \vee Q \equiv Q \vee P$<br>$P \wedge Q \equiv Q \wedge P$ |
| *associatività* | $(P \vee Q) \vee R \equiv P \vee (Q \vee R)$<br>$(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R)$ |
| *assorbimento* | $P \vee (P \wedge Q) \equiv P$<br>$P \wedge (P \vee Q) \equiv P$ |
| *distributività* | $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$<br>$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$ |
| *leggi di De Morgan* | $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$<br>$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$ |
| *doppia negazione* | $\neg\neg P \equiv P$ |

*Dimostrazione.* Tutte le equivalenze precedenti possono essere facilmente provate utilizzando la definizione di interpretazione semantica della logica proposizionale. Mostriamo, a titolo esemplificativo, una delle leggi di De Morgan:

$$\begin{aligned} v(\neg(P \vee Q)) = 1 &\iff v(P \vee Q) = 0 \\ &\iff v(P) = v(Q) = 0 \\ &\iff v(\neg P) = v(\neg Q) = 1 \\ &\iff v(\neg P \wedge \neg Q) = 1 \end{aligned}$$

Dunque, per ogni $v$, $v(\neg(P \vee Q)) = v(\neg P \wedge \neg Q)$, quindi per definizione di equivalenza semantica $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$.
Si possono ottenere dimostrazioni alternative utilizzando le tavole di verità. □

### 1.4.5 Completezza funzionale

Ogni fbf $P$ con proposizioni atomiche $A_1, \ldots, A_n$ definisce una funzione di verità $f_P : \{0,1\}^n \to \{0,1\}$ che associa ad ogni attribuzione di valori di verità per $A_1, \ldots, A_n$ il valore di verità di $P$. Formalmente:

**Definizione 1.26** *Sia $P$ una fbf contenente esattamente $n$ proposizioni atomiche distinte $A_1, \ldots, A_n$; la funzione $f_P : \{0,1\}^n \to \{0,1\}$ tale che $\forall(a_1, \ldots, a_n) \in \{0,1\}$, $f_P(a_1, \ldots, a_n) = v(P)$, dove $v$ è una interpretazione t.c. $v(A_i) = a_i$, $\forall A_i \in P$, è la* funzione di verità *di $P$.*

Allo stesso modo, anche i connettivi logici definiscono delle funzioni di verità, descritte dalle loro tabelle di verità. Se il connettivo è $n$-ario, chiameremo fun-

zione di verità del connettivo la funzione di verità della formula proposizionale ottenuta applicando il connettivo ad $n$ proposizioni atomiche distinte.

**Esempio 1.8** La funzione di verità del connettivo $\wedge$ ($f_\wedge$) è $f_{A \wedge B}$.

Concettualmente, ogni funzione $f : \{0,1\}^n \to \{0,1\}$ definisce un qualche connettivo $n$-ario. Ad esempio, se $n = 2$ vi sono $2^{(2^2)} = 16$ funzioni da $\{0,1\}^2$ in $\{0,1\}$. Dunque, in linea di principio, esistono 16 connettivi binari differenti, mentre ne abbiamo definiti solamente 3. Che possiamo dire degli altri connettivi?

**Definizione 1.27** *Dato un insieme di connettivi logici* $\mathbf{C}$*, e un connettivo* $\mathbf{c} \notin \mathbf{C}$*,* $\mathbf{c}$ *si dice (semanticamente)* derivabile *da* $\mathbf{C}$ *se esiste una formula proposizionale* $P$ *costruita con i soli connettivi di* $\mathbf{C}$ *tale che* $f_P = f_{\mathbf{c}}$*.*

In altri termini, un connettivo è derivabile se è possibile definirlo in funzione di altri connettivi.

**Esempio 1.9** La fbf $P = (A \wedge \neg B) \vee (\neg A \wedge B)$ definisce la funzione di verità descritta dalla seguente tavola di verità:

| $A$ | $B$ | $\neg A$ | $\neg B$ | $A \wedge \neg B$ | $\neg A \wedge B$ | $P$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |

Osserviamo che tale funzione risulta vera se e solo se i suoi argomenti hanno differenti valori di verità ($A$ è vera e $B$ è falsa o, viceversa, $A$ è falsa e $B$ è vera); per tale motivo è chiamata "o esclusivo" (corrisponde all' "aut" latino) e si è soliti indicarla con il simbolo $\veebar$. $\veebar$ è dunque un connettivo derivabile da $\{\neg, \wedge, \vee\}$.

**Esempio 1.10** La fbf $Q = (A \to B) \wedge (B \to A)$ definisce la funzione di di verità descritta dalla seguente tavola:

| $A$ | $B$ | $A \to B$ | $B \to A$ | $Q$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Tale funzione risulta vera se e solo se i suoi argomenti ($A$ e $B$) hanno lo stesso valore di verità (o sono entrambi veri o entrambi falsi); viene solitamente indicata con il simbolo $\leftrightarrow$. $\leftrightarrow$ è dunque un connettivo derivabile da $\{\wedge, \to\}$.

Il teorema che segue stabilisce delle utili equivalenze che permettono di definire (derivare) alcuni connettivi in termini di altri.

**Teorema 1.28**

1. $A \to B \equiv \neg A \vee B$
2. $A \vee B \equiv \neg A \to B$
3. $A \vee B \equiv \neg(\neg A \wedge \neg B)$
4. $A \wedge B \equiv \neg(\neg A \vee \neg B)$
5. $A \wedge B \equiv (((A \to \bot) \to \bot) \to (B \to \bot)) \to \bot$
6. $\neg A \equiv A \to \bot$
7. $\bot \equiv A \wedge \neg A$
8. $A \leftrightarrow B \equiv (A \to B) \wedge (B \to A)$
9. $A \underline{\vee} B \equiv (A \wedge \neg B) \vee (\neg A \wedge B)$

*Dimostrazione.* È lasciata al lettore per esercizio. □

**Definizione 1.29** *Un insieme di connettivi logici si dice* funzionalmente completo *se per ogni funzione* $f : \{0,1\}^n \to \{0,1\}$ *esiste una fbf* $P$ *costruita mediante questi t.c.* $f = f_P$.

In altri termini, un insieme di connettivi è funzionalmente completo se ogni altro connettivo è derivabile da essi.
Nel seguito dimostreremo che (Corollario 1.35) gli insiemi $\{\vee, \neg\}$ e $\{\wedge, \neg\}$ sono funzionalmente completi. Sapendo che un dato insieme di connettivi $\mathbf{C}$ è funzionalmente completo, per mostrare che un altro insieme di connettivi $\mathbf{C}$' lo è basta ovviamente dimostrare che ogni connettivo del primo insieme è derivabile nel secondo. L'esistenza di insiemi funzionalmente completi consente di restringere l'analisi a proposizioni espresse in termini di connettivi appartenenti ad uno qualsiasi di tali insiemi. La scelta di uno di questi è per lo più una questione di convenienza. Il vantaggio di utilizzare pochi connettivi è di dover testare un numero limitato di casi per dimostrare le proprietà delle formule costruite a partire da essi; d'altra parte, la definizione di un connettivo in modo derivato tende a nascondere le proprietà intinseche del connettivo stesso rendendone il significato intuitivo assai poco evidente.

**Esempio 1.11** Se consideriamo l'insieme $\{\to, \bot\}$ come primitivo (si veda, a tal proposito, l'esercizio 1.12), la proposizione $A \wedge B$, per il Teorema 1.28, ha la forma $(((A \to \bot) \to \bot) \to (B \to \bot)) \to \bot$ il cui significato risulta meno chiaro rispetto alla forma precedente.

In questo libro abbiamo adottato un compromesso tra coincisione e chiarezza, considerando l'insieme $\{\neg, \vee, \wedge, \to, \bot\}$ come primitivo.

### 1.4.6 Forme Normali

È spesso utile poter trasformare una fbf in un'altra ad essa equivalente che ha una qualche forma canonica prestabilita. Tipicamente, ciò si realizza sostituendo una componente della formula data con altre formule ad essa equivalenti, fino al raggiungimento della forma desiderata. Tale forma canonica è abitualmente detta *normale*, in quanto il procedimento di riscrittura dei sottocomponenti non può essere applicato ulteriormente.
Le forme normali che considereremo sono le cosiddette forme normali congiuntive e disgiuntive.
Una *disgiunzione* di fbf $P_1, P_2 \ldots, P_n$ è una formula del tipo $P_1 \vee P_2 \vee \cdots \vee P_n$ (in virtù dell'associatività del connettivo binario di disgiunzione, ometteremo le parentesi).
Analogamente, la *congiunzione* di $P_1, P_2 \ldots, P_n$ è una formula del tipo $P_1 \wedge P_2 \wedge \cdots \wedge P_n$.

**Definizione 1.30** *Un* letterale *è una proposizione atomica o la sua negazione.*

**Definizione 1.31** *Una fbf $P$ è detta in* forma normale congiuntiva *(fnc) se e solo se $P = P_1 \wedge \cdots \wedge P_n$ con $n \geq 1$, dove $\forall i = 1, \ldots, n$, $P_i$ è una disgiunzione di letterali.*

**Esempio 1.12**

$$A \wedge \neg B \wedge (A \vee C)$$

$$(\neg A \vee B \vee C) \wedge (\neg C \vee A)$$

sono fbf in forma normale congiuntiva.

**Definizione 1.32** *Una fbf $P$ è detta in* forma normale disgiuntiva *(fnd) se e solo se $P = P_1 \vee \cdots \vee P_n$ con $n \geq 1$, dove $\forall i = 1, \ldots, n$, $P_i$ è una congiunzione di letterali.*

**Esempio 1.13**

$$A \vee (\neg B \wedge C)$$

$$(A \wedge B) \vee (C \wedge \neg A) \vee C$$

sono fbf in forma normale disgiuntiva.

**Teorema 1.33** *Per ogni fbf $P$ esistono una forma normale congiuntiva $P^C$ ed una forma normale disgiuntiva $P^D$, tali che $P \equiv P^C$ e $P \equiv P^D$.*

*Dimostrazione.* Per passare da una fbf $P$ ad una forma normale congiuntiva o disgiuntiva equivalente ad essa è sufficiente utilizzare le equivalenze dimostrate in precedenza. Lo schema della procedura di trasformazione è il seguente:

**Passo 1** si eliminano i connettivi diversi da $\wedge, \vee, \neg$ utilizzando il Teorema 1.28,

**Passo 2** si utilizzano ripetutamente la legge della doppia negazione e le leggi di De Morgan (Teorema 1.25) per portare i simboli di negazione immediatamente davanti alle proposizioni atomiche,

**Passo 3** si utilizza più volte la distributività per convertire la fbf $P$ in congiunzioni di digiunzioni (per trovare $P^C$) o in disgiunzioni di congiunzioni (per trovare $P^D$).

□

**Esempio 1.14** Cerchiamo le forme normali disgiuntiva e congiuntiva equivalenti a $(A \vee \neg B) \rightarrow C$.

$$\begin{array}{rcl} (A \vee \neg B) \rightarrow C & \equiv & \neg(A \vee \neg B) \vee C \\ & \equiv & (\neg A \wedge \neg(\neg B)) \vee C \\ & \equiv & (\neg A \wedge B) \vee C \end{array}$$

$(\neg A \wedge B) \vee C$ è in forma normale disgiuntiva.

La forma normale congiuntiva è, invece, $\neg A \vee C \wedge (B \vee C)$.

Sia $P$ una fbf contenente le proposizioni atomiche atomiche $A_1, \ldots A_n$, è possibile trovare la forma normale disgiuntiva ($P^D$) ad essa equivalente utilizzando il seguente metodo:

- si costruisce la tavola di verità di $P$,
- ogni linea di essa che ha valore di verità 1 dà luogo ad una congiunzione i cui letterali sono determinati nel modo seguente: se nell'interpretazione $v$ che corrisponde a quella linea risulta $v(A_i) = 1$, allora viene inserito $A_i$ come letterale, altrimenti $\neg A_i$.

Per ottenere la forma normale congiuntiva ($P^C$) basta scambiare tra loro, nella procedura precedente, i ruoli di 0 ed 1 e quelli della disgiunzione e della congiunzione.

**Esempio 1.15** Data la formula $P$ descritta dalla seguente tavola di verità:

| $A$ | $B$ | $C$ | $P$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

- $P^D = (\neg A \wedge \neg B \wedge \neg C) \vee (A \wedge \neg B \wedge \neg C) \vee (A \wedge \neg B \wedge C)$

- $P^C = (A \vee B \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (A \vee \neg B \vee \neg C) \wedge$
  $(\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg B \vee \neg C)$.

Osserviamo che le formule prodotte con tale metodo ($P^C$ e $P^D$) non sono necessariamente quelle più brevi.

**Corollario 1.34** *L'insieme di connettivi* $\{\neg, \wedge, \vee\}$ *è funzionalmente completo.*

**Corollario 1.35** *Gli insiemi di connettivi* $\{\neg, \vee\}$ *e* $\{\neg, \wedge\}$ *sono funzionalmente completi.*

*Dimostrazione.* Segue dal corollario precedente essendo $\wedge$ derivabile da $\{\neg, \vee\}$ in quanto $A \wedge B \equiv \neg(\neg A \vee \neg B)$ e, viceversa, $\vee$ derivabile da $\{\neg, \wedge\}$ . $\square$

### 1.4.7 Dualità

Consideriamo i valori di verità di $\vee$ e $\wedge$:

$$A \wedge B \text{ è 1 se e solo se } A = B = 1$$
$$A \vee B \text{ è 0 se e solo se } A = B = 0$$

questi sono duali nel senso che entrambi derivano dal rovesciamento dei ruoli di 0 e 1 nell'altro. Vediamo di precisare meglio questa "dualità" tra i suddetti connettivi. Per fare ciò definiamo una trasformazione $^\perp$ tra formule ben formate nel seguente modo:

**Definizione 1.36** $^\perp : FBF \to FBF$ *soddisfa:*

1. $P^\perp = \neg P$ *se* $P$ *è una proposizione atomica*
2. $(P \wedge Q)^\perp = P^\perp \vee Q^\perp$
3. $(P \vee Q)^\perp = P^\perp \wedge Q^\perp$
4. $(\neg P)^\perp = \neg P^\perp$

**Esempio 1.16**

$$\begin{array}{rcl} ((A \wedge \neg B) \vee C)^\perp & = & (A \wedge \neg B)^\perp \wedge C^\perp \\ & = & (A^\perp \vee (\neg B)^\perp) \wedge \neg C \\ & = & (\neg A \vee \neg B^\perp) \wedge \neg C \\ & = & (\neg A \vee \neg\neg B) \wedge \neg C \\ & \equiv & (\neg A \vee B) \wedge \neg C \end{array}$$

La trasformazione $^\perp$ ha il seguente effetto:

**Lemma 1.37** $P^\perp \equiv \neg P$.

*Dimostrazione.* Per definizione di equivalenza semantica, $P^{\perp} \equiv \neg P$ se e solo se $\forall v,\ v(P^{\perp}) = v(\neg P) = 1 - v(P)$. Proviamo la tal cosa per induzione su $P$.
(*caso base*) Se $P$ è una proposizione atomica $v(P^{\perp}) = v(\neg P) = 1 - v(P)$.
Veniamo al caso induttivo:

$$\begin{aligned} v((P \vee Q)^{\perp}) &= v(P^{\perp} \wedge Q^{\perp}) \\ &= min(v(P^{\perp}), v(Q^{\perp})) \\ &= min(1 - v(P), 1 - v(Q)) \\ &= 1 - max(v(P), v(Q)) \\ &= 1 - v(P \vee Q) \end{aligned}$$

I rimanenti casi sono lasciati al lettore come esercizio. □

La trasformazione $^{\perp}$ non risponde pienamente ai nostri scopi. Quello che stiamo cercando è di poter effettivamente interscambiare i connettivi $\wedge$ e $\vee$. A tal fine si consideri una nuova funzione $^{d}$ che chiameremo it funzione di dualità:

**Definizione 1.38** *$^{d} : FBF \to FBF$ è definita nel seguente modo:*

1. *$P^d = P$ se $P$ è una proposizione atomica*
2. *$(P \wedge Q)^d = P^d \vee Q^d$*
3. *$(P \vee Q)^d = P^d \wedge Q^d$*
4. *$(\neg P)^d = \neg P^d$*

Introduciamo ora la nozione di *sostituzione simultanea*:

**Definizione 1.39** *$R[^{S_1,\ldots,S_n}/_{A_1,\ldots,A_n}]$ si ottiene sostituendo $S_i$ al posto di $A_i$ per tutti gli $i \leq n$ simultaneamente.*

**Lemma 1.40** *Per ogni insieme $A_1, \ldots A_n$ di simboli proposizionali atomici che compaiono in una proposizione $R$ si ha:*

$$R^d[^{\neg\neg A_1,\ldots,\neg\neg A_n}/_{A_1,\ldots,A_n}] = R^{\perp}[^{\neg A_1,\ldots,\neg A_n}/_{A_1,\ldots,A_n}]$$

*Dimostrazione.* È lasciata al lettore come esercizio. □

**Teorema 1.41 (Dualità)**
$P \equiv Q \iff P^d \equiv Q^d$.

*Dimostrazione.* Utilizzeremo la trasformazione $^{\perp}$ come passo intermedio. Siano $A_1, \ldots, A_n$ i simboli proposizionali atomici in $P$ e $Q$. Dal Lemma 1.40 e dal teorema di sostituzione segue che $P^d \equiv P^{\perp}[^{\neg A_1,\ldots,\neg A_n}/_{A_1,\ldots,A_n}]$ e $Q^d \equiv Q^{\perp}[^{\neg A_1,\ldots,\neg A_n}/_{A_1,\ldots,A_n}]$. Essendo $P \equiv Q$ risulta $\neg P \equiv \neg Q$, da cui, per il Lemma 1.37, $P^{\perp} \equiv Q^{\perp}$. Quindi

$$P^d \equiv P^{\perp}[^{\neg A_1,\ldots,\neg A_n}/_{A_1,\ldots,A_n}] \equiv Q^{\perp}[^{\neg A_1,\ldots,\neg A_n}/_{A_1,\ldots,A_n}] \equiv Q^d$$

L'inverso deriva immediatamente dal fatto che $P^{dd} = P$. □

**Esempio 1.17** Utilizzando il teorema di dualità è possibile dedurre la seconda legge di De Morgan dalla prima, così come le leggi di idempotenza, associatività e distributività della somma da quelle del prodotto.

Si osservi che in generale è possibile ottenere la tabella di verità della formula $P^d$ semplicemente scambiando ogni 0 in 1 e viceversa, a partire dalla tabella di verità di $P$ (il lettore lo dimostri per esercizio, procedendo per induzione su $P$). Ad esempio, la tabella di verità di $A \wedge B$ è

| $A$ | $B$ | $A \wedge B$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Invertendo 0 e 1 otteniamo la tabella di verità di $(A \wedge B)^d = A \vee B$:

| $A$ | $B$ | $A \vee B$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

Se una tabella di verità definisce un certo connettivo **c**, diremo che la tabella duale definisce il connettivo duale $\mathbf{c}^d$.

**Esempio 1.18** $\wedge$ e $\vee$ sono duali, e $\neg$ è duale di se stesso.

## 1.5 Cenni storici e bibliografici

Benchè esistano innumerevoli anticipazioni del pensiero logico moderno nelle opere di Leibniz (1646-1716) e vari altri studiosi, è solo con i lavori di Boole[10] e De Morgan[11] del 1847 che si hanno i primi significativi sviluppi verso una trattazione matematica formale e coerente di questa materia. L'approccio originario è di natura algebrica: si tratta di un calcolo per operazioni tra classi, esteso in seguito ad un calcolo di relazioni. Tale approccio fu proseguito, ed ulteriormente sviluppato, nei lavori di Peirce, Schröder e Peano.
La procedura di decisione basata su tabelle di verità è utilizzata in modo informale da Frege già nel 1879, ma si deve a Peirce (1885) la formalizzazione del metodo generale (si veda [PeiCP], vol.III, pp.223-225). Questa tecnica fu poi ampiamente utilizzata da Łukasiewicz [Luk20] e Post[Pos21], a cui si deve anche la sua generalizzazione a sistemi logici con più di due valori di verità.

[10] George Boole, *The mathematical analysis of Logic* (1847) e *An investigation of the laws of thought* (1854).
[11] Augustus De Morgan, *Formal Logic* (1847) e *Syllabus of a Proposed System of Logic* (1860).

Per un'introduzione all'argomento si vedano, tra gli altri, [RT58, Urq86]. In generale, logiche a più valori vengono utilizzate per rappresentare e manipolare informazioni incerte. Alcune aree di applicazione di tali logiche all'Informatica sono: Intelligenza Artificiale ([Tur84]), verifica della correttezza dei programmi ([Jon86]) e teoria dei codici correttori adattivi ([Mun92]).
Post ([Pos41]) fu il primo ad affrontare in modo sistematico il problema della completezza funzionale, cioè di insiemi completi di connettivi primitivi indipendenti fra loro.
Osserviamo, per inciso, che l'approccio semantico basato su tabelle di verità, seguito in questo volume, è un caso particolare e piuttosto artificioso di semantica su di una particolare algebra di Boole con esattamente due elementi. Purtroppo, per ragioni di concisione, si è dovuto rinunciare alla trattazione formale delle strutture algebriche sottostanti al calcolo logico (per un approccio semantico di tale tipo il lettore può consultare [Cur63]); il testo di riferimento tradizionale per gli aspetti più prettamente algebrici è l'immensa opera di Birkhoff ([Bir40]).
Le nozioni di forma normale disgiuntiva e congiuntiva sono state introdotte rispettivamente da Schröder (1877) e Peirce (1880), benchè le principali idee siano già presenti nei lavori di Boole.
Il principio di dualità è abitualmente accreditato a Schröder[12].
Il problema della soddisfacibilità per formule proposizionali (SAT) è stato uno dei primi esempi di problema $\mathcal{NP}$-completi introdotto in letteratura ([Coo71]). Karp [Kar72] ha dimostrato la rilevanza computazionale della nozione di $\mathcal{NP}$-completezza, fornendo un gran numero di esempi pratici. Per una ampia rassegna bibliografica su questi aspetti si vedano [HU79] e [GJ79].

## Esercizi

**1.1** Eliminare, quando è possibile, le parentesi nelle seguenti fbf:

1. $((A \wedge B) \rightarrow (\neg C))$
2. $(A \rightarrow (B \rightarrow (\neg C)))$
3. $((A \wedge B) \vee (C \rightarrow D))$
4. $\neg(A \vee ((\neg B) \rightarrow C)))$
5. $(A \rightarrow (B \vee (C \rightarrow D)))$
6. $(\neg((\neg(\neg(\neg A))) \wedge \bot))$
7. $(A \rightarrow (B \wedge ((\neg C) \vee D)))$

**1.2** Provare che ogni formula ben formata contiene lo stesso numero di parentesi aperte ("(") e chiuse (")").
(Suggerimento: si utilizzi il principio di induzione strutturale).

[12] E.Schröder. *Vorlesungen über di Algebra der Logik*, 3 vol. Leipzig, 1890-1905.

**1.3** Verificare, utilizzando le tavole di verità, quali delle seguenti formule ben formate sono delle tautologie:

1. $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$
2. $\neg(A \to \neg A)$
3. $A \vee \neg A$
4. $\bot \to A$
5. $\neg A \to (A \to B)$
6. $(A \wedge B) \wedge (\neg B \vee C)$
7. $A \vee B \to A \wedge B$
8. $(A \to C) \to ((B \to C) \to (A \vee B \to C))$
9. $(A \to B) \to ((B \to \neg C) \to \neg A)$

**1.4** Il seguente insieme di formule

$$A_1 \vee A_2, \neg A_2 \vee \neg A_3, A_3 \vee A_4, \neg A_4 \vee A_5$$

è soddisfacibile?

**1.5** In base alle ipotesi $(a, b, c, d)$ dell'esempio 1.6, è possibile concludere che Carlo è americano?. Si motivi la risposta.

**1.6** Provare che $B \vee C$ è soddisfacibile se e solo se $(B \vee A) \wedge (C \vee \neg A)$ lo è.

**1.7** Dimostrare che il problema della soddisfacibilità (SAT) si può ridurre al problema della 3-soddisfacibilità, cioè che per ogni $P \in FBF$ è possibile costruire una fbf $P'$ che è una congiunzione di disgiunzioni di al più 3 letterali, tale che $P$ è soddisfacibile se e solo se $P'$ lo è.
(Suggerimento: si usi il risultato dell'esercizio precedente per decomporre ricorsivamente ogni disgiunzione di letterali $A_1 \vee A_2 \cdots \vee A_k$, con $k \geq 3$ nella forma $(A_1 \vee A_2 \vee A) \wedge (A_3 \vee \cdots \vee A_k \vee \neg A)$ dove $A$ è una proposizione atomica che non occorre in $A_1, \cdots, A_k$).

**1.8** Stabilire se le seguenti affermazioni sono equivalenti:

1. $A \models B$
2. se $\models A$ allora $\models B$.

**1.9** Provare che:

1. $\bot \vee B \equiv B$
2. $\neg\bot \wedge B \equiv B$

3. $A \models A$
4. $A \models B$ e $B \models C$ implica $A \models C$
5. $\models A \to B$, allora $A \wedge B \equiv A$ e $A \vee B \equiv B$
6. $\models A$ implica $A \wedge B \equiv B$
7. $\models A$ implica $\neg A \vee B \equiv B$
8. se $A \models B$ e $A \models \neg B$, allora $\models \neg A$
9. se $A \models C$ e $B \models C$, allora $A \vee B \models C$.

**1.10** Stabilire se le seguenti affermazioni risultano vere:

1. se $A \models B$ allora $\neg A \models \neg B$
2. se $A \models B$ e $A \wedge B \models C$ allora $A \models C$
3. se $A \vee B \models A \wedge B$.
4. se $A \vee B \models A \wedge B$ allora $A \equiv B$.

**1.11** Sia $\triangle$ il connettivo definito dalla seguente tavola di verità:

| $A$ | $B$ | $A \triangle B$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

Esprimerlo in funzione di $\{\vee, \neg\}$.

**1.12** Provare che l'insieme $\{\to, \bot\}$ è funzionalmente completo sapendo che lo è $\{\wedge, \neg\}$.

**1.13** Sia $\sharp$ il connettivo ternario espresso dalla seguente funzione di verità:

$$f(a_1, a_2, a_3) = 1 \Longleftrightarrow a_1 + a_2 + a_3 \geq 2$$

Esprimerlo in funzione di $\{\wedge, \neg\}$.

**1.14** Sia $\boxdot$ il connettivo definito dalla seguente tavola di verità:

| $A$ | $B$ | $A \boxdot B$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

1. Esprimerlo in funzione di $\{\neg, \wedge\}$ e $\{\neg, \vee\}$.

2. Provare che, per ogni connettivo n-ario, esiste una formula associata ad esso contenente solo il connettivo $\boxdot$.

**1.15** Mostrare che l'insieme di connettivi $\{\wedge, \leftrightarrow, \veebar\}$ è funzionalmente completo e che nessuno dei suoi sottoinsiemi propri lo è.

**1.16** Stabilire se gli insiemi di connettivi $\{\wedge, \rightarrow\}$ e $\{\vee, \wedge\}$ sono funzionalmente completi.

**1.17** Trovare le forme normali congiuntive e disgiuntive equivalenti alle seguenti formule:

1. $(A \rightarrow B) \rightarrow (B \rightarrow \neg C)$
2. $\neg(A \rightarrow (B \rightarrow \neg C)) \wedge D$
3. $\neg(A \wedge B \wedge (C \rightarrow D))$
4. $A \vee (C \rightarrow \neg(D \wedge B))$
5. $\neg(A \leftrightarrow B)$

**1.18** Data la fbf $P$ descritta dalla seguente tavola di verità:

| $A$ | $B$ | $C$ | $P$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

trovare $P^D$ ed $P^C$.

**1.19** Trovare un criterio affinché una formula in fnc sia una tautologia. Fare altrettanto per una fnd.

**1.20** Provare che i due enunciati del teorema di compattezza sono equivalenti.

**1.21** Un grafo si dice *k-colorabile* ($k \in \mathcal{N}$) se esiste una partizione dei suoi vertici in $k$ insiemi distinti $V_1, \cdots, V_k$ tali che due vertici adiacenti non appartengono allo stesso insieme $V_i$. Provare che un grafo è k-colorabile se e solo se ogni suo sottografo finito lo è (cfr. [KK67]).
(Suggerimento: Considerare l'insieme di formule proposizionali su $\{1, \cdots, k\} \times V$

- $(i, a) \rightarrow \neg(j, a)$ per tutti gli $i$, $j \leq k$, $i \neq j$ e per ogni $a \in V$
- $(1, a) \vee (2, a) \vee \cdots \vee (k, a)$

- $(i, a) \to \neg(i, b)$ per tutti gli $a, b \in V$ adiacenti

e porre $v(i, a) = 1$ se e solo se $a \in V_i$).

**1.22** Si definisca il connettivo $\odot$ duale del connettivo di implicazione $\to$, dandone la tabella di verità.

1. Si esprima $\odot$ in funzione di $\{\neg, \wedge\}$.

2. Si estenda la funzione $(\ )^{\perp}$ a formule composte anche dai connettivi $\odot$ e $\to$, e si dimostri che per ogni $P$, $v(P^{\perp}) = 1 - v(P)$.

# Capitolo 2

# Sistemi Deduttivi

Si consideri la seguente "proposizione":

$(*)$ Se $A \to B$ e $A \to (B \to C)$ allora $A \to C$.

è possibile mostrare che è una tautologia costruendo la tabella di verità della formula corrispondente, cioè:

$$((A \to B) \wedge (A \to (B \to C))) \to (A \to C)$$

Tuttavia, se si incontrasse la proposizione $(*)$ in un qualche testo matematico, ci si aspetterebbe piuttosto di trovare una *dimostrazione* del seguente tipo:

*Dimostrazione.* Supponiamo che

2

1. $A \to B$
2. $A \to (B \to C)$

Vogliamo dimostrare che $A \to C$. Questo significa provare che se vale $A$ allora deve valere $C$. Supponiamo dunque che $A$ sia vera. Da $A$ e 1. possiamo concludere $B$. Da $A$ e 2. possiamo concludere $B \to C$. Infine, da $B$ e $B \to C$ concludiamo $C$. □

Il problema principale della logica formale è quello di definire dei sistemi di calcolo che permettano di sviluppare delle dimostrazioni del tipo precedente, cioè come una sequenza di passi elementari che, partendo dalle premesse, consenta di ottenere la conclusione.
Data inoltre una nozione semantica di *validità* (Definizione 1.9) delle formule logiche, ci si aspetta che il sistema formale sia *corretto* nel senso che non permetta di inferire proposizioni non valide (o, eliminando la doppia negazione, permetta di inferire solo proposizioni valide). Correlato a questo problema è quello della *completezza* del sistema formale: sapendo che il sistema è corretto, ci si chiede se ogni formula valida sia dimostrabile al suo interno.

In questo capitolo affronteremo il problema della definizione sintattica dei calcoli logici, introducendo alcuni dei principali sistemi formali di logica simbolica: Deduzione Naturale, Sistemi Assiomatici e Calcolo dei Sequenti.
La correttezza e completezza di questi saranno discusse nel capitolo seguente.

## 2.1 Proprietà intuitive dei sistemi deduttivi

Cerchiamo di individuare le proprietà generali che ci si aspetta siano soddisfatte da un sistema di calcolo.
Useremo in questa sezione la notazione

$$A_1, A_2, \ldots, A_n \vdash B$$

in modo informale, per indicare che nel sistema in oggetto siamo in grado di produrre una dimostrazione della conclusione $B$ a partire da un insieme (eventualmente vuoto) di premesse $A_1, A_2, \ldots, A_n$ (nel seguito useremo le lettere greche $\Gamma, \Delta, \ldots$ per indicare insiemi finiti di fbf).
Per alcuni connettivi logici è piuttosto semplice definire le proprietà attese dal sistema formale.
Consideriamo, ad esempio, il caso della congiunzione. Avendo la premessa $A \wedge B$ ci si aspetta di poter concludere sia $A$ che $B$. Viceversa, avendo entrambe le premesse $A$ e $B$ ci si aspetta di poter concludere $A \wedge B$.
Nella notazione precedente, il sistema formale deve dunque soddisfare le seguenti regole:

$$(\wedge e.1) \quad A \wedge B \vdash A$$

$$(\wedge e.2) \quad A \wedge B \vdash B$$

$$(\wedge i) \quad A, B \vdash A \wedge B$$

Si noti la differenza tra le regole $(\wedge e.1), (\wedge e.2)$ da un lato e la regola $(\wedge i)$ dall'altro. Le prime due dicono *cosa si può concludere da* una formula il cui connettivo principale è $\wedge$. Per ovvie ragioni, tali regole sono dette di *eliminazione* del connettivo. La terza regola esprime le condizioni necessarie per *poter concludere* una formula il cui connettivo principale è $\wedge$. Questa viene detta regola di *introduzione* del connettivo. Le lettere $e$ ed $i$ nei nomi che abbiamo associato alle regole sono un riferimento mnemonico a questa suddivisione.
Veniamo al caso della disgiunzione. Le regole di introduzione sono abbastanza ovvie: avendo $A$ si può concludere $A \vee B$ e, analogamente, avendo $B$ si può concludere $A \vee B$. Vi sono quindi le regole seguenti

$$(\vee i.1) \quad A \vdash A \vee B$$

$$(\vee i.2) \quad B \vdash A \vee B$$

Per quanto riguarda l'eliminazione di $\vee$, la situazione è assai meno chiara: che cosa si può *concludere da* una formula del tipo $A \vee B$? Assai poco, sembrerebbe,

dal momento che non è noto quale formula tra $A$ e $B$ sia vera. Supponiamo, tuttavia, che *sia $A$ che $B$* permettano di derivare una formula $C$, supponiamo cioè che $A \vdash C$ e $B \vdash C$ . In tal caso non interessa sapere chi tra $A$ e $B$ sia vera: avendo come premessa $A \vee B$ si sa che una di esse deve essere vera, e siccome entrambe permettono di derivare $C$, allora ci si aspetta che $C$ sia derivabile da $A \vee B$.
Questo ragionamento può essere espresso dalla regola

$$(\vee e) \quad \textit{se } A \vdash C \textit{ e } B \vdash C \textit{ allora } A \vee B \vdash C$$

Si noti che questa ha una struttura più complicata delle precedenti. Nel seguito, regole di tale tipo, che dipendono da alcune ipotesi di deducibilità all'interno del sistema, saranno dette regole *condizionali*. Le rimanenti, *elementari*. Inoltre, se $A \vdash B$ è una ipotesi di una regola condizionale, diremo che $A$ e $B$ sono rispettivamente premesse e conclusioni *sussidiarie* della regola stessa. Ad esempio, in $(\vee e)$, $A$ e $B$ sono premesse sussidiarie, e $C$ è una conclusione sussidiaria.
Consideriamo ora il connettivo di implicazione. Cosa si può concludere da una premessa del tipo $A \to B$? Naturalmente, che se si ha $A$ allora si può derivare $B$. Si ha dunque la seguente regola di eliminazione:

$$(\to e) \quad A, A \to B \vdash B$$

Questa è comunemente nota con il nome di *modus ponens*.
Viceversa, quando è possibile concludere $A \to B$? Anche in questo caso la risposta è naturale: quando si può derivare $B$ da una premessa $A$, che porta alla seguente regola condizionale di introduzione:

$$(\to i) \quad \textit{se } A \vdash B \textit{ allora } \vdash A \to B$$

Veniamo ora al connettivo di negazione, che è di gran lunga il più problematico dal punto di vista logico. Intuitivamente, la negazione di una formula $A$ è una nuova formula che esprime la falsità di $A$. Dal punto di vista di un sistema deduttivo, una formula è vera se si è in grado di fornirne una dimostrazione all'interno di esso, cioè se $\vdash A$. Questo indurrebbe a ritenere che $A$ sia falsa quando *non esiste* una dimostrazione di $A$, cioè $\nvdash A$; avremo quindi $\vdash \neg A$ se e solo se $\nvdash A$. Questa nozione di falsità, che chiameremo *non dimostrabilità*, è sicuramente accettabile da un punto di vista platonico, ma assai problematica dal punto di vista formale. Notiamo innanzitutto che essa sarebbe curiosamente "instabile": una proposizione vera (dimostrabile nel sistema) resta tale sotto ogni ipotesi aggiuntiva; ovviamente questo non è vero per le proposizioni non dimostrabili. Quindi, una formula *falsa* potrebbe diventare vera rispetto ad estensioni del sistema formale.
Un secondo problema è costituito dal fatto che, ragionevolmente, ci si attende che il sistema logico sia chiuso per istanziazioni, cioè che $\vdash A$ implichi $\vdash A'$ per ogni istanza $A'$ di $A$. Se non vogliamo cadere in ovvie contraddizioni, dovremmo allora intendere $\nvdash A$ nel senso che *nessuna istanza* di $A$ è dimostrabile, o anche

che nessuna formula con "la struttura" di $A$ è dimostrabile[1].
Tuttavia il problema principale legato alla nozione di falsità come *non dimostrabilità* è che nella teoria che abbiamo sviluppato fino ad ora non si ha a disposizione alcuno strumento (interno alla teoria) per parlare di "non dimostrabilità" (ovvero, non abbiamo considerato un sistema formale per $\nvdash$). Non solo: non è neppure evidente che una tale teoria debba necessariamente esistere, cioè che si possa catturare in modo effettivo il significato "platonico" della nozione di falsità (questo è ad esempio il caso, come vedremo, della logica del primo ordine).
A causa di tutte queste difficoltà, affronteremo il problema della formalizzazione delle regole logiche per il connettivo di negazione in modo differente.
Diciamo che una teoria formale è *inconsistente* quando permette di concludere qualunque formula. Una formula $A$ si può allora considerare *assurda* se l'ipotesi di $A$ rende inconsistente la teoria. Possiamo quindi ragionevolmente interpretare la nozione di falsità come assurdità.
Si introduce nel sistema formale una costante $\bot$ che denota inconsistenza (o falsità, nella nostra nuova accezione). Per definizione di $\bot$ si ammetterà nel sistema la regola di eliminazione:

$$(\bot e) \quad \bot \vdash A$$

La negazione è quindi introdotta dalla seguente regola condizionale:

$$(\neg i) \quad se \;\; A \vdash \bot \;\; allora \;\; \vdash \neg A$$

Ci si aspetta inoltre che la regola precedente sia "invertibile", nel senso che, avendo $A$, l'ipotesi aggiuntiva di $\neg A$ renda inconsistente il sistema. Questo dà la regola di eliminazione per la negazione:

$$(\neg e) \quad A, \neg A \vdash \bot$$

Esiste ancora una regola concernente la negazione che si potrebbe ragionevolmente aggiungere al sistema: se una formula $B$ è dimostrabile assumendo $A$, e la stessa formula è dimostrabile assumendo $\neg A$, allora l'ipotesi di $A$ o $\neg A$ sembrerebbe superflua e potremmo concludere che $B$ è comunque dimostrabile:

$$se \;\; \neg A \vdash B \;\; e \;\; A \vdash B \;\; allora \;\; \vdash B$$

Si osservi che grazie a $(\vee e)$ questa regola è equivalente all'assunzione $\vdash (A \vee \neg A)$ (*terzium non datur*). Ora, tale assunzione è del tutto ragionevole dal punto di vista della interpretazione semantica della nozione di "verità", ma lo è assai meno se si interpreta "verità" nel senso di dimostrabilità: intuitivamente, nulla ci dice che una dimostrazione di $A$ o di $\neg A$ debba necessariamente esistere.

[1] Supponiamo che il sistema sia chiuso per istanziazioni. Se esso permettesse di provare $\vdash A$, con $A$ variabile proposizionale, allora ogni formula sarebbe dimostrabile, ed il sistema sarebbe triviale. Supponiamo dunque $\nvdash A$. Allora, platonicamente, $\vdash \neg A$. Ma ancora siamo liberi di istanziare $A$, ottenendo ad esempio $\vdash \neg(A \to A)$ che è intuitivamente scorretto.

Per approfondire il problema, consideriamo alcune formulazioni alternative della regola precedente.
Nel caso particolare in cui $B \equiv A$, questa si può riscrivere nel seguente modo (nota come regola di Peirce):

$$se \ \ \neg A \vdash A \ \ allora \ \ \vdash A$$

Ma se vale $(\neg e)$, da $\vdash A$ si può concludere $\neg A \vdash \bot$. Viceversa, se vale $(\bot e)$, $\neg A \vdash \bot$ implica $\neg A \vdash A$. Dunque, se il sistema formale soddisfa $(\neg e)$ e $(\bot i)$, l'ipotesi $\neg A \vdash A$ è equivalente a $\neg A \vdash \bot$, e la regola di Peirce può essere riscritta nel modo seguente:

$$(RAA) \quad se \ \ \neg A \vdash \bot \ \ allora \ \ \vdash A$$

Questa è la nota legge di *reductio ad absurdum*, che è il principio delle prove per contraddizione: per "dimostrare" $A$ si fa vedere che la sua negazione conduce ad un assurdo. Questo passaggio è ragionevolmente sospetto: ciò che si può dire è che, per una ragione essenzialmente "semantica" (classicamente intesa ma, questa, altamente intuitiva), e quindi trascendente dal sistema formale, $A$ deve essere "valida". Tuttavia si può ragionevolmente concludere di avere in questo modo una *dimostrazione* di $A$?
La questione non è trascurabile. I sistemi logici che si ottengono assumendo o rifiutando la regola (RAA) o quelle ad essa equivalenti sono profondamente diversi. I primi sono detti sistemi classici, e saranno quelli che tratteremo in questo testo. I secondi sono noti come *sistemi intuizionisti*[2]. La differenza tra sistemi classici ed intuizionisti e la rilevanza matematica di questi ultimi si apprezza maggiormente a livello di calcolo dei predicati, dove ritorneremo brevemente sull'argomento. Per quel che concerne la logica proposizionale, la differenza più evidente è che nei sistemi intuizionisti, ogni volta che si ha una dimostrazione di $A \vee B$, questo significa che si ha o una dimostrazione di $A$ o una di $B$, che appunto esprime la natura *costruttiva* del connettivo di disgiunzione. Ovviamente tale proprietà non vale nei sistemi classici: si pensi a $\vdash A \vee \neg A$: in generale, nè $A$ nè $\neg A$ è dimostrabile.
Esiste ancora una regola importante che deve essere soddisfatta dal sistema deduttivo. In particolare, ci si aspetta che le dimostrazioni siano *componibili*: se da un insieme di ipotesi $\Gamma$ si deduce $A$, e da un insieme di ipotesi $\Gamma, A$ si deduce $B$, la composizione di queste due dimostrazioni deve risultare in una dimostrazione di $B$ dalle ipotesi $\Gamma$. Si noti che tale regola esprime una delle caratteristiche più naturali dei procedimenti dimostrativi: per provare un teorema complesso $B$, si può cominciare con il dimostrare dei lemmi più semplici ($A$, in questo caso), e poi provare $B$ utilizzando questi.
La regola precedente è nota in letteratura come *regola di taglio*[3]; nella nostra notazione può essere espressa nel modo seguente:

$$(taglio) \quad se \ \ \Gamma \vdash A \ \ e \ \ \Gamma, A \vdash B \ \ allora \ \ \Gamma \vdash B$$

[2] I sistemi intuizionisti, sono spesso chiamati *sistemi costruttivi*.
[3] *Cut rule*, in inglese. Il nome originale, utilizzato dal logico tedesco Gentzen è *schnitt*.

## 2.2 La Deduzione Naturale

Fino ad ora abbiamo semplicemente considerato un insieme di proprietà auspicabili per un sistema deduttivo che voglia modellare un procedimento di dimostrazione. Affrontiamo ora il problema di dare una definizione formale di tale sistema sotto forma di un calcolo logico.
Studieremo nel seguito varie formalizzazioni possibili. La prima che prendiamo in esame, introdotta da Gentzen [Gen34, Gen55], è nota con il nome di *Deduzione Naturale*[4].
Consideriamo innanzitutto le regole *elementari* del paragrafo precedente. L'idea di Gentzen è quella di dare una semplice rappresentazione di esse, scrivendo le premesse al di sopra di una linea orizzontale, e la conclusione al di sotto di questa. La linea orizzontale, che svolge il ruolo di $\vdash$, rappresenta dunque un passo elementare di inferenza.
In questa notazione, abbiamo le seguenti regole:

$$(\wedge e.1)\ \frac{A \wedge B}{A} \qquad (\wedge e.2)\ \frac{A \wedge B}{B} \qquad (\wedge i)\ \frac{A \quad B}{A \wedge B}$$

$$(\vee i.1)\ \frac{A}{A \vee B} \qquad (\vee i.2)\ \frac{B}{A \vee B} \qquad (\rightarrow e)\ \frac{A \quad A \rightarrow B}{B}$$

$$(\bot e)\ \frac{\bot}{A} \qquad (\neg e)\ \frac{A \quad \neg A}{\bot}$$

L'idea è che, qualora la conclusione di una regola coincide con la premessa di un'altra, queste possono essere composte, dando luogo ad una dimostrazione che ha la forma di albero. Si noti che tale tipo di composizione è giustificata dalla regola di taglio.
Se l'*insieme* delle foglie dell'albero è *contenuto* in $\Gamma$ e la radice è $C$, l'albero rappresenterà una dimostrazione di $\Gamma \vdash C$.

**Esempio 2.1** L'albero

$$\dfrac{\dfrac{\dfrac{A \wedge B}{B} \qquad \dfrac{A \wedge B}{A}}{B \wedge A} \qquad B \wedge A \rightarrow C}{\dfrac{C}{C \vee D}}$$

rappresenta una dimostrazione di $A \wedge B, B \wedge A \rightarrow C \vdash C \vee D$.

[4] Un classico riferimento bibliografico per la Deduzione Naturale è il libro di Prawitz [Pra65]; per una semplice introduzione a questo sistema di calcolo si veda [VDa80].

Veniamo al caso delle regole condizionali, cioè regole che dipendono da alcune ipotesi del tipo $\Gamma \vdash C$. La soluzione di Gentzen consiste nello scrivere le premesse sussidiarie ($\Gamma$) tra parentesi al di sopra della conclusione sussidiaria ($C$) che a sua volta diventa premessa della regola principale.
Avremo dunque le seguenti regole:

$$(\vee e)\ \frac{A \vee B \qquad \begin{matrix}[A]\\ C\end{matrix} \qquad \begin{matrix}[B]\\ C\end{matrix}}{C} \qquad\qquad (\rightarrow i)\ \frac{\begin{matrix}[A]\\ B\end{matrix}}{A \rightarrow B}$$

$$(\neg i)\ \frac{\begin{matrix}[A]\\ \bot\end{matrix}}{\neg A} \qquad\qquad (RAA)\ \frac{\begin{matrix}[\neg A]\\ \bot\end{matrix}}{A}$$

L'idea è che, durante l'applicazione di una di queste regole, ogni ipotesi $A$ che compare tra parentesi nella regola può essere *cancellata* nel sottoalbero la cui radice è la formula sottostante ad $[A]$ nella regola stessa (l'ipotesi cancellata sarà rappresentata tra parentesi quadre anche negli alberi di prova). Un albero di derivazione di radice $C$ il cui insieme di foglie *non cancellate* è *contenuto* in $\Gamma$ costituisce una prova di $\Gamma \vdash C$.
Consideriamo qualche esempio. Da $A$ posso derivare $B \rightarrow A$ mediante una applicazione di $(\rightarrow i)$.

$$\frac{A}{B \rightarrow A}$$

La regola consentirebbe inoltre di cancellare eventuali premesse (foglie) di tipo $B$ nel sottoalbero di radice $A$. Poiché in questo caso non ve ne sono (il sottoalbero è la foglia $A$ stessa), non si opera nessuna cancellazione. Continuiamo la derivazione con un'altra applicazione di $(\rightarrow i)$ per ottenere $A \rightarrow (B \rightarrow A)$. In questo caso, è possibile cancellare la foglia $A$ dal sottoalbero di radice $B \rightarrow A$.

$$\frac{\dfrac{[A]}{B \rightarrow A}}{A \rightarrow (B \rightarrow A)}$$

Si noti che l'albero non contiene foglie non cancellate: esso rappresenta una dimostrazione di $\vdash A \rightarrow (B \rightarrow A)$.
È bene che il lettore si soffermi a riflettere sul ruolo della cancellazione. Consideriamo, per fissare le idee, l'esempio precedente. Avendo $A$ come ipotesi è stata fornita una dimostrazione di $B \rightarrow A$. Ovviamente, per poter concludere $A \rightarrow (B \rightarrow A)$, non si ha più bisogno dell'ipotesi $A$, poiché, per così dire, è stata incorporata all'interno dell'asserto (che è appunto la funzione delle formule condizionali o implicative). Si noti anche, tuttavia, che non sarebbe scorretto mantenere l'ipotesi $A$: la regola di cancellazione *permette*, ma non *obbliga*, di cancellare le formule che appaiono tra parentesi quadre nello schema di regola.
Vediamo qualche altro esempio.

**Esempio 2.2** Proviamo che $\vdash (A \to (B \to C)) \to (A \wedge B \to C)$

$$\dfrac{\dfrac{\dfrac{[A \wedge B]}{B} \qquad \dfrac{\dfrac{[A \wedge B]}{A} \qquad [A \to (B \to C)]}{B \to C}}{C}}{\dfrac{A \wedge B \to C}{(A \to (B \to C)) \to (A \wedge B \to C)}}$$

Osserviamo che tutte le ipotesi sono cancellate. In particolare, le due occorrenze dell'ipotesi $A \wedge B$ sono cancellate durante l'introduzione di $A \wedge B \to C$, e l'ipotesi $A \to (B \to C)$ è cancellata durante l'introduzione di $(A \to (B \to C)) \to (A \wedge B \to C)$.

La Deduzione Naturale è un formalismo semplice ed elegante per la *rappresentazione* di dimostrazioni, ma piuttosto difficile da usare nella *ricerca* di queste. Vediamo, a titolo di esempio, come si potrebbe cercare di *costruire* l'albero di derivazione dell'esempio precedente. L'obiettivo è quello di dimostrare $(A \to (B \to C)) \to (A \wedge B \to C)$. Poiché il connettivo principale di questa formula è una implicazione, ci si può aspettare che l'ultima regola applicata (dal basso) sia stata $(\to i)$ (si noti tuttavia che questa non è la sola possibilità: ad esempio la regola "corretta" potrebbe essere, a priori, una regola di riduzione ad assurdo). Seguendo tale strategia, dobbiamo ora cercare una dimostrazione di $A \wedge B \to C$ avendo a disposizione $A \to (B \to C)$ come ipotesi. Ancora, questo può ridursi a cercare di dimostrare $C$ dalle ipotesi $A \to (B \to C)$ e $A \wedge B$. Ma ciò è semplice: avendo $A \wedge B$ si possono derivare sia $A$ che $B$, e dunque, mediante due applicazioni della regola $(\to i)$, si ottiene $C$ da $A \to (B \to C)$.

**Esempio 2.3** Proviamo che $A \to \neg\neg A$. Infatti

$$\dfrac{\dfrac{[A] \qquad [\neg A]}{\bot}}{\dfrac{\neg\neg A}{A \to \neg\neg A}}$$

Le ipotesi $\neg A$ e $A$ sono cancellate, rispettivamente con l'applicazione delle regole $(\neg i)$ e $(\to i)$.

**Esempio 2.4** Proviamo che $\vdash (A \to C) \to ((B \to C) \to ((A \vee B) \to C))$

$$\dfrac{\dfrac{\dfrac{\dfrac{[A \vee B] \qquad \dfrac{[A] \quad [A \to C]}{C} \qquad \dfrac{[B] \quad [B \to C]}{C}}{C}}{A \vee B \to C}}{(B \to C) \to ((A \vee B) \to C)}}{(A \to C) \to ((B \to C) \to ((A \vee B) \to C))}$$

Le ipotesi $A$ e $B$ sono cancellate applicando $(\vee e)$, mentre $A \vee B, B \to C$ e $A \to C$ sono cancellate, rispettivamente, durante l'introduzione di $A \vee B \to C$, $(B \to C) \to ((A \vee B) \to C)$ e $(A \to C) \to ((B \to C) \to ((A \vee B) \to C))$.

Invitiamo il lettore a cercare le dimostrazioni in Deduzione Naturale di alcune semplici tautologie logiche: vedrà che in molti casi è assai meno facile di quanto si possa immaginare.

Si osservi che la notazione che è stata adottata per la rappresentazione delle dimostrazioni, ed in particolare per la cancellazione delle ipotesi, è *ambigua*. Consideriamo, ad esempio, la seguente dimostrazione di $\vdash A \to (A \to A)$

$$\dfrac{\dfrac{[A]}{A \to A}}{A \to (A \to A)}$$

Non è chiaro se l'ipotesi $A$ sia stata cancellata durante la *prima* o la *seconda* applicazione di $(\to i)$. Questa ambiguità non è rilevante se si è semplicemente interessati al problema dell'esistenza di una dimostrazione. Il caso è differente se si è anche interessati alla dimostrazione in se stessa. Ad esempio, le due dimostrazioni precedenti di $\vdash A \to (A \to A)$ sono concettualmente molto diverse: la prima segue lo schema di dimostrazione di $\vdash B \to (A \to A)$, mentre la seconda quello di $\vdash A \to (B \to A)$.
Si noti che le dimostrazioni stesse hanno un ovvio interesse matematico. Se si pensa a un sistema informatico interattivo per il supporto alla dimostrazione o alla verifica automatica di teoremi, e dunque alla memorizzazione di ampie parti di teorie esistenti, risulta evidente che non basta un elenco più o meno strutturato di teoremi, ma si vuole poter rappresentare, unitamente ad essi, anche le loro dimostrazioni, come avviene in qualunque testo matematico. Ovviamente, dimostrazioni differenti dovrebbero avere rappresentazioni differenti.
Quella parte della logica che si occupa in modo preminente delle dimostrazioni, e dei problemi di rappresentazione, equivalenza, normalizzazione, ecc. va appunto sotto il nome di *teoria della dimostrazione*.
Nel caso della Deduzione Naturale, l'ambiguità notazionale evidenziata in precedenza può essere risolta, ad esempio, numerando (o etichettando) in modo univoco tutte le ipotesi, e specificando per ogni applicazione di una regola quali di queste sono eventualmente cancellate. In modo equivalente, è possibile

aggiungere dei puntatori da ogni regola che prevede cancellazione alle ipotesi cancellate durante l'applicazione della regola stessa.

## 2.3 Sistemi Assiomatici

L'idea alla base dei Sistemi Assiomatici[5] è quella di accettare come unica regola di inferenza il *modus ponens*

$$(\rightarrow e) \quad A, A \rightarrow B \vdash B$$

introducendo opportuni *assiomi* (particolari formule logiche di cui si assume la verità) in sostituzione delle altre regole.
Supponiamo per il momento di accettare non solo il modus ponens, ma anche la regola di introduzione del connettivo di implicazione:

$$(\rightarrow i) \quad \textit{se } \ A \vdash B \ \textit{ allora } \ \vdash A \rightarrow B$$

Si noti che la congiunzione di queste due regole implica che

$$A \vdash B \ \textit{ se e solo se } \ \vdash A \rightarrow B$$

Come conseguenza, ogni ipotesi $A \vdash B$ di una regola condizionale può essere rimpiazzata da $\vdash A \rightarrow B$. Ad esempio, la regola

$$(\vee e) \quad \textit{se } \ A \vdash C \ \textit{ e } \ B \vdash C \ \textit{ allora } \ A \vee B \vdash C$$

può essere riscritta nella forma

$$\textit{se } \ \vdash A \rightarrow C \ \textit{ e } \ \vdash B \rightarrow C \ \textit{ allora } \ A \vee B \vdash C$$

Ma a questo punto è possibile aggiungere direttamente le conclusioni sussidiarie come premesse della conclusione principale. Infatti, scrivere che se $A \rightarrow C$ e $B \rightarrow C$ sono dimostrabili, allora anche $A \vee B \vdash C$ lo è, è del tutto equivalente a scrivere che $A \vee B \vdash C$ è dimostrabile a condizione di avere anche le ipotesi $A \rightarrow C$ e $B \rightarrow C$, ovvero

$$A \rightarrow C, B \rightarrow C, A \vee B \vdash C$$

Quindi tutte le regole possono essere espresse in forma elementare.
Inoltre, ogni regola elementare della forma

$$A_1, A_2, \ldots, A_n \vdash B$$

può essere sostituita da

$$\vdash A_1 \rightarrow (A_2 \rightarrow \ldots (A_n \rightarrow B) \ldots)$$

[5] In letteratura, tali sistemi sono anche noti come *Sistemi alla Hilbert*.

Applicando il procedimento precedente alle regole del paragrafo 2.1, si ottengono quindi i seguenti assiomi:

$(\wedge e.1)$ $\quad \vdash (A \wedge B) \to A$
$(\wedge e.2)$ $\quad \vdash (A \wedge B) \to B$
$(\wedge i)$ $\quad \vdash A \to (B \to (A \wedge B))$

$(\vee i.1)$ $\quad \vdash A \to (A \vee B)$
$(\vee i.2)$ $\quad \vdash B \to (A \vee B)$
$(\vee e)$ $\quad \vdash (A \to C) \to ((B \to C) \to ((A \vee B) \to C))$

$(\bot e)$ $\quad \vdash \bot \to A$
$(\neg i)$ $\quad \vdash (A \to \bot) \to \neg A$
$(\neg e)$ $\quad \vdash A \to (\neg A \to \bot)$
$(RAA)$ $\quad \vdash (\neg A \to \bot) \to A$

Vediamo ora di eliminare la regola $(\to i)$. In altri termini, si vogliono trovare degli assiomi tali che il sistema deduttivo così ottenuto soddisfi come *metateorema* la regola

$$se \;\; A \vdash C \;\; allora \;\; \vdash A \to C$$

avendo a disposizione il modus ponens come unica regola di inferenza.
Supponiamo di avere una dimostrazione di $A \vdash C$ nel sistema con solo $(\to e)$, e vediamo che assiomi aggiungere per poter dimostrare $\vdash A \to C$.
Procediamo per induzione sul numero di applicazioni di $(\to e)$ nella dimostrazione di $A \vdash C$.
Se $(\to e)$ non è mai stato utilizzato (caso base dell'induzione), ci sono due possibilità: o $C = A$, oppure $C$ è un assioma. Nel primo caso, basterà aggiungere l'assioma $\vdash A \to A$. Nel secondo, esiste una dimostrazione di $C$, in quanto $C$ è un assioma. Dunque è sufficiente aggiungere il nuovo assioma $\vdash C \to (A \to C)$ per ottenere $\vdash (A \to C)$ con una applicazione di $(\to e)$.
Veniamo al caso induttivo. Supponiamo che la dimostrazione di $A \vdash C$ sia stata ottenuta mediante $n + 1$ applicazioni di modus ponens. Siccome questa è la sola regola di inferenza del sistema, $C$ deve essere la conclusione di una applicazione di modus ponens a partire da due premesse della forma $B \to C$ e $B$. Inoltre, queste due formule devono essere entrambe derivabili da $A$ (altrimenti non avremmo una dimostrazione di $A \vdash C$). Si ha quindi $A \vdash B \to C$ e $A \vdash B$. Queste due dimostrazioni conterranno un numero di applicazioni di modus ponens inferiore o uguale ad $n$. Possiamo dunque applicare l'ipotesi di induzione, concludendo che esistono due dimostrazioni di $\vdash A \to (B \to C)$ e $\vdash (A \to B)$, rispettivamente. Dunque, per ottenere $\vdash A \to C$ basta aggiungere l'assioma $\vdash (A \to (B \to C)) \to ((A \to B) \to (A \to C))$.
Riassumendo, la regola $(\to i)$ diventa derivabile a condizione di assumere i seguenti assiomi:

$(I) \quad \vdash A \to A$
$(K) \quad \vdash A \to (B \to A)$
$(S) \quad \vdash (A \to (B \to C)) \to ((A \to B) \to (A \to C))$

Il risultato precedente può essere generalizzato ad un numero arbitrario di ipotesi $A_1, \ldots, A_{n-1}, A_n$. In particolare, vale il seguente teorema, la cui dimostrazione formale (che si basa sul ragionamento precedente) è lasciata al lettore come esercizio.

**Teorema 2.1 (Deduzione)**
*Se* $A_1, \ldots, A_{n-1}, A_n \vdash B$*, allora* $A_1, \ldots, A_{n-1} \vdash A_n \to B$.

***Osservazione*** Gli assiomi del sistema logico sono in realtà degli *schemi* di assiomi. Questo significa che *ogni loro istanziazione* è un assioma. Ad esempio, $A \to (A \to A)$ è un assioma ottenuto per sostituzione di $A$ al posto di $B$ in $(K)$. Dunque, in realtà si hanno una *infinità* di assiomi: gli schemi sono un modo conveniente per rappresentare questa infinità in modo finitistico.

Anche nei Sistemi Assiomatici possiamo convenientemente rappresentare le dimostrazioni sotto forma di albero. In questo caso, le foglie possono essere ipotesi o assiomi. Un albero di derivazione di radice $B$ le cui foglie che non siano assiomi sono contenute in $\Gamma$ rappresenta una dimostrazione di $\Gamma \vdash B$.
Come esempio di albero di derivazione, mostriamo come l'assioma $(I)$ sia in realtà *ridondante*, nel senso che è derivabile dai rimanenti. Infatti, come caso particolare di $(S)$, sostituendo $A \to A$ al posto di $B$ ed $A$ al posto di $C$ si ha

$$S[^{A\to A}/_B, ^{A}/_C] \equiv (A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$$

Inoltre, come casi particolari di $(K)$ abbiamo

$$K[^{A\to A}/_B] \equiv A \to ((A \to A) \to A)$$

$$K[^{A}/_B] \equiv A \to (A \to A)$$

Dunque, il seguente albero di prova è una dimostrazione di $\vdash A \to A$:

$$\dfrac{\dfrac{S[^{A\to A}/_B, ^{A}/_C] \qquad K[^{A\to A}/_B]}{(A \to (A \to A)) \to (A \to A)} \qquad K[^{A}/_B]}{A \to A}$$

D'ora in avanti, eviteremo di specificare le particolari istanze degli assiomi utilizzate negli alberi di prova, cioè utilizzeremo gli assiomi in modo *polimorfo*. Nel caso dell'esempio precedente avremo dunque semplicemente

$$\dfrac{\dfrac{S \qquad K}{(A \to (A \to A)) \to (A \to A)} \qquad K}{A \to A}$$

È possibile dimostrare che questa notazione non è ambigua, nel senso che una volta fissata la radice dell'albero esiste sempre un modo *più generale* (cioè il meno istanziato possibile) per istanziare opportunamente gli assiomi in modo da ottenere un albero di derivazione abituale (a condizione, ovviamente, che lo "schema" di albero sia corretto).
In generale, è estremamente difficile sviluppare delle dimostrazioni nei Sistemi Assiomatici. Il teorema di deduzione è, in tal senso, uno strumento indispensabile.

**Esempio 2.5** Cerchiamo una dimostrazione di $\vdash A \vee \neg A$ nel calcolo assiomatico. Dimostriamo innanzi tutto che $\neg(A \vee \neg A), A \vdash \bot$:

$$\dfrac{\neg(A \vee \neg A) \qquad \dfrac{\dfrac{A \qquad A \to (A \vee \neg A)}{A \vee \neg A} \qquad A \vee \neg A \to (\neg(A \vee \neg A) \to \bot)}{\neg(A \vee \neg A) \to \bot}}{\bot}$$

In modo analogo, si dimostra che $\neg(A \vee \neg A), \neg A \vdash \bot$. Allora, per il teorema di deduzione, devono esistere due dimostrazioni di $\neg(A \vee \neg A) \vdash A \to \bot$ e $\neg(A \vee \neg A) \vdash \neg A \to \bot$. Da queste, utilizzando rispettivamente gli assiomi $(\neg i)$ e $(RAA)$ si ottengono in un passo di modus ponens le dimostrazioni di $\neg(A \vee \neg A) \vdash \neg A$ e $\neg(A \vee \neg A) \vdash A$. Utilizzando l'assioma $(\neg e)$, con due applicazioni di modus ponens si ricava quindi una dimostrazione di $\neg(A \vee \neg A) \vdash \bot$. Dunque, per il teorema di deduzione, esiste una dimostrazione di $\vdash \neg(A \vee \neg A) \to \bot$. Per modus ponens a partire dall'istanza $(\neg(A \vee \neg A) \to \bot) \to (A \vee \neg A)$ di $(RAA)$, si ottiene infine la dimostrazione cercata di $A \vee \neg A$.

### 2.3.1 Formule e tipi ⋆

Nel caso dei Sistemi Assiomatici si può introdurre una notazione, molto più compatta degli alberi, per rappresentare le dimostrazioni. Si osservi che il modus ponens permette di trasformare le dimostrazioni $t_1$ e $t_2$ di $A \to B$ ed $A$ in una dimostrazione di $B$. È possibile quindi introdurre un operatore binario "$\cdot$" e rappresentare con $(t_1 \cdot t_2)$ la dimostrazione di $B$ così ottenuta. Una interpretazione particolarmente suggestiva di questa rappresentazione è quella di vedere una dimostrazione $t$ di una formula $A$ come un oggetto *di tipo* $A$ (scriveremo $t : A$). La formula $A \to B$ può essere vista come l'insieme delle funzioni da $A$ in $B$, cioè come l'insieme di quelle trasformazioni che permettono di ottenere un oggetto di tipo $B$ (una prova di $B$) a partire da un oggetto di tipo $A$ (una prova di $A$). A questo punto, la regola di modus ponens dice semplicemente che *applicando* una funzione $t_1$ di tipo $A \to B$ ad un oggetto $t_2$ di tipo $A$ si ottiene un oggetto $(t_1 \cdot t_2)$ di tipo $B$.
In tal modo, si può dare una semplice rappresentazione lineare per tutte le dimostrazioni. Gli assiomi saranno rappresentati da particolari *costanti* del tipo opportuno (si possono utilizzare i nomi degli assiomi a questo scopo), mentre

eventuali foglie dell'albero differenti dagli assiomi saranno rappresentate da *variabili* (ancora, del tipo opportuno). Una dimostrazione sarà allora descritta da un qualche *polinomio* sul nostro alfabeto di variabili e costanti, cioè da un qualche termine $M$ costruito a partire da variabili e costanti per mezzo dell'operatore binario di applicazione (modus ponens). In particolare, dato un albero che rappresenta una derivazione $A_1, A_2, \ldots A_n \vdash B$, la stessa dimostrazione può essere rappresentata da un termine $M$ con variabili libere contenute in $x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n$. In questo caso si scriverà

$$x_1 : A_1, \ldots, x_{n-1} : A_{n-1}, x_n : A_n \vdash M : B$$

Ad esempio, utilizzando questa notazione, e sfruttando il polimorfismo degli assiomi, la dimostrazione di $I$ data in precedenza può essere semplicemente rappresentata dal termine chiuso $((S \cdot K) \cdot K)$, (nel seguito ometteremo il punto che denota l'applicazione e scriveremo semplicemente $(t_1\ t_2)$ al posto di $(t_1 \cdot t_2)$).
Analogamente, la dimostrazione di $\neg(A \vee \neg A), A \vdash \bot$ dell'Esempio 2.5, è rappresentata dal termine:

$$x : \neg(A \vee \neg A), y : A \vdash ((\neg e\ (\vee i.1\ y))\ x) : \bot$$

Ovviamente, non tutti i termini di questo linguaggio rappresentano delle dimostrazioni, cosí come non tutti gli alberi sono degli alberi di prova corretti. Riprendendo l'analogia precedente tra formule e tipi, l'ovvia condizione da rispettare è che il termine sia *ben tipato*.
Una interessante conseguenza del fatto di aver espresso le dimostrazioni come termini (di un linguaggio tipato), è il poter esprimere le equivalenze tra dimostrazioni come semplici uguaglianze tra questi termini. Consideriamo ad esempio l'assioma $K : A \to (B \to A)$. Questo può essere visto come una funzione che presi due argomenti $x : A$ e $y : B$ restituisce un risultato $((K\ x)\ y)$ di tipo $A$. Ovviamente, ci si aspetta che tale risultato sia lo stesso ricevuto in ingresso, come appare evidente dalla dimostrazione che daremmo di $K$ in Deduzione Naturale. In altri termini, le due dimostrazioni

$$x : A, y : B \vdash ((K\ x)\ y) : A \quad \text{e}$$

$$x : A, y : B \vdash x : A$$

sono logicamente equivalenti (benché strutturalmente diverse). Questa equivalenza può essere semplicemente espressa mediante l'equazione

$$(*) \qquad ((K\ x)\ y) = x$$

Consideriamo ora il caso di $S : (A \to (B \to C)) \to ((A \to B) \to (A \to C))$. Come descritto dal suo tipo, $S$ può essere visto come una funzione che presi in ingresso gli elementi $x : A \to (B \to C)$, $y : A \to B$ e $z : A$ resituisce un risultato $(((S\ x)\ y)\ z) : C$. Come si ottiene dai dati in ingresso un elemento di tipo $C$?

È necessario applicare $x$ a $z$ per ottenere una funzione $(x\ z) : B \to C$; quindi la si applica a $(y\ z) : B$. Come nel caso precedente, le due dimostrazioni

$$x : A \to (B \to C), y : A \to B, z : A \vdash (((S\ x)\ y)\ z) : C \quad \text{e}$$

$$x : A \to (B \to C), y : A \to B, z : A \vdash ((x\ z)\ (y\ z)) : C$$

sono logicamente equivalenti. Ciò può essere semplicemente espresso dall'equazione

$$(**) \qquad (((S\ x)\ y)\ z) = ((x\ z)\ (y\ z))$$

Il linguaggio di termini considerato in precedenza con l'aggiunta delle regole $(*)$ e $(**)$ è noto sotto il nome di *Logica Combinatoria* [CF58, HS86].
Come utile esercizio, il lettore può divertirsi a cercare altre equivalenze per i rimanenti assiomi del calcolo.

**Proposizione 2.2** *Dato un termine $M$ del linguaggio applicativo precedente, ed una variabile $x$, sia $\lambda^* x.M$ il termine definito induttivamente nel modo seguente:*

1. $\lambda^* x.x = I$;
2. $\lambda^* x.y = (K\ y)$ *per ogni variabile* $y \neq x$;
3. $\lambda^* x.C = (K\ C)$ *per ogni costante* $C$;
4. $\lambda^* x.(M\ N) = (S\ \lambda^* x.M)(\lambda^* x.N)$ .

*Se*

$$x_1 : A_1, \ldots, x_{n-1} : A_{n-1}, x_n : A_n \vdash M : B$$

*allora*

$$x_1 : A_1, \ldots, x_{n-1} : A_{n-1} \vdash \lambda^* x_n.M : A \to B$$

*Dimostrazione.* Si effettua per induzione strutturale su $M$. È lasciata al lettore come esercizio. □

Si noti che questo risultato costituisce una versione effettiva del teorema di deduzione. La trasformazione $\lambda^* x.M$ fornisce un modo algoritmico, anche se un pò lungo e noioso, per "costruire" dimostrazioni nel sistema assiomatico utilizzando il teorema di deduzione.
La $\lambda$-notazione dell'esempio precedente è tratta dal $\lambda$ calcolo[6], un formalismo introdotto da Church negli anni trenta per lo studio delle funzioni calcolabili[7]. Tale formalismo è basato solo sul concetto di applicazione di funzioni, inteso

[6] Il $\lambda$ calcolo costituisce il nucleo dei linguaggi di programmazione funzionali (es. *Lisp, ML, Scheme*).

[7] Storicamente, il $\lambda$ calcolo ha avuto una notevole importanza, in quanto è stata l'equivalenza tra il $\lambda$ calcolo ed il formalismo delle funzioni ricorsive, dimostrata da Kleene negli anni trenta (si veda, a tal proposito, [Odi89]) ad indurre Church a formulare la tesi che porta il suo nome e sulla quale si basa la *Teoria della Calcolabilità*.

nella sua accezione più generale, e senza porre limiti sulla natura degli argomenti che possono essere valori o, a loro volta, funzioni. Data la generalità del calcolo, Church ha introdotto la $\lambda$-notazione (da cui il $\lambda$ calcolo prende nome) per eliminare l'ambiguità esistente tra valori e funzioni. Infatti, supponiamo di scrivere $x$; come si fa a sapere se ci si riferisce alla funzione identità oppure al valore $x$? Nell' "usuale" calcolo matematico, il significato risulta chiaro dal contesto, al contrario, in un calcolo in cui l'operazione fondamentale è quella di applicazione di funzioni e dove non si pone alcun limite sulla natura degli argomenti, è necessario utilizzare una notazione differente per i due concetti. Nel particolare esempio in esame, la notazione introdotta da Church denota con $x$ il valore "$x$" e con $\lambda x.x$ la funzione che, preso un argomento "$x$", restituisce lo stesso. Vediamo ora quali sono gli "oggetti" (termini) di tale formalismo. Un termine del $\lambda$ calcolo è o una variabile, o l'applicazione $(M\ N)$ di due termini $M$ ed $N$, oppure la $\lambda$ *astrazione* $\lambda x.M$ di un termine $M$ rispetto alla variabile $x$. Intuitivamente, $\lambda x.M : A \to B$ è una funzione che associa ad un elemento generico $x : A$ ($x$ è il parametro formale della funzione $\lambda x.M$) il termine $M : B$ (in cui può comparire $x$). Dunque, il risultato della applicazione di $\lambda x.M : A \to B$ ad un dato $N : A$ ($N$, in questo caso, è il parametro attuale) consiste nel termine $M$ dove la variabile $x$ è stata rimpiazzata da $N$, cioè $M[^N/_x]$. Questo è espresso dalla seguente equazione, nota come $\beta$-equivalenza, che costituisce la principale regola della teoria equazionale del $\lambda$ calcolo:

$$(\beta) \qquad (\lambda x.M)N = M[^N/_x]$$

Osserviamo che tale regola costituisce la generalizzazione del concetto di applicazione di una funzione ad un argomento, al caso in cui questo può essere un termine qualsiasi (in particolare, anche una funzione).

**Esempio 2.6** Se si definiscono $K = \lambda x.\lambda y.x$ e $S = \lambda x.\lambda y.\lambda z.((x\ z)\ (y\ z))$ le due equazioni della Logica Combinatoria sono una ovvia conseguenza della regola $\beta$.

**Proposizione 2.3** *La trasformazione $\lambda^*$ definita nella Proposizione 2.2 è una buona codifica della $\lambda$-astrazione in Logica Combinatoria, nel senso che per tutti i termini $M$ ed $N$ di questa logica, $(\lambda^* x.M)N = M[^N/_x]$.*

*Dimostrazione.* Si effettua per induzione strutturale su $M$. La dimostrazione è lasciata al lettore come esercizio. □

Osserviamo che la scelta tra l'assumere la $\lambda$ astrazione come primitiva o come derivata (come nel caso della Logica Combinatoria) corrisponde alla scelta tra assumere o meno il principio di deduzione come principio primitivo della logica. Dunque, il $\lambda$ calcolo corrisponde ad un sistema logico in cui si assumono come primitivi sia il modus ponens che la regola di introduzione della implicazione:

$$(\to i)\ \frac{\begin{array}{c}[x : A]\\ M : B\end{array}}{\lambda x.M : A \to B}$$

In effetti, nel caso dell'implicazione, il $\lambda$ calcolo risolve in modo elegante il problema notazionale di ambiguità, messo in evidenza in precedenza, per le dimostrazioni della Deduzione Naturale (cfr. pagina 43).
La teoria equazionale del $\lambda$ calcolo rispecchia meglio della Logica Combinatoria le equivalenze logiche tra dimostrazioni (corrispondenza che è perfetta limitatamente ai connettivi di congiunzione e implicazione). Per questa ragione, il $\lambda$ calcolo può essere proficuamente utilizzato come linguaggio per la rappresentazione di dimostrazioni.
Purtroppo una trattazione anche solo introduttiva di questo formalismo richiederebbe un libro a parte, ed è assolutamente al di fuori degli intenti del presente volume. Il lettore interessato all'argomento può consultare [Chu41, Bar84].

### 2.3.2 Altri Sistemi Assiomatici

Il Sistema Assiomatico presentato in questo libro è stato tratto da [Cur63]. Esistono infinite altre formulazioni di calcoli logici di questo tipo che differiscono tra loro sia nella scelta degli assiomi che nell'insieme dei connettivi logici considerati come primitivi. Menzioniamo qui di seguito alcuni dei più celebri.
Nel caso in cui il sistema contenga i connettivi di congiunzione e disgiunzione, gli assiomi dati in precedenza sono piuttosto tradizionali; l'unica differenza è che abitualmente si preferisce sostituire ($\wedge i$) con l'assioma equivalente

$$\vdash (A \to B) \to ((A \to C) \to (A \to B \wedge C)$$

per ovvie ragioni di simmetria con ($\vee e$).
Il Sistema Assiomatico di Hilbert e Bernays, contenuto nella loro classica opera *Grundlagen der Mathematik* [HB34] introduce i seguenti assiomi per implicazione e negazione:

1. $A \to (B \to A)$
2. $(A \to (A \to B)) \to (A \to B)$
3. $(A \to B) \to ((B \to C) \to (A \to C))$
4. $(A \to B) \to (\neg B \to \neg A)$
5. $A \to \neg\neg A$
6. $\neg\neg A \to A$

Un celebre esempio di sistema logico che si serve solo dei connettivi di implicazione e negazione ed introduce gli altri per definizione si deve a Łukasiewicz ([Chu56]). Esso consta di tre soli assiomi:

1. $(A \to B) \to ((B \to C) \to (A \to C))$
2. $(\neg A \to A) \to A$
3. $A \to (\neg A \to B)$

Al contrario, Russel e Whitehead, nei *Principia Mathematica* [RW10] hanno scelto, come primitivi, i connettivi di negazione e disgiunzione, proponendo i seguenti assiomi[8]:

1. $\neg(A \vee A) \vee A$
2. $\neg A \vee (B \vee A)$
3. $\neg(A \vee B) \vee (B \vee A)$
4. $\neg(A \vee (B \vee C)) \vee (B \vee (A \vee C))$
5. $\neg(\neg A \vee B) \vee (\neg(C \vee A) \vee (A \vee B))$

In questo caso, la regola di modus ponens è sostituita dalla simile regola di *sostituzione*:

$$\frac{\neg A \vee B \qquad A}{B}$$

## 2.4 Relazione tra ND e H

Nei paragrafi precedenti sono stati introdotti due sistemi formali per rappresentare e sviluppare dimostrazioni. La domanda che sorge naturale è se i due sistemi permettono di ottenere gli stessi risultati, ovvero se l'insieme delle formule derivabili da un insieme di ipotesi $\Gamma$ sia lo stesso per entrambi i sistemi.
Dimostriamo per prima cosa che ogni formula derivabile da $\Gamma$ nel sistema assiomatico presentato nella sezione 2.3, al quale ci riferiremo come $H$, lo è anche in Deduzione Natuale, ovvero

**Teorema 2.4** *$\Gamma \vdash_H A$ implica $\Gamma \vdash_{ND} A$.*

*Dimostrazione.* Osserviamo che le prove in $H$ costituiscono un caso particolare di quelle in Deduzione Naturale. L'unica differenza è che le "foglie" dell'albero di prova possono essere anche assiomi. Dunque, per trasformare una dimostrazione nel sistema $H$ in una dimostrazione in Deduzione Naturale basta espandere gli assiomi nelle loro rispettive dimostrazioni in Deduzione Naturale. In altri termini, è sufficiente provare che ogni assioma di $H$ è dimostrabile in Deduzione Naturale.
Consideriamo l'assioma $A \to (A \to A \wedge B))$. La sua dimostrazione in Deduzione Naturale è

$$\dfrac{\dfrac{\dfrac{[A] \qquad [B]}{A \wedge B}}{B \to (A \wedge B)}}{A \to (B \to (A \wedge B))}$$

[8] Bernays notò, in seguito, che l'assioma 4. è in realtà ridondante.

Consideriamo $S$:

$$\dfrac{\dfrac{\dfrac{[A \to (B \to C)] \qquad [A]}{B \to C} \qquad \dfrac{[A \to B] \qquad [A]}{B}}{C}}{\dfrac{A \to C}{(A \to B) \to (A \to C)}}$$
$$\dfrac{}{(A \to (B \to C)) \to ((A \to B) \to (A \to C))}$$

La ricerca delle dimostrazioni in DN dei rimanenti assiomi è lasciata al lettore come esercizio. □

Veniamo quindi al viceversa, ovvero:

**Teorema 2.5** $\Gamma \vdash_{ND} A$ *implica* $\Gamma \vdash_H A$.

*Dimostrazione.* La dimostrazione si effettua per induzione sulla profondità dell'albero di prova di $\Gamma \vdash_{ND} A$.
(*caso base*) L'albero ha profondità 1, ossia è costituito dalla sola formula $A$. Questo implica che $A$ deve essere contenuta in $\Gamma$, e lo stesso albero è anche una dimostrazione di $\Gamma \vdash_H A$.
Veniamo al passo induttivo. Supponiamo che l'asserto sia vero per tutti gli alberi di prova di $\Gamma \vdash_{ND} A$ di profondità minore o uguale ad $n$, e dimostriamo che è vero anche per gli alberi di profondità $n+1$.
È necessario considerare dei sottocasi a seconda dell'ultima regola applicata nell'albero di derivazione. Vi saranno quindi tanti sottocasi quante sono le regole della Deduzione Naturale.

- Supponiamo che l'ultima regola applicata sia $(\wedge e.1)$. Questo significa che l'albero di prova per $\Gamma \vdash_{ND} A \wedge B$ ha profondità minore o uguale di $n$. Per ipotesi induttiva esiste dunque una dimostrazione di $\Gamma \vdash_H A \wedge B$. Utilizzando l'assioma $\vdash (A \wedge B) \to A$, con una applicazione di modus ponens si ottiene una dimostrazione di $\Gamma \vdash_H A$.

- Supponiamo che l'ultima regola applicata sia $(\vee e)$. Vi sono dunque degli alberi di prova per $\Gamma \vdash_{ND} C \vee D$, $\Gamma, C \vdash_{ND} A$ e $\Gamma, D \vdash_{ND} A$ di profondità minore o uguale ad $n$. Per ipotesi induttiva, risulta $\Gamma \vdash_H C \vee D$, $\Gamma, C \vdash_H A$ e $\Gamma, D \vdash_H A$. Per il teorema di deduzione, si ha $\Gamma \vdash_H (C \to A)$ e $\Gamma \vdash_H (D \to A)$. Utilizzando l'assioma $\vdash (C \to A) \to ((D \to A) \to ((C \vee D) \to A))$, mediante tre applicazioni di modus ponens si ottiene una dimostrazione di $\Gamma \vdash_H A$.

I rimanenti casi sono lasciati al lettore come esercizio. □

## 2.5 IL Calcolo dei Sequenti

Passiamo ora a considerare una terza classe di sistemi logici formali introdotta, come la Deduzione Naturale, da Gentzen [Gen34], e nota come Calcolo dei Sequenti (per delle trattazioni più recenti si vedano [Cur63, Tak75, Gal86, TS96]).

Questo tipo di calcolo differisce dai sistemi precedenti per due aspetti fondamentali:

1. mentre in Deduzione Naturale e nei Sistemi Assiomatici le regole di inferenza si applicano a formule logiche, nel Calcolo dei Sequenti queste si applicano ad asserzioni di derivabilità del tipo $A_1, \ldots, A_n \vdash B$, denominate *sequenti*. Una regola di inferenza in tale calcolo consente di derivare un nuovo sequente in funzione di altri sequenti assunti come premesse. In altre parole, la regola di inferenza esprime cosa si può derivare nel sistema in funzione di precedenti assunzioni di derivabilità (cosa molto simile a quanto è stato fatto nell'introduzione a questo capitolo, discutendo le proprietà che il sistema logico formale deve intuitivamente soddisfare).

2. tutte le regole sono tali che nuovi connettivi logici possono essere solo *introdotti* ma mai *eliminati*. Questo fornisce al sistema una struttura quasi costruttiva che ha molte importanti conseguenze sulla (meta) teoria delle dimostrazioni.

Cominciamo l'analisi del sistema.
Come già detto, un *sequente* è una espressione del tipo [9]

$$A_1, \ldots, A_n \vdash B$$

L'idea intuitiva è che il sequente afferma (ipotizza) la derivabilità logica di $B$ in funzione delle premesse $A_1, \ldots, A_n$.
In generale, useremo le lettere greche $\Delta, \Gamma, \ldots$ per esprimere delle sequenze *finite* di fbf.

**Definizione 2.6** *Una* regola di inferenza *ha la forma*

$$\frac{S_1}{S} \qquad \text{o} \qquad \frac{S_1 \qquad S_2}{S}$$

*dove $S_1, S_2$ ed $S$ sono sequenti. $S_1$ ed $S_2$ sono detti* premesse *della regola, mentre $S$ è detto* conclusione.

Tale regola asserisce che dato $S_1$ (rispettivamente, dati $S_1$ ed $S_2$) si può dedurre $S$. $S_1$ ed $S_2$ sono dette premesse, mentre $S$ è detta conclusione.
Nel Calcolo dei Sequenti, vi sono quattro gruppi di regole: assiomi, cut, regole strutturali e regole logiche. Vediamole in dettaglio.
La prima domanda che ci si pone è la seguente: quando si puó sostenere che un sequente $\Gamma \vdash B$ è *evidentemente* vero? Chiaramente, quando $B$ compare nell'insieme di ipotesi $\Gamma$. Questo conduce al seguente assioma:

$$(Ax) \qquad \Gamma \vdash B \qquad \textit{se } B \in \Gamma$$

[9] Vedremo, nel seguito, che in realtà saremo costretti a considerare una forma più generale di sequente.

La regola di taglio, già discussa a p. 39, ha la seguente forma:

$$(taglio) \quad \frac{\Gamma \vdash A \qquad A, \Gamma' \vdash B}{\Gamma, \Gamma' \vdash B}$$

Veniamo quindi alle *regole strutturali* che consentono di manipolare l'ordine ed il numero delle ipotesi. In particolare, queste stabiliscono che:

1. l'ordine delle ipotesi è irrilevante (regola di permutazione):

$$(perm) \qquad \frac{\Gamma \vdash A}{\Gamma' \vdash A}$$

   se $\Gamma'$ è una permutazione di $\Gamma$.

2. assumere due volte una ipotesi $B$ è equivalente ad assumere $B$ una sola volta (regola di contrazione):

$$(contr) \qquad \frac{\Gamma, B, B \vdash A}{\Gamma, B \vdash A}$$

3. è sempre possibile *aggiungere* nuove ipotesi (regola di indebolimento):

$$(indeb) \qquad \frac{\Gamma \vdash A}{\Gamma, B \vdash A}$$

Si noti che in virtù di tale regola, l'assioma può essere espresso nella seguente forma, più debole della precedente:

$$(Ax) \qquad A \vdash A$$

Le regole logiche sono quelle concernenti i connettivi logici. Come già accennato in precedenza, avremo in questo caso solo regole di *introduzione* dei connettivi, che saranno classificate in *destre* (r) o *sinistre* (l) a seconda che permettano di introdurre il connettivo logico nella parte destra o in quella sinistra del sequente che è conclusione della regola.
Riprendendo l'introduzione a questo capitolo, molte di queste regole sono del tutto intuitive:

$$(\wedge l.1)\quad \frac{\Gamma, A \vdash C}{\Gamma, A \wedge B \vdash C} \qquad (\wedge r)\quad \frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$$

$$(\wedge l.2)\quad \frac{\Gamma, B \vdash C}{\Gamma, A \wedge B \vdash C}$$

$$(\vee l)\quad \frac{\Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma, A \vee B \vdash C} \qquad (\vee r.1)\quad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}$$

$$(\vee r.2)\quad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}$$

$$(\rightarrow l)\quad \frac{\Gamma \vdash A \qquad \Gamma, B \vdash C}{\Gamma, A \rightarrow B \vdash C} \qquad (\rightarrow l)\quad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B}$$

Veniamo ora alla negazione. Anziché introdurre una costante $\perp$ che denota inconsistenza, si farà l'ipotesi che la parte destra del sequente possa essere vuota (cioè si ammetteranno dei sequenti del tipo $\Gamma \vdash$ ), interpretando l'assenza della conclusione come inconsistenza delle premesse.
La regola che permette di derivare qualunque formula in presenza di inconsistenza assume allora il seguente aspetto

$$\frac{\Gamma \vdash}{\Gamma \vdash A}$$

e può essere assimilata ad una regola di *indebolimento* a destra (una regola strutturale, dunque).
Le regole di introduzione della negazione sono:

$$(\neg l)\quad \frac{\Gamma \vdash A}{\Gamma, \neg A \vdash} \qquad (\neg r)\quad \frac{\Gamma, A \vdash}{\Gamma \vdash \neg A}$$

Il sistema presentato fino ad ora corrisponde all'insieme delle regole logiche che abbiamo chiamato *intuizioniste*, ed è noto come sistema $LJ$.
Per ottenere un sistema di *logica classica* è necessario aggiungere la regola di riduzione ad assurdo o una regola ad essa equivalente. Prima di discutere tale regola è bene tuttavia soffermarsi ad analizzare alcune proprietà importanti del calcolo che abbiamo appena sviluppato.
Nel calcolo appena presentato, la regola di taglio è ridondante. Il fatto che ogni dimostrazione di un sequente $\Gamma \vdash A$ che fa uso di questa regola possa essere riscritta in una dimostrazione dello stesso sequente senza uso di tagli costituisce uno dei maggiori meta-teoremi del Calcolo dei Sequenti, noto come Gentzen's Hauptsatz.
Data una dimostrazione priva di tagli, la proprietà più eclatante del Calcolo dei Sequenti, è la cosiddetta *proprietà della sottoformula*: una dimostrazione priva di tagli di $\Gamma \vdash A$ contiene solo sequenti le cui formule sono sottoformule delle

formule in $\Gamma, A$. La proprietà è una ovvia conseguenza della particolare struttura delle regole del calcolo, che fa sì che si abbiano solo regole di introduzione dei connettivi, ma mai regole di eliminazione. Dunque, una volta costruita una formula composta, non esiste modo di eliminarla nel proseguo dell'albero di derivazione. Come già accennato nell'introduzione, questa proprietà conferisce al Calcolo dei Sequenti una natura "costruttiva" che ha molte importanti ripercussioni sulla meccanizzazione del processo inferenziale e sulla ricerca automatica di dimostrazioni.
Torniamo ora alla legge di riduzione ad assurdo. La sua traduzione immediata porterebbe alla seguente regola:

$$(RAA) \quad \frac{\Gamma, \neg A \vdash}{\Gamma \vdash A}$$

Il problema è che essa viola la proprietà della sottoformula: per poter concludere $A$ si utilizza una formula più complessa: $\neg A$.
Si potrebbe pensare di risolvere il problema aggiungendo semplicemente l'assioma $\vdash A \vee \neg A$ che sappiamo essere logicamente equivalente alla regola di riduzione ad assurdo. In effetti, in presenza di tale assioma, $(RAA)$ diventa derivabile, come mostrato dal seguente albero di prova:

$$\dfrac{\vdash A \vee \neg A \qquad \dfrac{\dfrac{\Gamma, \neg A \vdash}{\Gamma, \neg A \vdash A} \qquad \dfrac{A \vdash A}{\Gamma, A \vdash A}}{\Gamma, A \vee \neg A \vdash A}}{\Gamma \vdash A} \; (taglio)$$

Si noti tuttavia che è stato necessario utilizzare la regola di taglio. In effetti, l'introduzione dell'assioma $\vdash A \vee \neg A$ ha una conseguenza altrettanto spiacevole della perdita della proprietà della sottoformula: in generale, la regola di taglio non risulta più essere eliminabile dal sistema.
Per risolvere il problema precedente si è costretti ad introdurre una modifica piuttosto cospicua nel sistema. In effetti, la ragione principale delle difficoltà che si sono incontrate nella trattazione della nozione classica di negazione sono imputabili alla evidente asimmetria del calcolo introdotto rispetto alla gestione della parte destra e sinistra dei sequenti.
Proviamo ad eliminare questa asimmetria. È necessario dunque considerare sequenti della forma

$$\Gamma \vdash \Delta$$

dove $\Gamma = A_1, A_2, \ldots, A_n$ e $\Delta = B_1, B_2, \ldots, B_m$ sono sequenze finite, eventualmente vuote, di formule.
Per ovvie ragioni di dualità, se la virgola nella parte sinistra del sequente ha il significato intuitivo di *congiunzione*, la virgola a destra deve esprimere *disgiunzione*. Dunque, un sequente del tipo precedente asserisce intuitivamente che da $A_1 \wedge A_2 \wedge \cdots \wedge A_n$ si è in grado di derivare $B_1 \vee B_2 \vee \cdots \vee B_m$.
Tutte le regole strutturali hanno ora un loro corrispettivo destro: l'insieme delle

conclusioni può essere permutato, contratto e indebolito.
Le regole di introduzione dei connettivi restano sostanzialmente immutate, a parte l'aggiunta di una sequenza pametrica di formule $\Delta$ in tutte le parti destre dei sequenti. In Figura 2.1 è riportato l'elenco completo delle regole del calcolo.

$$(Ax) \quad A \vdash A$$

$$(taglio) \quad \frac{\Gamma \vdash A, \Delta \qquad \Gamma', A \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$$

$$(perm-l) \quad \frac{\Gamma, A, B, \Gamma' \vdash \Delta}{\Gamma, B, A, \Gamma' \vdash \Delta} \qquad (perm-r) \quad \frac{\Gamma \vdash \Delta, A, B, \Delta'}{\Gamma \vdash \Delta, B, A, \Delta'}$$

$$(contr-l) \quad \frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta} \qquad (contr-r) \quad \frac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta}$$

$$(indeb-l) \quad \frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta} \qquad (indeb-r) \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, A}$$

$$(\wedge l.1) \quad \frac{\Gamma, A \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} \qquad (\wedge r) \quad \frac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}$$

$$(\wedge l.2) \quad \frac{\Gamma, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}$$

$$(\vee l) \quad \frac{\Gamma, A \vdash \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta} \qquad (\vee r.1) \quad \frac{\Gamma \vdash A, \Delta}{\Gamma \vdash A \vee B, \Delta}$$

$$(\vee r.2) \quad \frac{\Gamma \vdash B, \Delta}{\Gamma \vdash A \vee B, \Delta}$$

$$(\rightarrow l) \quad \frac{\Gamma \vdash A, \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \rightarrow B \vdash \Delta} \qquad (\rightarrow r) \quad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \rightarrow B, \Delta}$$

$$(\neg l) \quad \frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta} \qquad (\neg r) \quad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}$$

Figura 2.1: Regole del sistema LK

Il sistema così ottenuto è noto come sistema $LK$.

**Esempio 2.7** Nel sistema $LK$ la formula $A \vee \neg A$ è *derivabile*, infatti:

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{\vdash \neg A, A}}{\vdash A \vee \neg A, A}}{\vdash A \vee \neg A, A \vee \neg A}}{\vdash A \vee \neg A}$$

Vediamo qualche altro esempio di dimostrazione.

**Esempio 2.8** Proviamo che $\vdash (A \wedge B) \to (B \wedge A)$.

$$\dfrac{\dfrac{\dfrac{B \vdash B}{A \wedge B \vdash B} \qquad \dfrac{A \vdash A}{A \wedge B \vdash A}}{A \wedge B \vdash B \wedge A}}{\vdash (A \wedge B) \to (B \wedge A)}$$

**Esempio 2.9** Proviamo che $\neg(A \vee B) \vdash \neg A \wedge \neg B$.

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{\vdash A, \neg A}}{\vdash A \vee B, \neg A}}{\neg(A \vee B) \vdash \neg A} \qquad \dfrac{\dfrac{\dfrac{B \vdash B}{\vdash B, \neg B}}{\vdash A \vee B, \neg B}}{\neg(A \vee B) \vdash \neg B}}{\neg(A \vee B) \vdash \neg A \wedge \neg B}$$

Come già osservato in precedenza, le regole logiche del Calcolo dei Sequenti (a parte la regola di taglio, che verrà discussa nella prossima sezione) permettono unicamente di *introdurre* dei connettivi, ma mai di *eliminarli* (proprietà della sottoformula). La formula del sequente in cui è stato introdotto il connettivo è detta *formula principale* dell'inferenza logica. Le sottoformule costituenti la formula principale, sono dette *formule ausiliarie* dell'inferenza. Tutte le altre formule dei sequenti coinvolti nella regola logica assumono il ruolo di *parametri*; l'insieme dei parametri è detto *contesto* della regola. Ad esempio, nella regola

$$(\to l) \quad \dfrac{\Gamma \vdash A, \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \to B \vdash \Delta}$$

la formula principale è $A \to B$, le formule ausiliarie sono $A$ e $B$, mentre tutte le formule in $\Gamma, \Delta$ sono parametri ($\{\Gamma, \Delta\}$ è detto contesto).
Nel caso delle regole strutturali, considereremo tutte le formule come parametriche. Quando una formula ausiliaria compare nel sequente premessa dell'inferenza sullo stesso lato del sequente in cui compare la formula principale diremo che essa ha la stessa *polarità* della formula principale. In caso contrario, ha polarità inversa. Ad esempio, in $(\to l)$, $B$ ha la stessa polarità di $A \to B$, mentre $A$ ha polarità inversa. Si noti che la polarità dei costituenti dipende in realtà dal connettivo, e non dalla regola: anche nel caso di $(\to r)$, $B$ ha la stessa polarità di $A \to B$, mentre $A$ ha polarità inversa.

### 2.5.1 Eliminazione del taglio ⋆

Il sistema $LK$ gode della seguente proprietà:

**Teorema 2.7 (Gentzen's Hauptsatz)**
*La regola di taglio è eliminabile dal sistema LK, ovvero per ogni sequente* $\Gamma \vdash \Delta$ *dimostrabile in LK è possibile ottenere in modo effettivo una nuova dimostrazione dello stesso sequente che non contiene nessuna applicazione della regola di taglio.*

L'interesse del teorema è nella natura effettiva (algoritmica) del procedimento di eliminazione dei tagli. Questo risultato costituisce uno dei teoremi più rilevanti della moderna teoria della dimostrazione. La prova del teorema, piuttosto lunga e complessa, verrà solo accennata qui di seguito.

**Definizione 2.8** *Il* livello di un taglio *è la somma delle profondità delle deduzioni delle premesse della regola di taglio (senza contare, nella profondità, l'applicazione di regole strutturali).*

Per *complessità di una formula* si intende il numero di connettivi che compaiono all'interno di essa.

**Definizione 2.9** *Il* grado di un taglio *è la complessità della formula sulla quale avviene il taglio.*

**Idea della dimostrazione**
La dimostrazione si effettua per induzione sul grado del taglio con una sottoinduzione sul livello del taglio. Data una prova di un sequente contenente delle applicazioni della regola di taglio, tale prova viene trasformata in un'altra avente tagli di grado inferiore oppure di livello più basso. Vi sono due possibilità:

1. il taglio è *logico*, vale a dire che entrambe le regole utilizzate appena prima della regola del taglio sono regole logiche di introduzione per la formula sulla quale avviene il taglio[10].

2. il taglio è detto *strutturale* in tutti gli altri casi, cioè quando almeno una delle due regole immediatamente precedenti al taglio utilizza la formula di taglio $A$ in modo parametrico. In questo caso, diremo che la regola è parametrica in $A$.

L'idea della dimostrazione è quella di muovere le regole di taglio "verso l'alto" fino a ridursi a dei tagli con assiomi che possono essere eliminati semplicemente (questo è in effetti il caso di base della sottoinduzione sulle profondità).
In generale, nel caso di tagli strutturali, è sempre possibile spostare la regola che utilizza la formula di taglio $A$ in modo parametrico al di sotto della regola di taglio, costruendo una dimostrazione con un taglio di livello inferiore. Vediamo

[10] Si osservi che in questo caso le due regole di introduzione devono essere rispettivamente destre e sinistre.

un esempio di ciò. Supponiamo, per fissare le idee, di avere una prova in cui il taglio ha la seguente forma:

$$\dfrac{\dfrac{\Gamma, B \vdash C, \Delta, A}{\Gamma \vdash B \to C, \Delta, A} \qquad A, \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \vdash B \to C, \Delta, \Delta'}$$

la prova viene trasformata in

$$\dfrac{\dfrac{\Gamma, B \vdash C, \Delta, A \qquad A, \Gamma' \vdash \Delta'}{\Gamma, \Gamma', B \vdash C, \Delta, \Delta'}}{\Gamma, \Gamma' \vdash B \to C, \Delta, \Delta'}$$

che è una prova dello stesso sequente il cui taglio è di livello inferiore.
Nel caso di un taglio logico, questo viene sostituito da uno o piú tagli di grado inferiore. Vediamone degli esempi. Una prova in cui il taglio ha la forma

$$\dfrac{\dfrac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \to B, \Delta} \qquad \dfrac{\Gamma' \vdash A, \Delta' \qquad \Gamma', B \vdash \Delta'}{\Gamma', A \to B \vdash \Delta'}}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$$

viene ridotta a

$$\dfrac{\Gamma' \vdash A, \Delta' \qquad \dfrac{\Gamma, A \vdash B, \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, A, \Gamma' \vdash \Delta, \Delta'}}{\dfrac{\Gamma', \Gamma, \Gamma' \vdash \Delta', \Delta, \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}}$$

che è una prova dello stesso sequente avente due tagli di grado inferiore ($A$ e $B$ hanno una complessità minore di $A \to B$).
Data una prova in cui il taglio ha la forma

$$\dfrac{\dfrac{\Gamma \vdash A, \Delta}{\Gamma \vdash A \vee B, \Delta} \qquad \dfrac{\Gamma', A \vdash \Delta' \quad \Gamma', B \vdash \Delta'}{\Gamma', A \vee B \vdash \Delta'}}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$$

questa viene sostituita da

$$\dfrac{\Gamma \vdash A, \Delta \qquad \Gamma', A \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$$

che è una prova dello stesso sequente avente un taglio di grado inferiore.
In realtà non tutti i casi sono così semplici. I maggiori problemi si incontrano

quando la formula su cui si effettua il taglio è stata appena contratta in una delle due premesse, come nel caso seguente:

$$\dfrac{\dfrac{\Gamma \vdash \Delta, A, A}{\Gamma \vdash \Delta, A} \qquad A, \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$$

L'ovvia trasformazione sembrerebbe essere:

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash \Delta, A, A \qquad A, \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta', A} \qquad A, \Gamma' \vdash \Delta'}{\dfrac{\Gamma, \Gamma', \Gamma' \vdash \Delta, \Delta', \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}}$$

Tuttavia questa regola di riduzione, unitamente a quella simmetrica per gestire il caso di contrazione nella premessa destra, può dare luogo a sequenze infinite di riduzioni, come nel caso seguente (il lettore lo verifichi per esercizio):

$$\dfrac{\dfrac{\Gamma \vdash \Delta, A, A}{\Gamma \vdash \Delta, A} \qquad \dfrac{A, A, \Gamma' \vdash \Delta'}{A, \Gamma' \vdash \Delta'}}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$$

Per ovviare a questo problema è necessario considerare una forma generalizzata di taglio, in cui si permette il taglio simultaneo su più occorrenze diverse della stessa formula, ovvero

$$(multi - taglio) \qquad \dfrac{\Gamma \vdash \Delta, mA \qquad nA, \Gamma' \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$$

dove $nA$ denota $n$ occorrenze di $A$.
I dettagli piuttosto lunghi e complessi della dimostrazione dell'eliminazione di questa regola fuoriescono tuttavia dagli intenti del presente volume. Il lettore interessato può consultare [TS96, Gal91, Tak75]. □
Per concludere, osserviamo anche che nonostante la regola di taglio sia *eliminabile*, essa *non è derivabile* in $LK$, ovvero non è possibile esprimerla mediante una combinazione di altre regole del calcolo (ciò è una ovvia conseguenza della proprietà della sottoformula).

### 2.5.2 Sulle regole strutturali ⋆

Osserviamo che nelle regole di LK per $\wedge$ e $\vee$ aventi due premesse i contesti per ambedue le premesse sono gli stessi; tali regole sono dette *dipendenti dal contesto* (o *addittive*). Tuttavia, per la presenza delle regole strutturali, è possibile ottenere dei sistemi equivalenti ad LK nei quali alcune (o tutte) le regole dipendenti dal contesto sono sostituite da regole *libere dal contesto* (o *moltiplicative*)

nelle quali, cioè, i contesti di entrambe le premesse si assumono disgiunti e sono semplicemente "uniti" nella conclusione. Le versioni moltiplicative delle regole $(\wedge r)$ e $(\vee l)$ sono

$$(\wedge' r)\quad \frac{\Gamma \vdash A, \Delta \qquad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \wedge B, \Delta, \Delta'} \qquad (\vee' l)\quad \frac{\Gamma, A \vdash \Delta \qquad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \vee B \vdash \Delta, \Delta'}$$

Proviamo, per fissare le idee, che le due versioni della regola per l' $\wedge$ sono equivalenti, vale a dire che, in presenza delle regole strutturali sono derivabili l'una dall'altra. Infatti, supponiamo di disporre di $(\wedge r)$ e dimostriamo la derivabilità, da questa, di $(\wedge' r)$.

$$\frac{\dfrac{\Gamma \vdash A, \Delta}{\Gamma, \Gamma' \vdash A, \Delta, \Delta'} \qquad \dfrac{\Gamma' \vdash B, \Delta}{\Gamma, \Gamma' \vdash B, \Delta, \Delta'}}{\Gamma, \Gamma' \vdash A \wedge B, \Delta, \Delta'}$$

Si noti che, è stato necessario utilizzare le regole di indebolimento. Viceversa, supponiamo di disporre di $(\wedge' r)$ e dimostriamo la derivabilità, da questa, di $(\wedge r)$.

$$\frac{\dfrac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma, \Gamma \vdash A \wedge B, \Delta, \Delta}}{\Gamma \vdash A \wedge B, \Delta}$$

Si noti che è stato necessario utilizzare le regole di contrazione.
La dimostrazione dell'equivalenza delle due versioni della regola per $\vee$ è lasciata al lettore come esercizio.
Un discorso del tutto simile vale anche per le regole logiche ad una sola premessa. Ad esempio, le due regole $(\wedge l)$ possono essere sostituite dall'unica regola

$$(\wedge' l)\quad \frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}$$

e simmetricamente le due regole per $(\vee r)$ possono essere sostituite dall'unica regola

$$(\vee' r)\quad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta}$$

Anche in questo caso è facile dimostrare l'equivalenza tra le varie formulazioni, in presenza di regole strutturali. Consideriamo il caso di $\wedge$, lasciando al lettore come esercizio quelle per $\vee$.
Avendo $(\wedge l.1)$ e $(\wedge l.2)$ deriviamo $(\wedge' l)$ nel seguente modo:

$$\dfrac{\dfrac{\dfrac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B, B \vdash \Delta}}{\Gamma, A \wedge B, A \wedge B \vdash \Delta}}{\Gamma, A \wedge B \vdash \Delta}$$

Viceversa, avendo $(\wedge' l)$, possiamo derivare $(\wedge l.1)$ e $(\wedge l.2)$:

$$\dfrac{\dfrac{\Gamma, A \vdash \Delta}{\Gamma, A, B \vdash \Delta}}{\Gamma, A \wedge B \vdash \Delta} \qquad \dfrac{\dfrac{\Gamma, B \vdash \Delta}{\Gamma, A, B \vdash \Delta}}{\Gamma, A \wedge B \vdash \Delta}$$

Le regole $(\wedge' l)$ e $(\vee' r)$ devono essere considerate *moltiplicative*, mentre $(\wedge l.1)$, $(\wedge l.2)$, $(\vee r.1)$ e $(\vee r.2)$ sono *addittive*. La ragione di questo fatto è dovuta alla tecnica di eliminazione dei tagli. In particolare, per eliminare (o meglio "ridurre") un taglio tra due formule introdotte con regole della stessa "natura" (rispettivamente moltiplicative o addittive) non è necessario richiedere l'uso di regole strutturali. Ad esempio, il taglio

$$\dfrac{\dfrac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta} \qquad \dfrac{\Gamma', A, \vdash \Delta'}{\Gamma, A \wedge B \vdash \Delta'}}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$$

Può essere ridotto al seguente taglio "più semplice[11]" (senza aggiunta di regole strutturali)

$$\dfrac{\Gamma \vdash A, \Delta \qquad \Gamma' A, \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta}$$

Questo non sarebbe stato possibile se ad esempio il taglio fosse stato del seguente tipo:

$$\dfrac{\dfrac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta} \qquad \dfrac{\Gamma', A, B \vdash \Delta'}{\Gamma, A \wedge B \vdash \Delta'}}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$$

in quanto dovremmo operare delle contrazioni sulla prova "ridotta" (gli altri casi sono lasciati al lettore come esercizio).

Dunque le regole strutturali assumono un ruolo di primaria importanza nel Calcolo dei Sequenti. In particolare, la presenza o l'assenza di queste determina sistemi logici notevolmente differenti tra loro. Ad esempio, il sistema LJ, che corrisponde alla Logica Intuizionista (per delle trattazioni recenti si vedano, tra gli altri, [Tro73, Dum77, Gab81, VDa86]), cui abbiamo accennato in precedenza, si ottiene imponendo, nel sistema LK, che alla destra dei sequenti vi sia al più una formula; questo chiaramente impedisce la presenza delle regole strutturali a destra $((perm - r), (contr - r)$ e $(indeb - r))$.
In generale, logiche che non hanno tutte le regole strutturali, sono dette *Logiche Substrutturali* (si veda [DS93] per una panoramica).
Discuteremo brevemente alcune di queste. Sistemi nei quali sono assenti le regole di indebolimento sono detti *Logiche della Rilevanza* (si vedano [AB75, Dun86,

[11] Di grado inferiore.

ABD92]); il nome è dovuto essenzialmente al fatto che, nelle regole di inferenza, le premesse devono essere rilevanti per le conclusioni. Tali logiche sono utilizzate, in Intelligenza Artificiale, per modellare ragionamenti con incertezza.
Logiche nelle quali sono assenti le regole di contrazione, si vedano ad esempio [OK85, KW84], sono spesso chiamate *Logiche BCK*. Il nome deriva dal fatto che il frammento implicativo (cioè l'insieme delle formule contenenti soltanto il connettivo "$\rightarrow$") di tali logiche, può essere assiomatizzato dagli schemi di assiomi:

$$\text{B} : (A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C))$$

$$\text{C} : (A \rightarrow (B \rightarrow C)) \rightarrow (B \rightarrow (A \rightarrow C))$$

$$\text{K} : A \rightarrow (B \rightarrow A)$$

Le *Logiche Multivalore di Łukasiewicz*, menzionate nel paragrafo 1.5, accettano le regole di indebolimento e permutazione, mentre è possibile utilizzare le regole di contrazione solo su formule aventi delle caratteristiche prestabilite.
Nella *Logica Lineare*, introdotta da Girard [Gir87] (si vedano inoltre [Gir95, Tro92]), sono assenti sia le regole di indebolimento, che quelle di contrazione[12]. Ciò comporta, come discusso in precedenza, lo "sdoppiamento" dei connettivi $\wedge$ e $\vee$ nelle corrispondenti versioni addittive e moltiplicative. La Logica Lineare offre una visione molto più sottile della logica classica, enfatizzando in modo particolare il ruolo delle formule come vere e proprie "risorse" di calcolo, con innumerevoli applicazioni all'informatica.
Ricordiamo infine il *Calcolo di Lambek*, [Lam58], assolutamente privo di regole strutturali. Tale calcolo viene utilizzato per modellare importanti caratteristiche tipiche del linguaggio naturale (grammatiche categoriali).

### 2.5.3 Invertibilità delle regole logiche

Il Calcolo dei Sequenti, assai meglio della Deduzione Naturale, permette di costruire "all'indietro" le dimostrazioni di teoremi in modo piuttosto semplice e naturale.

**Esempio 2.10** Consideriamo il caso:

$$\vdash (A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$$

Il connettivo principale di questa formula è $\rightarrow$. Proviamo a rimuoverlo; applicando all'indietro la regola $(\rightarrow r)$ si ottiene il sequente:

$$A \rightarrow (B \rightarrow C) \vdash (A \rightarrow B) \rightarrow (A \rightarrow C)$$

Continuando ad applicare all'indietro $(\rightarrow r)$ ci si riduce in due passi a

$$A \rightarrow (B \rightarrow C), A \rightarrow B, A \vdash C$$

[12]Comunque, tali regole vengono reintrodotte, in maniera "controllata" utilizzando due nuovi connettivi "!" e "?".

Ora si è costretti ad operare sulla parte sinistra del sequente. Poiché la regola $(\to l)$ ha due antecedenti, questo costringerà a spezzare la dimostrazione in due sottodimostrazioni. Ad esempio, se si lavora sulla formula $A \to (B \to C)$, si otterranno i due sottosequenti

$$(1) \quad A \to B, A \vdash A, C \qquad\qquad (2) \quad B \to C, A \to B, A \vdash C$$

(1) si ottiene dall'assioma $A \vdash A$ mediante indebolimenti, quindi questa parte della dimostrazione è conclusa.
Passiamo a (2). Se si opera all'indietro con $(\to l)$ su $B \to C$ si ottengono

$$(2.1) \quad A \to B, A \vdash B, C \qquad\qquad (2.2) \quad C, A \to B, A \vdash C$$

(2.2) si ottiene dall'assioma $C \vdash C$ per indebolimenti. Per quanto riguarda (2.1), eliminando l'ultimo connettivo di implicazione si ottengono

$$(2.1.1) \quad A \vdash A, B, C \qquad\qquad (2.1.2) \quad B, A \vdash B, C$$

che sono ancora entrambi derivabili da assiomi tramite passi di indebolimento. Questo conclude la dimostrazione, che ha dunque la seguente forma:

$$\dfrac{\dfrac{\dfrac{A \vdash A}{A \to B, A \vdash A, C} \quad \dfrac{\dfrac{\dfrac{A \vdash A}{A \vdash A, B, C} \quad \dfrac{B \vdash B}{B, A \vdash B, C}}{A \to B, A \vdash B, C} \quad \dfrac{C \vdash C}{C, A \to B, A \vdash C}}{B \to C, A \to B, A \vdash C}}{A \to (B \to C), A \to B, A \vdash C}}{A \to (B \to C), A \to B \vdash A \to C}}{A \to (B \to C) \vdash (A \to B) \to (A \to C)}}{\vdash (A \to (B \to C)) \to ((A \to B) \to (A \to C))}$$

Nell'esempio precedente la scelta della regola da applicare è stata effettuata in modo casuale. In certi casi, la situazione non è sempre così ovvia.
Consideriamo, ad esempio, il sequente $\vdash A \vee \neg A$ (che sappiamo essere un teorema di LK). L'unico connettivo presente è $\vee$. Se si prova ad eliminarlo, si ottengono o il sottosequente $\vdash A$ oppure il sottosequente $\vdash \neg A$. Tuttavia, nessuno di questi due sequenti è dimostrabile in LK.
In realtà, la regola corretta da applicare in questo caso è una regola di *contrazione* (si veda la dimostrazione nell'Esempio 2.7). In taluni casi, è possibile ovviare a questo tipo di problemi considerando formulazioni alternative, ma logicamente equivalenti, del sistema logico.
Ad esempio, come osservato nel capitolo precedente, le due regole per $(\vee r)$ possono essere sostituite dall'unica regola

$$(\vee r - II) \quad \dfrac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta}$$

In questo calcolo, la dimostrazione di $\vdash A, \neg A$ è semplicemente:

$$\dfrac{\dfrac{A \vdash A}{\vdash A, \neg A}}{\vdash A \vee \neg A}$$

Simmetricamente, le due regole $(\wedge l)$ possono essere sostituite dall'unica regola

$$(\wedge l - II) \quad \dfrac{\Gamma, A, B \vdash \Delta}{\Gamma A \wedge B \vdash \Delta}$$

Chiameremo questa nuova formulazione del sistema LK, formulazione II[13] (e la precedente formulazione I).

Le regole introdotte in precedenza risolvono il problema per $\vdash A, \neg A$. Tuttavia, chi assicura che un problema simile non si verifichi per altre formule ed altri connettivi? In generale, la proprietà desiderabile è che ogniqualvolta si ha un albero di prova che termina in un sequente che contiene una formula composta $M$ che appare come parametro nell'ultima regola dell'inferenza, sia sempre possibile riorganizzare l'albero in modo tale da concludere la dimostrazione con una regola di introduzione per $M$. Questo significa che l'ultima regola dell'albero di prova originario e quella di introduzione per $M$ possono essere *invertite*. Dal punto di vista della ricerca all'indietro della dimostrazione, questo assicurerebbe che comunque si sceglie il connettivo da eliminare, tale scelta è corretta.
Proviamo a formalizzare meglio il problema. Sia dato un albero di prova $t$ di $\Gamma \vdash \Delta$ e supponiamo che in esso appaia una formula composta $M$ che non è stata introdotta dall'ultima regola dell'albero. Per fissare le idee, supponiamo che $M$ compaia nella parte destra del sequente (il caso opposto è simmetrico). Dunque, $\Delta$ ha la forma $\Delta', M$ (a meno di permutazioni). Supponiamo che esista un'unica regola $R$ che permetta di introdurre $M$ (questo è il caso della formulazione II, ma non della formulazione I di LK). $R$ avrà un certo numero di premesse; indichiamo con $N_i$ le formule ausilarie di $M$ nella $i$-esima premessa della regola. L'idea è quella di costruire, a partire dall'albero originario $t$, dei nuovi alberi di dimostrazione per $\Gamma \vdash \Delta', N_i$ o $\Gamma, N_i \vdash \Delta'$, a seconda della polarità di $N_i$ rispetto ad $M$, in modo tale da poter concludere $\Gamma \vdash \Delta$ con un'applicazione di $R$.
A tal fine, sia $t'$ il sottoalbero di $t$ che contiene degli avi parametrici di $M$. Supponiamo di cancellare $M$ da questo sottoalbero e di sostituirlo con $N_i$ (se $N_i$ ha polarità inversa rispetto a $M$, si deve operare l'inserimento nella parte opposta del sequente rispetto a quella in cui si trova $M$). Le foglie del sottoalbero $t'$ sono i punti in cui $M$ è creato. Questo può avvenire o tramite una regola di indebolimento (ed in tal caso non vi sono problemi nell'introdurre $N_i$ al posto di $M$), oppure mediante un'applicazione della regola $R$. In quest ultimo caso,

[13] Dal punto di vista strettamente logico, la formulazione II del Calcolo dei sequenti non è molto elegante in quanto "mischia" regole addittive e moltiplicative. Tuttavia, come vedremo, questa formulazione semplifica notevolmente la ricerca automatica di prove, e soprattutto permette una dimostrazione estremamente semplice del teorema di completezza.

quando si rimpiazza $M$ con $N_i$, il sequente cosí ottenuto *coincide* necessariamente con la $i$-esima premessa di quella applicazione di $R$, e ci si ricongiunge quindi all'albero di prova originario. Dunque, nel caso della formulazione II del sistema LK, è sempre possibile operare l'inversione suddetta. In generale, quando un procedimento di questo tipo è applicabile, si dirà che la regola $R$ è *direttamente invertibile* nel sistema logico.
Vediamo un esempio di inversione.

**Esempio 2.11** Si consideri il seguente albero $t$:

$$\dfrac{\dfrac{A \vdash A}{A \to B, A \vdash A, C} \qquad \dfrac{\dfrac{\dfrac{A \vdash A}{A \vdash A, B, C} \quad \dfrac{B \vdash B}{B, A \vdash B, C}}{A \to B, A \vdash B, C} \quad \dfrac{C \vdash C}{C, A \to B, A \vdash C}}{B \to C, A \to B, A \vdash C}}{\dfrac{A \to (B \to C), A \to B, A \vdash C}{A \to (B \to C), A \to B \vdash A \to C}}$$

In esso, l'ultima regola applicata è $(\to r)$, con formula principale $A \to C$. Vogliamo trovare una dimostrazione dello stesso sequente nella quale l'ultima regola applicata sia $(\to l)$, con formula principale $A \to B$. Il sottoalbero di $t$ che contiene in modo parametrico $A \to B$ è il seguente:

$$\dfrac{\dfrac{A \to B, A \vdash A, C \qquad \dfrac{A \to B, A \vdash B, C \quad C, A \to B, A \vdash C}{B \to C, A \to B, A \vdash C}}{A \to (B \to C), A \to B, A \vdash C}}{A \to (B \to C), A \to B \vdash A \to C}$$

Rimpiazzando opportunamente $A$ e $B$ al posto di $A \to B$ in accordo alle loro polarità, si ottengono i due sottoalberi seguenti:

$$\dfrac{\dfrac{A \vdash A, A, C \qquad \dfrac{A \vdash A, B, C \quad C, A \vdash A, C}{B \to C, A \vdash A, C}}{A \to (B \to C), A \vdash A, C}}{A \to (B \to C) \vdash A, A \to C}$$

$$\dfrac{\dfrac{B, A \vdash A, C \qquad \dfrac{B, A \vdash B, C \quad C, B, A \vdash C}{B \to C, B, A \vdash C}}{A \to (B \to C), B, A \vdash C}}{A \to (B \to C), B \vdash A \to C}$$

Entrambi gli alberi possono essere ora ricongiunti a sottoalberi dell'albero ori-

ginario, per dar luogo a degli alberi di prova completi. In particolare, avremo:

$$\dfrac{\dfrac{}{A \vdash A}\quad \dfrac{\dfrac{}{A \vdash A}\quad \dfrac{}{C \vdash C}}{B \to C, A \vdash A, C}}{\dfrac{A \to (B \to C), A \vdash A, C}{A \to (B \to C) \vdash A, A \to C}}$$

$$\dfrac{\dfrac{}{A \vdash A}\quad \dfrac{\dfrac{}{B \vdash B}\quad \dfrac{}{C \vdash C}}{B \to C, B, A \vdash C}}{\dfrac{A \to (B \to C), B, A \vdash C}{A \to (B \to C), B \vdash A \to C}}$$

Infine, da questi due alberi, con una applicazione di $(\to l)$ si ottiene la nuova dimostrazione cercata di $A \to (B \to C), A \to B \vdash A \to C$ in cui le regole di introduzione dell'implicazione per $A \to (B \to C)$ e $A \to B$ sono state invertite.

## 2.6 Cenni storici e bibliografici

La prima formulazione di un calcolo proposizionale come sistema logico formale (di natura assiomatica) si deve a Frege, nei suoi *Begriffschrifts* del 1879. La rilevanza di questi lavori passò per lungo tempo inosservata; i primi ad apprezzarne il significato ed a riprenderne il metodo furono, dopo oltre venti anni, Russel [Rus03] e Whitehead [RW10]. Da quel momento e fino alla metà degli anni trenta si assiste ad un continuo proliferare di differenti formulazioni, sempre di natura assiomatica, essenzialmente rivolte ad investigare problemi di economia, indipendenza o semplicemente ad approfondire e separare il significato logico dei singoli connettivi. Alcuni dei più celebri di questi sistemi sono stati brevemente discussi nel paragrafo 2.3.2. Notiamo che l'uso di *schemi* per questi calcoli fu introdotto da Von Neumann [VNe27] solo nel 1927. Prima di allora si tendeva ad utilizzare una esplicita regola di *sostituzione* che permetteva di instanziare variabili proposizionali con formule ben formate.
L'enunciato e la prima dimostrazione del teorema di deduzione si devono ad Herbrand [Her30]. L'idea di usare il principio di deduzione come regola primitiva di inferenza è di Gentzen [Gen34]. Tale regola, e più in generale tutte le regole condizionali che prevedono cancellazione di alcune ipotesi, hanno un carattere meno primitivo delle altre, che le rende legittimamente "sospette" da un punto di vista strettamente logistico (così come definito da Frege e Russel). Tuttavia, è innegabile la naturalezza di questo approccio; assumere come primitivo il principio di deduzione significa formalizzare l'usuale pratica matematica che consiste nel dimostrare una implicazione assumendo l'ipotesi e dimostrando il conseguente.

La regola di deduzione riacquista pienamente il suo carattere elementare nel Calcolo dei Sequenti, dovuto ancora a Genzten [Gen34], sotto forma di regola di introduzione a destra del connettivo di implicazione. Il prezzo da pagare è quello di internalizzare nel linguaggio logico stesso il simbolo di inferenza: le asserzioni elementari non sono più dunque proposizioni, ma *sequenti* della forma $\Gamma \vdash \Delta$ che intuitivamente esprimono relazioni di *derivabilità* logica.
Il teorema di eliminazione dei tagli è il teorema principale ("*Hauptsatz*") di Gentzen [Gen34]. L'interesse della regola di taglio risiede, oltre che nella sua naturalezza, nel fatto che corrisponde alla normale pratica di scomporre una dimostrazione complessa in funzione di lemmi più semplici, dimostrati indipendentemente. Dunque, la ridondanza della regola di taglio, e la natura costruttiva della sua dimostrazione, assicurano che questo procedimento non aggiunge nulla al potere espressivo della logica in esame.
L'analogia tra tipi e proposizioni è anche nota come isomorfismo di Curry-Howard (Howard [How80] è stato il primo a esplicitarla, ma le idee generali risalgono a Curry). Il punto di partenza di tale analogia consiste nel vedere una formula $A$ come un tipo, ed una dimostrazione di $A$ come un termine di tipo $A$. L'aspetto più interessante di ciò risiede nel fatto che il procedimento di eliminazione dei tagli nel sistema logico può essere messo in relazione con un analogo procedimento di normalizzazione dei termini. L'analogia è particolarmente forte tra il frammento logico implicativo ed il $\lambda$ calcolo: in questo caso, l'eliminazione dei tagli corrisponde esattamente al processo di $\beta$-riduzione del $\lambda$-termine. Per una introduzione a questi argomenti si veda [GLT90].
Quella parte della logica che tratta della teoria della dimostrazione ha ricevuto un nuovo impulso a partire dai primi anni settanta proprio grazie alle sua rilevanza in informatica. La data d'inizio di questa rinascita può essere fatta coincidere con il fondamentale risultato di Girard [Gir72, Gir86] di eliminazione dei tagli per il calcolo intuizionista di ordine superiore (sebbene il merito di averne colto la rilevanza informatica debba piuttosto essere accreditato a Reynolds [Rey74]). Da quel momento l'isomorfismo di Curry-Howard è stato il tema conduttore di una enorme quantità di ricerche condotte in informatica negli anni ottanta su sistemi di tipo, polimorfismo e sottotipi, stimolate dalla sinergia di teorie ed approcci diversi: $\lambda$ calcolo, programmazione funzionale, teoria delle dimostrazioni, teoria delle categorie. Il lettore interessato ad una introduzione a questi aspetti può consultare [LS86, AL91] e la bibliografia ivi menzionata.

## Esercizi

**2.1** Provare che $(RAA)$ è equivalente a $\neg\neg A \vdash A$.

**2.2** Dimostrare, in un sistema deduttivo, che le seguenti proposizioni sono derivabili:

1. $(A \wedge B) \wedge C \rightarrow A \wedge (B \wedge C)$
2. $A \wedge B \rightarrow A \vee B$

3. $\neg(A \wedge B) \rightarrow \neg A \vee \neg B$
4. $\neg\neg A \rightarrow A$
5. $A \rightarrow \neg\neg A$
6. $\neg A \vee \neg\neg A$
7. $\neg\neg(A \vee \neg A)$
8. $\neg(A \wedge \neg A)$
9. $(A \rightarrow B) \vee (B \rightarrow A)$
10. $\neg A \vee B \rightarrow (\neg\neg B) \vee \neg A$
11. $\bot \rightarrow A$

**2.3** Quali dei teoremi del precedente esercizio sono derivabili nel sistema intuizionista? Motivare la risposta.

**2.4** Mostrare l'equivalenza dei Sistemi Assiomatici presentati in 2.3 e 2.3.2.

**2.5** Provare che:

1. $\vdash (A \rightarrow \neg A) \rightarrow \neg A$
2. $\vdash (A \rightarrow B) \wedge (A \rightarrow \neg B) \rightarrow \neg A$
3. $\vdash ((A \rightarrow B) \rightarrow A) \rightarrow A$ ) (*legge di Peirce*)
4. $A \vdash \neg(\neg A \wedge \neg B)$
5. $\neg(A \wedge \neg B), A \vdash B$
6. $\vdash (A \rightarrow B) \vee (B \rightarrow A)$

**2.6** Mostrare che se $\vdash A$, allora $\vdash B \rightarrow A$.

**2.7** Sia $^- : FBF \rightarrow FBF$ tale che $\forall A, P, Q \in FBF$

- $\overline{A} = \neg A$ se $A$ è una proposizione atomica
- $\overline{P \wedge Q} = \overline{P} \wedge \overline{Q}$
- $\overline{P \vee Q} = \overline{P} \vee \overline{Q}$
- $\overline{P \rightarrow Q} = \overline{P} \rightarrow \overline{Q}$
- $$\overline{\neg P} = \begin{cases} P & \text{se } P \text{ è una proposizione atomica} \\ \neg\overline{P} & \text{altrimenti} \end{cases}$$

1. Determinare $\overline{(A \wedge \neg B) \vee \neg(A \wedge C))}$ e $\overline{(A \vee \neg B) \wedge (\neg A \wedge B)}$.

2. È vero che $\overline{\overline{P}} = P$, $\forall P \in FBF$? Motivare la risposta.

3. Provare che $\forall P \in FBF \ \vdash P, \overline{P}$ e $P, \overline{P} \vdash$.

**2.8** Si considerino le seguenti proposizioni:

1. Se Franco è un elettricista, allora Giorgio è un idraulico oppure Elena è un'insegnante.

2. Se Giorgio è un idraulico, allora Lucia non fa la commessa oppure Elena è un'insegnante.

3. Se Lucia fa la commessa, allora Franco è un elettricista.

4. Lucia fa la commessa.

5. Elena è un'insegnante.

Dimostrare che $a, b, c, d \vdash e$ utilizzando la Deduzione Naturale ed il Calcolo dei Sequenti.

**2.9** Si consideri in seguente ragionamento:

a Se Antonio è di Agrigento o Bruno è di Bologna, allora Carlo è di Cremona oppure Davide è di Domodossola.

b Se Antonio è di Agrigento allora Carlo non è di Cremona.

Dunque
(c) Davide è di Domodossola oppure Antonio non è di Agrigento.

Se ne verifichi la correttezza utilizzando la Deduzione Naturale ed il Calcolo dei Sequenti.

**2.10** Definire le regole della Deduzione Naturale per il connettivo $\leftrightarrow$; quindi provare che:

1. $\vdash A \vee A \leftrightarrow A$

2. $\vdash A \wedge A \leftrightarrow A$

3. $\vdash A \vee (A \wedge B) \leftrightarrow A$

4. $\vdash A \wedge (A \vee B) \leftrightarrow A$

# Capitolo 3

# Correttezza e Completezza

Abbiamo sinora introdotto la sintassi della logica proposizionale, la sua semantica sotto forma di interpretazioni ed alcuni calcoli logici di inferenza simbolica. Vogliamo ora studiare la relazione esistente tra il calcolo proposizionale e la sua semantica, ovvero tra le nozioni di *derivabilità* e *validità*. Ovviamente, ci si aspetta che ogni teorema, cioè ogni formula derivabile in un sistema formale, sia anche valida (sia una tautologia), ovvero che $\vdash A$ implichi $\models A$. Questo assicurerebbe che il sistema di calcolo formale è *corretto*, nel senso che non permette di inferire cose semanticamente non valide. Tuttavia, possiamo essere più esigenti: vogliamo che il calcolo sia anche *completo*, nel senso che permetta di derivare *ogni* tautologia. In altre parole, il sistema risulta completo se $\models A$ implica $\vdash A$. Se si vede il calcolo logico come una estensione degli aspetti sintattici del sistema formale (attinenti al simbolo di derivabilità $\vdash$), si può parlare semplicemente di due livelli: uno sintattico, ed uno semantico. Il Teorema di completezza stabilisce la relazione tra questi due livelli, creando una completa analogia tra nozioni sintattiche (nella accezione precedente) e semantiche. Tale relazione si può riassumere brevemente nella seguente tabella, a cui il lettore può fare costante riferimento durante la lettura di questo capitolo:

| sintassi (calcolo) | semantica |
|---|---|
| $\Gamma \vdash A$<br>derivabilità | $\Gamma \models A$<br>conseguenza semantica |
| $\vdash A$<br>teorema | $\models A$<br>tautologia |
| $\Gamma \nvdash \bot$<br>consistenza | $\Gamma \not\models \bot$<br>soddisfacibilità |
| $\Gamma \vdash \bot$<br>inconsistenza | $\Gamma \models \bot$<br>insoddisfacibilità |

Presenteremo nel seguito tre diverse dimostrazioni di correttezza e completezza (una per ogni sistema formale). Ognuna di queste è basata su un approccio differente e mette in luce nuove proprietà del calcolo proposizionale. Tali dimostrazioni, inoltre, permetteranno di spiegare alcune importanti sfumature

nell'enunciato del Teorema di completezza (completezza forte, debole, finita). Le prime due dimostrazioni possono essere facilmente adattate ad ognuno dei sistemi logici considerati. Al contrario, la terza, che è anche la più intuitiva, si basa su alcune proprietà caratteristiche del Calcolo dei Sequenti (in particolare della formulazione II). La prima dimostrazione non è *costruttiva*: data una tautologia $A$ ($\models A$) si prova che nel sistema deduttivo in oggetto deve necessariamente esistere una derivazione di $A$, ma non si fornisce tale derivazione in modo esplicito ed effettivo. Al contrario, le rimanenti sono entrambe costruttive.

## 3.1 Deduzione Naturale

**Teorema 3.1 (Correttezza)**
*Se* $\Gamma \vdash P$ *allora* $\Gamma \models P$.

*Dimostrazione.* Per induzione sulla profondità dell'albero di prova per $\Gamma \vdash P$.
(*caso base*) L'albero è costituito dalla sola radice $P$, allora, per definizione, $P \in \Gamma$, che implica $\Gamma \models P$.
Veniamo al caso induttivo. Dobbiamo distinguere tanti sottocasi quante sono le possibili regole di inferenza che concludono l'albero di prova di $\Gamma \vdash P$. Ne vedremo solo alcuni: i rimanenti sono lasciati al lettore come esercizio.

1. $P = P_1 \wedge P_2$ e l'ultima regola di deduzione applicata è $(\wedge i)$. Allora devono esistere due alberi di prova per $\Gamma \vdash P_1$ e $\Gamma \vdash P_2$. Per ipotesi induttiva, $\Gamma \models P_1$ e $\Gamma \models P_2$, che implica $\Gamma \models P_1 \wedge P_2$.

2. $P = P_1 \rightarrow P_2$ e l'ultima regola applicata è $(\rightarrow i)$. Dunque esiste un albero di prova per $\Gamma, P_1 \vdash P_2$ e per ipotesi induttiva $\Gamma, P_1 \models P_2$. Sia $v$ un modello di $\Gamma$. Se $v(P_1) = 1$, allora $v(P_2) = 1$ per l'ipotesi precedente, e dunque anche $v(P_1 \rightarrow P_2) = 1$. Viceversa se $v(P_1) = 0$, per definizione di interpretazione, risulta ancora $v(P_1 \rightarrow P_2) = 1$. Dunque $\Gamma \models P_1 \rightarrow P_2$.

3. L'ultima regola applicata è una regola di riduzione ad assurdo ($RAA$). Allora esiste un albero di prova per $\Gamma, \neg P \vdash \bot$ e per ipotesi induttiva $\Gamma, \neg P \models \bot$. Questo significa che nessun modello soddisfa $\Gamma, \neg P$. Supponiamo di avere un modello per $\Gamma$: necessariamente, in questo modello deve essere $v(P) = 1$, poiché in caso contrario avremmo un modello per $\Gamma, \neg P$. Dunque $\Gamma \models P$.

□

Il Teorema di *correttezza* assicura che ciò che è deducibile da certe premesse, mediante le regole della Deduzione Naturale, è conseguenza logica delle premesse. In particolare tale calcolo consente di derivare *solo* formule valide (tautologie); infatti:

**Corollario 3.2** *Se* $\vdash P$ *allora* $\models P$.

L'inverso del teorema precedente, e cioè se $\Gamma \models P$ allora $\Gamma \vdash P$ (*completezza*) è assai più complicato, e richiede l'introduzione di alcuni nozioni e risultati preliminari; infatti, a priori, non è affatto ovvio che le regole della Deduzione Naturale siano sufficienti per ottenere, da un insieme di premesse, tutte le conseguenze semantiche dello stesso, ed in particolare, dall'insieme vuoto di premesse, *tutte* le formule valide. Ad esempio, il sistema intuizionista, ottenuto da quello classico rimuovendo la sola regola $RAA$ non è completo rispetto alla nozione classica di validità semantica basata su tabelle di verità.

**Definizione 3.3** *Un insieme di proposizioni* $\Gamma$ *è* consistente *se* $\Gamma \nvdash \bot$*; è* inconsistente *se non è consistente, cioè* $\Gamma \vdash \bot$*.*

Si noti che la nozione di *consistenza* è prettamente "sintattica" (cioè relativa al sistema di calcolo formale). A priori, niente assicura che ogni insieme di proposizioni consistente sia anche soddisfacibile e viceversa. Un verso di questa corrispondenza (soddisfacibilità implica consistenza) è un semplice corollario del Teorema di correttezza (Lemma 3.5); l'altro verso è assai più delicato.

**Lemma 3.4** *Le seguenti condizioni risultano equivalenti:*

1. $\Gamma$ *è inconsistente;*
2. *esiste una proposizione* $P$ *tale che* $\Gamma \vdash P$ *e* $\Gamma \vdash \neg P$*;*
3. *per ogni proposizione* $P$*,* $\Gamma \vdash P$*.*

*Dimostrazione.*

1. (1. $\Rightarrow$ 3.). Da $\Gamma \vdash \bot$, applicando $(\bot\ e)$, si ottiene una dimostrazione di $\Gamma \vdash P$ per ogni $P$.

2. (3. $\Rightarrow$ 2.). Immediata.

3. (2. $\Rightarrow$ 1.). Per ipotesi esistono le dimostrazioni di $\Gamma \vdash P$ e $\Gamma \vdash \neg P$; applicando $(\bot i)$ si ottiene una dimostrazione di $\Gamma \vdash \bot$.

□

**Lemma 3.5** *Se un insieme di proposizioni* $\Gamma$ *è soddisfacibile, allora è consistente.*

*Dimostrazione.* Supponiamo che $\Gamma$ sia soddisfacibile, cioè esiste un'interpretazione $v$ tale che $v(P) = 1$, $\forall P \in \Gamma$. Ovviamente $v$ non può soddisfare $\bot$, in quanto per definizione di interpretazione deve essere $v(\bot) = 0$. Allora, $\Gamma \not\models \bot$ che, per il Teorema di correttezza, implica $\Gamma \nvdash \bot$. Dunque $\Gamma$ è consistente. □

**Lemma 3.6**

1. $\Gamma, \neg P \vdash \bot \ \Rightarrow\ \Gamma \vdash P$*;*
2. $\Gamma, P \vdash \bot \ \Rightarrow\ \Gamma \vdash \neg P$*.*

*Dimostrazione.* Seguono, rispettivamente, da $(RAA)$ e $(\rightarrow i)$. $\square$

**Definizione 3.7** *Un insieme* $\Gamma$ *è detto* consistente massimale *se e solo se:*

1. $\Gamma$ *è consistente;*

2. *se* $\Gamma \subseteq \Gamma'$ *e* $\Gamma'$ *è consistente, allora* $\Gamma = \Gamma'$.

Per dimostrare la completezza delle regole della Deduzione Naturale, proveremo che:

- ogni insieme di proposizioni consistente è soddisfacibile (Teorema 3.11); per fare ciò, sarà necessario dimostrare delle semplici proprietà degli insiemi consistenti massimali (Lemmi 3.8 e 3.9) e che

- ogni insieme consistente è contenuto in uno consistente massimale (Teorema 3.10).

**Lemma 3.8** *Ogni insieme consistente massimale è chiuso rispetto alla derivabilità, cioè se* $\Gamma \vdash P$ *allora* $P \in \Gamma$.

*Dimostrazione.* Supponiamo $\Gamma \vdash P$ e $P \notin \Gamma$. Poichè per ipotesi $\Gamma$ è consistente massimale, $\Gamma \cup \{P\}$ deve essere inconsistente. Ma per il Lemma 3.6 $\Gamma \cup \{P\} \vdash \bot$ implica $\Gamma \vdash \neg P$. Dunque $\Gamma$ sarebbe inconsistente, in quanto permetterebbe di derivare sia $P$ che $\neg P$, contraddicendo l'ipotesi. $\square$

**Lemma 3.9** *Sia* $\Gamma$ *un insieme consistente massimale. Per tutte le proposizioni* $P$ *e* $Q$*:*

1. $P \in \Gamma$ *oppure* $\neg P \in \Gamma$*;*

2. $P \wedge Q \in \Gamma \Leftrightarrow P \in \Gamma$ *e* $Q \in \Gamma$*;*

3. $P \vee Q \in \Gamma \Leftrightarrow P \in \Gamma$ *o* $Q \in \Gamma$*;*

4. $P \rightarrow Q \in \Gamma \Leftrightarrow (\neg P \in \Gamma$ o $Q \in \Gamma)$.

*Dimostrazione.*

1. Sia $\Gamma' = \Gamma \cup \{P\}$. Se $\Gamma'$ è inconsistente, allora per i Lemmi 3.6 e 3.8 $\neg P \in \Gamma$. Viceversa, se $\Gamma'$ è consistente, $\Gamma' = \Gamma$ per la massimalità di quest'ultimo, e dunque $P \in \Gamma$.

2. Supponiamo che $P \wedge Q \in \Gamma$. Poiché $P \wedge Q \vdash P$, $P \wedge Q \vdash Q$ e $\Gamma$ è chiuso rispetto alla derivabilità, $P \in \Gamma$ e $Q \in \Gamma$. Viceversa, se $P \in \Gamma$ e $Q \in \Gamma$ deve essere $P \wedge Q \in \Gamma$, poichè $P, Q \vdash P \wedge Q$ e $\Gamma$ è chiuso rispetto alla derivabilità.

3. Supponiamo che $P \vee Q \in \Gamma$, e dimostriamo che se $P \notin \Gamma$ allora necessariamente $Q \in \Gamma$. Per il punto 1. se $P \notin \Gamma$, allora $\neg P \in \Gamma$. Il seguente albero mostra che $P \vee Q, \neg P \vdash Q$:

$$\dfrac{P \vee Q \qquad \dfrac{\dfrac{[P] \quad \neg P}{\bot}}{Q} \qquad [Q]}{Q}$$

Essendo $\Gamma$ chiuso rispetto alla derivabilità, $Q \in \Gamma$. Viceversa, se $P \in \Gamma$ o $Q \in \Gamma$, allora $P \vee Q \in \Gamma$ in quanto $P \vdash P \vee Q$, $Q \vdash P \vee Q$ e $\Gamma$ è chiuso rispetto alla derivabilità.

4. Supponiamo che $P \to Q \in \Gamma$. Vogliamo provare che $\neg P \in \Gamma$ oppure $Q \in \Gamma$. Se $\neg P \in \Gamma$, l'asserto è verificato; in caso contrario, per il punto 1., risulta $P \in \Gamma$; essendo $Q$ derivabile da $P \to Q, P$ e $\Gamma$ è chiuso rispetto alla derivabilità, segue che $Q \in \Gamma$. Viceversa, supponiamo che $\neg P \in \Gamma$ o $Q \in \Gamma$, vogliamo provare che $P \to Q \in \Gamma$. Se per assurdo $P \to Q \notin \Gamma$, allora, per il punto 1., $\neg(P \to Q) \in \Gamma$, da cui $\Gamma \vdash \neg(P \to Q)$; se $\neg P \in \Gamma$, è facile provare che $\neg P \vdash P \to Q$ (il lettore lo faccia per esercizio), quindi $\Gamma \vdash P \to Q$ e dunque $\Gamma$ è inconsistente; se $Q \in \Gamma$, con un'applicazione di $(\to i)$, $Q \vdash P \to Q$, da cui segue che $\Gamma$ è inconsistente.

□

**Teorema 3.10** *Ogni insieme consistente $\Gamma$ è contenuto in uno consistente massimale.*

*Dimostrazione.* Sia $P_0, P_1, \cdots$ un'enumerazione di tutte le formule ben formate. Definiamo una successione non decrescente di insiemi $\{\Gamma_i\}_{i \in \mathcal{N}}$ nel seguente modo:

- $\Gamma_0 = \Gamma$
- $\Gamma_{n+1} = \begin{cases} \Gamma_n \cup \{P_n\} & \text{se } \Gamma_n \cup \{P_n\} \text{ è consistente} \\ \Gamma_n & \text{altrimenti} \end{cases}$

Sia $\Gamma^* = \bigcup_{i=0}^{\infty} \{\Gamma_i\}$. Vogliamo provare che $\Gamma^*$ è consistente massimale. Dimostriamone anzitutto la consistenza. Supponiamo che $\Gamma^* \vdash \bot$, allora esiste un albero di prova che permette di derivare $\bot$ da un insieme *finito* $\Delta$ di ipotesi in $\Gamma^*$. Necessariamente esiste un qualche $\Gamma_k$ che contiene $\Delta$, per cui risulta $\Gamma_k \vdash \bot$, ma questo è assurdo in quanto tutti i $\Gamma_i$ sono insiemi consistenti per definizione. Proviamo ora che $\Gamma^*$ è massimale. Supponiamo che esista un insieme consistente $\Gamma'$ tale che $\Gamma^* \subseteq \Gamma'$. Sia $P_m$ una formula in $\Gamma'$. Poiché $\Gamma_m \subseteq \Gamma^* \subseteq \Gamma'$, e $\Gamma'$ è consistente, allora $\Gamma_m \cup \{P_m\}$ è consistente, e quindi $\Gamma_{m+1} = \Gamma_m \cup \{P_m\}$, che implica $P_m \in \Gamma^*$. Dunque $\Gamma^* = \Gamma'$. □

**Teorema 3.11** *Un insieme di proposizioni $\Gamma$ è consistente se e solo se è soddisfacibile.*

*Dimostrazione.* ($\Rightarrow$) Per il Teorema 3.10, $\Gamma$ è contenuto in uno consistente massimale $\Gamma^*$. Sia $v$ l'interpretazione definita nel seguente modo: per ogni formula atomica $A$, $v(A) = 1 \Leftrightarrow A \in \Gamma^*$; proviamo che per ogni proposizione $P$, $v(P) = 1$ se e solo se $P \in \Gamma^*$. Da ciò segue che $v$ è un modello per $\Gamma^*$ e quindi per $\Gamma$. Per induzione sulla struttura di $P$.
*(caso base)* Se $P$ è una proposizione atomica l'asserto è vero per come è definita l'interpretazione $v$. Veniamo al caso induttivo.

1. Se $P = \neg P_1$. Allora

$$\begin{array}{lcll} v(\neg P_1) = 1 & \Leftrightarrow & v(P_1) = 0 & \\ & \Leftrightarrow & P_1 \notin \Gamma^* & \text{ipotesi induttiva} \\ & \Leftrightarrow & \neg P_1 \in \Gamma^* & \text{Lemma 3.9.1} \end{array}$$

2. Se $P = P_1 \wedge P_2$. Allora

$$\begin{array}{lcll} v(P_1 \wedge P_2) = 1 & \Leftrightarrow & v(P_1) = v(P_2) = 1 & \\ & \Leftrightarrow & P_1 \in \Gamma^* \text{ e } P_2 \in \Gamma^* & \text{ipotesi induttiva} \\ & \Leftrightarrow & P_1 \wedge P_2 \in \Gamma^* & \text{Lemma 3.9.2} \end{array}$$

3. Se $P = P_1 \vee P_2$. Allora

$$\begin{array}{lcll} v(P_1 \vee P_2) = 1 & \Leftrightarrow & v(P_1) = 1 \text{ oppure } v(P_2) = 1 & \\ & \Leftrightarrow & P_1 \in \Gamma^* \text{ oppure } P_2 \in \Gamma^* & \text{ipotesi induttiva} \\ & \Leftrightarrow & P_1 \vee P_2 \in \Gamma^* & \text{Lemma 3.9.3} \end{array}$$

4. Se $P = P_1 \rightarrow P_2$. Allora

$$\begin{array}{lcll} v(P_1 \rightarrow P_2) = 1 & \Leftrightarrow & v(P_1) = 0 \text{ oppure } v(P_2) = 1 & \\ & \Leftrightarrow & P_1 \notin \Gamma^* \text{ oppure } P_2 \in \Gamma^* & \text{ipotesi induttiva} \\ & \Leftrightarrow & \neg P_1 \in \Gamma^* \text{ oppure } P_2 \in \Gamma^* & \text{Lemma 3.9.1} \\ & \Leftrightarrow & P_1 \rightarrow P_2 \in \Gamma^* & \text{Lemma 3.9.4} \end{array}$$

Dunque $v(P) = 1\ \forall P \in \Gamma^*$; essendo $\Gamma \subseteq \Gamma^*$, $v$ è un modello per $\Gamma$.
($\Leftarrow$) Segue dal Lemma 3.5. $\square$

**Teorema 3.12 (Completezza)**
*Se* $\Gamma \models P$ *allora* $\Gamma \vdash P$.

*Dimostrazione.* Se $\Gamma \models P$ allora, per definizione di conseguenza semantica, per ogni interpretazione $v$, da $v(\Gamma) = 1$ segue $v(P) = 1$ (e quindi $v(\neg P) = 0$); dunque non può esistere un'interpretazione che soddisfa $\Gamma, \neg P$. Per il teorema precedente, $\Gamma, \neg P$ è inconsistente, da cui segue, per il Lemma 3.6, che $\Gamma \vdash P$. $\square$

Abbiamo dunque dimostrato che ciò che è conseguenza semantica di un dato insieme di premesse, è deducibile da queste mediante le regole della Deduzione Naturale.
Il teorema precedente è spesso indicato, in letteratura, come Teorema di completezza forte, per distinguerlo da quello di completezza debole, la cui formulazione è la seguente:

**Teorema 3.13 (Completezza debole)**
$\models P$ *se* $\vdash P$.

*Dimostrazione.* Segue dal Teorema di completezza. □

Il suesposto teorema, unitamente al Corollario 3.2, assicura che l'insieme delle formule derivabili in Deduzione Naturale coincide con quello delle tautologie. Dunque nella logica proposizionale, trarre delle inferenze per via sintattica o per via semantica è del tutto equivalente. Alla luce di ciò possiamo concludere che, data una qualunque formula della logica proposizionale, si è in grado di stabilire se questa è un teorema o meno (basta costruire la sua tabella di verità e controllare che l'ultima colonna sia costituita solo da "1").
Osserviamo che procedendo in tal modo non si *ottiene* una dimostrazione della formula, ma si stabilisce solamente che tale dimostrazione *esiste*.
In particolare, la prova del Teorema 3.13 *non* è *costruttiva*. Nei paragrafi 3.2 e 3.3 verranno presentate due diverse dimostrazioni della completezza[1] del calcolo proposizionale, che, al contrario, sono costruttive.
Come importante corollario del Teorema di completezza (forte), si ottiene una dimostrazione alternativa del Teorema di compattezza, già provato in altro modo nel primo capitolo:

**Corollario 3.14 (Teorema di compattezza)**
$\Gamma \models P$ *se e solo se esiste un sottoinsieme finito* $\Delta$ *di* $\Gamma$ *tale che* $\Delta \models P$.

*Dimostrazione.* ($\Leftarrow$) Immediata.
($\Rightarrow$) Supponiamo che $\Gamma \models P$; per il Teorema di completezza $\Gamma \vdash P$, dunque esiste un albero di prova che permette di derivare $P$ da un insieme finito $\Delta$ di ipotesi in $\Gamma$; per il Teorema di correttezza $\Delta \models P$. □

## 3.2 Sistema Assiomatico

Avendo dimostrato che la nozione di derivabilità in Deduzione Naturale coincide con quella nel Sistema Assiomatico, la correttezza e la completezza di quest'ultimo seguono come ovvio corollario. Presentiamo tuttavia una dimostrazione diretta di questo fatto, per evidenziare i molteplici approcci possibili al problema.
Come nel caso della Deduzione Naturale, la correttezza del calcolo assiomatico risulta semplice: basta provare che gli assiomi sono tautologie (sappiamo già che la regola di modus ponens è semanticamente corretta). Tale verifica è banale, ed è lasciata al lettore come esercizio.
Veniamo dunque alla completezza. Cominciamo con il dimostrare il seguente lemma:

**Lemma 3.15** *Sia* $P$ *una formula ben formata ed* $A_1, \ldots, A_n$ *le proposizioni atomiche che vi compaiono. Sia* $v$ *un'interpretazione. Definiamo, per ogni*

[1] Finita.

*proposizione $Q$,*

$$\overline{Q} = \begin{cases} Q & \text{se } v(Q) = 1 \\ \neg Q & \text{se } v(Q) = 0 \end{cases}$$

*Allora:*

$$\overline{A_1}, \ldots, \overline{A_n} \vdash \overline{P}$$

*Dimostrazione.* Per induzione sulla struttura di $P$.

- $P = A_i$ con $i \in \{1, \cdots, n\}$. Dobbiamo dimostrare che $\overline{A_i} \vdash \overline{A_i}$, ma questo è banalmente vero, sia che $\overline{A_i} = A_i$, sia che $\overline{A_i} = \neg A_i$.

- $P = \bot$. In questo caso $\overline{P} = \neg\bot$. Come istanza di $(\neg\ i)$ abbiamo $\vdash (\bot \to \bot) \to \neg\bot$. Poichè da $\mathrm{I} \vdash \bot \to \bot$, per modus ponens risulta $\vdash \neg\bot$.

- $P = \neg Q$. Sia $v$ una generica interpretazione. Consideriamo due sottocasi, a seconda che $v(P) = 1$ o $v(P) = 0$. Se $v(P) = 1$, allora $v(Q) = 0$ e, per definizione, $\overline{Q} = \neg Q$. Per ipotesi induttiva,

$$\overline{A_1}, \ldots, \overline{A_n} \vdash \overline{Q}$$

Ma $\overline{Q} = \neg Q = P = \overline{P}$, e dunque

$$\overline{A_1}, \ldots, \overline{A_n} \vdash \overline{P}$$

Viceversa, supponiamo che $v(P) = 0$. In questo caso $v(Q) = 1$ e $\overline{Q} = Q$. Per ipotesi induttiva,

$$\overline{A_1}, \ldots, \overline{A_n} \vdash Q$$

Osserviamo ora che $Q \vdash \neg\neg Q$, come mostrato dal seguente albero:

$$\dfrac{\dfrac{Q \qquad Q \to (\neg Q \to \bot)\ \ (\neg e)}{\neg Q \to \bot} \qquad\qquad (\neg Q \to \bot) \to \neg\neg Q\ \ (\neg i)}{\neg\neg Q}$$

Componendo i due alberi di prova precedenti, si ricava dunque una dimostrazione di

$$\overline{A_1}, \ldots, \overline{A_n} \vdash \neg\neg Q = \overline{P}$$

- $P = Q \wedge R$. Come in precedenza, distinguiamo due sottocasi. Se $v(P) = v(Q \wedge R) = 1$ allora $v(Q) = v(R) = 1$. Per ipotesi induttiva risulta

$$\overline{A_1}, \ldots, \overline{A_n} \vdash Q \qquad \overline{A_1}, \ldots, \overline{A_n} \vdash R$$

Componendo i due alberi di prova precedenti con il seguente albero

$$\dfrac{R \qquad \dfrac{Q \qquad Q \to (R \to (Q \wedge R))\ \ (\wedge i)}{R \to (Q \wedge R)}}{Q \wedge R}$$

otteniamo la dimostrazione di

$$\overline{A_1}, \ldots, \overline{A_n} \vdash Q \wedge R = \overline{P}$$

Viceversa, se $v(P) = 0$ allora $v(Q) = 0$ oppure $v(R) = 0$. Esaminiamo solo la prima possibilità (l'altro caso è simmetrico). Se $v(Q) = 0$, allora $\overline{Q} = \neg Q$ e per ipotesi induttiva

$$\overline{A_1}, \ldots, \overline{A_n} \vdash \neg Q$$

Consideriamo ora il seguente albero:

$$\dfrac{\dfrac{\dfrac{Q \wedge R \qquad Q \wedge R \to Q}{Q} \qquad Q \to (\neg Q \to \bot)}{\neg Q \to \bot} \qquad \neg Q}{\bot}$$

Questo rappresenta una dimostrazione di $\neg Q, Q \wedge R \vdash \bot$. Per il Teorema di deduzione, esiste una dimostrazione di $\neg Q \vdash Q \wedge R \to \bot$ da cui, mediante una applicazione di $(\neg i)$, otteniamo una dimostrazione di $\neg Q \vdash \neg(Q \wedge R)$. Componendo quest'ultima con $\overline{A_1}, \ldots, \overline{A_n} \vdash \neg Q$ otteniamo la dimostrazione desiderata di $\overline{A_1}, \ldots, \overline{A_n} \vdash \neg(Q \wedge R) = \overline{P}$.

- $P = Q \vee R$. Come è stato fatto in precedenza, distinguiamo due sottocasi. Se $v(P) = 1$ allora $v(Q) = 1$ oppure $v(R) = 1$. Consideriamo il primo caso (il rimanente è simmetrico). Se $v(Q) = 1$ allora $\overline{Q} = Q$ e per ipotesi induttiva esiste una dimostrazione di

$$\overline{A_1}, \ldots, \overline{A_n} \vdash Q$$

Da questa, utilizzando l'assioma $(\vee i)$, mediante un'applicazione di modus ponens, otteniamo una dimostrazione di

$$\overline{A_1}, \ldots, \overline{A_n} \vdash Q \vee R = \overline{P}$$

Viceversa, supponiamo che $v(P) = 0$. Allora deve essere $v(Q) = 0$ e $v(R) = 0$, quindi $\overline{Q} = \neg Q$ e $\overline{R} = \neg R$. Per ipotesi induttiva esistono due dimostrazioni di

$$\overline{A_1}, \ldots, \overline{A_n} \vdash \neg Q \qquad \overline{A_1}, \ldots, \overline{A_n} \vdash \neg R$$

Da queste, mediante un'applicazione di modus ponens ad istanze di $(\neg e)$, otteniamo due dimostrazioni di

$$\overline{A_1}, \ldots, \overline{A_n} \vdash Q \to \bot \qquad \overline{A_1}, \ldots, \overline{A_n} \vdash R \to \bot$$

e dall'assioma $(Q \to \bot) \to ((R \to \bot) \to ((Q \vee R) \to \bot))$, con due ulteriori passi di modus ponens, si ricava una dimostrazione di

$$\overline{A_1}, \ldots, \overline{A_n} \vdash (Q \vee R) \to \bot$$

Infine, dall'assioma $((Q \vee R) \to \bot) \to \neg(Q \vee R)$ si ottiene, mediante modus ponens, la dimostrazione cercata di

$$\overline{A_1}, \ldots, \overline{A_n} \vdash \neg(Q \vee R) = \overline{P}$$

- $P = Q \to R$. Distinguiamo tre casi:

  1. Se $v(R) = 1$ allora $v(P) = 1$, $\overline{R} = R$ e $\overline{P} = Q \to R$. Per ipotesi induttiva esiste una dimostrazione di

     $$\overline{A_1}, \ldots, \overline{A_n} \vdash R$$

     Dall'assioma $R \to (Q \to R)$, mediante una applicazione di modus ponens, si ottiene una dimostrazione di

     $$\overline{A_1}, \ldots, \overline{A_n} \vdash Q \to R = \overline{P}$$

  2. Se $v(Q) = 0$ allora $v(P) = 1$, $\overline{Q} = \neg Q$ e $\overline{P} = Q \to R$. Per ipotesi induttiva esiste una dimostrazione di

     $$\overline{A_1}, \ldots, \overline{A_n} \vdash \neg Q$$

     Consideriamo il seguente albero

     $$\dfrac{\dfrac{\dfrac{Q \to (\neg Q \to \bot) \qquad Q}{\neg Q \to \bot} \qquad \neg Q}{\bot} \qquad \bot \to R}{R}$$

     Questo rappresenta una dimostrazione di $\neg Q, Q \vdash R$. Per il Teorema di deduzione esiste una dimostrazione di $\neg Q \vdash Q \to R$, che, composta con $\overline{A_1}, \ldots, \overline{A_n} \vdash \neg Q$, fornisce la dimostrazione desiderata di

     $$\overline{A_1}, \ldots, \overline{A_n} \vdash Q \to R = \overline{P}$$

  3. Se $v(Q) = 1$ e $v(R) = 0$ allora $v(P) = 0$, $\overline{Q} = Q$, $\overline{R} = \neg R$ e $\overline{P} = \neg(Q \to R)$. Per ipotesi induttiva esistono due dimostrazioni di

     $$\overline{A_1}, \ldots, \overline{A_n} \vdash Q \qquad \overline{A_1}, \ldots, \overline{A_n} \vdash \neg R$$

     È sufficiente dimostrare che $Q, \neg R \vdash \neg(Q \to R)$. Proviamo innanzi tutto che $Q, \neg R, Q \to R \vdash \bot$. Questo è espresso dal seguente albero di prova:

     $$\dfrac{\dfrac{\dfrac{Q \qquad Q \to R}{R} \qquad R \to (\neg R \to \bot)}{\neg R \to \bot} \qquad \neg R}{\bot}$$

Per il Teorema di deduzione, esiste una dimostrazione di $Q, \neg R \vdash (Q \to R) \to \bot$, e da questa, utilizzando l'assioma

$$((Q \to R) \to \bot) \to \neg(Q \to R)$$

mediante una applicazione di modus ponens, si ottiene la dimostrazione desiderata di

$$\overline{A_1}, \ldots, \overline{A_n} \vdash \neg(Q \to R) = \overline{P}$$

□

**Teorema 3.16 (Completezza debole)**
*Per ogni proposizione $P$, se $\models P$ allora $\vdash P$.*

*Dimostrazione.* Sia $\models P$ e siano $A_1, \ldots, A_n$ le proposizioni atomiche che vi compaiono. Per il lemma precedente, per ogni interpretazione $v$,

$$\overline{A_1}, \ldots, \overline{A_{n-1}}, \overline{A_n} \vdash \overline{P}$$

Essendo per ipotesi $P$ una tautologia, $v(P) = 1$ e quindi

$$\overline{A_1}, \ldots, \overline{A_{n-1}}, \overline{A_n} \vdash P$$

In particolare, se $v(A_n) = 1$ risulta

$$\overline{A_1}, \ldots, \overline{A_{n-1}}, A_n \vdash P$$

Mentre se $v(A_n) = 0$ risulta

$$\overline{A_1}, \ldots, \overline{A_{n-1}}, \neg A_n \vdash P$$

Per il Teorema di deduzione, esistono due dimostrazioni

$$\overline{A_1}, \ldots, \overline{A_{n-1}} \vdash A_n \to P \qquad \overline{A_1}, \ldots, \overline{A_{n-1}} \vdash \neg A_n \to P$$

Consideriamo il seguente albero:

$$\dfrac{\dfrac{A_n \vee \neg A_n \qquad \dfrac{\neg A_n \to P \qquad \dfrac{A_n \to P \qquad (A_n \to P) \to ((\neg A_n \to P) \to (A_n \vee \neg A_n \to P))}{(\neg A_n \to P) \to (A_n \vee \neg A_n \to P)}}{A_n \vee \neg A_n \to P}}{P}$$

Componendolo con le due dimostrazioni precedenti e con la dimostrazione di $A_n \vee \neg A_n$ (Esempio 2.5) si ottiene, per ogni $v$, una dimostrazione di

$$\overline{A_1}, \ldots, \overline{A_{n-1}} \vdash P$$

Abbiamo dunque eliminato l'ipotesi $\overline{A_n}$. Iterando il procedimento appena descritto sulle restanti ipotesi si ottiene una dimostrazione di $\vdash P$. □

***Osservazione*** Tale teorema viene chiamato Teorema di completezza debole per distinguerlo dal Teorema di completezza, la cui formulazione fa riferimento alla nozione di conseguenza logica.

Osserviamo che la prova del Teorema 3.16 è *costruttiva* in quanto seguendone i vari passi è possibile ricavare una dimostrazione per $P$.

**Teorema 3.17 (Completezza finita)**
*Sia* $\Gamma$ *un insieme finito di proposizioni, se* $\Gamma \models P$ *allora* $\Gamma \vdash P$.

*Dimostrazione.* Segue dai teoremi di deduzione e di completezza debole. □

Per provare il Teorema di completezza (forte) è necessario utilizzare il Teorema di compattezza (Teorema 1.18).

**Teorema 3.18 (Completezza)**
*Se* $\Gamma \models P$ *allora* $\Gamma \vdash P$.

*Dimostrazione.* Segue dal Teorema di completezza finita e dal Teorema di compattezza. □

## 3.3 Calcolo dei Sequenti

Al fine di enunciare in modo elegante i Teoremi di correttezza e completezza per il Calcolo dei Sequenti è opportuno generalizzare la nozione di soddisfacibilità semantica nel modo seguente:

**Definizione 3.19** *Dati due insiemi di formule* $\Gamma$ *e* $\Delta$*, si dice che* $\Delta$ *è* soddisfatto in $\Gamma$ *(*$\Gamma \models \Delta$*), se e solo se* $\forall v \;\; ((\forall P_i \in \Gamma \; v(P_i) = 1) \Rightarrow \exists Q \in \Delta, v(Q) = 1)$.

In questa sezione considereremo la formalizzazione II del calcolo.

**Teorema 3.20 (Correttezza)**
*Se* $\Gamma \vdash \Delta$ *allora* $\Gamma \models \Delta$.

*Dimostrazione.* Al solito, la dimostrazione consiste nel verificare che gli assiomi sono validi e che le regole di inferenza preservano la validità. Nel nostro caso l'unico assioma è $A \vdash A$, ed ovviamente $A \models A$.
Il fatto che le regole strutturali preservano la validità è immediato. Nel dimostrare che le regole di inferenza logiche fanno altrettanto è sufficiente considerare il caso in cui tutti i parametri sulla sinistra della conclusione hanno valore 1, e quelli sulla destra hanno valore 0, poichè, in caso contario, l'asserto è banalmente verificato. La dimostrazione procede per ispezione delle varie regole, osservando che le ipotesi di validità delle formule ausiliarie forzano un valore di verità per la formula principale che rende valido il sequente. Presentiamone

solo alcune, le rimanenti sono lasciate al lettore come esercizio. Consideriamo la regola

$$(\vee l) \quad \frac{\Gamma, A \vdash \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}$$

Se i parametri in $\Gamma$ hanno valore 1 e quelli in $\Delta$ valore 0, l'unico caso in cui $\Gamma, A \models \Delta$ e $\Gamma, B \models \Delta$ è che $v(A) = v(B) = 0$. Ma in tale situazione, anche $v(A \vee B) = 0$ e dunque $\Gamma, A \vee B \models \Delta$.
Consideriamo la regola

$$(\rightarrow r) \quad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \rightarrow B, \Delta}$$

Se i parametri in $\Gamma$ hanno valore 1, quelli in $\Delta$ valore 0, e $\Gamma, A \models B, \Delta$ vi sono solo due possibilità (non esclusive): o $v(A) = 0$ oppure $v(B) = 1$. In entrambi i casi $v(A \rightarrow B) = 1$ e dunque $\Gamma \models A \rightarrow B, \Delta$. $\square$

Naturalmente, il Teorema di correttezza vale anche per la formulazione I del Calcolo dei Sequenti. L'interesse della formulazione II, in questo caso, è che non solo le regole preservano la validità, ma anche la non validità. In altri termini, preservano la validità anche se lette "all'indietro" (scambiando cioè premesse e conclusioni).

**Teorema 3.21** *Per ogni regola di inferenza della formulazione II del Calcolo dei Sequenti, le premesse sono valide se e solo se la conclusione è valida.*

*Dimostrazione.* ($\Rightarrow$) Segue dal teorema precedente.
($\Leftarrow$) Considereremo solo alcune regole, lasciando le restanti al lettore.

$$(\rightarrow l) \quad \frac{\Gamma \vdash A, \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \rightarrow B \vdash \Delta}$$

Supponiamo che $\Gamma, A \rightarrow B \models \Delta$. Vogliamo dimostrare che $\Gamma \models A, \Delta$ e $\Gamma, B \models \Delta$. L'unico caso interessante è quando tutte le formule in $\Gamma$ hanno valore 1 e quelle in $\Delta$ valore 0. In tale situazione, poichè per ipotesi $\Gamma, A \rightarrow B \models \Delta$, deve essere $v(A \rightarrow B) = 0$. Questo implica che $v(A) = 1$ e $v(B) = 0$, e quindi ovviamente $\Gamma \models A, \Delta$ e $\Gamma, B \models \Delta$.

$$(\vee r) \quad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta}$$

Al solito, ci si restringe al caso in cui tutte le formule in $\Gamma$ hanno valore 1 e quelle in $\Delta$ valore 0. Se $\Gamma \models A \vee B, \Delta$, deve essere $v(A \vee B) = 1$, per cui o $v(A) = 1$ oppure $v(B) = 1$. In entrambi i casi $\Gamma \models A, B, \Delta$. $\square$

***Osservazione*** Il teorema precedente non vale nel caso della formulazione I del Calcolo dei Sequenti. Infatti, consideriamo la regola

$$(\vee r) \quad \frac{\Gamma \vdash A, \Delta}{\Gamma \vdash A \vee B, \Delta}$$

supponiamo che tutte le formule in $\Gamma$ hanno valore 1, quelle in $\Delta$ valore 0 ed inoltre $v(A) = 0$ e $v(B) = 1$. Allora $\Gamma \models A \vee B, \Delta$, ma $\Gamma \not\models A, \Delta$.

**Teorema 3.22 (Completezza finita)**
*Sia* $\Gamma$ *un insieme finito di proposizioni, se* $\Gamma \models \Delta$ *allora* $\Gamma \vdash \Delta$.

*Dimostrazione.* Supponiamo che $\Gamma \models \Delta$. A partire dal sequente $\Gamma \vdash \Delta$ costruiamo all'indietro un albero di derivazione applicando regole logiche fino a ridursi a sequenti con solo formule atomiche. Poiché ogni premessa di una regola logica contiene meno connettivi della conclusione, il procedimento termina necessariamente. Dunque, dopo un numero finito di passi, ci si riduce ad un albero le cui foglie sono sequenti del tipo $\Gamma' \vdash \Delta'$ dove tutte le formule in $\Gamma'$ e $\Delta'$ sono atomiche. Poiché le regole del calcolo preservano la validità anche quando sono applicate all'indietro, tutti questi sequenti devono essere validi. Ma l'unica possibilità affinché $\Gamma' \models \Delta'$ è che $\Gamma'$ e $\Delta'$ contengano una sottoformula in comune (in caso contrario è banale trovare un'interpretazione $v$ tale che $v(\Gamma') = 1$ e $v(\Delta') = 0$). Tuttavia, se $\Gamma'$ e $\Delta'$ contengono una formula in comune $A$, il sequente $\Gamma' \vdash \Delta'$ è facilmente derivabile dall'assioma $A \vdash A$ per indebolimenti e permutazioni. Dunque possiamo sempre completare l'albero di prova, ottenendo una dimostrazione di $\Gamma \vdash \Delta$. □

Come nel paragrafo precedente, per provare il Teorema di completezza (forte) è necessario utilizzare il Teorema di compattezza (Teorema 1.18).

**Teorema 3.23 (Completezza)**
*Se* $\Gamma \models P$ *allora* $\Gamma \vdash P$.

*Dimostrazione.* Segue dal Teorema di completezza finita e dal Teorema di compattezza. □

**Corollario 3.24** *Le regole di contrazione sono ridondanti nella formulazione II del Calcolo dei Sequenti.*

Dunque in tale formulazione, le regole di contrazione sono "internalizzate" in quelle logiche.

**Corollario 3.25** *Il Calcolo dei Sequenti è logicamente equivalente alla Deduzione Naturale ed al Sistema Assiomatico.*

La dimostrazione del Teorema di completezza (finita) mostra, tra le altre cose, la ridondanza della regola di taglio nel caso della formulazione II del Calcolo dei Sequenti. Tuttavia, come già accennato in precedenza, l'interesse del teorema non è tanto nell'*esistenza* di una dimostrazione senza tagli per un dato sequente, ma nel procedimento effettivo di eliminazione che permette di trasformare attraverso riscritture elementari, una dimostrazione con tagli in una, priva di tagli, ad essa equivalente.

## 3.4 Cenni storici e bibliografici

I primi calcoli logici sono stati introdotti allo scopo di sviluppare al loro *interno* tutti i possibili ragionamenti. Per tale motivo, inizialmente, non si è cercata una giustificazione di tali calcoli all'*esterno* degli stessi, in termini del significato delle formule. Bernays ([Ber26]) è stato il primo ad utilizzare nozioni semantiche per giustificare i sistemi di Whitehead e Russel fornendo per essi una dimostrazione di completezza. La prima dimostrazione pubblicata e per lungo tempo, la più influente, è stata quella di Post ([Pos21]). Questi ha mostrato che ogni formula ben formata è equivalente (semanticamente) ad una in forma normale disgiuntiva. Quindi ha aggiunto degli assiomi al sistema del *Principia Mathematica* in modo tale da produrre, per ogni formula valida $P$, la forma normale disgiuntiva, $P^d$, equivalente ad essa ed una derivazione $P^d \to P$ ($\models P \Rightarrow \vdash P^d$). Unendo le due dimostrazioni si ottiene una prova[2] *costruttiva* di $P$. Successivamente è stato provato che gli assiomi aggiunti da Post erano in realtà derivabili da quelli già presenti nel *Principia Mathematica*. La dimostrazione di completezza presentata nel paragrafo 3.1 è dovuta ad Henkin ([Hen54]). La prova di Post presenta due svantaggi rispetto a quest'ultima: non può essere facilmente generalizzata a calcoli logici di altra natura e non consente di dimostrare la completezza (forte), che richiederebbe assunzioni non costruttive. Le dimostrazioni dei paragrafi 3.2 e 3.3 sono rispettivamente dovute a Kalmar ([Kal35]) e Ketonen (si veda [Cur63]). Per una panoramica sulle molteplici prove costruttive del Teorema di completezza si veda [Sur73].

## Esercizi

**3.1** Stabilire quali dei seguenti insiemi sono consistenti:

1. $\{A \to B, C \to D, D \to E, E \to \neg A\}$
2. $\{A \vee B \to C, A, B \to \neg C, B\}$
3. $\{A \vee B \to C, A, B \to \neg C, C\}$

**3.2** Dimostrare che un insieme $\Gamma$ è consistente massimale se e solo se esiste un'unica interpretazione $v$ tale che $v(P) = 1,\ \forall P \in \Gamma$.

**3.3** Provare che un insieme $\Gamma$ è consistente massimale se e solo se sono verificate le seguenti condizioni:

1. $\Gamma$ è consistente;
2. Per ogni proposizione $P$, $P \in \Gamma$ oppure $\neg P \in \Gamma$.

**3.4** Mostrare che se $\alpha \to \beta \in \Gamma$ e $\Gamma, \neg\alpha$ e $\Gamma, \beta$ sono inconsistenti, allora $\Gamma$ è inconsistente.

[2] Teorema di completezza debole.

**3.5** Provare che l'insieme $\Gamma = \{P \mid v(P) = 1\}$, dove $v$ è un'interpretazione fissata, è un insieme consistente massimale.

**3.6** Sia $\Gamma$ un insieme consistente massimale; provare che la funzione $v : FBF \to \{0, 1\}$ definita nel seguente modo: per ogni proposizione $P$, $v(P) = 1 \Leftrightarrow P \in \Gamma$, è un'interpretazione (cioè verifica le condizioni stabilite nella Definizione 1.5).

**3.7** Dimostrare che $\Gamma \nvdash Q$ se e solo se esiste un'interpretazione $v$ tale che $v(P) = 1$ per ogni $P \in \Gamma$ e $v(Q) = 0$.

**3.8** Riformulare le dimostrazioni del paragrafo 3.1 in *un* Sistema Assiomatico e nel Calcolo dei Sequenti.

**3.9** Provare la completezza dei Sistemi Assiomatici menzionati in 2.3.2.

**3.10** Riformulare le dimostrazioni del paragrafo 3.2 in Deduzione Naturale e nel Calcolo dei Sequenti.

**3.11** Mostrare che le seguenti affermazioni sono equivalenti:

1. $A \vdash B$
2. $A \models B$
3. $\models A \to B$
4. $\vdash A \to B$

**3.12** Provare che le seguenti proposizioni sono dei teoremi:

1. $\neg(A \wedge \neg A)$
2. $(A \vee B) \to (B \vee A)$
3. $(\neg A \to A) \to A$
4. $A \to (B \to B)$
5. $(\neg A \wedge A) \to B$
6. $(\neg B \wedge (A \to B)) \to \neg A$
7. $\neg B \to ((A \to B) \to \neg A)$
8. $(A \to B) \to ((A \to \neg B) \to \neg A)$
9. $((A \wedge B) \to C) \to (A \to (B \to C))$
10. $(A \to (B \to C)) \to (B \to (A \to C))$
11. $(A \to (B \vee C)) \to ((A \to B) \vee (A \to C))$
12. $(A \to (B \wedge C)) \to ((A \to B) \wedge (A \to C))$

# Capitolo 4

# Logica dei predicati

## 4.1 Introduzione

Nei capitoli precedenti abbiamo studiato le proprietà sintattiche e semantiche dei connettivi proposizionali, cioè di "operazioni" che consentono di combinare tra loro proposizioni più semplici per formare proposizioni composte. In tali operazioni, le proposizioni sono assunte ed analizzate come entità elementari, concetti astratti che denotano un valore di verità "vero" o "falso", e l'enfasi è posta sul modo in cui il valore di verità della proposizione composta dipende dai valori di verità dei componenti.
Dal punto di vista formale, il calcolo proposizionale è tuttavia poco espressivo. In particolare, i suoi meccanismi sono insufficienti per gestire ragionamenti che coinvolgono *generalità* e nozioni correlate, cioè la possibilità di esprimere che una certa proprietà $P$ sussiste *per tutti* gli "oggetti" del dominio inteso del discorso o, che *esiste* almeno un oggetto che gode della proprietà $P$, ed altre affermazioni simili. Ciò è dovuto al fatto che la logica proposizionale si limita a considerare le proposizioni composte come funzioni di quelle atomiche, senza scomporre ulteriormente queste ultime, benché esse non costituiscano gli elementi più semplici del ragionamento e posseggano a loro volta una struttura interna che assume un ruolo di primaria importanza nelle deduzioni. Consideriamo, ad esempio, la seguente derivazione:

$$\frac{\begin{array}{c}\text{ogni numero intero è un numero razionale}\\ \text{1 è un numero intero}\end{array}}{\text{1 è un numero razionale}}$$

questa è intuitivamente corretta; tuttavia, se riscritta utilizzando la notazione simbolica della logica proposizionale, assume una forma del tipo:

$$A \wedge B \rightarrow C$$

e non vi è alcun motivo per cui tale formula risulti vera. Questo avviene appunto perché non vengono considerate le strutture interne delle proposizioni atomiche

$A, B$ e $C$.
Per poter ovviare a questi problemi, è necessario introdurre degli operatori di una natura sensibilmente differente dai connettivi proposizionali, tradizionalmente noti con il nome di *quantificatori*. Tra questi, i più utili dal punto di vista matematico sono il quantificatore universale ($\forall$) e quello esistenziale ($\exists$). La semantica intesa per tali connettivi richiede un sistema basato su due classi distinte di oggetti: i "termini" ed i "predicati". I termini rappresentano gli "oggetti" del dominio del discorso; i predicati sono le relazioni che possono sussistere tra questi oggetti. Formule quantificate del tipo $\forall xA$ o $\exists xA$ esprimono rispettivamente il fatto che la proprietà $A$ sussiste per ogni generico elemento $x$ del dominio, o che esiste almeno un elemento $x$ che gode della proprietà $A$.
La logica così ottenuta è nota come logica dei predicati (o logica del primo ordine), il cui nome deriva dal fatto che una proposizione (atomica) presenta, in genere, una struttura interna del tipo soggetto−predicato. Essa può essere considerata come un'estensione di quella proposizionale e risulta sufficientemente espressiva da permettere la formalizzazione di gran parte dei ragionamenti che ricorrono nel linguaggio naturale e nella matematica e da costituire la base di diversi utili linguaggi di programmazione, tra i quali Prolog (si veda a tal proposito il capitolo 6) ed SQL.
Più precisamente, la logica dei predicati estende quella proposizionale con le nozioni aggiuntive di: quantificatori, predicati, funzioni, variabili e costanti.

- I *quantificatori* introdotti sono: il quantificatore *universale* e quello *esistenziale*; il primo permette di considerare la *generalità* degli oggetti del dominio inteso del discorso, come ad esempio nella sentenza:

  tutti i numeri interi sono numeri razionali

  mentre il secondo consente di esprimere l'*esistenza* di oggetti in una data relazione, come:

  *esiste* un numero razionale che non è intero.

- I *predicati* consentono di esprimere proprietà e relazioni su insiemi di oggetti del dominio. Se ad esempio consideriamo i numeri naturali, possiamo essere interessati ad esprimere la proprietà che determinati numeri sono primi. Avremo dunque sentenze della forma:

  5 è un numero primo
  4 è un numero primo
  $x$ è un numero primo

  Si noti che la prima di queste sentenze risulta vera, la seconda falsa, mentre per quanto concerne la terza, non è possibile stabilire se è vera o falsa: questa è una funzione logica che diventa vera o falsa quando la *variabile* $x$ viene rimpiazzata da un oggetto del dominio del discorso (che in questo caso è l'insieme dei numeri naturali).

- Le *funzioni* permettono di definire nuovi "oggetti" del dominio in termini di quelli preesistenti e di individuarli univocamente. Ad esempio, sui numeri naturali possiamo considerare la funzione $s$ che associa ad ogni intero

$x$ il suo successore $x + 1$; se consideriamo invece un dominio di individui, possiamo essere interessati alla funzione che associa ad ogni persona suo padre, e così via.

In analogia con quanto è stato fatto per la logica proposizionale, procediamo alla trattazione formale della logica dei predicati presentandone prima la sintassi e poi la semantica.

## 4.2 Sintassi

Definiamo anzitutto l'alfabeto per la logica dei predicati.
In generale, per ragioni di chiarezza e leggibilità, indicheremo le variabili con lettere minuscole terminali dell'alfabeto $(x, y, z, \dots)$; le costanti con lettere minuscole iniziali $(a, b, c, \dots)$, e le funzioni con lettere intermedie $(f, g, h, \dots)$. In questo capitolo, i simboli di predicato saranno rappresentati da lettere maiuscole iniziali dell'alfabeto $(A, B, C, \dots)$. Nel caso di predicati e funzioni, si utilizzeranno spesso degli apici per indicarne l' *arietà*, cioè il numero degli argomenti. Si scriverà, ad esempio, $A^3$ e $f^2$ per indicare rispettivamente che $A$ è un predicato ternario ed $f$ è una funzione binaria. Le lettere maiuscole $P, Q, \cdots$ (eventualmente indiciate) verranno utilizzate come meta-simboli per le formule della logica del primo ordine.
I *quantificatori* sono rappresentati dai seguenti simboli:

| | | |
|---|---|---|
| $\forall$ | " per ogni" | quantificatore universale |
| $\exists$ | "esiste" | quantificatore esistenziale |

**Esempio 4.1** La sentenza "$x$ è un numero pari" si può formalizzare come $A^1(x)$, dove $A^1$ è il predicato "... numero pari" attribuito ad $x$ (soggetto); quindi $A^1(4)$ rappresenta "4 è un numero pari".
La proposizione "$x+1$ è maggiore di $x$" utilizzando la notazione simbolica della logica dei predicati, diventa $A^2(f^2(x, 1), x)$,dove $A^2$ è il predicato "... maggiore di ..." ed $f^2$ la funzione che presi due argomenti esegue la loro somma ($f^2(x, y) = x + y$).

**Definizione 4.1** *Un* linguaggio del primo ordine *è definito da un alfabeto composto da:*

- *un insieme di simboli di costante* $a, b, c \dots$
- *un insieme infinito, VAR, di simboli di variabile* $x, y, z, \dots$
- *un insieme di simboli di funzione* $f, g, h, \dots$
- *un insieme di simboli di predicato* $A, B, C, \dots$
- *connettivi :* $\wedge, \vee, \neg, \rightarrow, \bot$
- *quantificatori:* $\forall, \exists$

- *simboli ausiliari: "(" , ")".*

In generale, non si faranno particolari assunzioni sulla cardinalità degli insiemi di simboli per costanti, funzioni e predicati[1]. Abitualmente, tuttavia, si richiede che funzioni e predicati siano in numero finito. Al contrario, si accettano insiemi di costanti di cardinalità arbitraria, perchè può essere interessante considerare teorie logiche per descrivere modelli di qualunque cardinalità, ed in tal caso può essere utile avere una costante differente per ogni elemento del modello (si veda, a tal proposito, il paragrafo 5.4). Per ragioni tecniche, è inoltre opportuno disporre di una infinità (numerabile) di variabili, in modo da avere continua disponibilità di variabili "fresche", cioè variabili che non appaiono nelle formule già considerate.
Si noti comunque che l'alfabeto deve essere fissato una volta per tutte. Non esiste dunque un unico linguaggio del primo ordine, ma ogni alfabeto definisce un linguaggio differente. I linguaggi differiscono tra loro per i diversi simboli di costante, funzione e predicato definiti nell'alfabeto; ovviamente la differenza rilevante non è nei nomi scelti per tali simboli, ma nel loro numero ed arietà (ciò che è comunemente nota come "*segnatura*" del linguaggio). Fissato un linguaggio, si può ora definire l'insieme dei suoi *termini*:

**Definizione 4.2** *TER è il minimo insieme $X$ tale che:*

1. *ogni costante appartiene ad $X$*
2. *ogni variabile appartiene ad $X$*
3. *se $t_1, t_2, \ldots t_n \in X$ ed $f^n$ è un simbolo di funzione del linguaggio $\Rightarrow$ $f^n(t_1, \ldots, t_n) \in X$ con $n \geq 1$.*

***Osservazione*** È possibile considerare le costanti come funzioni di arità 0.

**Definizione 4.3** *Sia $\mathcal{L}$ un linguaggio del primo ordine. La* cardinalità *di $\mathcal{L}$, che si indica con $|\mathcal{L}|$, è la cardinalità dell'insieme degli elementi che lo costituiscono.*

Dopo aver presentato i termini, procediamo alla definizione formale delle formule ben formate di un dato linguaggio del primo ordine.

**Definizione 4.4** *FBF è il minimo insieme $X$ tale che:*

1. *$\bot \in X$*
2. *$t_1, \ldots, t_n \in$ TER ed $A^n$ è un simbolo di predicato del linguaggio $\Rightarrow$ $A^n(t_1, \ldots, t_n) \in X$ con $n \geq 0$*
3. *$P \in X \Rightarrow (\neg P) \in X$*
4. *$P, Q \in X \Rightarrow (P \wedge Q), (P \vee Q), (P \rightarrow Q) \in X$*

[1] Ovviamente, se si considerano solo insiemi finiti di formule, si può sempre supporre che l'alfabeto sia finito.

5. $P \in X \Rightarrow ((\forall x)P), ((\exists x)P) \in X$

Le fbf dei punti 1 e 2 sono dette formule atomiche o semplicemente *atomi*.

**Esempio 4.2** Trasformare in fbf le seguenti sentenze:

1. Ogni numero naturale è un numero intero
2. Esiste un numero che non è naturale
3. Per ogni numero $x$ esiste un numero $y$ tale che $x < y$

Denotando con $A^1$ il predicato "...numero naturale", con $B^1$ "...numero intero" e con $C^2$ " $\ldots < \ldots$ ", le precedenti proposizioni risultano:

1. $((\forall x)(A^1(x) \rightarrow B^1(x)))$
2. $((\exists x)(\neg A^1(x)))$
3. $((\forall x)((\exists y)C^2(x, y))$

Per aumentare la leggibilità delle fbf si utilizzano le regole di precedenza dei connettivi introdotte nel paragrafo 1.3, con in aggiunta quelle per i quantificatori, per eliminare, quando è possibile, le parentesi. Le precedenze sono espresse nel seguente modo[2]:

$$\forall = \exists = \neg > \wedge > \vee > \rightarrow$$

In base a tali convenzioni, le fbf dell'esempio sopra si possono riscrivere, rispettivamente, come:

1. $\forall x(A^1(x) \rightarrow B^1(x))$
2. $\exists x \neg A^1(x)$
3. $\forall x \exists y C^2(x, y)$

Inoltre, nel seguito, ometteremo abitualmente l'apice $n$ nelle espressioni $f^n(t_1, \ldots, t_n)$ o $A^n(t_1, \ldots, t_n)$, poiché è implicitamente definito dal numero degli argomenti.

### 4.2.1 Sottoformule

In analogia con quanto fatto per la logica proposizionale, vediamo la definizione formale di *sottoformula*:

**Definizione 4.5** *Sia $P$ una fbf*

*1. Se $P$ è $\perp$ oppure $A(t_1, \ldots, t_n)$, allora $P$ stessa è la sua sola sottoformula.*

*2. Se $P$ è $\neg P_1$, allora le sottoformule di $P$ sono $P$ stessa e quelle di $P_1$.*

[2]Per semplicità, indicheremo con il simbolo "=" l'espressione "$\cdots$ ha la stessa precedenza di $\cdots$", e con il simbolo ">" l'espressione "$\cdots$ ha precedenza maggiore di $\cdots$".

*3. Se $P$ è $(P_1 \wedge P_2), (P_1 \vee P_2), (P_1 \rightarrow P_2)$ allora le sue sottoformule sono $P$ stessa e quelle di $P_1$ e $P_2$.*

*4. Se $P$ è $(\forall x P_1), (\exists x P_1)$, allora le sue sottoformule sono $P$ stessa e quelle di $P_1$.*

**Esempio 4.3** Sia $P$ la fbf

$$\exists x A(f(x), y) \wedge \neg \forall y B(f(y), y)$$

Le sue sottoformule sono: $P$, $\exists x A(f(x), y)$, $A(f(x), y)$, $\neg \forall y B(f(y), y)$, $\forall y B(f(y), y)$, $B(f(y), y)$.

### 4.2.2 Induzione

Definiamo ora i Principi di Induzione Strutturale per i termini e le formule ben formate.

**Teorema 4.6** *Sia $\mathcal{A}$ una proprietà, allora $\mathcal{A}(t)$*[3] per ogni $t \in$ *TER se:*

*1. $\mathcal{A}$ è verificata per tutti i simboli di variabile e costante.*

*2. $\forall t_1, \ldots, t_n \in$ TER, $\mathcal{A}(t_1), \mathcal{A}(t_2), \ldots, \mathcal{A}(t_n) \Rightarrow \mathcal{A}(f^n(t_1, \ldots, t_n))$ per tutti i simboli di funzione $f^n$* con $n \geq 1$.

*Dimostrazione.* Sia $Y = \{l \in \text{TER} \mid \mathcal{A}(l)\}$, allora $Y$ soddisfa 1-3 della Definizione 4.2. Quindi TER $\subseteq Y$, ossia $\forall t \in$ TER, $\mathcal{A}(t)$. □

**Teorema 4.7** *Sia $\mathcal{A}$ una proprietà, allora $\mathcal{A}(P)$* per ogni $P \in$ *FBF se:*

*1. $\mathcal{A}$ è verificata per tutti gli atomi.*

*2.* $\forall P_1 \in \text{FBF}, \ \mathcal{A}(P_1) \Rightarrow \mathcal{A}((\neg P_1))$.

*3.* $\forall P_1, P_2 \in \text{FBF}, \ \mathcal{A}(P_1), \mathcal{A}(P_2) \Rightarrow \mathcal{A}(P_1 \wedge P_2), \mathcal{A}(P_1 \vee P_2), \mathcal{A}(P_1 \rightarrow P_2)$

*4.* $\forall P_1 \in \text{FBF} \ \mathcal{A}(P_1) \Rightarrow \mathcal{A}((\forall x_i) P_1), \mathcal{A}((\exists x_i) P_1) \quad \forall i$

*Dimostrazione.* Analoga a quella del teorema precedente. □

### 4.2.3 Variabili libere e legate

Consideriamo una formula quantificata del tipo $\forall x Q$. Intuitivamente, essa esprime che la formula $Q$, in cui può comparire la variabile $x$, è vera per ogni elemento $x$ del dominio inteso del discorso. Dato dunque un termine $t$ del linguaggio, ci si aspetta che se è vero $\forall x Q$, allora rimpiazzando le occorrenze di $x$ in $Q$ con il termine $t$ (ovvero operando la *sostituzione* $Q[^t/_x]$) si ottiene ancora una sentenza corretta. I quantificatori stabiliscono dei *legami* tra la variabile quantificata e le eventuali occorrenze della stessa variabile all'interno del corpo della formula. Lo

[3] $\mathcal{A}$ è verificata

studio formale di questi legami costituisce la principale difficoltà sintattica della logica dei predicati, e richiede una definizione piuttosto meticolosa di nozioni quali *campo d'azione*, *occorrenze libere* e *legate* di variabili, *sostituzione* etc. che saranno oggetto di questa sezione.
La prima nozione che dobbiamo discutere è quella di *campo d'azione* (scope) di un quantificatore, che definisce la *portata* della quantificazione, ovvero la sottoespressione sintattica su cui essa ha effetto. Per definizione, il campo d'azione di un quantificatore è la fbf immediatamente alla sua destra. Ad esempio, il campo di azione del quantificatore esistenziale nella fbf $(\exists x)(P \to Q) \to R$ è la sottoformula $P \to Q$.
Una variabile che compare nel campo di azione di un quantificatore si dirà *legata*; altrimenti *libera*. Ad esempio, nella formula $\forall x A(x, y)$, $x$ è legata mentre $y$ è libera.
Formalmente, l'insieme delle variabili libere di una formula ben formata è definito nel modo seguente:

**Definizione 4.8** *Sia $t \in TER$; l'insieme $FV(t)$ delle* variabili libere *in $t$ è definito per induzione come segue:*

- $FV(x_i) = \{x_i\}$ *per una variabile* $x_i$.
- $FV(c) = \emptyset$ *per una costante* $c$.
- $FV(f(t_1, \ldots, t_n)) = FV(t_1) \cup \cdots \cup FV(t_n)$ *per una funzione n-aria* $f$.

*Per ogni fbf $P$ l'insieme $FV(P)$ delle* variabili libere *di $P$ è definito da:*

- $FV(\bot) = \emptyset$.
- $FV(A(t_1, \ldots, t_n)) = FV(t_1) \cup \cdots \cup FV(t_n)$ *per un predicato n-ario.*
- $FV(\neg P_1) = FV(P_1)$.
- $FV(P_1 \wedge P_2) = FV(P_1) \cup FV(P_2)$.
- $FV(P_1 \vee P_2) = FV(P_1) \cup FV(P_2)$.
- $FV(P_1 \to P_2) = FV(P_1) \cup FV(P_2)$.
- $FV(\forall x P_1) = FV(P_1) - \{x\}$.
- $FV(\exists x P_1) = FV(P_1) - \{x\}$.

**Definizione 4.9** *Una fbf $P$ è detta* chiusa *se $FV(P) = \emptyset$; aperta altrimenti.*

Lasciamo al lettore la cura di definire l'insieme BV delle variabile legate di una formula.

Si noti che per una fbf $P$, l'intersezione di $\mathrm{FV}(P)$ e $\mathrm{BV}(P)$ non è necessariamente vuoto; vale a dire che una stessa variabile può occorrere in $P$ sia libera che legata. Sia ad esempio $P$ la fbf $\forall x(Q(x, y) \to R(x)) \wedge \forall y(\neg Q(x, y) \to \forall z R(z))$; allora $\mathrm{FV}(P) = \{x, y\}$ e $\mathrm{BV}(P) = \{x, y, z\}$. Al contrario, ogni *occorrenza* di un certa variabile è o libera o legata.

***Osservazione*** Può accadere che una variabile compaia nel campo d'azione di più di un quantificatore per essa, come nella formula $\forall x(A(x) \rightarrow \forall x B(x))$ dove la seconda occorrenza della variabile $x$ è nel campo d'azione di entrambi i quantificatori. In questo caso, si assume che il legame corretto sia con il quantificatore più interno.

Il campo d'azione dei quantificatori determina in modo essenziale l'interpretazione stessa della formula, come risulterà evidente dalla semantica formale definita nel prossimo paragrafo. Tuttavia, facendo riferimento all'interpretazione intuitiva dei quantificatori, è opportuno che il lettore svolga fin d'ora numerosi esercizi di formalizzazione in logica del primo ordine di semplici enunciati tratti sia dal linguaggio matematico che da quello naturale. Il problema principale consiste appunto nel *posizionamento* corretto dei quantificatori all'interno della formula.

**Esempio 4.4** Asserire

> "Per ogni intero $x$ esiste un numero $y$ tale che se $x$ non è il successore di $y$ allora $x = 0$"

è una cosa ben diversa dal dire

> "Per ogni intero $x$ se esiste un numero $y$ tale che $x$ non è il successore di $y$ allora $x = 0$"

In particolare, pur non avendo introdotto la definizione formale di interpretazione nella logica del primo ordine, la prima asserzione è intuitivamente vera: o $x = 0$ e dunque l'implicazione è vera perchè il conseguente è vero, oppure $x \neq 0$ ed in tal caso basta prendere come $y$ il predecessore di $x$ e l'implicazione è vera in quanto la premessa $x \neq s(y)$ risulta falsa. Al contrario, la seconda asserzione è intuitivamente falsa. Consideriamo ad esempio $x = 1$: ovviamente esiste un $y$ per cui 1 non è successore di $y$ (dunque la premessa è vera), ma $1 \neq 0$ e dunque il conseguente è falso.
Veniamo al problema della loro formalizzazione logica. Supponiamo di utilizzare un linguaggio del primo ordine con una costante 0, una funzione unaria $s$ che denota la funzione successore ed un predicato binario $E(x, y)$ che denota la relazione di uguaglianza.
La prima asserzione sarà formalizzata dalla formula

$$\forall x \exists y(\neg E(x, s(y)) \rightarrow E(x, 0))$$

mentre la seconda assume la forma

$$\forall x(\exists y \neg E(x, s(y)) \rightarrow E(x, 0))$$

che sono ben diverse tra loro!

Un altro errore comune è quello di invertire inopinatamente l'ordine dei quantificatori in una formula, che al contario gioca un ruolo essenziale.

**Esempio 4.5** Dire che

"Per ogni $x$, esiste un $y$ tale che $x$ è minore di $y$"

è una cosa completamente diversa dall'affermare

"Esiste un $y$ tale che, per ogni $x$, $x$ è minore di $y$"

Quest'ultima sentenza asserisce l'esistenza di un massimo rispetto alla relazione d'ordine, ed è intuitivamente falsa se si considerano come oggetti del dominio del discorso i numeri naturali (mentre, al contrario, la prima asserzione è banalmente vera).
In un linguaggio del primo ordine, le due asserzioni suddette assumono, rispettivamente, la seguente forma:

$$\forall x \exists y A(x, y)$$

$$\exists y \forall x A(x, y)$$

### 4.2.4 Sostituzione

I nomi delle variabili quantificate assumono il ruolo di *parametri formali*; questo significa che in una formula del tipo $\forall x P$ o $\exists x P$ si può cambiare il nome $x$ in un nome $y$ senza modificare il significato intuitivo della formula, a condizione di cambiare in modo consistente i nomi di tutte le occorrenze della variabile $x$ in $P$ legate dal quantificatore. Ovviamente, il nuovo nome $y$ non doveva comparire libero in $P$ poichè altrimenti si andrebbero a legare delle occorrenze di variabili che non erano legate nella formula originaria.
Illustriamo la tal cosa con un esempio tratto dalla matematica.

**Esempio 4.6** Si consideri l'integrale

$$f(y) = \int_a^b x \cdot y \, dx$$

intuitivamente la variabile $x$, nella classificazione precedentemente discussa, è "legata", mentre $y$ è "libera". È possibile sstituire $z$ al posto di $x$ (quindi l'integrale diventerebbe $f(y) = \int_a^b z \cdot y \, dz$); non è però ammesso sostituire $y$ al posto di $x$, perché in tal modo si otterrebbe una funzione differente.

Vediamo ora la definizione formale di sostituzione di una variabile $x$ con un termine $t$ in una formula $P$, che denoteremo con $P[^t/_x]$.
Definiamo innanzi tutto la *sostituzione* all'interno di termini.

**Definizione 4.10** *Siano $s, t \in TER$, allora $s[^t/_x]$, ossia il termine ottenuto rimpiazzando tutte le occorrenze di $x$ in $s$ con $t$, è definito nel seguente modo:*

- *se $s$ è una variabile $y$, allora*

$$y[^t/_x] = \begin{cases} y & \text{se } y \neq x \\ t & \text{se } y = x \end{cases}$$

- *se $s$ è una costante $c$, allora* $c[^t/_x] = c$
- $f(t_1, \ldots, t_n)[^t/_x] = f(t_1[^t/_x], \ldots, t_n[^t/_x])$

Nel definire $P[^t/_x]$ esiste un problema piuttosto delicato che deve essere discusso. In particolare, si vuole evitare che una qualche variabile libera in $t$ venga indebitamente legata da qualche quantificatore di $P$ durante la sostituzione. Consideriamo ad esempio la formula $P$ data da $\exists x(x < y)$, e la sostituzione $P[^x/_y]$. Intuitivamente si sta semplicemente ridenominando una variabile e ci si aspetta che il significato della formula non cambi. Tuttavia, applicando in modo semplicistico la nozione di sostituzione, otterremmo la formula $\exists x(x < x)$ che ha un significato intuitivo differente dalla precedente. Ad esempio, pur non avendo ancora introdotto la definizione formale di interpretazione per la logica dei predicati, $\exists x(x < x)$ risulta intuitivamente falsa se si considera una qualunque struttura ordinata, mentre la formula di partenza, vale a dire $\exists x(x < y)$, potrebbe non esserlo (dipende dall'interpretazione che si dà alla variabile libera $y$). Ciò accade in quanto la variabile (libera) $y$ diventa legata dopo la sua sostituzione con la variabile $x$. Un modo differente di vedere il problema è il seguente. Abbiamo già osservato che si possono ridenominare liberamente le variabili legate senza alterare la semantica attesa della formula. Dunque $\exists x(x < y)$ è logicamente equivalente ad $\exists z(z < y)$. Se ora si sostituisce $x$ per $y$ in tale formula, si ottiene $\exists z(z < x)$ che, intuitivamente, ha lo stesso "significato" di $\exists x(x < y)$. Tale considerazione suggerisce la soluzione al problema. Allorchè si ha una sostituzione del tipo $(\forall x P)[t/y]$ o $(\exists x P)[t/y]$ e la variabile $x$ compare libera in $t$, è necessario ridenominare in modo opportuno la variabile $x$ in modo da evitare "collisioni" indesiderate.
Consideriamo ora la definizione formale.

**Definizione 4.11** *Siano $P \in FBF$ e $t \in TER$, allora $P[^t/_x]$ è definito, per induzione strutturale su $P$, nel seguente modo:*

- $\bot[^t/_x] = \bot$
- $A(t_1, \ldots, t_n)[^t/_x] = A(t_1[^t/_x], \ldots, t_n[^t/_x])$
- $(\neg P_1)[^t/_x] = \neg P_1[^t/_x]$
- $(P_1 \wedge P_2)[^t/_x] = P_1[^t/_x] \wedge P_2[^t/_x]$
- $(P_1 \vee P_2)[^t/_x] = P_1[^t/_x] \vee P_2[^t/_x]$
- $(P_1 \to P_2)[^t/_x] = P_1[^t/_x] \to P_2[^t/_x]$
- $$(\forall y P_1)[^t/_x] = \begin{cases} \forall y(P_1[^t/_x]) & \text{se } x \neq y \text{ e } y \notin FV(t) \\ \forall z(P_1[^z/_y][^t/_x]) & \text{se } x \neq y,\ y \in FV(t) \text{ e } z \text{ non occorre in } P_1, t \\ \forall y P_1 & \text{se } x = y \end{cases}$$

- $$(\exists y P_1)[^t/_x] = \begin{cases} \exists y(P_1[^t/_x]) & \text{se } x \neq y \text{ e } y \notin FV(t) \\ \exists z(P_1[^z/_y][^t/_x]) & \text{se } x \neq y,\ y \in FV(t)\ z \text{ non occorre in } P_1, t \\ \exists y P_1 & \text{se } x = y \end{cases}$$

***Osservazione*** Dalla definizione precedente segue che la sostituzione ha effetto solo se si applica a variabili libere.

**Esempio 4.7** Sia $P$ la fbf $\forall x(A(x) \rightarrow B(f_1(y), x))$ e sia $t$ il termine $f_2(x)$, allora

$$P[^t/_y] = \forall z(A(z) \rightarrow B(f_1(f_2(x)), z))$$

mentre $P[^t/_x] = P$ essendo $x$ una variabile legata.

**Esercizio 4.12** *Scrivere la definizione formale di sostituzione simultanea.*

Il risultato della sostituzione simultanea di $t_1, \ldots, t_n$ al posto di $x_1, \ldots, x_n$ in un termine $t$ si indica con $t[^{t_1,\ldots,t_n}/_{x_1,\ldots,x_n}]$ (con $P[^{t_1,\ldots,t_n}/_{x_1,\ldots,x_n}]$ quella in una fbf $P$). Si osserva che la sostituzione simultanea può essere diversa dalle corrispondenti sostituzioni ripetute ( cioè $((\ldots(t[^{t_1}/_{x_1}])[^{t_2}/_{x_2}]\ldots)[^{t_n}/_{x_n}]$); infatti, se si considera

$$A(x_0, x_1)[^{x_1,x_0}/_{x_0,x_1}] = A(x_1, x_0)$$

mentre

$$(A(x_0, x_1)[^{x_1}/_{x_0}])[^{x_0}/_{x_1}] = A(x_1, x_1)[^{x_0}/_{x_1}] = A(x_0, x_0)$$

## 4.3 Semantica

Il linguaggio introdotto nel paragrafo precedente, come tutti i linguaggi formali, è solo un insieme di simboli, associati in base a particolari regole, del tutto privi di significato. Ci proponiamo ora di interpretare tali simboli.
Nella logica proposizionale un'interpretazione è un assegnamento di valori di verità alle proposizioni atomiche che viene opportunamente esteso a quelle composte; in quella del primo ordine, data la maggior complessità del linguaggio, ciò non risulta sufficiente. Infatti, per definire un'interpretazione nella logica dei predicati, è necessario specificare due entità: un *dominio*, sul quale assumeranno i valori le variabili, ed un *assegnamento* che associa funzioni ai simboli di funzione e predicati ai simboli di predicato. Più formalmente:

**Definizione 4.13** *Una* struttura *è una coppia* $\mathcal{A} = (D_\mathcal{A}, I_\mathcal{A})$ *dove* $D_\mathcal{A}$ *è un qualunque insieme non vuoto detto* dominio (o universo) *ed* $I_\mathcal{A}$ *un'* assegnazione *che associa:*

1. *ad ogni simbolo di costante* $c$ *un elemento* $I_\mathcal{A}(c) \in D_\mathcal{A}$
2. *ad ogni simbolo di funzione* $f^k$ *con* $k \geq 1$ *una funzione*

$$I_\mathcal{A}(f^k) : D_\mathcal{A}^k \rightarrow D_\mathcal{A}$$

*3. ad ogni simbolo di predicato* $B^k$ *con* $k \geq 1$ *una relazione*

$$I_{\mathcal{A}}(B^k) : D_{\mathcal{A}}^k \to \{0,1\}$$

Osserviamo che la funzione $I_{\mathcal{A}}(B^k) : D_{\mathcal{A}}^k \to \{0,1\}$ si può vedere, in modo equivalente, come un opportuno sottoinsieme di $D_{\mathcal{A}}^k$; in particolare come quel sottoinsieme di cui $I_{\mathcal{A}}(B^k)$ è la funzione caratteristica.

**Notazione**
Nel prosieguo, scriveremo semplicemente $c^{\mathcal{A}}$, $f^{\mathcal{A}}$ e $B^{\mathcal{A}}$ al posto (rispettivamente) di $I_{\mathcal{A}}(c)$, $I_{\mathcal{A}}(f)$ e $I_{\mathcal{A}}(B)$.

Non essendo possibile interpretare una formula ben formata contenente variabili libere, si assegneranno ad esse degli elementi del dominio; in tal modo il valore di verità di una fbf dipenderà dallo specifico assegnamento scelto.

**Definizione 4.14** *Un* ambiente *per* $\mathcal{A}$ *è una funzione dalle variabili a* $D_{\mathcal{A}}$. *Indicheremo con* $ENV_{\mathcal{A}} = \{\xi_i^{\mathcal{A}} \mid \xi_i^{\mathcal{A}} : \mathrm{VAR} \to D_{\mathcal{A}}\}$ *l'insieme di tutti gli ambienti per* $\mathcal{A}$.

**Notazione**
$\xi^{\mathcal{A}}[{}^{a}/_{x}]$ è l'ambiente $\xi^{\mathcal{A}}$ salvo che ad $x$ si associa $a$; vale a dire che $\forall y \neq x$, $\xi^{\mathcal{A}}[{}^{a}/_{x}](y) = \xi^{\mathcal{A}}(y)$ e $\xi^{\mathcal{A}}[{}^{a}/_{x}](x) = a$.

**Definizione 4.15** *Un'* interpretazione *per un linguaggio* $\mathcal{L}$ *del primo ordine è una coppia* $(\mathcal{A}, \xi^{\mathcal{A}})$ *dove* $\mathcal{A}$ *è una struttura e* $\xi^{\mathcal{A}}$ *un ambiente per* $\mathcal{A}$.

**Esempio 4.8** Sia $P$ la fbf $\forall x A(x, f(x)) \vee B(g(c, y))$; $A$ è un predicato binario, $B$ un predicato unario, $f$ una funzione unaria, $g$ binaria e $c$ una costante, $y \in \mathrm{FV}(P)$. Sia $(\mathcal{A}, \xi^{\mathcal{A}})$ un'interpretazione la cui restrizione a $P$ è la seguente:

- $D_{\mathcal{A}} = \mathcal{N} = \{0, 1, \dots\}$
- $I_{\mathcal{A}}(A) = A^{\mathcal{A}} = \{(m, n) \mid m, n \in D_{\mathcal{A}} \text{ e } m > n\}$
- $I_{\mathcal{A}}(B) = B^{\mathcal{A}} = \{n \in D_{\mathcal{A}} \mid n \text{ è } primo\}$
- $I_{\mathcal{A}}(f) = f^{\mathcal{A}} = n \mapsto n + 1$
- $I_{\mathcal{A}}(g) = g^{\mathcal{A}} = (m, n) \mapsto m + n$
- $I_{\mathcal{A}}(c) = c^{\mathcal{A}} = 2$
- $\xi^{\mathcal{A}}(y) = 1$

Nell'interpretazione precedente, la formula $P$ assume il "significato"

"Per ogni intero $x$, $x > x + 1$ oppure $2 + 1$ è un numero primo"

che è intuitivamente vera in quanto 3 è primo.
Tuttavia, non abbiamo per ora alcuno strumento formale per stabilire il valore di verità di $P$.
Osserviamo inoltre che, in generale, il valore di verità di una formula della logica dei predicati può dipendere sia dalla struttura che dall'ambiente. Ad esempio, se si considera la struttura $\mathcal{B}$ t.c.

$$I_{\mathcal{B}}(B) = B^{\mathcal{B}} = \{n \in D_{\mathcal{B}} \mid n = 0\}$$

$P$ risulta falsa (qualunque sia l'ambiente). La stessa cosa avviene se si interpreta $P$ nella struttura $\mathcal{A}$ con $\xi(y) = 2$.
Definiamo quindi formalmente il *valore di verità* di una fbf rispetto ad una data interpretazione.

**Definizione 4.16** *Sia $P$ una fbf ed $(\mathcal{A}, \xi^{\mathcal{A}})$ un'interpretazione. Per ogni termine $t$ che occorre in $P$, denotiamo il suo valore nell'interpretazione $(\mathcal{A}, \xi^{\mathcal{A}})$ con $[\![t]\!]^{\mathcal{A}}_{\xi}$; la funzione $[\![\ ]\!]^{\mathcal{A}}_{\xi} : \mathrm{TER} \to D_{\mathcal{A}}$ è definita in modo induttivo come segue:*

- *se $t$ è una variabile $x$, allora $[\![t]\!]^{\mathcal{A}}_{\xi} = \xi^{\mathcal{A}}(x)$.*
- *se $t$ è una funzione $f(t_1, \ldots t_k)$,* dove $t_1, \ldots, t_k \in \mathrm{TER}$ *ed $f$ è un simbolo di funzione di arità $k$ con $k \geq 0$[4], allora*
  $[\![t]\!]^{\mathcal{A}}_{\xi} = f^{\mathcal{A}}([\![t_1]\!]^{\mathcal{A}}_{\xi}, \ldots, [\![t_k]\!]^{\mathcal{A}}_{\xi})$.

*Definiamo ora il* valore di verità *di $P$ in una interpretazione $(\mathcal{A}, \xi^{\mathcal{A}})$, che verrà denotato con $v^{(\mathcal{A},\xi^{\mathcal{A}})}(P)$.*
*La funzione di valutazione $v^{(\mathcal{A},\xi^{\mathcal{A}})} : \mathrm{FBF} \to \{0, 1\}$ è definita induttivamente come segue:*

- $v^{(\mathcal{A},\xi^{\mathcal{A}})}(\bot) = 0$.
- $v^{(\mathcal{A},\xi^{\mathcal{A}})}(P_1(t_1, \ldots, t_k)) = P_1^{\mathcal{A}}([\![t_1]\!]^{\mathcal{A}}_{\xi}, \ldots, [\![t_k]\!]^{\mathcal{A}}_{\xi})$.
- $v^{(\mathcal{A},\xi^{\mathcal{A}})}(P_1 \wedge P_2) = min(v^{(\mathcal{A},\xi^{\mathcal{A}})}(P_1), v^{(\mathcal{A},\xi^{\mathcal{A}})}(P_2))$.
- $v^{(\mathcal{A},\xi^{\mathcal{A}})}(P_1 \vee P_2) = max(v^{(\mathcal{A},\xi^{\mathcal{A}})}(P_1), v^{(\mathcal{A},\xi^{\mathcal{A}})}(P_2))$.
- $v^{(\mathcal{A},\xi^{\mathcal{A}})}(P_1 \to P_2) = max(1 - v^{(\mathcal{A},\xi^{\mathcal{A}})}(P_1), v^{(\mathcal{A},\xi^{\mathcal{A}})}(P_2))$.
- $v^{(\mathcal{A},\xi^{\mathcal{A}})}(\forall x P_1) = min\{v^{(\mathcal{A},\xi^{\mathcal{A}[a/x])}}(P_1) \mid a \in D_{\mathcal{A}}\}$.
- $v^{(\mathcal{A},\xi^{\mathcal{A}})}(\exists x P_1) = max\{v^{(\mathcal{A},\xi^{\mathcal{A}[a/x])}}(P_1) \mid a \in D_{\mathcal{A}}\}$.

Si noti che il quantificatore universale può essere visto come una congiunzione iterata. Supponiamo, infatti, che $D_{\mathcal{A}} = \{a_1, \ldots a_n\}$, allora la formula ben formata $\forall x P(x)$ ha lo stesso "significato" di $P^{\mathcal{A}}(a_1) \wedge \cdots \wedge P^{\mathcal{A}}(a_n)$, di qui $v^{(\mathcal{A},\xi^{\mathcal{A}})}(\forall x P) = min\{v^{(\mathcal{A},\xi^{\mathcal{A}[a/x])}}(P) \mid a \in D_{\mathcal{A}}\}$.
Al contrario, il quantificatore esistenziale si può considerare come una disgiunzione iterata: se $D_{\mathcal{A}} = \{a_1, \ldots, a_n\}$, allora $\exists x P(x)$ corrisponde a $P^{\mathcal{A}}(a_1) \vee \cdots \vee P^{\mathcal{A}}(a_n)$.

[4] Si veda 4.2.

***Osservazione*** Tale definizione, al contrario di quanto avviene per la logica proposizionale, non consente di determinare in modo effettivo il valore di verità di una fbf in un'interpretazione $\mathcal{A}$, in quanto se $D_{\mathcal{A}}$ è infinito, occorrerebbero infiniti controlli.

Il valore di verità di una fbf $P$ in una data interpretazione $(\mathcal{A}, \xi^{\mathcal{A}})$ dipende solo dalla restrizione di $\xi^{\mathcal{A}}$ all'insieme delle variabili libere di $P$. Infatti:

**Lemma 4.17** *Sia $t$ un termine t.c $FV(t) = \{y_1, \ldots, y_n\}$ e sia $\mathcal{A}$ una struttura. Per tutti gli ambienti $\xi_1^{\mathcal{A}}, \xi_2^{\mathcal{A}}$ t.c. $\xi_1^{\mathcal{A}}(y_i) = \xi_2^{\mathcal{A}}(y_i)\ \forall i = 1, \ldots, n$, risulta $[\![t]\!]_{\xi_1}^{\mathcal{A}} = [\![t]\!]_{\xi_2}^{\mathcal{A}}$.*

*Dimostrazione.* Si effettua per induzione sulla struttura del termine $t$.
(*caso base*)

- $t$ è una costante $c$, allora $[\![t]\!]_{\xi_1}^{\mathcal{A}} = c^{\mathcal{A}} = [\![t]\!]_{\xi_2}^{\mathcal{A}}$.
- $t$ è una variabile $y$, allora $[\![t]\!]_{\xi_1}^{\mathcal{A}} = \xi_1(y) = \xi_2(y) = [\![t]\!]_{\xi_2}^{\mathcal{A}}$.

Veniamo al caso induttivo.
$t$ è una funzione $f(t_1, \ldots, t_k)$, allora $[\![t]\!]_{\xi_1}^{\mathcal{A}} = f^{\mathcal{A}}([\![t_1]\!]_{\xi_1}^{\mathcal{A}}, \ldots, [\![t_k]\!]_{\xi_1}^{\mathcal{A}})$ supponiamo, per ipotesi induttiva, che $[\![t_i]\!]_{\xi_1}^{\mathcal{A}} = [\![t_i]\!]_{\xi_2}^{\mathcal{A}}\ \forall i = 1, \ldots, k$ allora $[\![t]\!]_{\xi_1}^{\mathcal{A}} = [\![t]\!]_{\xi_2}^{\mathcal{A}}$. □

**Proposizione 4.18** *Siano $P$ una fbf tale che $FV(P) = \{y_1, \ldots, y_n\}$ ed $\mathcal{A}$ una struttura; per tutti gli ambienti $\xi_1^{\mathcal{A}}, \xi_2^{\mathcal{A}}$ tali che $\xi_1^{\mathcal{A}}(y_i) = \xi_2^{\mathcal{A}}(y_i),\ \forall i = 1, \ldots, n$ risulta $v^{(\mathcal{A}, \xi_1^{\mathcal{A}})}(P) = v^{(\mathcal{A}, \xi_2^{\mathcal{A}})}(P)$.*

*Dimostrazione.* Si effettua per induzione sulla struttura della fbf $P$. (*caso base*)

- Se $P$ è $\perp$ l'asserto è verificato.
- Se $P$ è $P_1(t_1, \ldots, t_k)$, essendo, per il lemma precedente, $[\![t_i]\!]_{\xi_1}^{\mathcal{A}} = [\![t_i]\!]_{\xi_2}^{\mathcal{A}}\ \forall i = 1, \ldots, k$ segue che $v^{(\mathcal{A}, \xi_1^{\mathcal{A}})}(P) = v^{(\mathcal{A}, \xi_2^{\mathcal{A}})}(P)$.

Veniamo al caso induttivo.

- Se $P$ è $P_1 \wedge P_2$, supponiamo, per ipotesi induttiva, che sia vero l'asserto per $P_1$ e $P_2$, allora $v^{(\mathcal{A}, \xi_1^{\mathcal{A}})}(P) = max(v^{(\mathcal{A}, \xi_1^{\mathcal{A}})}(P_1), v^{(\mathcal{A}, \xi_1^{\mathcal{A}})}(P_2)) = max(v^{(\mathcal{A}, \xi_2^{\mathcal{A}})}(P_1),\ v^{(\mathcal{A}, \xi_2^{\mathcal{A}})}(P_2)) = v^{(\mathcal{A}, \xi_2^{\mathcal{A}})}(P)$.
- Se $P$ è $P_1 \vee P_2$ oppure $P_1 \rightarrow P_2$ la dimostrazione è analoga a quella del caso precedente.
- Se $P$ è $\forall x P_1$. $v^{(\mathcal{A}, \xi_1^{\mathcal{A}})}(P) = 1$ se e solo se $v^{(\mathcal{A}, \xi_1^{\mathcal{A}}[^a/_x])}(P_1) = 1\ \forall a \in D_{\mathcal{A}}$. Essendo $\xi_1 = \xi_2$ su $\mathrm{FV}(P_1) = \mathrm{FV}(P) - \{x\}$, per ipotesi induttiva $v^{(\mathcal{A}, \xi_1^{\mathcal{A}}[^a/_x])}(P_1) = v^{(\mathcal{A}, \xi_2^{\mathcal{A}}[^a/_x])}(P_1)\ \forall a \in D_{\mathcal{A}}$ che implica $v^{(\mathcal{A}, \xi_1^{\mathcal{A}})}(P) = v^{(\mathcal{A}, \xi_2^{\mathcal{A}})}(P)$.
- Se $P$ è $\exists x P_1$ la dimostrazione è analoga al caso precedente e viene lasciata al lettore come esercizio.

□

### 4.3.1 Soddisfacibilità, validità e modelli

**Definizione 4.19** *Sia $P$ una formula ben formata:*

1. *$P$ è* soddisfatta *in una struttura $\mathcal{A}$, rispetto all'ambiente $\xi^{\mathcal{A}}$, se $v^{(\mathcal{A},\xi^{\mathcal{A}})}(P) =$ 1. In tale situazione scriveremo $(\mathcal{A}, \xi^{\mathcal{A}}) \models P$.*

2. *$P$ è* soddisfacibile *in $\mathcal{A}$ se esiste un ambiente $\xi^{\mathcal{A}}$ tale che $(\mathcal{A}, \xi^{\mathcal{A}}) \models P$.*

3. *$P$ è* vera in $\mathcal{A}$ *se per ogni ambiente $\xi^{\mathcal{A}}$ si ha $(\mathcal{A}, \xi^{\mathcal{A}}) \models P$. In tale situazione, diremo che $\mathcal{A}$ è un* modello *per $P$, e scriveremo $\mathcal{A} \models P$.*

4. *$P$ è* soddisfacibile *se esiste una struttura $\mathcal{A}$ tale che $P$ è* soddisfacibile *in $\mathcal{A}$.*

5. *$P$ è* valida *se è vera in ogni struttura. In questo caso scriveremo $\models P$.*

Osserviamo che la nozione di formula valida rappresenta la controparte nella logica del primo ordine, della nozione di tautologia (Definizione 1.9). È bene tuttavia tener presente che, mentre per verificare se una formula è una tautologia è possibile utilizzare direttamente la definizione di interpretazione in logica proposizionale, per stabilire se una formula della logica del primo ordine è valida, la semantica non è di grande aiuto! Assumono pertanto una importanza ben maggiore, che non nel caso proposizionale, la nozione di teorema (cap.5), una volta dimostrati i teoremi di correttezza e completezza, ed i metodi di refutazione (cap.6).

La definizione precedente si estende ad insiemi (eventualmente infiniti) di fbf, come segue:

**Definizione 4.20** *Sia $\Gamma$ un insieme di formule ben formate:*

1. *$\Gamma$ è* soddisfacibile *se esiste una struttura $\mathcal{A}$ ed un assegnamento $\xi$ tale che per ogni formula $P \in \Gamma$ si ha $(\mathcal{A}, \xi^{\mathcal{A}}) \models P$.*

2. *Una struttura $\mathcal{A}$ è un* modello *di $\Gamma$ se per ogni $P \in \Gamma$ si ha $\mathcal{A} \models P$. (cioè se $\mathcal{A}$ è un modello per ciascuna delle formule in $\Gamma$). Se $\mathcal{A}$ è un modello di $\Gamma$ scriveremo $\mathcal{A} \models \Gamma$.*

3. *$\Gamma$ è* valido *se ogni struttura è un modello per $\Gamma$. In tale situazione scriveremo $\models \Gamma$.*

Diamo ora la generalizzazione al primo ordine della nozione di *conseguenza semantica*.

**Definizione 4.21** *Dato un insieme di formule $\Gamma$ ed una formula $Q$, diremo che $Q$ è una* conseguenza semantica *di $\Gamma$, e scriveremo $\Gamma \models Q$, se per ogni struttura $\mathcal{A}$ ed ogni ambiente $\xi^{\mathcal{A}}$ tali che per ogni $P \in \Gamma$ si ha $(\mathcal{A}, \xi^{\mathcal{A}}) \models P$, risulta anche $(\mathcal{A}, \xi^{\mathcal{A}}) \models Q$.*

È utile considerare in modo esplicito la negazione delle nozioni di soddisfacibilità e validità.

**Definizione 4.22**

*1. Una fbf $P$ è* falsa in *una struttura $\mathcal{A}$ se e solo se non è soddisfacibile in $\mathcal{A}$, cioè se non esiste nessun ambiente $\xi^{\mathcal{A}}$ tale che $(\mathcal{A}, \xi^{\mathcal{A}}) \models P$. In questo caso scriveremo $\mathcal{A} \not\models P$.*

*2. Una fbf $P$ è* insoddisfacibile *(o* contraddittoria*) se e solo se non è soddisfacibile (ovvero è falsa in ogni struttura).*

Lasciamo al lettore la cura di estendere le definizioni precedenti a insiemi di formule ben formate.

**Esempio 4.9** (Paradosso del barbiere)
In un piccolo paese esiste un solo barbiere. Per definizione, il barbiere rade tutti e solo coloro che non si radono da soli. Ma allora, chi rade il barbiere? Se si rade da solo, allora non pùo radersi in quanto il barbiere rade *solo* coloro che non si radono da soli. Viceversa, se non si rade da solo, visto che il barbiere rade *tutti* coloro che non si radono da soli, dovrebbe radersi lui stesso. In effetti, la definizione precedente di barbiere è contraddittoria.
Vediamo come tale ragionamento si formalizza nella logica del primo ordine. Indichiamo con $R(x, y)$ il fatto che "$x$ rade $y$", e utilizziamo una costante $b$ per denotare il barbiere. Il fatto che il barbiere rade tutti coloro che non si radono da soli è formalizzato dalla formula

$$\forall x(\neg R(x, x) \rightarrow R(b, x))$$

Viceversa, il fatto che il barbiere rade solo coloro che non si radono da soli è formalizzato dalla formula

$$\forall x(R(b, x) \rightarrow \neg R(x, x))$$

Dunque, il barbiere è "definito" dalla congiunzione delle due formule precedenti, o più semplicemente dalla formula

$$\forall x(R(b, x) \leftrightarrow \neg R(x, x))$$

Lasciamo come esercizio per il lettore la dimostrazione formale che questa formula è contraddittoria.

Dalle definizioni precedenti segue immediatamente che:

**Teorema 4.23** *Siano $\Gamma$ un insieme di fbf e $P$ una fbf.*

*1. $P$ è valida se e solo se $\neg P$ è insoddisfacibile.*

*2. $P$ è soddisfacibile se e solo se $\neg P$ non è valida.*

*3. $\Gamma \models P$ se e solo se $\Gamma \cup \{\neg P\}$ è insoddisfacibile.*

***Osservazione*** Affermare che una formula $P$ non è valida non implica che essa sia contraddittoria (cioè la validità di $\neg P$), ma semplicemente che la sua negata $\neg P$ è soddisfacibile.

### 4.3.2 Proprietà della relazione di soddisfacibilità

**Definizione 4.24** *Data una fbf $P$. Sia $FV(P) = \{x_1, \ldots, x_k\}$. Si definisce*

- chiusura universale *di $P$, $Cl(P) = \forall x_1, \ldots, x_k P$.*
- chiusura esistenziale *di $P$, $Ex(P) = \exists x_1 \cdots x_n P$.*

**Lemma 4.25** *$\mathcal{A} \models P$ se e solo se $\mathcal{A} \models Cl(P)$.*

*Dimostrazione.* Immediata. □

**Teorema 4.26** *Sia $P$ una formula ben formata, $P$ è valida se e solo se $Cl(P)$ lo è.*

*Dimostrazione.* Segue dal lemma precedente. □

**Teorema 4.27** *Sia $P$ una formula ben formata, $P$ è soddisfacibile se e solo se $Ex(P)$ lo è.*

*Dimostrazione.* È lasciata al lettore come esercizio. □

Il lemma che segue permette di stabilire che le proprietà della relazione di soddisfacibilità (Definizione 4.19) corrispondono al significato intuitivo dei connettivi; questa caratteristica verrà ampiamente utilizzata nelle dimostrazioni successive.

**Lemma 4.28** *Per ogni $P$ e $Q \in FBF$, ed ogni interpretazione $(\mathcal{A}, \xi^{\mathcal{A}})$:*

1. $(\mathcal{A}, \xi^{\mathcal{A}}) \models P \wedge Q$ se e solo se $(\mathcal{A}, \xi^{\mathcal{A}}) \models P$ e $(\mathcal{A}, \xi^{\mathcal{A}}) \models Q$.
2. $(\mathcal{A}, \xi^{\mathcal{A}}) \models P \vee Q$ se e solo se $(\mathcal{A}, \xi^{\mathcal{A}}) \models P$ o $(\mathcal{A}, \xi^{\mathcal{A}}) \models Q$.
3. $(\mathcal{A}, \xi^{\mathcal{A}}) \models \neg P$ se e solo se $(\mathcal{A}, \xi^{\mathcal{A}}) \not\models P$.
4. $(\mathcal{A}, \xi^{\mathcal{A}}) \models P \rightarrow Q$ se e solo se $((\mathcal{A}, \xi^{\mathcal{A}}) \models P \Rightarrow (\mathcal{A}, \xi^{\mathcal{A}}) \models Q)$.
5. $(\mathcal{A}, \xi^{\mathcal{A}}) \models \forall x P$ se e solo se $(\mathcal{A}, \xi^{\mathcal{A}}[^a/_x]) \models P$ per ogni $a \in D_{\mathcal{A}}$.
6. $(\mathcal{A}, \xi^{\mathcal{A}}) \models \exists x P$ se e solo se $(\mathcal{A}, \xi^{\mathcal{A}}[^a/_x]) \models P$ per qualche $a \in D_{\mathcal{A}}$.

*Dimostrazione.* Segue dalla Definizione 4.16; mostriamo a titolo esemplificativo la prova dei seguenti casi:

*4.* ($\Rightarrow$) Supponiamo che $(\mathcal{A}, \xi^{\mathcal{A}}) \models P$, i.e. $v^{(\mathcal{A},\xi^{\mathcal{A}})}(P) = 1$, allora essendo $v^{(\mathcal{A},\xi^{\mathcal{A}})}(P \rightarrow Q) = max(1 - v^{(\mathcal{A},\xi^{\mathcal{A}})}(P), v^{(\mathcal{A},\xi^{\mathcal{A}})}(Q)) = 1$, per la Def. 4.16, risulta $v^{(\mathcal{A},\xi^{\mathcal{A}})}(Q) = 1$, dunque $(\mathcal{A}, \xi^{\mathcal{A}}) \models Q$.
($\Leftarrow$) Viceversa, sia $(\mathcal{A}, \xi^{\mathcal{A}}) \models P \Rightarrow (\mathcal{A}, \xi^{\mathcal{A}}) \models Q$ e supponiamo per assurdo che $(\mathcal{A}, \xi^{\mathcal{A}}) \not\models P \rightarrow Q$; ciò vuol dire che $v^{(\mathcal{A},\xi^{\mathcal{A}})}(P \rightarrow Q) = max(1 - v^{(\mathcal{A},\xi^{\mathcal{A}})}(P), v^{(\mathcal{A},\xi^{\mathcal{A}})}(Q)) = 0$ da cui segue $v^{(\mathcal{A},\xi^{\mathcal{A}})}(Q) = 0$ e $v^{(\mathcal{A},\xi^{\mathcal{A}})}(P) = 1$ che contraddice l'ipotesi.

*6.* $(\mathcal{A}, \xi^{\mathcal{A}}) \models \forall x P$ se e solo se $min\{v^{(\mathcal{A},\xi^{\mathcal{A}}[^a/_x])}(P) \mid \text{per ogni } a \in D_{\mathcal{A}}\} = 1$ se e solo se per ogni $a \in D_{\mathcal{A}}$, $v^{(\mathcal{A},\xi^{\mathcal{A}}[^a/_x])}(P) = 1$ se e solo se $(\mathcal{A}, \xi^{\mathcal{A}}[^a/_x]) \models P$ per ogni $a \in D_{\mathcal{A}}$.
□

In altri termini, il lemma precedente asserisce che nelle dimostrazioni è possibile sostituire i connettivi con i corrispettivi termini nel metalinguaggio ed interpretare gli atomi verificando le relazioni nella struttura.

**Esempio 4.10** Sia $P$ la formula ben formata $\forall xy \exists z C(f(x,y),z)$. Sia $(\mathcal{A}, \xi^{\mathcal{A}})$ un'interpretazione tale che:

- $D_{\mathcal{A}} = \mathcal{N}$
- $f^{\mathcal{A}} = (x,y) \mapsto x - y$
- $C^{\mathcal{A}} = \{(n,m) \mid n, m \in D_{\mathcal{A}} \text{ e } n = m\}$

$(\mathcal{A}, \xi^{\mathcal{A}}) \models \forall xy \exists z C(f(x,y),z)$ se e solo se per ogni $n, m \in \mathcal{N}$ esiste un numero naturale $p$ tale che $p = n - m$; e questo è chiaramente vero.

Per dimostrare la soddisfacibilità di una formula basta esibire un particolare modello per essa; se la formula non è particolarmente complicata, è spesso possibile trovare in modo intuitivo una interpretazione che la soddisfi.
Al contrario, per dimostrarne la validità è necessario considerare tutte le possibili interpretazioni per la formula. In questo caso, la semantica non aiuta molto: essenzialmente essa riduce i quantificatori ai rispettivi termini nel metalinguaggio o più formalmente ad operazioni algebriche di minimi e massimi. Illustriamo la tal cosa con un esempio.

**Esempio 4.11** Consideriamo la formula $P$ data da

$$\exists x \forall y A(x,y) \rightarrow \forall y \exists y A(x,y)$$

Vogliamo dimostrare che $P$ è valida. Consideriamo una generica interpretazione $(\mathcal{A}, \xi^{\mathcal{A}})$. Per il Lemma 4.28, $(\mathcal{A}, \xi^{\mathcal{A}}) \models P$ se e solo se il fatto che per qualche $a \in D_{\mathcal{A}}$ si ha che per ogni $b \in D_{\mathcal{A}}$, $v^{(\mathcal{A},\xi^{\mathcal{A}}[^a/_x][^b/_y])}(A(x,y)) = 1$ implica che per ogni $c \in D_{\mathcal{A}}$ si ha che per qualche $d \in D_{\mathcal{A}}$,
$v^{(\mathcal{A},\xi^{\mathcal{A}}[^d/_x][^c/_y])}(A(x,y)) = 1$ A questo punto è necessario "ragionare" nel metalinguaggio, in accordo alla semantica intuitiva di "per ogni" e "per qualche". Fissiamo in particolare un $a \in D_{\mathcal{A}}$ tale che per ogni $b \in D_{\mathcal{A}}$ risulti

$$v^{(\mathcal{A},\xi^{\mathcal{A}}[^a/_x][^b/_y])}(A(x,y)) = 1$$

Questo significa che ogni elemento del dominio è in relazione $A$ con $a$. Dunque per ogni $c \in D_{\mathcal{A}}$ si ha che per qualche $d \in D_{\mathcal{A}}$ (in particolare per per l'elemento $a$ suddetto), $v^{(\mathcal{A},\xi^{\mathcal{A}}[^d/_x][^c/_y])}(A(x,y)) = 1$.

Tutto questo non risulta particolarmente intuitivo, nè divertente. Vedremo nel prossimo capitolo come sia possibile ridurre il problema semantico della validità al problema sintattico della dimostrabilità. In altri termini, se vogliamo mostrare la validità di una formula, la cosa più semplice da fare è spesso dimostrarla formalmente in un qualche calcolo logico.
Al contrario, la semantica è estremamente utile per dimostrare la *non validità* di una formula $P$ (mentre è spesso complicato provare la *non dimostrabilità* in un calcolo logico). A tal fine, è sufficiente esibire una interpretazione che *non soddisfa* $P$.

**Esempio 4.12** Consideriamo la formula $P$ data da

$$\exists x(A(x) \to B(x)) \to (\exists x A(x) \to \exists x B(x))$$

Vogliamo trovare una interpretazione che non soddisfa $P$. Essendo $P$ una formula del tipo $P_1 \to P_2$, l'interpretazione deve soddisfare la premessa $P_1 = \exists x(A(x) \to B(x))$ ma non la conclusione $P_2 = \exists x A(x) \to \exists x B(x)$. Questo significa che l'interpretazione deve soddisfare $\exists x A(x)$ ma non $\exists x B(x)$. A questo punto, l'interpretazione del predicato $B$ è necessariamente una relazione vuota (mentre l'interpretazione di $A$ è necessariamente non vuota). Riconsideriamo allora $P_1 = \exists x(A(x) \to B(x))$. Poichè il predicato $B$ è sempre falso, l'unico modo per rendere vera l'implicazione interna, è che per qualche $x$ l'interpretazione di $A(x)$ risulti falsa. Dunque basta interpretare $A$ come un qualunque predicato che non è nè sempre vero nè sempre falso. Ad esempio, preso come dominio l'insieme dei numeri naturali, si può interpretare $A(x)$ come $\{x \mid x \text{ è un numero pari }\}$ e $B(x) = \emptyset$.
Il lettore verifichi per esercizio che questa interpretazione non soddisfa $P$.

### 4.3.3 Equivalenza semantica

Il concetto di equivalenza (semantica) nella logica dei predicati è il seguente:

**Definizione 4.29** *Due formule ben formate $P$ e $Q$ sono* (semanticamente) equivalenti *(si scrive $P \equiv Q$) se per tutte le interpretazioni $(\mathcal{A}, \xi^{\mathcal{A}})$ di $P$ e $Q$ si ha:*

$$v^{(\mathcal{A},\xi^{\mathcal{A}})}(P) = v^{(\mathcal{A},\xi^{\mathcal{A}})}(Q)$$

***Osservazione*** Due fbf sono equivalenti se e solo se $\models P \leftrightarrow Q$.

Per come è stata definita la funzione di valutazione $v^{(\mathcal{A},\xi^{\mathcal{A}})}$, tutte le equivalenze semantiche dimostrate per la logica proposizionale valgono anche in quella dei predicati.
Come primo esempio interessante di equivalenza semantica, dimostriamo che ridenominando una variabile quantificata in modo consistente all'interno del campo d'azione del quantificatore si ottiene una nuova formula semanticamente

equivalente a quella di partenza. A questo scopo, abbiamo bisogno del seguente lemma, noto come lemma di sostituzione[5].

**Lemma 4.30**

1. *Se $y$ non occorre in $t$, allora $[\![t]\!]^{\mathcal{A}}_{\xi[^a/_x]} = [\![t[^y/_x]]\!]^{\mathcal{A}}_{\xi[^a/_y]}$;*

2. *Se $y$ non occorre in $P$, allora $v^{(\mathcal{A},\xi^{\mathcal{A}}[^a/_x])}(P) = v^{(\mathcal{A},\xi^{\mathcal{A}}[^a/_y])}(P[y/x])$.*

*Dimostrazione.*

1. La dimostrazione si effettua mediante una semplice induzione sulla struttura del termine $t$, ed è lasciata come esercizio per il lettore.

2. Per induzione sulla struttura della formula $P$. Gli unici casi interessanti sono quelli concernenti i quantificatori. Trattiamo qui solo il caso in cui $P$ sia $\forall z P_1$. Il caso in cui $P$ è $\exists z P_1$ è del tutto analogo.
Dobbiamo dimostrare che

$$v^{(\mathcal{A},\xi^{\mathcal{A}}[^a/_x])}(\forall z P_1) = v^{(\mathcal{A},\xi^{\mathcal{A}}[^a/_y])}((\forall z P_1)[^y/_x])$$

Vi sono due possibilità:

- Se $z = x$, per definizione di sostituzione $(\forall z P_1)[^y/_x] = \forall z P_1$. Ovviamente, $x \notin FV(\forall z P_1)$, e quindi

$$v^{(\mathcal{A},\xi^{\mathcal{A}}[^a/_x])}(\forall z P_1) = v^{(\mathcal{A},\xi^{\mathcal{A}})}(\forall z P_1)$$

  Inoltre, per ipotesi, $y \notin FV(\forall z P_1)$, e dunque anche

$$v^{(\mathcal{A},\xi^{\mathcal{A}}[^a/_y])}((\forall z P_1)[^y/_x]) = v^{(\mathcal{A},\xi^{\mathcal{A}}[^a/_y])}(\forall z P_1) = v^{(\mathcal{A},\xi^{\mathcal{A}})}(\forall z P_1)$$

- Se $x \neq z$. Poichè $y$ non occorre in $\forall z P_1$ possiamo supporre $y \neq z$. Allora $(\forall z P_1)[^y/_x] = \forall z P_1[^y/_x]$. Per ipotesi induttiva, per ogni ambiente $\xi^{\mathcal{A}}$

$$v^{(\mathcal{A},\xi^{\mathcal{A}}[^a/_x])}(P_1) = v^{(\mathcal{A},\xi^{\mathcal{A}}[^a/_y])}(P_1[^y/_x])$$

  indipendentemente dall'interpretazione di $z$, da cui segue l'asserto.

□

**Teorema 4.31** *Siano $P$ una formula ben formata e $z$ una variabile che non occorre in $P$. Allora*

1. $\exists x P \equiv \exists z P[^z/_x]$

[5] La presente versione del lemma di sostituzione è leggermente più debole di quella abituale. In particolare, l'ipotesi che la variabile $y$ non occorra in $t$ o in $P$ può essere indebolita all'ipotesi che $y$ non occorra *libera* in $t$ o $P$. Tuttavia, il presente enunciato, la cui dimostrazione è più semplice, è sufficiente per i nostri scopi.

*2.* $\forall x P \equiv \forall z P[^{z}/_{x}]$

*Dimostrazione.* Segue immediatamente dal lemma di sostituzione. □

Vediamo ora delle utili proprietà che coinvolgono i quantificatori. Il teorema che segue rappresenta una generalizzazione delle Leggi di De Morgan; esso consente di spostare i quantificatori attraverso il connettivo ($\neg$);

**Teorema 4.32** *Sia $P$ una formula ben formata:*

*1.* $\neg\forall x P \equiv \exists x \neg P$

*2.* $\neg\exists x P \equiv \forall x \neg P$

*3.* $\forall x P \equiv \neg\exists x \neg P$

*4.* $\exists x P \equiv \neg\forall x \neg P$

*Dimostrazione.*

1. Per il Lemma 4.28 $(\mathcal{A}, \xi^{\mathcal{A}}) \models \neg\forall x P$ se e solo se $(\mathcal{A}, \xi^{\mathcal{A}}) \not\models \forall x P$, se e solo se non per tutti i $b \in D_{\mathcal{A}}$ $(\mathcal{A}, \xi^{\mathcal{A}}[^{b}/_{x}]) \models P$, se e solo se esiste un $b \in D_{\mathcal{A}}$ tale che $(\mathcal{A}, \xi^{\mathcal{A}}[^{b}/_{x}]) \not\models P$, se e solo se esiste $b \in D_{\mathcal{A}}$ tale che $(\mathcal{A}, \xi^{\mathcal{A}}[^{b}/_{x}]) \models \neg P$, se e solo se $(\mathcal{A}, \xi^{\mathcal{A}}) \models \exists x \neg P$.

2. Simile alla 1.

3. e 4. Si possono ottenere da 1. e 2. □

L'ordine dei quantificatori dello stesso tipo è irrilevante; inoltre, un quantificatore che lega una variabile che non occorre libera nella fbf che si trova nel suo campo d'azione, può essere cancellato. Più formalmente:

**Teorema 4.33** *Sia $P$ una formula ben formata, allora:*

*1.* $\forall x \forall y P \equiv \forall y \forall x P$

*2.* $\exists x \exists y P \equiv \exists y \exists x P$

*3.* $\forall x P \equiv P$ *se* $x \notin FV(P)$

*4.* $\exists x P \equiv P$ *se* $x \notin FV(P)$

*Dimostrazione.* È lasciata al lettore come esercizio. □

Abbiamo in precedenza osservato che $\forall$ e $\exists$ rappresentano una generalizzazione rispettivamente dei connettivi $\wedge$ e $\vee$; dunque non sorprende il fatto che $\forall$ (e $\exists$) sia distributiva su $\wedge$ (rispettivamente su $\vee$). Mentre $\forall$ (e $\exists$) è distributiva su $\vee$ (rispettivamente su $\wedge$) se sono verificate determinate condizioni. Vediamo, infatti, il seguente teorema:

**Teorema 4.34** *Siano $P_1$ e $P_2$ formule ben formate, allora*

1. $\forall x(P_1 \wedge P_2) \equiv \forall x P_1 \wedge \forall x P_2$
2. $\exists x(P_1 \vee P_2) \equiv \exists x P_1 \vee \exists x P_2$
3. $\forall x(P_1 \vee P_2) \equiv \forall x P_1 \vee P_2$ *se* $x \notin FV(P_2)$
4. $\exists x(P_1 \wedge P_2) \equiv \exists x P_1 \wedge P_2$ *se* $x \notin FV(P_2)$

*Dimostrazione.* I primi due casi sono immediati; proviamo il caso *4.* e lasciamo il *3.* come esercizio per il lettore.
Per il Lemma 4.28, per ogni $(\mathcal{A}, \xi^{\mathcal{A}})$, risulta $(\mathcal{A}, \xi^{\mathcal{A}}) \models \exists x P_1 \wedge P_2$ se e solo se $(\mathcal{A}, \xi^{\mathcal{A}}) \models \exists x P_1$ e $(\mathcal{A}, \xi^{\mathcal{A}}) \models P_2$, se e solo se $\exists b \in D_{\mathcal{A}}$ tale che $(\mathcal{A}, \xi^{\mathcal{A}}[^b/_x]) \models P_1$ e (poiché $x \notin \mathrm{FV}(P_2)$) $(\mathcal{A}, \xi^{\mathcal{A}}[^b/_x]) \models P_2$, se e solo se $\exists b \in D_{\mathcal{A}}$ tale che $(\mathcal{A}, \xi^{\mathcal{A}}[^b/_x]) \models P_1 \wedge P_2$, se e solo se $(\mathcal{A}, \xi^{\mathcal{A}}) \models \exists x(P_1 \wedge P_2)$. □

Osserviamo che se $x \in \mathrm{FV}(P_2)$

- $\forall x(P_1 \vee P_2) \not\equiv \forall x P_1 \vee \forall x P_2$
- $\exists x(P_1 \wedge P_2) \not\equiv \exists x P_1 \wedge \exists x P_2$

Ad esempio, dire che

> " per ogni intero x, x è pari oppure x è dispari"
> $(\forall x(P(x) \vee D(x)))$

è una cosa ben diversa dal dire

> "ogni intero x è pari oppure ogni intero x è dispari"
> $(\forall x P(x) \vee \forall x D(x))$

Allo stesso modo, dire

> "esiste un intero x pari ed esiste un intero x dispari"
> $(\exists x P(x) \wedge \exists x D(x))$

non è logicamente equivalente a

> "esiste un intero x che è pari e dispari"
> $(\forall x(P(x) \wedge D(x)))$

Tuttavia, è sempre possibile spostare i quantificatori attraverso i connettivi $\wedge$ e $\vee$, ridenominando in modo opportuno le variabili. In particolare:

**Teorema 4.35** *Siano $P_1$ e $P_2$ formule ben formate, allora*

1. $\mathcal{Q}_1 x P_1 \vee \mathcal{Q}_2 x P_2 \equiv \mathcal{Q}_1 x \mathcal{Q}_2 z(P_1 \vee P_2[^z/_x])$
2. $\mathcal{Q}_3 x P_1 \wedge \mathcal{Q}_4 x P_2 \equiv \mathcal{Q}_3 x \mathcal{Q}_4 z(P_1 \wedge P_2[^z/_x])$

*con $\mathcal{Q}_i \in \{\exists, \forall\}$ e $z \notin FV(P_1) \cup FV(P_2)$.*

*Dimostrazione.* Segue dai Teoremi 4.34 e 4.31. □

In modo analogo si possono spostare i quantificatori all'esterno del connettivo di implicazione ($\rightarrow$); infatti

**Teorema 4.36** *Se $x \notin FV(Q)$*

1. $\forall x P \rightarrow Q \equiv \exists x(P \rightarrow Q)$
2. $\exists x P \rightarrow Q \equiv \forall x(P \rightarrow Q)$
3. $Q \rightarrow \exists x P \equiv \exists x(Q \rightarrow P)$
4. $Q \rightarrow \forall x P \equiv \forall x(Q \rightarrow P)$

*Dimostrazione.* È lasciata al lettore come esercizio. □

Si noti che l'ipotesi $x \notin \mathrm{FV}(Q)$ del teorema precedente può essere sempre verificata, a condizione di ridenominare in modo opportuno le variabili legate. Ad esempio, se $y \notin FV(B)$

$$\forall x A(x) \rightarrow B(x) \equiv \forall y A(y) \rightarrow B(x) \equiv \exists y(A(y) \rightarrow B(x))$$

***Osservazione*** La logica dei predicati è chiamata anche logica del primo ordine in quanto i quantificatori agiscono solo sulle variabili individuali; al contrario, in un *linguaggio del secondo ordine* è possibile avere variabili funzionali e predicative. Ad esempio $\forall P(P(x, y) \rightarrow P(y, x))$ è una formula della logica del secondo ordine il cui "significato" è il seguente: nel modello in cui si interpreta tale formula, ogni volta che vale $P(x, y)$ per un certo predicato $P$, vale anche $P(y, x)$.

### 4.3.4 Forma Normale Prenessa

Nel paragrafo 1.4.6 sono state introdotte due forme normali per la logica proposizionale: la forma normale congiuntiva e quella disgiuntiva. Nella logica del primo ordine è utile considerare una nuova forma normale, detta prenessa, in cui tutti i quantificatori compaiono "in testa" alla formula.

**Definizione 4.37** *Una formula ben formata $P$ è in* forma normale prenessa *se ha la forma $\mathcal{Q}_1 x_1 \mathcal{Q}_2 x_2 \ldots \mathcal{Q}_n x_n P_1$ con $n \geq 0$, dove $\mathcal{Q}_i \in \{\exists, \forall\}$ e $P_1$ non contiene quantificatori.*
*$\mathcal{Q}_1 x_1 \mathcal{Q}_2 x_2 \ldots \mathcal{Q}_n x_n$ viene detto* prefisso*, mentre $P_1$* matrice *della formula $P$.*

**Esempio 4.13** Le seguenti formule ben formate sono in forma normale prenessa: $\forall x \exists y \forall z(A(x, y) \vee B(z) \rightarrow C(x))$, $\forall x \exists z(\neg A(x) \rightarrow B(z, x))$

**Teorema 4.38** *Per ogni $P \in FBF$ esiste una forma normale prenessa $P^P$ equivalente.*

*Dimostrazione.* Per induzione sulla struttura di $P$.
(*caso base*) Se $P$ è un atomo allora è già in forma normale prenessa. Veniamo al caso induttivo.

• Se $P$ è $\neg P_1$. Supponiamo, per ipotesi induttiva che $P_1^P$ sia la formula in forma normale prenessa equivalente a $P_1$; l'asserto segue dal Teorema 4.32.

• Se $P$ è $P_1 \wedge P_2$ oppure $P_1 \vee P_2$ o $P_1 \rightarrow P_2$. Supponiamo, per ipotesi induttiva che $P_1^P$ e $P_2^P$ siano le formule in forma normale prenessa equivalenti, rispettivamente, a $P_1$ ed a $P_2$; l'asserto segue dai Teoremi 4.35 e 4.36.

• Se $P$ è $\mathcal{Q}xP_1$ con $\mathcal{Q} \in \{\forall, \exists\}$. Supponiamo, per ipotesi induttiva, che $P_1^P$ sia la formula in forma normale prenessa equivalente a $P_1$, allora $\mathcal{Q}xP^P$ è in forma normale prenessa.

□

**Esempio 4.14** Cerchiamo la forma normale prenessa di $\forall x A(x) \rightarrow \neg\forall y B(y)$.

$$\begin{aligned} \forall x A(x) \rightarrow \neg\forall y B(y) &\equiv \forall x A(x) \rightarrow \exists y \neg B(y) \\ &\equiv \exists y(\forall x A(x) \rightarrow \neg B(y)) \\ &\equiv \exists y \exists x(A(x) \rightarrow \neg B(y)) \end{aligned}$$

ed $\exists y \exists x(A(x) \rightarrow \neg B(y))$ è in forma normale prenessa.

### 4.3.5 Forma di Skolem

È possibile eliminare i quantificatori esistenziali in una formula ben formata introducendo opportunamente dei simboli di funzione. La formula che ne risulta si dice essere in forma di Skolem.
A differenza delle forme normali discusse in precedenza, una formula ben formata non è equivalente alla corrispondente forma di Skolem; tuttavia quest'ultima gode della seguente proprietà: è soddisfacibile se e solo se lo è la formula di partenza.
Prima di mostrare la definizione formale di forma di Skolem vediamo di presentarla in modo intuitivo.

- Consideriamo la seguente fbf

  $$\exists x A(x)$$

  il cui significato, per il Lemma 4.28 è esprimibile con una frase del tipo "esiste $x$ per cui vale $A(x)$", cioè si afferma l'esistenza di un certo "oggetto" avente una data proprietà. È possibile eliminare il quantificatore esistenziale introducendo un simbolo di costante, supponiamo, per fissare le idee, $c$, ottenendo la fbf $A(c)$; in tal modo si crea un nome per l' "oggetto" di cui si è predicata l'esistenza e se ne afferma la proprietà ($A$).

- Quando in una fbf sono presenti dei quantificatori universali, l'eliminazione di quelli esistenziali non è altrettanto semplice. Consideriamo, per fissare le idee, la fbf $P$

  $$\forall x \exists y B(x, y).$$

  Sia $(\mathcal{A}, \xi^{\mathcal{A}})$ un'interpretazione tale che:

- $D_{\mathcal{A}} = \mathcal{N}$
- $B^{\mathcal{A}} = \{(x, y) \mid y \text{ è multiplo di x}\}$

per il Lemma 4.28 $(\mathcal{A}, \xi^{\mathcal{A}})$ è un modello per $P$ in quanto "per ogni $x \in \mathcal{N}$ esiste $y \in \mathcal{N}$ tale che $y$ è multiplo di $x$"; se eliminassimo il quantificatore esistenziale introducendo un simbolo di costante, ad esempio $d$, ci troveremmo ad affermare che "ogni numero naturale ha come multiplo $d$". In realtà, per eliminare i quantificatori esistenziali che si trovano nel campo di azione di quelli universali, è necessario introdurre dei simboli di funzione che esprimono, in sostanza, come ciò di cui si predica l'esistenza dipende da quello che le variabili rappresentano. Tornando alla fbf $P$, è possibile introdurre un simbolo di funzione $f$ trasformandola in $\forall x B(x, f(x))$ la cui interpretazione in $(\mathcal{A}, \xi^{\mathcal{A}})$ è "per ogni $x \in \mathcal{N}$, $f(x)$ è un multiplo di $x$". Tale formula non è equivalente alla precedente, tuttavia è ancora soddisfacibile: basta interpretare il simbolo di funzione $f$ con una qualunque funzione che associa ad un intero $n$ un suo multiplo.

**Definizione 4.39** *Sia $P$ una fbf in forma normale prenessa, i.e.*
*$P$ è $\mathcal{Q}_1 x_1 \mathcal{Q}_2 x_2 \ldots \mathcal{Q}_n x_n P_1$; si dice* forma di Skolem *di $P$, e si indica con $P^S$, la fbf ottenuta con il seguente procedimento, detto skolemizzazione: per ogni occorrenza $\mathcal{Q}_t$ di un quantificatore esistenziale*

- *se $\mathcal{Q}_t$ non è nel campo di azione di alcun quantificatore universale, si introduce una costante $c$ che non compare in $P_1$, si cancella $\mathcal{Q}_t x_t$ dal prefisso di $P$ e si sostituisce $c$ ad $x_t$ in $P_1$ (cioè $P_1[{}^{c}/_{x_t}]$);*

- *siano, invece, $\mathcal{Q}_{i,1}, \ldots \mathcal{Q}_{i,m}$ con $m > 0$ i quantificatori universali nel cui campo di azione si trova $\mathcal{Q}_t$, allora si sceglie un simbolo di costante $g^m$ che non compare in $P_1$, si cancella $\mathcal{Q}_t x_t$ dal prefisso di $P$ e si sostituisce $g(x_{i,1}, \ldots, x_{i,m})$ ad $x_t$ in $P_1$ (cioè $P_1[{}^{g(x_{i,1},\ldots,x_{i,m})}/_{x_t}]$).*

*Le costanti e le funzioni introdotte in tal modo sono dette rispettivamente* costanti *e* funzioni di Skolem.

**Esempio 4.15** La fbf $\exists x \forall y \forall z \exists t B(x, y, z, t)$ ha la seguente forma di Skolem:

$$\forall y \forall z B(c, y, z, f(y, z))$$

Come già accennato in precedenza, la trasformazione di una formula in forma di Skolem non preserva l'equivalenza.
Infatti, per fissare le idee, consideriamo la fbf $\exists x A(x)$ discussa in precedenza. Non è esattamente la stessa cosa asserire $\exists x A(x)$ oppure $A(c)$. Per convincersi di ciò basta considerare, ad esempio, un'interpretazione $(\mathcal{A}, \xi^{\mathcal{A}})$ tale che $D_{\mathcal{A}} = \mathcal{N}$, $c^{\mathcal{A}} = 2$ e $A^{\mathcal{A}} = \{x \in \mathcal{N} \mid x \text{ è il più piccolo numero naturale}\}$, in tale interpretazione la prima formula risulta vera mentre la seconda no. Tuttavia è possibile provare che

**Teorema 4.40** *Per ogni formula ben formata $P$, $P$ è soddisfacibile se e solo se $P^S$ lo è.*

*Dimostrazione.* È sufficiente mostrare che ogniqualvolta si elimina un quantificatore esistenziale dal prefisso di una fbf $P_1$ utilizzando il procedimento illustrato nella Definizione 4.39, si ottiene una fbf $P_1'$ che è soddisfacibile se e solo se lo è $P_1$. Infatti, siano

$$P_1 = \mathcal{Q}_1 x_1 \dots \mathcal{Q}_t x_t \dots \mathcal{Q}_n x_n A(x_1, \dots, x_t, \dots, x_n) \quad \text{e}$$

$P_1' = \mathcal{Q}_1 x_1 \dots \mathcal{Q}_{t-1} x_{t-1} \mathcal{Q}_{t+1} x_{t+1} \dots \mathcal{Q}_n x_n A(x_1, \dots, x_{t-1}, f(x_{i,1}, \dots, x_{i,m}), x_{t+1}, \dots, x_n)$ dove $\mathcal{Q}_{i,1} \dots \mathcal{Q}_{i,m}$ sono i quantificatori universali nel cui campo di azione si trova $\exists\, x_t$.
Supponiamo che $P_1$ sia insoddisfacibile, allora anche $P_1'$ lo è in quanto, in caso contrario esisterebbe un'interpretazione che è un modello per $P_1'$ e dunque tale per cui, in base al Lemma 4.28, risulterebbe che per ogni $x_{i,1}, \dots, x_{i,m}$ esiste $f(x_{i,1}, \dots, x_{i,m})$ che verifica

$$\mathcal{Q}_{t+1} x_{t+1} \dots \mathcal{Q}_n x_n A(x_1, \dots, x_{t-1}, f(x_{i,1}, \dots, x_{i,m}), x_{t+1}, \dots, x_n)$$

Ma questo è assurdo avendo supposto che $P_1$ è insoddisfacibile.
Viceversa, se esiste un'interpretazione $(\mathcal{A}, \xi^{\mathcal{A}})$ che è un modello per $P_1$, allora per ogni $x_{i,1}, \dots, x_{i,m} \in D_{\mathcal{A}}$ esiste un certo $x_t \in D_{\mathcal{A}}$ che verifica

$$\mathcal{Q}_{t+1} x_{t+1} \dots \mathcal{Q}_n x_n A(x_1, \dots, x_{t-1}, x_t, x_{t+1}, \dots, x_n)$$

Quindi posta $f^{\mathcal{A}} = (x_{i,1}, \dots, x_{i,m}) \mapsto x_t$, $(\mathcal{A}, \xi^{\mathcal{A}})$ risulta un modello di $P_1'$. □

**Proposizione 4.41** *Sia $P$ una fbf. È possibile trasformarla in una formula $P'$ chiusa ed in forma di Skolem, soddisfacibile se e solo se $P$ lo è.*

*Dimostrazione.* Sia $\mathrm{FV}(P) = \{x_1, \cdots, x_n\}$. Per il Teorema 4.27 $P$ è soddisfacibile se e solo se $\exists x_1 \cdots x_n P$ lo è. Posto $P' = (\exists x_1 \cdots x_n P)^S$, L'asserto segue dal teorema precedente. □

### 4.3.6 Esempi di linguaggi del primo ordine

## Uguaglianza

Un predicato di particolare rilevanza matematica è l'uguaglianza. A causa della sue peculiarità, tale predicato viene spesso considerato, dal punto di vista logico, come un particolare simbolo logico dell'alfabeto del linguaggio (che dunque richiede una specifica interpretazione) piuttosto che come un semplice simbolo di predicato. Parleremo in questo caso di *logiche con uguaglianza*. Se si vuole trattare l'uguaglianza come un normale simbolo di predicato, si deve allora cercare di assiomatizzarne le proprietà caratteristiche. Queste possono essere descritte al primo ordine nel modo seguente:

1. $\forall x (x = x)$;
2. $\forall xy (x = y \rightarrow y = x)$;

3. $\forall xyz(x = y \wedge y = z \rightarrow x = z)$;

4. per ogni simbolo di funzione n-ario $f^n$ del linguaggio,
   $\forall x_1 \dots x_n y_1 \dots y_n$
   $(x_1 = y_1 \wedge \dots \wedge x_n = y_n \rightarrow f^n(x_1, \dots, x_n) = f^n(y_1, \dots, y_n))$;

5. per ogni simbolo di predicato n-ario $P^n$ del linguaggio,
   $\forall x_1 \dots x_n y_1 \dots y_n$
   $(x_1 = y_1 \wedge \dots \wedge x_n = y_n \rightarrow (P^n(x_1, \dots, x_n) \rightarrow P^n(y_1, \dots, y_n)))$.

I primi tre assiomi asseriscono che l'uguaglianza è una relazione di equivalenza; gli ultimi due, che è una congruenza rispetto a tutti i simboli di funzioni e predicati del linguaggio in esame.
Ovviamente tali assiomi sono validi nelle logiche con uguaglianza.
È importante sottolineare che questi non sono sufficenti per garantire che il predicato "=" sia effettivamente interpretato come identità in una data struttura. Infatti, 1-3 esprimono il fatto che "=" è una relazione di equivalenza, mentre 4 e 5 equivalgono ad affermare che "=" è una congruenza. Tuttavia tali assiomi descrivono in modo completo il senso logico intuitivo del predicato di uguaglianza.
D'ora in poi, quando utilizzeremo il simbolo "=", assumeremo implicitamente che sussistano anche gli assiomi precedenti.

## Tipi di dato

Tutti i principali tipi di dato utilizzati in informatica (liste, alberi, pile, etc.) possono essere facilmente formalizzati al primo ordine. Consideriamo, ad esempio, il caso delle pile (stack). Il linguaggio conterrà una costante *nil* per indicare la pila vuota, e una funzione binaria $push(s, a)$ che presi in input uno stack ed un elemento restituisce lo stack al quale è stato aggiunto l'elemento sulla sommità. Le operazioni fondamentali sono le funzioni $top(s)$ e $pop(s)$ che restituiscono rispettivamente l'elemento in cima alla pila e la pila alla quale è stato tolto l'ultimo elemento inserito. Queste possono essere semplicemente formalizzate dagli assiomi:

1. $\forall xy(top(push(x, y)) = y)$
2. $\forall xy(pop(push(x, y)) = x)$
3. $\forall x(push(pop(x), top(x)) = x)$

## Relazioni di ordinamento

L'alfabeto si compone di due predicati binari: $=$ e $\leq$. Un *ordinamento parziale* è una struttura che soddisfa i seguenti assiomi:

1. $\forall xyz(x \leq y \wedge y \leq z \rightarrow x \leq z)$
2. $\forall xy(x \leq y \wedge y \leq x \leftrightarrow x = y)$

L'esistenza di un elemento minimo è formalizzata dal predicato $\exists x \forall y (x \leq y)$. L'ordinamento è detto totale se $\forall xy(x \leq y \vee y \leq x)$.
Possiamo *definire* il predicato binario $x < y$ come abbreviazione sintattica per $x \leq y \ \wedge \neg(x = y)$. Diremo allora che una relazione di ordinamento è *densa* se $\forall xy(x < y \rightarrow \exists z(x < z \wedge z < y))$.

## Gruppi

L'alfabeto è composto da un simbolo di costante $e$, un simbolo di funzione unario $(\ )^{-1}$ (funzione di inversione), ed un simbolo di funzione binario $\cdot$ (composizione).
Come unico predicato si considera quello di uguaglianza.
Un *gruppo* è un modello di:

1. $\forall xyz((x \cdot y) \cdot z = x \cdot (y \cdot z))$
2. $\forall x(x \cdot e = x \wedge e \cdot x = x)$
3. $\forall x(x \cdot x^{-1} = e \wedge x^{-1} \cdot x = e)$

Il gruppo è detto *abeliano* se vale la proprietà commutativa, i.e.

$$\forall xy(x \cdot y = y \cdot x)$$

L'unicità dell'elemento neutro di un gruppo è espressa dal predicato

$$\forall xy((y \cdot x = y \wedge x \cdot y = y) \rightarrow y = e)$$

In generale, dato un predicato $P$, l'esistenza ed unicità di un elemento $x$ per cui vale $P$ (abitualmente indicato con la notazione $\exists! x P$) può essere definita al primo ordine come

$$\exists x(P(x) \wedge \forall z(P(z) \rightarrow x = z))$$

## Aritmetica

L'alfabeto si compone di una costante 0, un simbolo di funzione unario $s$ per il successore, e due simboli di funzione binaria $+$ e $*$ per somma e prodotto.
Come unico simbolo di predicato si considera l'uguaglianza. La seguente formalizzazione dell'aritmetica, che nel seguito indicheremo con PA, si deve al matematico italiano Peano:

1. $\forall x(\neg 0 = s(x))$
2. $\forall xy(s(x) = s(y) \rightarrow x = y)$
3. $\forall x(x + 0 = x)$
4. $\forall xy(x + s(y) = s(x + y))$
5. $\forall x(x * 0 = 0)$

6. $\forall xy(x * s(y) = x + (x * y))$
7. $(P(0) \wedge \forall x(P(x) \rightarrow P(s(x)))) \rightarrow \forall x P(x)$

L'ultimo "assioma" è in realtà uno *schema* di assioma: $P(x)$ può essere un qualunque predicato definibile nel linguaggio che contiene un'unica variabile libera $x$.
L'interpretazione intesa di PA è la struttura ordinaria dei numeri naturali nella quale "=" è interpretato come l'uguaglianza tra numeri naturali, 0 come il naturale 0, $s$ come la funzione successore, $+$ e $*$ rispettivamente come somma e prodotto.
Tale interpretazione, detta *modello standard* dell'aritmetica, non è l'unica possibile; nel paragrafo 5.4.5 discuteremo l'esistenza di interpretazioni di natura differente.

## Insiemi $\star$

Vediamo come è possibile formalizzare la nozione intuitiva di insieme nella logica del primo ordine. L'assiomatizzazione della Teoria degli Insiemi è particolarmente "delicata". Supponiamo di introdurre un predicato binario (infisso) $x \in y$ per denotare l'appartenenza di $x$ ad $y$.
Una proprietà naturale che ci si aspetta è che, dato un qualunque predicato $P(x)$, si possa formare l'insieme degli elementi $x$ per cui vale $P(x)$, cioè che esista un insieme $z$ (intuitivamente, $z = \{x \mid P(x)\}$) tale che $x \in z \leftrightarrow P(x)$, che conduce al seguente schema di assioma, noto come *assioma di comprensione*

$$\exists z \forall x(x \in z \leftrightarrow P(x))$$

Questo fu introdotto da Cantor alla fine del secolo scorso. Nel 1902, Russel mostrò in una lettera a Frege come questo assioma conduca in realtà ad un paradosso, noto appunto come paradosso di Russel. L'idea è estremamente semplice. Poiché lo schema di comprensione vale per ogni predicato $P(x)$, allora vale anche per $P(x) = \neg(x \in x)$. In particolare, si ha

$$\exists z \forall x(x \in z \leftrightarrow \neg(x \in x))$$

È facile dimostrare che questa formula è insoddisfacibile. Infatti, supponiamo per assurdo che esista una interpretazione $(\mathcal{A}, \xi^{\mathcal{A}})$ che la soddisfa. Allora deve esistere un elemento $a \in D_{\mathcal{A}}$ tale che $(\mathcal{A}, \xi^{\mathcal{A}}[^a/_z]) \models \forall x(x \in z \leftrightarrow \neg(x \in x))$. Poichè questa formula deve essere soddisfatta per ogni $x$, in particolare deve essere soddisfatta interpretando $x$ con $a$, cioè si deve avere $(\mathcal{A}, \xi^{\mathcal{A}}[^a/_z][^a/_x]) \models x \in z \leftrightarrow \neg(x \in x)$. Ma questo è equivalente a dire $(\mathcal{A}, \xi^{\mathcal{A}}[^a/_x]) \models x \in x \leftrightarrow \neg(x \in x)$, che è ovviamente impossibile, indipendentemente dall'interpretazione di $\in$. In termini meno formali, consideriamo l'insieme $W = \{x | x \notin x\}$ (l'insieme di tutti gli insiemi che non contengono se stessi come elemento). Per come è definito $W$, si ha $W \in W \leftrightarrow W \notin W$, che è chiaramente contradditorio.
Dunque, l'assioma di comprensione, nella forma generale suggerita da Cantor,

conduce ad un assurdo. Una soluzione a questo paradosso fu suggerita da Zermelo, che osservò come nella pratica matematica si utilizzi abitualmente il principio di comprensione solo per formare dei sottoinsiemi di insiemi già definiti in precedenza. In altri termini, *sapendo* che $y$ è un insieme, si può definire il sottoinsieme degli elementi di $y$ che godono di una certa proprietà $P(x)$. Questo può essere assiomatizzato al primo ordine dal seguente schema di assioma, noto come *assioma di separazione*[6]:

$$\forall y \exists z \forall x (x \in z \leftrightarrow x \in y \wedge P(x))$$

ovvero, per ogni insieme $y$ esiste un insieme $z$ tale che i suoi elementi sono esattamente gli elementi di $y$ che inoltre godono della proprietà $P$.
Dopo questa breve introduzione, definiamo formalmente l'assiomatizzazione della teoria degli insiemi proposta da Zermelo e successivamente perfezionata da Fraenkel (l'uso di un linguaggio formalizzato per la formulazione della teoria di Zermelo-Fraenkel fu proposto per la prima volta da Skolem, nel 1922). Il linguaggio utilizzato contiene:

- un simbolo di relazione binaria $\in$ detto *appartenenza*
- un simbolo di relazione binaria $\subseteq$ detto *inclusione*
- una costante $\emptyset$ che denota l'*insieme vuoto*
- un simbolo di funzione binaria $\{x, y\}$ che denota l'*insieme della coppia non ordinata* di elementi $x$ e $y$
- un simbolo di funzione $\mathcal{P}(x)$ che denota l'*insieme delle parti* di $x$
- un simbolo di funzione $\cup x$ che denota l'unione degli elementi di $x$

La teoria è costituita dai seguenti assiomi:

**Estensionalità** Due insiemi che hanno gli stessi elementi sono uguali

$$\forall x \forall y (\forall z (z \in x \leftrightarrow z \in y) \rightarrow x = y)$$

**Insieme vuoto** Specifica che l'insieme vuoto non contiene elementi

$$\forall z \neg (z \in \emptyset)$$

**Coppia** Specifica il significato di $\{x, y\}$

$$\forall x \forall y \forall z (z \in \{x, y\} \leftrightarrow z = x \vee z = y)$$

Si noti che $\{x, x\}$ è l'insieme singoletto per $x$:

$$\forall x \forall z (z \in \{x, x\} \leftrightarrow z = x)$$

Scriveremo dunque $\{x\}$ come abbreviazione per $\{x, x\}$.

[6] *Aussonderungs axiom.*

**Inclusione** Definisce il predicato di inclusione

$$\forall x \forall y (x \subseteq y \leftrightarrow \forall z (z \in x \rightarrow z \in y))$$

**Potenza** Specifica che $\mathcal{P}(x)$ è l'insieme delle parti di $x$

$$\forall x \forall z (z \in \mathcal{P}(x) \leftrightarrow z \subseteq x)$$

**Separazione** Per ogni insieme $y$ esiste un insieme $z$ ottenuto separando da $y$ gli elementi che godono di una proprità $P$

$$\forall y \exists z \forall x (x \in z \leftrightarrow x \in y \wedge P(x))$$

**Infinito** Tale assioma postula l'esistenza di insiemi infiniti (numerabili)

$$\exists z (\emptyset \in z \wedge \forall x (x \in z \rightarrow \{x\} \in z))$$

In base a tale assioma, $\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \ldots$ devono appartenere tutti a $z$.

**Rimpiazzamento** Afferma che l'immagine di un insieme tramite una funzione è ancora un insieme. Ricordiamo che una funzione è una particolare relazione $P(x, y)$ totale e univoca; ciò si formalizza nel modo seguente

$$\forall x (\exists y (P(x, y) \wedge \forall z (P(x, z) \rightarrow y = z)))$$

o anche

$$\forall x \exists! y P(x, y)$$

L'assioma di rimpiazzamento stabilisce che se $P$ è una funzione allora per ogni insieme $x$ esiste un insieme $y$ che coincide con l'imagine di $x$ via $P$. Ovvero, dato un qualunque elemento $z$ questo appartiene a $y$ se e solo se esiste un $t$ tale che $t \in x$ e $P(t, z)$. Esplicitamente

$$\forall x \exists! y P(x, y) \rightarrow \forall x \exists y \forall z (z \in y \leftrightarrow \exists t (t \in x \wedge P(t, z)))$$

L'assioma di rimpiazzamento è stato il principale contributo di Fraenkel alla presente teoria.

**Regolarità** Questo assioma afferma che ogni insieme non vuoto è disgiunto da almeno uno dei suoi elementi. Il fatto che due insiemi $x$ e $y$ sono disgiunti può essere espresso dicendo che se un elemento appartiene ad $x$ allora non appartiene ad $y$, ovvero dalla formula $\forall z (z \in x \rightarrow \neg (z \in y))$. Abbiamo dunque

$$\forall x (x \neq \emptyset \rightarrow \exists y (y \in x \wedge \forall z (z \in x \rightarrow \neg (z \in y))))$$

Esistono svariati altri principi della teoria degli insiemi che sono consistenti con gli assiomi precedenti ma non ne sono conseguenza logica (sono dunque *indipendenti* da essi). Tra questi, ricordiamo l'*assioma della scelta* e l'*ipotesi del continuo*, che sono abitualmente trattati separatamente per via della loro minore evidenza e della natura poco costruttiva. il lettore interessato ad approfondire l'argomento può consultare [VDD78, Lev79, Kun80].

## Esercizi

**4.1** Descrivere le strutture dati coda (queue) ed albero in un linguaggio del primo ordine (con uguaglianza).

**4.2** Determinare le variabili libere nelle seguenti fbf:

1. $\forall y \exists x A(x, y) \rightarrow B(x, y)$
2. $\exists x \exists y (A(x, y) \rightarrow B(x, y))$
3. $\neg \forall y \exists x A(y) \rightarrow (B(x, y) \wedge \forall z C(x, z))$
4. $\exists x \exists y (A(x, y) \rightarrow B(x)) \rightarrow \forall z C(z) \vee D(z)$

**4.3** Data la fbf $P$ $\exists x \forall y P(y, f(x), g(z))$. Definire un'interpretazione che è un modello per $P$ ed un'interpretazione che non lo è.

**4.4** Consideriamo le seguenti fbf:

- $\forall x P(x, x)$
- $\forall x \forall y (P(x, y) \rightarrow P(y, x))$
- $\forall x \forall y \forall z (P(x, y) \wedge P(y, z) \rightarrow P(x, z))$

Mostrare che nessuna di tali formule è conseguenza (semantica) delle altre due. (Suggerimento: trovare delle interpretazioni che sono dei modelli per due delle formule date ma non per la terza).

**4.5** Sia $(\mathcal{A}, \xi^{\mathcal{A}})$ un'interpretazione tale che:

- $D_{\mathcal{A}} = \mathcal{N}$
- $A^{\mathcal{A}} = \{(n, m) \mid n, m \in D_{\mathcal{A}} \text{ e } n \geq m\}$
- $B^{\mathcal{A}} = \{(n, m, p) \mid n, m, p \in D_{\mathcal{A}} \text{ e } n + m = p\}$

dire quali delle seguenti formule sono vere in $(\mathcal{A}, \xi^{\mathcal{A}})$:

1. $\forall x \forall y \forall z (B(x, y, z) \rightarrow B(y, x, z))$
2. $\forall x \exists y (B(x, x, y) \rightarrow B(y, x, y))$
3. $\exists x \exists y (A(x, y) \vee \neg B(y, x, y))$
4. $\forall x \exists y A(x, y) \rightarrow \forall x B(x, x, c)$
5. $\exists x \forall y B(x, y, x)$
6. $\forall x \forall y A(x, y)$
7. $\exists x \exists y A(x, y)$

8. $\forall x \exists y \neg A(x,y) \rightarrow \exists y \neg A(y,c)$

**4.6** Provare che per ogni fbf $P$

$$v^{(\mathcal{A},\xi^{\mathcal{A}})}(P[^{t}/_{x}]) = v^{(\mathcal{A},\xi^{\mathcal{A}}[^{[\![t]\!]^{\mathcal{A}}_{\xi}}/_{x}])}(P)$$

dove $t$ è un termine (*Lemma di traslazione*).
(Suggerimento: si utilizzi l'induzione sulla struttura di $P$).

**4.7** Stabilire quali delle formule seguenti sono valide e quali semplicemente soddisfacibili. Nel secondo caso, fornire un esempio di interpretazione che non ne è un modello.

1. $(\exists x A(x) \rightarrow \forall x B(x)) \rightarrow \forall x (A(x) \rightarrow B(x))$
2. $(\exists x A(x) \rightarrow \exists x B(x)) \rightarrow \forall x (A(x) \rightarrow B(x))$
3. $(\exists x A(x) \rightarrow \exists x B(x)) \rightarrow \exists x (A(x) \rightarrow B(x))$
4. $\exists x (A(x) \rightarrow B(x)) \rightarrow (\forall x A(x) \rightarrow \forall x B(x))$
5. $\exists x (A(x) \rightarrow B(x)) \rightarrow (\forall x A(x) \rightarrow \exists x B(x))$
6. $\exists x (A(x) \rightarrow B(x)) \rightarrow (\exists x A(x) \rightarrow \exists x B(x))$
7. $\forall x (A(x) \rightarrow B(x)) \rightarrow (\exists x A(x) \rightarrow \forall x B(x))$
8. $\forall x (A(x) \rightarrow B(x)) \rightarrow (\exists x A(x) \rightarrow \exists x B(x))$
9. $\forall x (A(x) \rightarrow B(x)) \rightarrow (\forall x A(x) \rightarrow \forall x B(x))$
10. $(\forall x A(x) \rightarrow \forall x B(x)) \rightarrow \forall x (A(x) \rightarrow B(x))$
11. $(\forall x A(x) \rightarrow \exists x B(x)) \rightarrow \exists x (A(x) \rightarrow B(x))$
12. $(\forall x A(x) \rightarrow \exists x B(x)) \rightarrow \forall x (A(x) \rightarrow B(x))$

**4.8** Si considerino le seguenti formule:

1. $\forall x \exists y A(x,y) \rightarrow \exists x \forall y A(x,y)$
2. $\exists x \forall y A(x,y) \rightarrow \forall x \exists y A(x,y)$
3. $\exists x \forall y A(x,y) \rightarrow \exists x \forall y A(x,y)$
4. $\forall x \exists y A(x,y) \rightarrow \forall x \exists y A(x,y)$

quali di queste sono valide?

**4.9** Provare che $\models \neg \exists y \forall x (\neg R(x,x) \leftrightarrow R(y,x))$.

**4.10** Provare che (*Teorema di deduzione semantica*):

1. $P \models Q$ se e solo se $\models P \rightarrow Q$
2. $P_1, \ldots, P_{n-1}, P_n \models Q$ se e solo se $P_1, \ldots, P_{n-1} \models P_n \rightarrow Q$

**4.11** Dimostrare che:

1. Se $\models A(y)$ allora $\models \forall x A(x)$, ma $A(y) \not\models \forall x A(x)$.
2. $P \wedge Q \models R$ se e solo se $P \models R$ e $Q \models R$.
3. $\{\forall x(A(x) \rightarrow B(x)), \forall x A(x)\} \models \forall x B(x)$

**4.12** Trasformare le seguenti formule ben formate in forma di Skolem:

1. $\forall y(\exists x A(x,y) \rightarrow B(y,x)) \wedge \ \exists y\ (\forall x C(x,y) \vee B(x,y))$
2. $\exists x \forall y \exists z D(x,y,z) \vee \exists x \forall y A(x,y) \wedge \neg\ \exists x \exists y B(x,y)$
3. $\neg\ (\forall x \exists y A(x,y) \rightarrow \exists x \exists y B(x,y)) \wedge \forall x \neg\ \exists y B(y,x)$

**4.13** Definire una procedura per trasformare una fbf in una forma di Skolem nella quale siano invertiti i ruoli di $\forall$ e $\exists$ (*forma di Skolem duale*). Provare che una fbf $P$ è valida se e solo se la corrispondente formula che non contiene quantificatori unversali lo è.

**4.14** Si scriva una tautologia $P$ tale che la skolemizzata di $P$ ($P^S$) non sia una tautologia.

**4.15** Provare che la formula ben formata

$$\forall x P(x, f(x)) \wedge \forall y \neg P(y,y) \wedge \forall u v w((P(u,v) \wedge P(v,w)) \rightarrow P(u,w))$$

è soddisfacibile ma non possiede un modello con dominio finito.

**4.16** Una fbf si dice *monadica* se contiene solo simboli di predicato unari (monadici); provare che ogni formula monadica è equivalente alla composizione, mediante l'utilizzo dei connettivi $\wedge$ e $\vee$, di formule della forma $\forall x P(x)$ e $\exists x P(x)$ dove $P$ è una formula che non contiene quantificatori.
(Suggerimento: si utilizzino le forme normali congiuntive e disgiuntive ed il Teorema 4.34).

**4.17** Sia $^\circ$ la trasformazione da formule della logica dei predicati a formule della logica proposizionale, definita nel seguente modo:

- Se $P$ è $A^n(t_1, \cdots, t_n)$, allora $P^\circ$ è la lettera proposizionale $A^n$.
- Se $P$ è $\neg P_1$, allora $P^\circ$ è $\neg P_1^\circ$.
- Se $P$ è $P_1 \rightarrow P_2$ o $P_1 \vee P_2$ o $P_1 \wedge P_2$, allora $P^\circ$ è, rispettivamente, $P_1^\circ \rightarrow P_2^\circ$ o $P_1^\circ \vee P_2^\circ$ o $P_1^\circ \wedge P_2^\circ$.
- Se $P$ è $\forall x P_1$ o $\exists x P_1$, allora $P^\circ$ è $P_1^\circ$.

Dare un esempio di formula $P$ non valida ma tale che $P^\circ$ sia una tautologia.

**4.18** Definire la sintassi e la semantica della logica del primo ordine con uguaglianza.

**4.19** Data la formula della logica dei predicati con uguaglianza:

$$\forall pq((A(p,a) \wedge p \neq a) \rightarrow \exists xy(f(x,y) = f(q,q) \wedge g(x,y) = p))$$

Stabilire se è soddisfatta nelle seguenti interpretazioni:

1. $(\mathcal{A}, \xi^{\mathcal{A}})$, dove $D_{\mathcal{A}} = \{1,3,5,7,9,\dots\}$, $a^{\mathcal{A}} = 19$, $f^{\mathcal{A}} = (x,y) \mapsto x^y$, $g^{\mathcal{A}} = (x,y) \mapsto x \cdot y$ e $A^{\mathcal{A}} = \{(n,m) \mid n,m \in D_{\mathcal{A}} \text{ e } n \leq m\}$.

2. $(\mathcal{B}, \xi^{\mathcal{B}})$, dove $D_{\mathcal{B}} = \mathcal{N}$, $a^{\mathcal{B}} = 1$, $f^{\mathcal{B}} = (x,y) \mapsto x+y$, $g^{\mathcal{B}} = (x,y) \mapsto x \cdot y$ e $A^{\mathcal{B}} = \{(n,m) \mid n,m \in D_{\mathcal{B}} \text{ e } n \leq m\}$.

3. $(\mathcal{C}, \xi^{\mathcal{C}})$, dove $D_{\mathcal{C}} = \mathcal{R}$[7], $a^{\mathcal{C}} = 0$, $f^{\mathcal{C}} = (x,y) \mapsto x+y$, $g^{\mathcal{C}} = (x,y) \mapsto x \cdot y$ e $A^{\mathcal{C}} = \{(n,m) \mid n,m \in D_{\mathcal{C}} \text{ e } n \geq m\}$.

[7] Insieme dei numeri reali.

# Capitolo 5

# Il Calcolo del Primo ordine

In analogia con quanto è stato fatto per il calcolo proposizionale, cominciamo con il discutere le proprietà logiche intuitive dei quantificatori. Premettiamo, tuttavia, un'importante convenzione sintattica che consentirà di evitare alcuni noiosi problemi di collisione tra variabili libere e legate. In particolare, considerando calcoli deduttivi al primo ordine, assumiamo di identificare *sintatticamente* formule che differiscono soltanto per la ridenominazione di variabili legate. Questo significa che le due formule $\forall x A$ e $\forall y A[{}^{y}/_{x}]$ sono considerate sintatticamente indistinguibili, ovvero in ogni momento si può operare il rimpiazzamento dell'una con l'altra.
Iniziamo quindi a considerare le regole di eliminazione ed introduzione del quantificatore universale. Sapendo che per ogni $x$ vale un certo predicato $A$ si può concludere che $A$ deve valere per ogni possibile istanziazione di $x$ con un termine arbitrario del linguaggio in esame. Questo conduce direttamente alla seguente regola di *eliminazione* del quantificatore:

$$\forall x A \vdash A[{}^{t}/_{x}]$$

Viceversa, quando si può concludere $\forall x A$? Nel caso in cui è stato dimostrato che $A$ vale per un "generico" $x$. Ora, il ruolo di "elemento generico" in un linguaggio del primo ordine è svolto, appunto, dalle variabili. Dunque ci si aspetta di avere una regola del tipo:

$$(*) \quad se\ \Gamma \vdash A[{}^{y}/_{x}]\ \ allora\ \ \Gamma \vdash \forall x A$$

L'unico problema è che la variabile $y$[1], che dovrebbe denotare un "elemento generico" potrebbe al contario apparire libera anche in qualche formula di $\Gamma$. Questo farebbe sì che essa perderebbe la sua "genericità", dovendo soddisfare le proprietà richieste da queste ipotesi. Ad esempio, sapendo che un generico $y$ è numero pari, ci si aspetta di poter concludere che il suo successore è dispari, che si può riassumere nell'asserto

$$P(y) \vdash D(s(y))$$

[1] $y$ è detta *eigenvariable* della formula.

Tuttavia, non è ovviamente vero che

$$P(y) \vdash \forall x D(s(x))$$

La condizione che è quindi necessario aggiungere alla regola $(*)$ è che *la variabile $y$ non deve comparire libera in* $\Gamma$.
Il caso del quantificatore esistenziale è duale. Sapendo che per un qualche termine $t$ vale $A[^t/_x]$ è possibile ovviamente concludere $\exists x A$, come espresso dalla seguente regola di introduzione:

$$A[^t/_x] \vdash \exists x A$$

Nel caso del quantificatore esistenziale, i problemi riguardano la regola di eliminazione: cosa si può concludere sapendo $\exists x A$? Veramente poco, dato che non è noto *per quale* $x$ sia vera $A$. Tuttavia, supponendo che *comunque si prenda* un generico elemento $y$, l'ipotesi $A[^y/_x]$ è sufficiente per dimostrare una proprietà $C$, allora la mera esistenza di un tale $y$ è sufficiente per concludere[2] $C$. Tale ragionamento è espresso dalla seguente regola:

$$\textit{se } \Gamma, A[^y/_x] \vdash C \;\; \textit{allora } \Gamma, \exists x A \vdash C$$

Anche in questo caso, la "genericità" dell'elemento $y$ deve essere garantita mediante l'ipotesi addizionale che $y$ non compaia libera in $\Gamma$ o $C$. Consideriamo infatti ancora l'asserto "se un generico $y$ è pari, allora il suo successore è dispari", rappresentato come

$$P(y) \vdash D(s(y))$$

Ovviamente sarebbe scorretto concludere

$$\exists x P(x) \vdash D(s(y))$$

(se esiste un numero pari, allora il successore di un generico $y$ è dispari).

Come nel caso del calcolo proposizionale, i *sistemi intuizionisti* differiscono da quelli classici solo per l'assenza della regola di riduzione ad assurdo (RAA). A livello predicativo, la differenza principale di questi sistemi rispetto a quelli classici è che nei primi, se si ha una dimostrazione di $\exists x P(x)$ allora esiste necessariamente un termine $t$ (detto *testimone*), per cui è derivabile $P[^t/_x]$; ovviamente tale caratteristica non è presente nei sistemi classici dove si può provare $\exists x P(x)$ deducendolo per contraddizione. In questo senso, le dimostrazioni nei sistemi intuizionisti, al contrario di quelli classici, sono *costruttive*: se si dimostra che esiste un termine che gode di una certa proprietà si è anche in grado di esibirlo in modo effettivo.
Per apprezzare pienamente le motivazioni filosofiche e matematiche dei sistemi intuizionisti vediamo un semplice esempio. Consideriamo la sentenza

> "esiste un uomo tale che, se lui possiede un cappello, allora tutti gli uomini possiedono un cappello".

[2] Si noti l'analogia con il caso della disgiunzione nel calcolo proposizionale.

che può essere formalizzata dalla formula

$$\exists x(C(x) \to \forall y C(y))^3$$

La frase sembra chiaramente falsa: non si vede ragione per cui un tale uomo debba esistere. Tuttavia, dal punto di vista della logica classica, la sentenza precedente è facilmente dimostrabile. Infatti, esistono due casi possibili: o ogni uomo ha il cappello, e quindi l'implicazione è vera qualunque sia $x$ in quanto è vero il conseguente, oppure esiste un uomo senza cappello. Per questo uomo la premessa dell'implicazione è falsa, e quindi, la sentenza è vera anche in questo caso (mostreremo in seguito la dimostrazioni formale di ciò, in alcuni calcoli logici). Tale dimostrazione non è costruttiva, in quanto non permette di esibire l'uomo $u$ per cui $C(u) \to \forall y C(y)$. Nei sistemi intuizionisti, la formula $\exists x(C(x) \to \forall y C(y))$ non è dimostrabile.

## 5.1 La Deduzione Naturale

Seguendo la stessa procedura utilizzata per i connettivi proposizionali è possibile tradurre in regole di Deduzione Naturale le proprietà intuitive dei quantificatori appena discusse.
Nel caso del quantificatore universale si ottengono rispettivamente:

$$(\forall i)\ \frac{A[^y/_x]}{\forall x A} \qquad\qquad (\forall e)\ \frac{\forall x A}{A[^t/_x]}$$

dove $x$ e $y$ sono variabili, e $t$ è un generico termine del linguaggio. Inoltre, in $(\forall i)$, deve valere l'ipotesi ausiliaria che $y$ non compaia libera in nessuna delle foglie non cancellate del sottoalbero di radice $A[^y/_x]$.
Le regole per il quantificatore esistenziale sono appena più complicate:

$$(\exists i)\ \frac{A[^t/_x]}{\exists x A} \qquad\qquad (\exists e)\ \frac{\exists x A \qquad \overset{[A[^y/_x]]}{C}}{C}$$

In $(\exists e)$, la variabile $y$ non può comparire libera in $C$, nè in nessuna delle foglie non cancellate del sottoalbero di radice $C$ (a parte, ovviamente, $A[^y/_x]$).
Vediamo, qui di seguito, qualche esempio di derivazione.

**Esempio 5.1** $\forall x(A \to B(x)) \vdash A \to \forall x B(x)$

$$\dfrac{\dfrac{\dfrac{\forall x(A \to B(x))}{A \to B(x)} \qquad [A]}{B(x)}}{\forall x B(x)}}{A \to \forall x B(x)}$$

[3] Si noti che è diversa da $\exists x C(x) \to \forall y C(y)$ (se esiste un uomo con il cappello allora tutti gli uomini hanno un cappello). In realtà, $\exists x(C(x) \to \forall y C(y))$ è logicamente equivalente, dal punto di vista classico, a $\forall x C(x) \to \forall y C(y)$, che è una ovvia tautologia.

**Esempio 5.2** $\exists x \forall y A(x,y) \vdash \forall y \exists x A(x,y)$

$$\dfrac{\dfrac{\exists x \forall y A(x,y) \qquad \dfrac{\dfrac{[\forall y A(x,y)]}{A(x,y)} \qquad \dfrac{[\forall x \neg A(x,y)]}{\neg A(x,y)}}{\bot}}{\dfrac{\bot}{\neg \forall x \neg A(x,y)}} \qquad \dfrac{\dfrac{[\neg \exists x A(x,y)] \qquad \dfrac{[A(x,y)]}{\exists x A(x,y)}}{\bot}}{\dfrac{\neg A(x,y)}{\forall x \neg A(x,y)}}}{\dfrac{\dfrac{\bot}{\exists x A(x,y)}}{\forall y \exists x A(x,y)}}$$

**Esempio 5.3** $\vdash \forall x(A(x) \to B) \to (\exists x A(x) \to B)$

$$\dfrac{\dfrac{\dfrac{\dfrac{[\forall x(A(x) \to B)]}{A(x) \to B} \qquad [A(x)]}{B} \qquad [\exists x A(x)]}{B}}{\dfrac{\exists x A(x) \to B}{\forall x(A(x) \to B) \to (\exists x A(x) \to B)}}$$

**Esempio 5.4** ("paradosso" dell'uomo col cappello).
$\vdash \exists x(C(x) \to \forall y C(y))$

$$\dfrac{\dfrac{[\neg \exists x(C(x) \to \forall y C(y))] \qquad \dfrac{\dfrac{\dfrac{\dfrac{[\neg C(x)] \qquad [C(x)]}{\bot}}{\forall y C(y)}}{C(x) \to \forall y C(y)}}{\exists x(C(x) \to \forall y C(y))}}{\dfrac{\dfrac{\dfrac{\bot}{C(x)}}{\forall y C(y)}}{C(x) \to \forall y C(y)}} \qquad [\neg \exists x(C(x) \to \forall y C(y))]}{\dfrac{\bot}{\exists x(C(x) \to \forall y C(y))}}$$

**Esempio 5.5** Dimostriamo che nell'aritmetica di Peano $\forall x \neg(x = s(x))$; vale a dire che $PA \vdash \forall x\ \neg(x = s(x))$ dove PA sono gli assiomi di p.116. Sia $P(x) = \neg(x = s(x))$. L'idea è quella di provare l'asserto per induzione.

(*caso base*) Si vuole dimostrare che $P(0) = \neg(0 = s(0))$. Ma questa è una semplice conseguenza dell'assioma $\forall x \neg(0 = s(x))$; infatti:

$$\frac{\forall x \neg(0 = s(x))}{\neg(0 = s(0))}$$

Veniamo al caso induttivo. Dobbiamo provare $\forall x(P(x) \to P(s(x)))$, cioè $\forall x(\neg(x = s(x)) \to \neg(s(x) = s(s(x))))$. A questo scopo si utilizza l'assioma $\forall xy(s(x) = s(y) \to x = y)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\forall xy(s(x) = s(y) \to x = y)}{\forall y(s(x) = s(y) \to x = y)}}{s(x) = s(s(x)) \to x = s(x)} \quad [s(x) = s(s(x))]}{x = s(x)} \quad [\neg(x = s(x))]}{\bot}}{\neg(s(x) = s(s(x)))}}{\neg(x = s(x)) \to \neg(s(x) = s(s(x)))}}{\forall x(\neg(x = s(x)) \to \neg(s(x) = s(s(x))))}$$

L'asserto si ottiene per modus ponens utilizzando l'assioma di induzione

$$(P(0) \wedge \forall x(P(x) \to P(s(x)))) \to \forall x P(x)$$

***Osservazione*** Il sistema di Deduzione Naturale per la logica dei predicati con uguaglianza si ottiene aggiungendo alle regole mostrate in precedenza, le seguenti, che corrispondono agli assiomi di p.114:

1. $\dfrac{}{x = x}$
2. $\dfrac{x = y}{y = x}$
3. $\dfrac{x = y \qquad y = z}{x = z}$
4. $\dfrac{x_1 = y_1, \ldots, x_n = y_n}{f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)}$
5. $\dfrac{x_1 = y_1, \ldots, x_n = y_n}{P(x_1, \ldots, x_n) \to P(y_1, \ldots, y_n)}$

dove $y_1, \ldots y_n$ sono libere per $x_1, \ldots, x_n$ in $P$.

### 5.1.1 Correttezza e Completezza $\star$

Dopo aver presentato le regole della Deduzione Naturale per la logica del primo ordine, affrontiamo il problema della correttezza e della completezza di tale calcolo (e di quelli ad esso equivalenti). Come nel caso proposizionale, si tratta di stabilire se ogni formula derivabile utilizzando le regole della Deduzione Naturale è anche valida (correttezza) e, viceversa, che ogni formula valida è derivabile nel calcolo in esame (completezza).

**Teorema 5.1 (Correttezza)**
*Se* $\Gamma \vdash P$ *allora* $\Gamma \models P$.

*Dimostrazione.* Per induzione sulla profondità dell'albero di prova per $\Gamma \vdash P$.
(*caso base*) L'albero è costituito dalla sola radice $P$, allora, per definizione, $P \in \Gamma$ e quindi $\Gamma \models P$.
Veniamo al caso induttivo. È necessario distinguere tanti sottocasi quante sono le possibili regole di inferenza che concludono l'albero di prova di $\Gamma \vdash P$. Discuteremo solo quelli relativi ai quantificatori, per i rimanenti si veda la dimostrazione del Teorema 3.1.

1. $P$ è $\forall x P_1$ e l'ultima regola di deduzione applicata è $(\forall i)$. Allora esiste un albero di prova per $\Gamma \vdash P_1[^y/_x]$ dove $y$ non appare libera in $\Gamma$; per ipotesi induttiva $\Gamma \models P_1[^y/_x]$, vale a dire che per ogni interpretazione $(\mathcal{A}, \xi^{\mathcal{A}})$ tale che $(\mathcal{A}, \xi^{\mathcal{A}}) \models \Gamma$ risulta $(\mathcal{A}, \xi^{\mathcal{A}}) \models P_1[^y/_x]$. Poichè $y$ non è libera in $\Gamma$, $(\mathcal{A}, \xi^{\mathcal{A}}) \models \Gamma$ se e solo se, comunque si scelga $b$ nel dominio $D_{\mathcal{A}}$ della struttura, $(\mathcal{A}, \xi^{\mathcal{A}}[^b/_y]) \models \Gamma$. Dunque, $\forall b \in D_{\mathcal{A}}$, $(\mathcal{A}, \xi^{\mathcal{A}}) \models \Gamma$ implica $(\mathcal{A}, \xi^{\mathcal{A}}[^b/_y]) \models P_1[^y/_x]$ e, portando all'interno il quantificatore, $(\mathcal{A}, \xi^{\mathcal{A}}) \models \Gamma$ implica che $\forall b \in D_{\mathcal{A}}$, $(\mathcal{A}, \xi^{\mathcal{A}}[^b/_y]) \models P_1[^y/_x]$. Ma $v^{(\mathcal{A}, \xi^{\mathcal{A}}[^b/_y])}(P_1[^y/_x]) = v^{(\mathcal{A}, \xi^{\mathcal{A}}[^b/_x])}(P_1)$ (il lettore lo dimostri per esercizio). In conclusione, $(\mathcal{A}, \xi^{\mathcal{A}}) \models \Gamma$ implica che $\forall b \in D_{\mathcal{A}} \quad (\mathcal{A}, \xi^{\mathcal{A}}[^b/_x]) \models P_1$, ovvero, per il Lemma 4.28, che $(\mathcal{A}, \xi^{\mathcal{A}}) \models \Gamma$ implica $(\mathcal{A}, \xi^{\mathcal{A}}) \models \forall x P_1$. Dunque $\Gamma \models \forall x P_1$.

2. $P$ è $P_1[^t/_x]$ e l'ultima regola di deduzione applicata è $(\forall e)$. Allora esiste un albero di derivazione relativo a $\Gamma \vdash \forall x P_1$; per ipotesi induttiva $\Gamma \models \forall x P_1$. Vogliamo dimostrare che, per ogni termine $t$, $\Gamma \models P_1[^t/_x]$. Per definizione, $\Gamma \models \forall x P_1$ se e solo se per ogni interpretazione $(\mathcal{A}, \xi^{\mathcal{A}})$ tale che $(\mathcal{A}, \xi^{\mathcal{A}}) \models \Gamma$ risulta $(\mathcal{A}, \xi^{\mathcal{A}}) \models \forall x P_1$, ovvero, per il Lemma 4.28, che comunque si scelga $b \in D_{\mathcal{A}}$, $(\mathcal{A}, \xi^{\mathcal{A}}[^b/_x]) \models P_1$. In particolare, $(\mathcal{A}, \xi^{\mathcal{A}}[^{[\![t]\!]^{\mathcal{A}}_{\xi}}/_x]) \models P_1$, ma per l'Esercizio 4.6, $v^{(\mathcal{A}, \xi^{\mathcal{A}}[^{[\![t]\!]^{\mathcal{A}}_{\xi}}/_x])}(P_1) = v^{(\mathcal{A}, \xi^{\mathcal{A}})}(P_1[^t/_x])$, e dunque $(\mathcal{A}, \xi^{\mathcal{A}}) \models P_1[^t/_x]$.

Le regole per il quantificatore esistenziale sono lasciate al lettore come esercizio. □

**Corollario 5.2** *Se* $\vdash P$, *allora* $\models P$.

Vediamo quindi l'inverso del precedente teorema, e cioè se $\Gamma \models P$ allora $\Gamma \vdash P$ (completezza). Tale dimostrazione, che ad una prima lettura può risultare alquanto complicata, ha numerose applicazioni, alcune delle quali verranno discusse nel paragrafo 5.4. Come nel caso proposizionale (paragrafo 3.1) per dimostrare che se $\Gamma \models P$ allora $\Gamma \vdash P$, proveremo che ogni insieme consistente di fbf è soddisfacibile (Teorema del modello).
In particolare, gli enunciati e le dimostrazioni concernenti gli insiemi consistenti massimali sviluppati per la logica proposizionale, si estendono senza alcuna modifica alla logica dei predicati (la verifica di ciò è lasciata al lettore come esercizio).
Utilizzando i summenzionati risultati, presentiamo ora una dimostrazione di completezza, che costituisce la naturale generalizzazione al primo ordine di quella del paragrafo 3.1.
Si procederà nel seguente modo:

- proveremo che per ogni insieme consistente massimale (Definizione 3.7) e di Henkin (Definizione 5.3) esiste un modello (Corollario 5.7)

- mostreremo che ogni insieme consistente può essere esteso in uno avente le summenzionate caratteristiche (Proposizione 5.10 e Teorema di Lindembaum).

**Definizione 5.3** *Sia $\mathcal{L}$ un linguaggio del primo ordine. Un insieme di fbf $\Gamma$ definito su $\mathcal{L}$ si dice* insieme di Henkin *se per ogni formula di $\mathcal{L}$ della forma $\exists x P(x)$ esiste un termine $t$, detto* testimone*, tale che $(\exists x P(x) \rightarrow P[^t/_x]) \in \Gamma$.*

Sottolineamo il fatto che un insieme di Henkin $\Gamma$ deve avere un testimone per ogni formula $\exists x P(x)$ *del linguaggio* sul quale è definito, e non solo per le formule esistenziali che appartengono a $\Gamma$.
La proprietà principale degli insiemi di Henkin è la seguente:

**Proposizione 5.4** *Sia $\Gamma$ un insieme di Henkin. Allora $\Gamma \vdash \exists x P$ se e solo se esiste un termine $t$ tale che $\Gamma \vdash P[^t/_x]$.*

*Dimostrazione.* ($\Leftarrow$) Segue da ($\exists i$).
($\Rightarrow$) Per definizione di insieme di Henkin, deve esistere un termine $t$ tale che $\exists x P \rightarrow P[^t/_x] \in \Gamma$. Dunque, $\Gamma \vdash P[^t/_x]$ per ($\rightarrow e$). $\square$

Ricordiamo che un insieme consistente massimale gode della seguente proprietà: se $\Gamma \vdash P$, allora $P \in \Gamma$ (chiusura per derivabilità). Sia $\Gamma$ un insieme consistente massimale e di Henkin. Vogliamo trovare un'interpretazione che è un modello per $\Gamma$; l'idea è quella di utilizzare la sua stessa sintassi.

**Definizione 5.5** *Sia $\mathcal{L}$ il linguaggio sul quale è definito $\Gamma$ e TER($\mathcal{L}$) l'insieme dei suoi termini. L'* interpretazione canonica *(o interpretazione dei termini) $(\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma})$ è definita nel seguente modo:*

- $D_{\mathcal{A}_\Gamma} = \{t \mid t \in TER(\mathcal{L})\}$

- *Per ogni simbolo di costante* $c$

$$c^{\mathcal{A}_\Gamma} = c$$

- *Per ogni simbolo di funzione* $f$ *di arità* $k$

$$f^{\mathcal{A}_\Gamma}(t_1, \ldots, t_k) = f(t_1, \ldots, t_k)$$

- *Per ogni simbolo di predicato* $A$ *ad* $n$ *argomenti*

$$A^{\mathcal{A}_\Gamma} = \{(t_1, \ldots, t_n) \in TER^n(\mathcal{L}) \mid A(t_1, \ldots, t_n) \in \Gamma\}$$

- *Per ogni variabile* $x$

$$\xi^{\mathcal{A}_\Gamma}(x) = x$$

Proveremo che se $\Gamma$ è un insieme consistente massimale e di Henkin allora $(\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma})$ è un modello per esso.
Intuitivamente, si considerano insiemi consistenti massimali e di Henkin in quanto, per questi, ciò si effettua in modo del tutto naturale; infatti, supponiamo, per fissare le idee, che $B$ sia un predicato unario, se $B(x) \in \Gamma$ e $B(x) \to B(y) \in \Gamma$, affinché $(\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma})$ sia un modello per $\Gamma$, $B(y)$ deve appartenere a $\Gamma$ (dunque $\Gamma$ deve essere chiuso per derivabilità); inoltre, se $\exists x B(x) \in \Gamma$, perché $(\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma})$ sia un modello per $\Gamma$, deve esserci un "testimone", cioè termine $t$ tale che $B(t) \in \Gamma$ (quindi $\Gamma$ deve essere un insieme di Henkin).

**Proposizione 5.6** *Sia* $\Gamma$ *un insieme consistente massimale e di Henkin ed* $(\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma})$ *l'interpretazione canonica.* $(\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma}) \models P$ *se e solo se* $\Gamma \vdash P$ *per ogni fbf* $P$*.*

*Dimostrazione.* Per induzione sulla struttura di $P$.
(*caso base*) Se $P$ è una formula atomica $P_1(t_1, \ldots, t_n)$, allora $(\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma}) \models P_1(t_1, \ldots, t_n)$ se e solo se, per come è definita l'interpretazione canonica, $P(t_1, \ldots, t_n) \in \Gamma$, se e solo se $\Gamma \vdash P_1(t_1, \ldots, t_n)$ essendo $\Gamma$ un insieme consistente massimale.
Se $P = \bot$ l'asserto è banalmente verificato.
Veniamo al caso induttivo.

- $P$ è $\neg P_1$. Allora

| | | | |
|---|---|---|---|
| $(\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma}) \models \neg P_1$ | $\Leftrightarrow$ | $(\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma}) \not\models P_1$ | Lemma 4.28 |
| | $\Leftrightarrow$ | $\Gamma \not\vdash P_1$ | ipotesi induttiva |
| | $\Leftrightarrow$ | $P_1 \notin \Gamma$ | essendo $\Gamma$ cons. mass. |
| | $\Leftrightarrow$ | $\neg P_1 \in \Gamma$ | Lemma 3.9.1 |
| | $\Leftrightarrow$ | $\Gamma \vdash \neg P_1$ | essendo $\Gamma$ cons. mass. |

- $P$ è $P_1 \to P_2$. Allora

$$
\begin{array}{lll}
(\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma}) \models P_1 \to P_2 & \Leftrightarrow & ((\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma}) \models P_1 \Rightarrow \\
 & & \quad (\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma}) \models P_2) \quad \text{Lemma 4.28} \\
 & \Leftrightarrow & \Gamma \vdash P_1 \Rightarrow \Gamma \vdash P_2 \quad \text{ipotesi induttiva} \\
 & \Leftrightarrow & P_1 \in \Gamma \Rightarrow P_2 \in \Gamma \quad \text{essendo } \Gamma \text{ cons. mass.} \\
 & \Leftrightarrow & \neg P_1 \in \Gamma \text{ o } P_2 \in \Gamma \\
 & \Leftrightarrow & P_1 \to P_2 \in \Gamma \quad \text{Lemma 3.9.4} \\
 & \Leftrightarrow & \Gamma \vdash P_1 \to P_2 \quad \text{essendo } \Gamma \text{ cons. mass.}
\end{array}
$$

- $P$ è $\exists x P_1(x)$. Allora

$$
\begin{array}{lll}
(\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma}) \models \exists x P_1(x) & \Leftrightarrow & \text{esiste un termine t tale} \\
 & & \text{che } (\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma}[^t/_x]) \models P_1 \quad \text{Lemma 4.28} \\
 & \Leftrightarrow & (\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma}[^{[\![t]\!]_\xi^{\mathcal{A}_\Gamma}}/_x]) \models P_1 \quad \text{Esercizio 5.6} \\
 & \Leftrightarrow & (\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma}) \models P_1[^t/_x] \quad \text{Esercizio 4.6} \\
 & \Leftrightarrow & \text{esiste un termine t tale} \\
 & & \text{che } \Gamma \vdash P_1[^t/_x] \quad \text{ipotesi induttiva} \\
 & \Leftrightarrow & \Gamma \vdash \exists x P_1 \quad \text{Proposizione 5.4}
\end{array}
$$

I rimanenti casi sono lasciati al lettore come esercizio. □

**Corollario 5.7** *Sia* $\Gamma$ *un insieme consistente massimale e di Henkin, allora* $(\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma}) \models \Gamma$ *e dunque* $\Gamma$ *è soddisfacibile.*

***Osservazione*** Se si considera un linguaggio $\mathcal{L}'$ del primo ordine con uguaglianza (cfr. p.114) è necessario ridefinire l'interpretazione dei termini. Infatti, supponiamo ad esempio che $\Gamma$ sia $\{f(x) = f(y)\}$; $\Gamma$ è ovviamente soddisfacibile, tuttavia l'interpretazione definita in 5.5 non ne è un modello in quanto $f^{\mathcal{A}_\Gamma}(x) = f(x) \neq f(y) = f^{\mathcal{A}_\Gamma}(y)$. L'idea è di introdurre una relazione di equivalenza sui termini di $\mathcal{L}'$ e definire il dominio $D_{\mathcal{A}_\Gamma}$ come l'insieme delle classi di equivalenza di tale relazione. L'interpretazione canonica $(\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma})$ per un linguaggio del primo ordine con uguaglianza si definisce allora nel seguente modo: Sia $\sim$ la relazione binaria definita sui termini di $\mathcal{L}'$ in modo tale che

$$t_1 \sim t_2 \text{ sse } t_1 = t_2 \in \Gamma$$

$\sim$ è una relazione di equivalenza (la verifica di ciò è lasciata al lettore come esercizio). Sia $\bar{t}$ la classe di equivalenza di $t$, vale a dire

$$\bar{t} = \{t' \in TER(\mathcal{L}') \mid t' \sim t\}$$

La nuova interpretazione è data da:

- $D_{\mathcal{A}_\Gamma} = \{\bar{t} \mid t \in TER(\mathcal{L})'\}$
- Per ogni simbolo di costante $c$

$$c^{\mathcal{A}_\Gamma} = \bar{c}$$

- Per ogni simbolo di funzione $f$ di arità $n$

$$f^{\mathcal{A}_\Gamma}(\overline{t_1}, \ldots, \overline{t_n}) = \overline{f(t_1, \ldots, t_n)}$$

- Per ogni simbolo di predicato $P$ ad $n$ argomenti

$$P^{\mathcal{A}_\Gamma} = \{(\overline{t_1}, \ldots, \overline{t_n}) \mid P(t_1, \ldots, t_n) \in \Gamma\}$$

- Per ogni variabile $x$

$$\xi^{\mathcal{A}_\Gamma}(x) = \overline{x}$$

Essendo $\sim$ una congruenza rispetto alle funzioni ed ai predicati (la dimostrazione di ciò è lasciata al lettore come esercizio), l'interpretazione è ben definita, cioè le definizioni di $f^{\mathcal{A}_\Gamma}$ e $P^{\mathcal{A}_\Gamma}$ sono indipendenti dalla scelta dei termini $t_1, \ldots, t_n$ come rappresentanti delle classi.
La Proposizione 5.6, per un linguaggio del primo ordine con uguaglianza, si dimostra in modo analogo.

Dopo aver provato che gli insiemi consistenti massimali e di Henkin sono soddisfacibili, mostriamo che ogni insieme consistente può essere esteso in uno consistente massimale e di Henkin.

**Definizione 5.8** *Sia $\Gamma$ un insieme di fbf definito su un linguaggio $\mathcal{L}$ del primo ordine; per ogni fbf $P$ della forma $\exists x P_1(x)$ si aggiunge ad $\mathcal{L}$ una costante $c_P$, detta* testimone *tale che se $P \neq P'$ allora $c_P \neq c_{P'}$. Sia $\mathcal{L}^*$ il linguaggio risultante. $\Gamma^*$ è l'insieme di fbf dato da $\Gamma \cup \{\exists x P_1(x) \to P_1(c_P) \mid \exists x P_1(x)$ è una formula chiusa con testimone $c_P\}$.*

**Lemma 5.9** *$\Gamma$ è un insieme consistente se e solo se $\Gamma^*$ lo è.*

*Dimostrazione.* ($\Rightarrow$) Sia $\exists x P(x) \to P(c)$ una delle nuove formule introdotte come indicato dalla Definizione 5.8; supponiamo che $\Delta, \exists x P(x) \to P(c) \vdash Q$, dove $Q$ è una fbf che non contiene la costante $c$ e $\Delta$ un insieme di fbf nessuna delle quali contenente $c$. Allora $\Delta \vdash Q$. Infatti

1. $\Delta \vdash (\exists x P(x) \to P(c)) \to Q$ (Teorema di deduzione)

2. $\Delta \vdash (\exists x P(x) \to P(y)) \to Q$ dove $y$ è una variabile che non occorre nella derivazione associata (Esercizio 5.9)

3. $\Delta \vdash \forall y((\exists x P(x) \to P(y)) \to Q)$ (da $(\forall i)$[4])

4. $\Delta \vdash \exists y(\exists x P(x) \to P(y)) \to Q$ (Esempio 5.3)

5. $\Delta \vdash (\exists x P(x) \to \exists y P(y)) \to Q$ (Esercizio 5.1)

6. $\vdash \exists x P(x) \to \exists y P(y)$ (Esercizio 5.1)

[4]Osserviamo che tale regola è applicabile in quanto $y$ non occorre in $\Delta$.

7. $\Delta \vdash Q$ (da 5. e 6.)

Supponiamo che $\Gamma^* \vdash \bot$; allora per definizione di derivabilità $\Gamma \cup \{\sigma_1, \ldots, \sigma_n\} \vdash \bot$ dove $\sigma_i$ con $i = 1, \ldots, n$ sono nuove formule del tipo $\exists x P(x) \to P(c)$. Proviamo, per induzione su $n$, che $\Gamma \vdash \bot$.
(*caso base*) $n = 0$ l'asserto è verificato.
Veniamo al caso induttivo. Supponiamo che $\Gamma \cup \{\sigma_1, \ldots, \sigma_{n+1}\} \vdash \bot$, allora, posto $\overline{\Gamma} = \Gamma \cup \{\sigma_1, \ldots, \sigma_n\}$ risulta $\overline{\Gamma}, \sigma_{n+1} \vdash \bot$, per la dimostrazione precedente $\Gamma \cup \{\sigma_1, \ldots, \sigma_n\} \vdash \bot$. L'asserto segue dall'ipotesi induttiva.
($\Leftarrow$) Immediata. □

Osserviamo che se $\Delta$ è un insieme di fbf, $\Delta^*$ *non* è detto che sia un insieme di Henkin in quanto arricchendo il linguaggio con nuovi simboli di costante vengono aggiunte nuove formule del tipo $\exists x Q(x)$ che possono non avere un testimone. Per ottenere un insieme di Henkin l'idea è quella di iterare il procedimento indicato nella Definizione 5.8 infinite volte.

**Proposizione 5.10** *Sia $\Gamma$ un insieme consistente, definiamo $\Gamma_0 := \Gamma$, $\Gamma_{n+1} := \Gamma_n^*$, allora $\Gamma_\omega = \bigcup_{n \geq 0} \Gamma_n$ è un insieme di Henkin consistente.*

*Dimostrazione.* Indichiamo con $\mathcal{L}_m$ (rispettivamente $\mathcal{L}_\omega$) il linguaggio di $\Gamma_m$ (rispettivamente $\Gamma_\omega$).

- $\Gamma_\omega$ è un insieme di Henkin. Infatti, sia $\exists x P(x) \in \mathcal{L}_\omega$, allora $\exists x P(x) \in \mathcal{L}_n$ per un certo $n$; per definizione $\exists x P(x) \to P(c_\tau) \in \Gamma_{n+1}$ per un dato $c_\tau$. Quindi $\exists x P(x) \to P(c_\tau) \in \Gamma_\omega$.

- $\Gamma_\omega$ è consistente. Infatti, supponiamo per assurdo che ciò non sia vero, allora $\Gamma_\omega \vdash \bot$, vale a dire che esiste un $n$ tale che $\Gamma_n \vdash \bot$; l'asserto segue dal Lemma 5.9, per induzione su $n$.

□

**Teorema 5.11 (di Lindembaum)**
*Ogni insieme consistente $\Gamma$ di fbf è contenuto in uno consistente massimale.*

La dimostrazione del Teorema di Lindembaum si può effettuare in modo analogo a quella del Teorema 3.10. Tuttavia quest'ultima è essenzialmente basata sulla numerabilità delle formule contenute in $\Gamma$. Più in generale, si può fornire una dimostrazione di tale teorema valida per insiemi consistenti di cardinalità arbitraria utilizzando il lemma di Zorn[5].

[5]Tale lemma asserisce che dato un insieme parzialmente ordinato $\mathcal{X}$, se ogni suo sottoinsieme totalmente ordinato ha un limite superiore in $\mathcal{X}$, allora $\mathcal{X}$ ha almeno un elemento massimale.
Il lemma di Zorn è equivalente all'*assioma della scelta* (per ogni insieme $\mathcal{Y}$ esiste almeno una funzione $f : (\mathcal{P}(\mathcal{Y}) - \emptyset) \to \mathcal{Y}$, detta funzione di scelta, tale che $\forall S \in \mathcal{P}(\mathcal{Y}) - \emptyset,\ f(S) \in S$). Una dimostrazione di tale equivalenza si può trovare, ad esempio, in [Kur65].

*Dimostrazione.* Sia $\Gamma$ un insieme consistente di fbf. Si consideri la famiglia $\mathcal{F}$ di tutti gli insiemi consistenti contenenti $\Gamma$ parzialmente ordinati per inclusione ($\subseteq$). Vogliamo provare che $\mathcal{F}$ ha un elemento massimale. Infatti, ogni catena[6]in $\mathcal{F}$ ha un limite superiore dato dall'unione di tutti gli elementi della catena (Esercizio 5.8), allora, per il Lemma di Zorn, $\mathcal{F}$ ha un elemento massimale $\overline{\Gamma}$ che ovviamente contiene $\Gamma$ ed è consistente massimale. □

***Osservazione*** In generale $\Gamma$ non ha un'unica estensione consistente massimale.

**Corollario 5.12** *Se $\Gamma$ è un insieme di Henkin, allora $\overline{\Gamma}$ lo è.*

*Dimostrazione.* Infatti, per ogni formula del tipo $\exists x P(x)$ per cui esiste un testimone $c$ tale che $\exists x P(x) \rightarrow P(c) \in \Gamma$, allora banalmente $\exists x P(x) \rightarrow P(c) \in \overline{\Gamma}$. □

**Teorema 5.13 (del Modello)**
*Ogni insieme consistente $\Gamma$ è soddisfacibile.*

*Dimostrazione.* Per la Proposizione 5.10, $\Gamma$ può essere esteso ad un insieme di Henkin consistente che, per il Teorema di Lindembaum è contenuto in un insieme consistente massimale. L'asserto segue dai Corollari 5.7 e 5.12. □

**Teorema 5.14 (Completezza)**
*Se $\Gamma \models P$ allora $\Gamma \vdash P$.*

*Dimostrazione.* Supponiamo che $\Gamma \models P$, allora $\Gamma \cup \{\neg P\}$ è insoddisfacibile; per il Teorema del modello $\Gamma, \neg P$ è inconsistente, da cui segue, per (RAA), che $\Gamma \vdash P$. □

**Corollario 5.15 (Completezza debole)**
*Se $\models P$ allora $\vdash P$.*

*Dimostrazione.* Segue dal Teorema di completezza. □

Osserviamo che tale dimostrazione non è costruttiva.

Abbiamo dunque stabilito la correttezza e la completezza della Deduzione Naturale (e dei calcoli logici ad essa equivalenti), vale a dire che una formula è valida se e solo se è dimostrabile in tale calcolo. Tuttavia, ricordiamo che nella logica del primo ordine, al contrario di quanto avviene in quella proposizionale, non è possibile stabilire la validità di una formula per via semantica in quanto la Definizione 4.19.5 richiede di controllare la verità della formula in esame in *ogni* possibile interpretazione (ed ovviamente il numero di queste è infinito); inoltre, se la formula contiene dei quantificatori ed il dominio di una data interpretazione è infinito, anche la determinazione della verità della formula in tale

[6]Famiglia di $\mathcal{F}$ totalmente ordinata per inclusione.

interpretazione può richiedere infiniti controlli.

Nel paragrafo 5.3 discuteremo una dimostrazione alternativa del Teorema di completezza (debole) che è semi-costruttiva: verrà infatti presentato un algoritmo in grado di trovare una prova nel Calcolo dei Sequenti per ogni formula valida. Anticipiamo che, nel caso in cui la formula non lo sia, tale algoritmo può non terminare. Questo è un limite intrinseco della logica del primo ordine dove, come vedremo nel paragrafo 5.4.1, è impossibile stabilire, mediante un procedimento meccanico, se una formula è valida o meno.

## 5.2 Sistemi Assiomatici

Esistono varie formulazioni di Sistemi Assiomatici del primo ordine che differiscono tra loro in base agli assiomi scelti ed al fatto che si insista o meno sull'avere il modus ponens come unica regola di inferenza. L'approccio più comune è quello di ammettere non solo il modus ponens

$$\frac{A \to B \qquad A}{B}$$

ma anche quella di generalizzazione (introduzione del quantificatore universale), ovvero:

$$\frac{A[^y/_x]}{\forall x A}$$

dove $y$ è, al solito, una variabile che non deve comparire libera nelle foglie dell'albero di prova di radice $A[^y/_x]$ ($y$ è abitualmente chiamata *indeterminata* nella terminologia dei Sistemi Assiomatici).
Una scelta tradizionale di (schemi di) assiomi è la seguente:

$(\Pi_0)$ $\vdash \forall x A \to A[^t/_x]$

$(\Pi_1)$ $\vdash (\forall x(C \to A)) \to (C \to \forall x A)$, dove $x \notin FV(C)$

$(\Sigma_0)$ $\vdash A[^t/_x] \to \exists x A$

$(\Sigma_1)$ $\vdash (\forall x(A \to C)) \to (\exists x A \to C)$, dove $x \notin FV(C)$

**Teorema 5.16 (Deduzione)**
*Se* $\Gamma, A \vdash B$*, allora* $\Gamma \vdash A \to B$*.*

*Dimostrazione.* La dimostrazione ricalca fedelmente quella del relativo teorema per il calcolo proposizionale. Ci limitiamo pertanto a trattare l'unico caso aggiuntivo, quello in cui $B$ sia ottenuto tramite una applicazione di $(\forall i)$. In tale situazione, $B$ è una formula del tipo $\forall x C$ e $\Gamma, A \vdash C[^y/_x]$ per una qualche indeterminata $y$. Per ipotesi induttiva, $\Gamma \vdash A \to C[^y/_x]$, ed applicando $(\forall i)$ risulta anche $\Gamma \vdash \forall y(A \to C[^y/_x])$. Poiché $y$ non è libera in $A$, si possono utilizzare l'assioma $(\Pi_1)$ ed il modus ponens, ottenendo $\Gamma \vdash A \to \forall y C[^y/_x]$. $\square$

***Osservazione*** La regola di generalizzazione utilizzata in questa trattazione non è quella che solitamente si incontra nei testi che si occupano dell'argomento. Infatti, in genere, nei sistemi alla Hilbert tale regola ha la forma

$$(Gen) \qquad \frac{A[^{y}/_{x}]}{\forall x A}$$

senza porre alcuna restrizione su $x$.
In tal caso, però, il Teorema di deduzione non vale con piena generalità, come mostra il seguente controesempio:
per $(Gen)$, $A(x) \vdash \forall x A(x)$ ma $A(x) \rightarrow \forall x A(x)$ non è una formula valida (il lettore lo verifichi per esercizio). È quindi necessario formulare tale teorema con una restrizione sufficiente ad escludere controesempi analoghi a quello riportato. In particolare, il Teorema di deduzione viene enunciato nel seguente modo: Se $\Gamma, A \vdash B$ e la derivazione di $B$ da $\Gamma, A$ non contiene alcuna applicazione di $(Gen)$ a variabili libere in $A$, allora $\Gamma \vdash A \rightarrow B$.

Anche nel caso predicativo è piuttosto semplice dimostrare l'equivalenza tra il Sistema Assiomatico ed il calcolo di Deduzione Naturale.

**Teorema 5.17** $\Gamma \vdash_{H} P$ *implica* $\Gamma \vdash_{ND} P$.

*Dimostrazione.* È sufficiente provare che gli assiomi sono dimostrabili in Deduzione Naturale (il lettore lo faccia per esercizio). □

**Teorema 5.18** $\Gamma \vdash_{ND} P$ *implica* $\Gamma \vdash_{H} P$.

*Dimostrazione.* È necessario dimostrare che le regole della Deduzione Naturale sono derivabili nel Sistema Assiomatico. Infatti $(\forall e)$ segue da modus ponens e $(\Pi_0)$, mentre $(\exists\ i)$ segue da modus ponens e $(\Sigma_0)$. Per quanto riguarda le regole dei connettivi logici, abbiamo già visto che queste erano derivabili non appena lo era $(\rightarrow i)$. Lo stesso vale per $(\exists e)$. Supponiamo, infatti, di avere una dimostrazione di $\Gamma, A[^{y}/_{x}] \vdash B$ dove $y$ non compare libera in $\Gamma, B$. Per $(\rightarrow i)$, si ha una dimostrazione di $\Gamma \vdash A[^{y}/_{x}] \rightarrow B$, ed applicando $(\forall i)$ si ottiene $\Gamma \vdash \forall y(A[^{y}/_{x}] \rightarrow B)$. Ma $y$ non è libera in $B$, dunque è possibile usare l'assioma $(\Sigma_1)$ ed il modus ponens per ottenere una dimostrazione di $\Gamma \vdash \exists y(A[^{y}/_{x}]) \rightarrow B$. Con una ulteriore applicazione di modus ponens abbiamo infine $\Gamma, \exists y(A[^{y}/_{x}]) \vdash B$, che era quanto si voleva dimostrare $(\exists y(A[^{y}/_{x}]) = \exists x A)$. La derivabilità di $(\rightarrow i)$ corrisponde al Teorema di deduzione. □

Come ovvio corollario di tale teorema seguono la correttezza e la completezza del Sistema Assiomatico in esame.

***Osservazione*** Un Sistema Assiomatico per la logica del primo ordine con uguaglianza si ricava aggiungendo agli assiomi appena presentati quelli di p.114.

## 5.3 IL Calcolo dei Sequenti

La versione al primo ordine del Calcolo dei Sequenti si ottiene aggiungendo le seguenti regole logiche di inferenza per i quantificatori:

$$(\forall l)\quad \frac{\Gamma, A[^t/_x] \vdash \Delta}{\Gamma, \forall x A \vdash \Delta} \qquad (\forall r)\quad \frac{\Gamma \vdash A[^y/_x], \Delta}{\Gamma \vdash \forall x A, \Delta}$$

$$(\exists l)\quad \frac{\Gamma, A[^y/_x] \vdash \Delta}{\Gamma, \exists x A \vdash \Delta} \qquad (\exists r)\quad \frac{\Gamma \vdash A[^t/_x], \Delta}{\Gamma \vdash \exists x A, \Delta}$$

Come condizione ausiliaria per le regole $(\forall r)$ e $(\exists l)$ è necessario richiedere che la variabile $y$ non compaia libera nel sequente conclusione.
Come nel caso del calcolo proposizionale, la *versione intuizionista* si ottiene imponendo la restrizione che nella parte destra di ogni sequente possa apparire al più una formula.
Vediamo qualche esempio di derivazione:

**Esempio 5.6** $\exists x(A \to B(x)) \vdash A \to \exists x B(x)$

$$\dfrac{\dfrac{\dfrac{A \vdash A, \exists x B(x) \qquad \dfrac{B(x), A \vdash B(x)}{B(x), A \vdash \exists x B(x)}}{A \to B(x), A \vdash \exists x B(x)}}{A \to B(x) \vdash A \to \exists x B(x)}}{\exists x(A \to B(x)) \vdash A \to \exists x B(x)}$$

**Esempio 5.7** $\neg\forall x A(x) \vdash \exists x \neg A(x)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdash A(x), \neg A(x)}{\vdash A(x), \exists x \neg A(x)}}{\vdash \forall x A(x), \exists x \neg A(x)}}{\neg\forall x A(x) \vdash \exists x \neg A(x)}}$$

La dimostrazione di correttezza semantica delle regole del Calcolo dei Sequenti è simile a quella per le regole della Deduzione Naturale, ed è dunque lasciata al lettore come semplice esercizio. Enunciamo semplicemente il risultato, per comodità di riferimento.

**Teorema 5.19 (Correttezza)**
$\Gamma \vdash \Delta \;\Rightarrow\; \Gamma \models \Delta$.

### 5.3.1 Invertibilità

Le regole $(\forall l)$ e $(\exists r)$ presentano problemi di invertibilità che complicano la ricerca "all'indietro" dell'albero di prova.
Si consideri, ad esempio, la dimostrazione del sequente $A(t_1) \vee A(t_2) \vdash \exists x A(x)$:

$$\dfrac{\dfrac{A(t_1) \vdash A(t_1)}{A(t_1) \vdash \exists x A(x)} \qquad \dfrac{A(t_2) \vdash A(t_2)}{A(t_2) \vdash \exists x A(x)}}{A(t_1) \vee A(t_2) \vdash \exists x A(x)}$$

Non esiste alcuna dimostrazione di esso che termini con una applicazione di $(\exists r)$. Infatti, la premessa di tale regola dovrebbe avere la forma $A(t_1) \vee A(t_2) \vdash A(t_3)$, ma, ovviamente, tale sequente non è dimostrabile (questa è una conseguenza immediata della *completezza* del calcolo che sarà discussa nella prossima sezione).
Ricordando che il connettivo esistenziale è una forma di disgiunzione infinita, per il modo in cui si è risolto l'analogo problema di invertibilità per il connettivo binario di disgiunzione nel caso proposizionale, si può tentare di sostituire $(\exists r)$ con uno *schema* di regola del seguente tipo:

$$(\exists r') \quad \dfrac{\Gamma \vdash A[^{t_1}/_x], \ldots, A[^{t_n}/_x], \Delta}{\Gamma \vdash \exists x A, \Delta}$$

(lasciamo al lettore la cura di dimostrare che tale regola risolve il problema nel caso precedente).
Un'altra possibilità è di vedere la formula $\exists x A(x)$ come una sorta di "generatore" delle sue possibili alternative, che conduce all'unica regola:

$$(\exists r'') \quad \dfrac{\Gamma \vdash A[^{t}/_x], \exists x A, \Delta}{\Gamma \vdash \exists x A, \Delta}$$

Simmetricamente, $(\forall l)$ può essere sostituita da

$$(\forall l'') \quad \dfrac{\Gamma, A[^{t}/_x], \forall x A \vdash \Delta}{\Gamma, \forall x A, \vdash \Delta}$$

Utilizzando $(\exists r'')$ si può riscrivere la dimostrazione di $A(t_1) \vee A(t_2) \vdash \exists x A(x)$ nel modo seguente:

$$\dfrac{\dfrac{\dfrac{A(t_1) \vdash A(t_1), A(t_2), \exists x A(x) \qquad A(t_2) \vdash A(t_1), A(t_2), \exists x A(x)}{A(t_1) \vee A(t_2) \vdash A(t_1), A(t_2), \exists x A(x)}}{A(t_1) \vee A(t_2) \vdash A(t_2), \exists x A(x)}}{A(t_1) \vee A(t_2) \vdash \exists x A(x)}$$

È facile dimostrare che le nuove regole sono logicamente equivalenti a quelle di partenza. Inoltre, esse godono di un'ulteriore proprietà che è stata utilizzata

nella dimostrazione di completezza del calcolo proposizionale: le regole $(\exists r'')$ e $(\forall l'')$ sono *reversibili*, nel senso che i sequenti conclusione sono semanticamente validi se e solo se lo sono le rispettive premesse.
Da questo punto di vista, è sempre possibile concludere una dimostrazione il cui sequente finale contiene una formula quantificata esistenzialemente nella parte destra (rispettivamente, quantificata universalmente nella parte sinistra) con una applicazione di $(\exists r'')$ (rispettivamente $(\forall l'')$). Tuttavia tale applicazione potrebbe semplicemente costituire del lavoro inutile e complicare l'albero di prova.
Si consideri, ad esempio, la seguente dimostrazione, che utilizza le regole originarie, del sequente $\forall x A(x) \vdash \forall x(A(x) \vee B)$:

$$\dfrac{\dfrac{\dfrac{A(x) \vdash A(x), B}{A(x) \vdash A(x) \vee B}}{\forall x A(x) \vdash A(x) \vee B}}{\forall x A(x) \vdash \forall x(A(x) \vee B)}$$

Cercando di invertire le due applicazioni di $(\forall l)$ e $(\forall r)$ si otterrebbe:

$$\dfrac{\dfrac{\dfrac{A(x) \vdash A(x), B}{A(x) \vdash A(x) \vee B}}{A(x) \vdash \forall x(A(x) \vee B)}}{\forall x A(x) \vdash \forall x(A(x) \vee B)}$$

Il problema è che, in tale situazione, l'applicazione di $(\forall r)$ è *scorretta* in quanto $x$ appare libera nel sequente conclusione. In questo caso, anche le nuove regole non sono sufficienti a risolvere il reale problema di inversione. È vero infatti che è sempre possibile applicare come ultima regola $(\forall l'')$, come nell'albero seguente (dove $t$ è un qualunque termine che non contiene $x$)

$$\dfrac{\dfrac{\dfrac{\dfrac{A(t), A(x) \vdash A(x), B}{A(t), A(x) \vdash A(x) \vee B}}{A(t), \forall x A(x) \vdash A(x) \vee B}}{A(t), \forall x A(x) \vdash \forall x(A(x) \vee B)}}{\forall x A(x) \vdash \forall x(A(x) \vee B)}$$

Tuttavia è stato semplicemente effettuato del lavoro inutile, introducendo la formula aggiuntiva $A(t)$ nella parte sinistra dei sequenti che non entra in gioco in alcun modo nella dimostrazione. Ovviamente, non si è risolto il reale problema di invertibilità, che in questo caso è intrinseco al sistema logico.

### 5.3.2 Un algoritmo di ricerca automatica

Il lavoro aggiuntivo prodotto dalle regole $(\forall l'')$ e $(\exists r'')$ può creare dei rami infiniti durante il tentativo costruzione all'indietro di un albero di prova, portando alla

non terminazione dell'algoritmo di ricerca anche in presenza di dimostrazioni finite. Ad esempio, nel caso del sequente $\forall x A(x) \vdash \forall x(A(x) \vee B)$, la ricerca potrebbe procedere nel modo seguente:

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{A(t_1), A(t_2), A(t_3), \forall x A(x) \vdash \forall x(A(x) \vee B)}}{A(t_1), A(t_2), \forall x A(x) \vdash \forall x(A(x) \vee B)}}{A(t_1), \forall x A(x) \vdash \forall x(A(x) \vee B)}}{\forall x A(x) \vdash \forall x(A(x) \vee B)}$$

Nell'algoritmo di ricerca di una dimostrazione si dovrebbe dunque garantire che, in presenza di un sequente a cui si possono applicare numerose regole logiche, nessuna di queste sia privilegiata rispetto alle altre. È possibile immaginare svariate tecniche (ed innumerevoli euristiche) per affrontare il problema. Poiché le complesse problematiche connesse all'efficienza di tali algoritmi esulano dallo scopo della presente trattazione, la soluzione che adotteremo è basata sul seguente approccio, piuttosto *naif*:

1. dato il sequente $\Gamma \vdash \Delta$ l'algoritmo cercherà ad ogni passo pari di applicare all'indietro una regola logica sulla prima formula (composta) di $\Gamma$, e ad ogni passo dispari sulla prima formula (composta) di $\Delta$;

2. le formule ausiliarie delle regole logiche prodotte dalla loro applicazione all'indietro saranno poste *in coda* nei sequenti che sono premesse della regola.

Esplicitamente, le regole che verranno applicate (da leggersi "all'indietro") sono riportate nella Figura 5.1 ($\Xi$ rappresenta una sequenza di *formule atomiche*). L'algoritmo si arresta lungo un ramo di ricerca non appena si genera un sequente $S$ che soddisfa una delle proprietà seguenti:

1. (terminazione con successo) $S$ è un quasi-assioma, cioè un sequente in cui la stessa formula appare sia nella parte destra che in quella sinistra (ci si può eventualmente restringere al caso di formule atomiche). In tale situazione diremo che il ramo è *chiuso*.

2. (terminazione con fallimento) $S$ è un sequente $\Xi_1 \vdash \Xi_2$ composto unicamente da formule atomiche e tale che nessuna formula appare sia in $\Xi_1$ che in $\Xi_2$.

Nel caso di terminazione con successo lungo un ramo di ricerca, si passerà ad esplorare i rami ancora aperti. Viceversa, in caso di terminazione con fallimento lungo un qualche ramo è possibile interrompere l'intera ricerca concludendo che il sequente non è dimostrabile. L'albero di ricerca è *chiuso* quando tutti i suoi rami sono chiusi.
Evidenziamo ancora il fatto che l'algoritmo potrebbe anche non terminare affatto, perdendosi lungo un ramo di ricerca infinito; tale problema non dipende

$$(\wedge l)\quad \frac{\Xi, \Gamma, A, B \vdash \Delta}{\Xi, A \wedge B, \Gamma \vdash \Delta} \qquad (\wedge r)\quad \frac{\Gamma \vdash \Xi, \Delta, A \quad \Gamma \vdash \Xi, \Delta, B}{\Gamma \vdash \Xi, A \wedge B, \Delta}$$

$$(\vee l)\quad \frac{\Xi, \Gamma, A \vdash \Delta \quad \Xi, \Gamma, B \vdash \Delta}{\Xi, A \vee B, \Gamma \vdash \Delta} \qquad (\vee r)\quad \frac{\Gamma \vdash \Xi, \Delta, A, B}{\Gamma \vdash \Xi, A \vee B, \Delta}$$

$$(\rightarrow l)\quad \frac{\Xi, \Gamma \vdash \Delta, A \quad \Xi, \Gamma, B \vdash \Delta}{\Xi, A \rightarrow B, \Gamma \vdash \Delta} \qquad (\rightarrow r)\quad \frac{\Gamma, A \vdash \Xi, B, \Delta}{\Gamma \vdash \Xi, A \rightarrow B, \Delta}$$

$$(\neg l)\quad \frac{\Xi, \Gamma \vdash \Delta, A}{\Xi, \neg A, \Gamma \vdash \Delta} \qquad (\neg r)\quad \frac{\Gamma, A \vdash \Xi, \Delta}{\Gamma \vdash \Xi, \neg A, \Delta}$$

$$(\forall l)\quad \frac{\Xi, \Gamma, A[{}^{t}/_{x}], \forall x A \vdash \Delta}{\Xi, \forall x A, \Gamma \vdash \Delta} \qquad (\forall r)\quad \frac{\Gamma \vdash \Xi, \Delta, A[{}^{y}/_{x}]}{\Gamma \vdash \Xi, \forall x A, \Delta}$$

$$(\exists l)\quad \frac{\Xi, \Gamma, A[{}^{y}/_{x}] \vdash \Delta}{\Xi, \exists x A, \Gamma \vdash \Delta} \qquad (\exists r)\quad \frac{\Gamma, \vdash \Xi, \Delta, A[{}^{t}/_{x}], \exists x A}{\Gamma \vdash \Xi, \exists x A, \Delta}$$

Figura 5.1: Formulazione di HK per un algoritmo di ricerca all'indietro

dal particolare algoritmo in esame ma, come risulterà chiaro nel paragrafo 5.4.1, è un limite intrinseco della logica del primo ordine.
Il principale problema che ancora resta da risolvere è la scelta della variabile $y$ e del termine $t$ nelle regole dei quantificatori.
Per quanto riguarda $y$ in $(\forall r)$ e $(\exists l)$ è necessario garantire che essa non appaia libera nel sequente conclusione. Una semplice soluzione consiste nel fissare un'enumerazione di tutte le variabili e considerare la prima variabile avente tale caratteristica.
In $(\forall l)$ e $(\exists r)$, $t$ può essere un termine generico del linguaggio. Il problema è che si deve garantire che tutti i termini possano potenzialmente essere presi in esame. A tale scopo fissiamo una certa enumerazione $t_1, t_2, \ldots, t_n, \ldots$ dei termini del linguaggio. Allorchè si applicherà una delle regole $(\forall l)$ o $(\exists r)$ si avrà cura di scegliere il primo termine della enumerazione che non è stato ancora utilizzato per generare una formula ausiliaria per il dato quantificatore lungo il cammino dell'albero di ricerca che conduce dal sequente in esame fino alla radice (si ricordi che si sta costruendo l'albero "all'indietro").
Introduciamo ora alcune nozioni e risultati che verranno utilizzati nel prossimo paragrafo.

**Definizione 5.20** *Data una interpretazione* $(\mathcal{A}, \xi^{\mathcal{A}})$*, ed un sequente* $\Gamma \vdash \Delta$ *diremo che una formula* $P$ *del sequente è un* costituente positivo *se appare nella parte destra e* $v^{(\mathcal{A},\xi^{\mathcal{A}})}(P) = 1$ *oppure se appare nella parte sinistra e* $v^{(\mathcal{A},\xi^{\mathcal{A}})}(P) = 0$*. Se* $P$ *ha caratteristiche opposte è detto* costituente negativo.

**Proposizione 5.21** *Sia $(\mathcal{A}, \xi^{\mathcal{A}})$ una interpretazione e $P$ un costituente positivo di $\Gamma \vdash \Delta$. Se $P$ è del tipo $\neg A$, $A \wedge B$, $A \vee B$, $A \to B$, $\forall x A$ in $\Delta$, o $\exists x A$ in $\Gamma$, allora in ogni sequente premessa di una regola di introduzione per $P$ una delle formule ausiliarie della regola è ancora un costituente positivo.*

*Dimostrazione.* Si effettua per casi sulle varie regole. Ne vedremo solo alcuni, i rimanenti sono lasciati al lettore come esercizio.

1. $P$ è $A \wedge B$, e $P$ si trova nella parte sinistra del sequente. La regola di introduzione per $P$ è dunque:

$$(\wedge l) \quad \frac{\Xi, \Gamma, A, B \vdash \Delta}{\Xi, A \wedge B, \Gamma \vdash \Delta}$$

Poiché $P$ è un costituente positivo e si trova nella parte sinistra, deve essere $v^{(\mathcal{A},\xi^{\mathcal{A}})}(P) = 0$. Questo implica che o $v^{(\mathcal{A},\xi^{\mathcal{A}})}(A) = 0$, oppure $v^{(\mathcal{A},\xi^{\mathcal{A}})}(B) = 0$. Poichè sia $A$ che $B$ sono nella parte sinistra della premessa, uno di essi è un costituente positivo.

2. $P$ è $A \wedge B$, e $P$ si trova nella parte destra del sequente. La regola di introduzione per $P$ è dunque:

$$(\wedge r) \quad \frac{\Gamma \vdash \Xi, \Delta, A \qquad \Gamma \vdash \Xi, \Delta, B}{\Gamma \vdash \Xi, A \wedge B, \Delta}$$

Poiché $P$ è un costituente positivo e si trova nella parte destra, deve essere $v^{(\mathcal{A},\xi^{\mathcal{A}})}(P) = 1$. Questo implica che $v^{(\mathcal{A},\xi^{\mathcal{A}})}(A) = 1$, e $v^{(\mathcal{A},\xi^{\mathcal{A}})}(B) = 1$. Poichè sia $A$ che $B$ sono nella parte destra delle rispettive premesse, entrambi sono costituenti positivi.

3. Se $P$ è $A \to B$ e $P$ si trova nella parte sinistra del sequente. La regola di introduzione per $P$ è:

$$(\to l) \quad \frac{\Xi, \Gamma \vdash \Delta, A \qquad \Xi, \Gamma, B \vdash \Delta}{\Xi, A \to B, \Gamma \vdash \Delta}$$

Poiché $P$ è un costituente positivo e si trova nella parte sinistra, deve essere $v^{(\mathcal{A},\xi^{\mathcal{A}})}(P) = 0$, che implica $v^{(\mathcal{A},\xi^{\mathcal{A}})}(A) = 1$ e $v^{(\mathcal{A},\xi^{\mathcal{A}})}(B) = 0$. Poichè $A$ che $B$ sono rispettivamente nella parte sinistra ed in quella destra delle premesse, entrambi sono costituenti positivi.

4. Se $P$ è $\forall x A$ e $P$ si trova nella parte destra del sequente. La regola di introduzione per $P$ è:

$$(\forall r) \quad \frac{\Gamma \vdash \Xi, \Delta, A[{}^{y}/_{x}]}{\Gamma \vdash \Xi, \forall x A, \Delta}$$

Inoltre deve essere $v^{(\mathcal{A},\xi^{\mathcal{A}})}(P) = 1$ che implica $v^{(\mathcal{A},\xi^{\mathcal{A}})}(A[{}^{y}/_{x}]) = 1$. Poichè $A[{}^{y}/_{x}]$ è ancora nella parte destra della premessa, è un costituente positivo.

□

### 5.3.3 Completezza

Utilizziamo l'algoritmo di ricerca precedente per dimostrare la completezza[7] del Calcolo dei Sequenti (nella formulazione usata nell'algoritmo, che tuttavia è banalmente equivalente alle precedenti), vale a dire che $\Gamma \models \Delta$ implica $\Gamma \vdash \Delta$. Equivalentemente (dal punto di vista classico), si vuole dimostrare che $\Gamma \not\models \Delta$ oppure $\Gamma \vdash \Delta$.
Supponiamo di cominciare l'algoritmo di ricerca all'indietro con il sequente $\Gamma \vdash \Delta$. Vi sono due possibilità: o l'algoritmo genera un un albero chiuso, oppure no. Nel primo caso, l'albero costituisce una dimostrazione di $\Gamma \vdash \Delta$. Dunque, per dimostrare la completezza è sufficiente provare che se l'algoritmo non conduce ad un albero chiuso, allora $\Gamma \not\models \Delta$, ovvero esiste una qualche interpretazione che soddisfa tutte le formule in $\Gamma$ ma nessuna formula in $\Delta$.
Supponiamo dunque che l'algoritmo di ricerca non conduca ad un albero chiuso. Vi sono due possibilità: o esiste un ramo dell'albero che porta ad un sequente $\Xi_1 \vdash \Xi_2$ le cui formule sono tutte atomiche e nessuna formula è comune a $\Xi_1$ e $\Xi_2$, oppure esiste almeno un ramo infinito. Consideriamo questo ramo, e sia $S_0, S_1, S_2, \ldots S_n, \ldots$, la lista (possibilmente infinita) dei sequenti corrispondenti ai nodi dell'albero, partendo dalla radice.
Siano $\mathcal{M}$ l'insieme di tutte le formule atomiche che appaiono nella parte sinistra di un qualche sequente $S_i$, ed $\mathcal{N}$ l'insieme di tutte le formule atomiche che appaiono nella parte destra di un qualche sequente $S_i$.

**Lemma 5.22** $\mathcal{M} \cap \mathcal{N} = \emptyset$

*Dimostrazione.* Per come sono definite le regole di inferenza, allorchè una formula atomica appare su un lato di un qualche sequente $S_i$, apparirà necessariamente sullo stesso lato per tutti i sequenti $S_j$ per $j \geq i$. Dunque se esistesse una formula in comune a $\mathcal{M}$ e $\mathcal{N}$ esisterebbe anche un qualche sequente $S_k$ quasi primo, contraddicendo il fatto che stiamo considerando un ramo non chiuso dell'albero. □

Vogliamo ora definire un'interpretazione $\mathcal{A}_\mathcal{M}$ in cui non valga $\Gamma \models \Delta$.
Tale interpretazione è nota come interpretazione dei termini o canonica (Definizione 5.5). La ragione di questo nome è che come dominio $D_{\mathcal{A}_\mathcal{M}}$ della struttura si considera l'insieme dei termini del linguaggio $TER(\mathcal{L})$, e l'interpretazione di un termine è ... il termine stesso. Più precisamente, l'interpretazione di un simbolo di costante $c$ è ... $c$ stessa, l'interpretazione di un simbolo di funzione $f^n$ è quella funzione da $TER^n(\mathcal{L})$ in $TER(\mathcal{L})$ che presi $n$ termini $t_1, \ldots, t_n$ gli associa il termine $f(t_1, \ldots, t_n)$. Come ambiente si sceglie $\xi$ tale che per ogni variabile $x$, $\xi(x) = x$. Risulta $[\![t]\!]_\xi^{\mathcal{A}_\mathcal{M}} = t$ (Esercizio 5.6). Per quanto riguarda i predicati, un simbolo di proposizione $P^n$ è interpretato come quella funzione $P^{\mathcal{A}_\mathcal{M}}$ da $TER^n(\mathcal{L})$ ai booleani tale che $P^{\mathcal{A}_\mathcal{M}}(t_1, \ldots, t_n) = 1$ se e solo se $P(t_1, \ldots, t_n) \in \mathcal{M}$.

**Definizione 5.23** *Si definisce* ordine *di una formula $P$ il numero di quantificatori e connettivi proposizionali che vi compaiono.*

[7] Nel caso in cui l'insieme di formule di partenza sia finito.

**Lemma 5.24** *Se un sequente $S_i$ contiene un costituente positivo di ordine $n+1$, esiste un sequente $S_j$ con $j \geq i$ che contiene un costituente positivo di ordine $n$.*

*Dimostrazione.* Consideriamo il sequente $S_i$ e sia $P$ il costituente positivo. Per come sono definite le regole logiche tale costituente resterà sullo stesso lato del sequente finchè non interverrà come formula principale di una regola. Sia $S_k$ il sequente in questione, ed $R$ la regola applicata per generare $S_{k+1}$. Se tale regola è di introduzione di un connettivo logico, $(\forall r)$ o $(\exists l)$, l'asserto è una conseguenza immediata della Proposizione 5.21, scelto $j = k + 1$.
Supponiamo allora che $R$ sia $(\exists r)$ e $P$ sia $\exists x A$. Poiché il costituente positivo $P$ non potrà mai essere eliminato, il ramo di ricerca in oggetto non può essere un ramo di fallimento e deve necessariamente essere infinito. Questo implica che lungo il ramo $\exists x B$ sarà utilizzato infinite volte come formula principale di $(\exists r)$, e dunque che per ogni termine $t$ del linguaggio si genererà una formula del tipo $B[^t/_x]$ nella parte destra. Se $\exists x B$ era un componente positivo, essendo nella parte destra doveva avere valore di verità 1. Per il Lemma 4.28, se $(\mathcal{A}_\mathcal{M}, \xi^{\mathcal{A}_\mathcal{M}}) \models \exists x B$ allora esiste un elemento del dominio, cioè un termine $s$ per cui $(\mathcal{A}_\mathcal{M}, \xi^{\mathcal{A}_\mathcal{M}}[^s/_x]) \models B$. Ma $v^{(\mathcal{A}_\mathcal{M}, \xi^{\mathcal{A}_\mathcal{M}}[^s/_x])}(B) = v^{(\mathcal{A}_\mathcal{M}, \xi^{\mathcal{A}_\mathcal{M}})}(B^{[[s]]_\xi^{\mathcal{A}_\mathcal{M}}}/_x])$, e per il Lemma di traslazione $= v^{(\mathcal{A}_\mathcal{M}, \xi^{\mathcal{A}_\mathcal{M}})}(B[^s/_x])$. Poichè necessariamente esisterà un qualche sequente $S_j$ in cui $B[^s/_x]$ verrà generato (ed eventualmente sulla destra), si ha l'asserto. Il caso di $(\forall l)$ è del tutto simmetrico. □

Si può ora concludere facilmente il Teorema di completezza. Supponiamo infatti per assurdo che nell'interpretazione $(\mathcal{A}_\mathcal{M}, \xi^{\mathcal{A}_\mathcal{M}})$, $\Gamma \models \Delta$. Questo implica che il sequente $S_0 = \Gamma \vdash \Delta$ deve contenere almeno un costituente positivo. Sia $n$ il suo ordine. Se $n > 0$, per il lemma precedente necessariamente esiste un qualche sequente $S_j$ con un costituente positivo di ordine $n - 1$, e così via, finché non si ottiene un sequente con un costituente positivo di ordine 0. Ma questo è assurdo poichè per come è stata definita l'interpretazione, tutti i costituenti di ordine 0 (cioè formule atomiche) sono necessariamente negativi (se il costituente è sulla sinistra allora appartiene a $\mathcal{M}$ e dunque ha valore 1, mentre se è sulla destra, allora appartiene a $\mathcal{N}$, quindi non appartiene a $\mathcal{M}$ ed ha valore 0).

### 5.3.4 Discussione dell'algoritmo

L'algoritmo mostrato in precedenza è particolarmente inefficiente.
Come dovrebbe essere chiaro dalla dimostrazione di completezza, gli unici problemi sono dovuti alle regole $(\forall r)$ e $(\exists l)$. È possibile dunque cominciare con l'applicare tutte le rimanenti regole e rimandare l'applicazione di queste ultime. Si noti che questa parte è necessariamente terminante, poichè tutte le regole, ad esclusione di $(\forall r)$ e $(\exists l)$, fanno decrescere l'ordine di una qualche formula del sequente. Alla fine, ci si riduce necessariamente a sequenti in una "forma normale" in cui nella parte sinistra vi sono solo formule atomiche o formule quantificate universalmente, mentre nella parte destra si hanno ancora formule atomiche o formule quantificate esistenzialmente. A questo punto si è costretti ad applicare $(\forall r)$ oppure $(\exists l)$ (che si applicheranno alternativamente, in modo

da operare una volta sulla parte destra ed una volta su quella sinistra del sequente). Tale applicazione genererà una copia della formula quantificata (che verrà messa in coda alla lista delle formule quantificate dello stesso tipo, in modo da avere priorità minore rispetto alle altre) ed una istanza della formula stessa che può essere processata immediatamente in modo da riottenere una nuova forma normale.
Ovviamente il problema principale è ancora la scelta del termine con cui creare l'istanza. In questo caso, si possono tentare delle euristiche, in modo da guidare la scelta in base alle altre formule che compaiono nel sequente, tenendo presente che l'obiettivo è quello di ottenere un quasi-assioma. Ad esempio, se si è nella situazione

$$\forall x A(x) \vdash A(t)$$

si può immediatamente decidere di istanziare $\forall x A(x)$ ad $A(t)$, dato che tale formula compare nella parte opposta.
Un esempio appena più complesso è costituito dal sequente $\forall x B(x, f(a)) \vdash \exists y B(g(y), y)$. Una sua possibile dimostrazione è la seguente:

$$\dfrac{\dfrac{B(g(f(a)), f(a)), \forall x B(x, f(a)) \vdash B(g(f(a)), f(a)), \exists y B(g(y), y)}{B(g(f(a)), f(a)), \forall x B(x, f(a)) \vdash \exists y B(g(y), y)}}{\forall x B(x, f(a)) \vdash \exists y B(g(y), y)}$$

Già in questo caso, tuttavia, non è del tutto chiaro come orientare in modo automatico la scelta delle istanziazioni operate, cioè $[^{g(f(a))}/_{x}]$ e $[^{f(a)}/_{y}]$.
Un possibile approccio consiste nel *ritardare* questa scelta, introducendo delle *metavariabili* (che denoteremo con lettere maiuscole) che intuitivamente rappresentano dei termini generici. Ad esempio, nel caso precedente, si procederebbe nel modo seguente:

$$\dfrac{\dfrac{B(X, f(a)), \forall x B(x, f(a)) \vdash B(g(Y), Y), \exists y B(g(y), y)}{B(X, f(a)), \forall x B(x, f(a)) \vdash \exists y B(g(y), y)}}{\forall x B(x, f(a)) \vdash \exists y B(g(y), y)}$$

A questo punto, vi sono le due formule atomiche $B(X, f(a))$ e $B(g(Y), Y)$ in parti opposte del sequente che appaiono come dei buoni candidati per generare un quasi assioma. L'unico problema è quello di capire se ammettono una istanza comune, cioè se è possibile istanziare $X$ e $Y$ in modo tale da rendere le due formule sintatticamente identiche. Questo problema è noto con il nome di *unificazione*, e verrà trattato formalmente nel prossimo capitolo (in un contesto differente). Per il momento, osserviamo semplicemente che il problema dell'unificazione è risolubile, e fornirebbe, nel caso precedente, la soluzione corretta.
L'uso di metavariabili comporta tuttavia delle complicazioni nelle regole $(\forall r)$ e $(\exists r)$ in quanto queste prevedono che la variabile $y$ introdotta (all'indietro) non compaia libera nel sequente conclusione della regola. Se il sequente conteneva una qualche metavariabile X, è necessario garantire che $X$ non venga in

seguito istanziata ad un termine che contiene occorrenze libere di $y$. Una possibile soluzione consiste nell'esplicitare la dipendenza di $y$ da $X$, scrivendo ad esempio $y(X)$. Questo impedisce che $X$ possa essere in seguito istanziata ad un termine che contiene $y(X)$, poichè si avrebbe un ovvio problema di circolarità (una meta-variabile non può mai essere unificata ad un termine che contiene una occorrenza della meta-variabile stessa).
Ovviamente, se $X$ viene istanziata ad un termine $t$ che a sua volta dipende da altre metavariabili $X_1, \ldots, X_n$, anche ogni variabile che dipendeva da $X$ viene ora a dipendere da $X_1, \ldots, X_n$.
Riassumendo:

- quando si applica una regola del tipo $(\forall l)$ o $(\exists r)$ si introduce una nuova metavariabile;
- quando si applica una regola del tipo $(\forall r)$ o $(\exists l)$ si introduce una variabile che non occorre ancora nel resto dell'albero e che dipende da tutte le metavariabili $X_1, \ldots, X_n$ che appaiono nel sequente conclusione;
- si ammette l'instanziazione di $X$ a $t$ solo se $t$ non dipende da $X$;
- l'instanziazione di una metavariabile $X$ ad un termine $t$ deve risultare nella sostituzione di $X$ con $t$ in tutto l'albero di prova; inoltre ogni variabile che dipendeva da $X$ dipenderà ora da da tutte le metavaribili che compaiono in $t$.

Il lettore interessato ad approfondire queste problematiche, può consultare [Pau92] e la bibliografia ivi menzionata.

## 5.4 Applicazioni del Teorema di completezza

### 5.4.1 Il problema della decisione

La prima applicazione del Teorema di completezza riguarda il problema di decidere se una formula della logica dei predicati è valida.
Come abbiamo già osservato in precedenza, nella logica del primo ordine, al contrario di quanto avviene in quella proposizionale, non è (a priori) possibile stabilire la validità di una formula per via semantica sia per l'infinità delle possibili interpretazioni, che per la natura possibilmente infinita dei domini che, in presenza di quantificatori, richiederebbe infiniti controlli. Al contrario, il problema della *derivabilità* di una formula in un qualche calcolo logico è chiaramente semi-decidibile, nel senso che si è in grado di fornire un algoritmo che risponde affermativamente se una formula $P$ è effettivamente derivabile ($\vdash P$), ma che potrebbe non terminare in caso contrario (e dunque, non è *mai* possibile concludere $\nvdash P$). Osserviamo infatti che, data la natura finitista dei calcoli, è possibile *enumerare* tutte le formule derivabili. Ad esempio, in Deduzione Naturale, si può cominciare con l'enumerare tutte le formule derivabili con alberi di profondità 1, poi passare agli alberi di profondità 2, e cosí via. Se la formula

$P$ è derivabile, allora prima o poi comparirà in tale enumerazione e si è dunque in grado di rispondere in modo affermativo; in caso contrario, l'algoritmo continuerà ad enumerare nuove formule derivabili considerando alberi di profondità sempre maggiore, senza tuttavia poter fornire una risposta negativa.
L'algoritmo di ricerca all'indietro utilizzato nella dimostrazione di completezza per il Calcolo dei Sequenti è un altro esempio di programma che è in grado di semi-decidere la derivabilità di una formula del primo ordine: se una dimostrazione esiste si è sicuri prima o poi di trovarla; in caso contrario, l'algoritmo potrebbe non terminare affatto.
Dal Teorema di completezza segue che $\vdash P \Leftrightarrow \models P$, dunque è possibile concludere che anche la nozione di validità è semi-decidibile.

La domanda che ci si pone è la seguente: il problema della decisione è anche *decidibile*, vale a dire, data una qualunque formula $P$ della logica dei predicati, è possibile rispondere "si" se $P$ è valida e "no" se $P$ non lo è? La risposta, a differenza di quanto accade nella logica proposizionale, è in questo caso negativa, e non nel senso che, allo stato dell'arte, non si conosce un procedimento di decisione per formule della logica dei predicati, ma nel senso ben più forte che si può dimostrare che tale procedimento *non può esistere*. Questo sconcertante risultato, dimostrato da Church nel 1936 [Chu36], si enuncia nel seguente modo:

**Teorema 5.25 (di Church)**
*Non esiste alcun algoritmo (metodo, procedura generale ...) che consente di decidere la validità di una qualunque formula della logica del primo ordine.*

La dimostrazione di tale teorema richiede la conoscenza di nozioni e risultati, propri della cosiddetta *Teoria della Calcolabilità*[8], che esulano dai scopi del presente volume. Effettuiamo, tuttavia, alcune considerazioni marginali su questo risultato.
Si è dunque visto che è possibile soltanto semi-decidere la validità di una qualunque formula della logica del primo ordine. Vediamo cosa si può dire sulla soddisfacibilità. Sappiamo che è possibile dare un'enumerazione effettiva di tutte le formule derivabili. Se fossimo anche in grado di enumerare in modo effettivo tutte le formule *non* derivabili, allora avremmo anche un algoritmo di decisione. Infatti, per decidere se una formula $P$ è derivabile (valida) o meno, basterebbe attivare i due algoritmi di enumerazione. In questo caso, si ha la sicurezza che $P$ comparirà dopo un tempo finito (quantunque grande) in una delle due liste e, a seconda dell'enumerazione in cui compare, si può decidere se è derivabile (valida) o meno. Inoltre, l'esistenza simultanea di due algoritmi di decisione per l'insieme delle formule derivabili e per quelle non derivabili è anche una condizione necessaria per la decidibilità dell'insieme delle espressioni derivabili. Infatti, avendo un algoritmo di decisione, e fissata una enumerazione di tutte le formule, le potremmo considerare consecutivamente decidendo per ognuna di esse se questa sia o meno derivabile. In questo modo si generano in modo effettivo le due successioni distinte delle formule derivabili e di quelle non

[8] Per una introduzione a tale teoria si vedano i libri di Rogers [Rog67] e Odifreddi [Odi89].

derivabili.
Il Teorema di Church asserisce per l'appunto che *non può esistere* un algoritmo in grado di enumerare in modo effettivo l'insieme delle formule non derivabili (non valide).
Sottolineamo il fatto che la non esistenza di un procedimento *generale* di decisione non preclude, tuttavia, la possibilità che per particolari sottoinsiemi di formule un tale procedimento possa esistere[9].
Ricordiamo che il problema della non validità di una formula è (computazionalmente) equivalente a quello della soddisfacibilità. Infatti, una formula $P$ è non valida se e solo se la sua negata $\neg P$ è soddisfacibile. Da cui segue che

**Corollario 5.26** *Il problema della soddisfacibilità di una qualunque formula della logica del primo ordine non è nemmeno semi-decidibile.*

$P$ ($\neg P$) è valida se e solo se $\neg P$ (rispettivamente, $P$) è contraddittoria (o, per completezza, insoddisfacibile). Dunque, l'insieme delle formule contraddittorie (o insoddisfacibili) è semi-decidibile.

### 5.4.2 Compattezza $\star$

Nella presente sezione discuteremo alcune delle possibili applicazioni del Teorema di completezza (forte), Teorema 5.14, alla *teoria dei modelli*; questa studia le relazioni esistenti tra gli insiemi di formule e le interpretazioni che li soddisfano. Uno dei principali risultati nella suddetta teoria è il Teorema di compattezza la cui dimostrazione è una diretta conseguenza del Teorema di completezza. Infatti:

**Teorema 5.27 (Compattezza)**
*Un insieme di fbf $\Gamma$ è soddisfacibile se e solo se ogni suo sottoinsieme finito lo è.*

Una sua formulazione equivalente (Esercizio 1.20) è la seguente:

**Teorema 5.28** *Un insieme di fbf $\Gamma$ è insoddisfacibile se e solo se esiste un suo sottoinsieme finito $\Delta$ che lo è.*

*Dimostrazione.* ($\Leftarrow$) Immediata.
($\Rightarrow$) Supponiamo che $\Gamma$ sia insoddisfacibile, per il Teorema del modello, $\Gamma$ è inconsistente, cioè $\Gamma \vdash \bot$. Allora esistono $P_0, \ldots, P_n \in \Gamma$ tali che $P_0, \ldots, P_n \vdash \bot$; per il Teorema di correttezza $P_0, \ldots, P_n \models \bot$, quindi $\{P_0, \ldots, P_n\}$ è insoddisfacibile. $\square$

Questo consente di provare un importante risultato, il Teorema di Löwenheim-Skolem, il quale mette in evidenza, tra le altre cose, i limiti del potere espressivo della logica del primo ordine (par. 5.4.3).

[9] Esistono classi assai vaste di formule della logica del primo ordine (si veda, a tal proposito, l'Esercizio 6.5 o il capitolo 46 di [Chu56]) per le quali è possibile stabilire se sono valide o meno.

Nel seguito utilizzeremo alcuni risultati elementari sui cardinali, le cui dimostrazioni si possono trovare in un qualunque testo di *teoria degli insiemi* (si veda, a tal proposito, la bibliografia consigliata nel paragrafo 5.6).

Per cardinalità di un'interpretazione $(\mathcal{A}, \xi^{\mathcal{A}})$ si intende la cardinalità del suo dominio, cioè $|D_{\mathcal{A}}|$. Diremo che $(\mathcal{A}, \xi^{\mathcal{A}})$ è finita o infinita, numerabile o non numerabile se $D_{\mathcal{A}}$ ha la proprietà corrispondente.

Sia $\Gamma$ un insieme di fbf definito su un linguaggio $\mathcal{L}$ del primo ordine. Siano $\overline{\Gamma}_\omega$ un insieme consistente massimale contenente $\Gamma_\omega$ (cfr. Proposizione 5.10) e $(\mathcal{A}_{\overline{\Gamma}_\omega}, \xi^{\mathcal{A}_{\overline{\Gamma}_\omega}})$ l'interpretazione canonica (Definizione 5.5).

**Proposizione 5.29** $(\mathcal{A}_{\overline{\Gamma}_\omega}, \xi^{\mathcal{A}_{\overline{\Gamma}_\omega}})$ *ha la cardinalità di* $\mathcal{L}$.

*Dimostrazione.* Sia $|\mathcal{L}| = \mu$. Occorre provare che $D_{\mathcal{A}_{\overline{\Gamma}_\omega}} = \{t \in TER(\mathcal{L}_\omega)\}$ ha cardinalità $\mu$. Per fare ciò si utilizzerà l'*assioma della scelta* nella seguente forma: siano $\mu$ e $\lambda$ cardinali infiniti, allora

$$\mu + \lambda = \mu \cdot \lambda = max(\mu, \lambda).$$

Dimostriamo, per induzione su $n$, che $\mathcal{L}_n$ ha cardinalità $\mu$, da cui segue che $\mathcal{L}_\omega$ ha cardinalità $\aleph_0 \cdot \mu = \mu$.
(*caso base*) $\mathcal{L}_0 = \mathcal{L}$. L'asserto è verificato per ipotesi. Supponiamo, per ipotesi induttiva, che $|\mathcal{L}_n| = \mu$, quindi, per ogni $k \in \mathcal{N}$, il numero di fbf di lunghezza $k$ formate con simboli in $\mathcal{L}_n$ è al più $\mu^k = \mu$. Dunque il numero totale di espressioni costruite con simboli di $\mathcal{L}$ è al più $\aleph_0 \cdot \mu = \mu$. Da ciò deriva immediatamente che il numero di testimoni aggiunti al passo $n+1$ è al più $\mu$ e quindi $|\mathcal{L}_{n+1}| = |\mathcal{L}_n| + \mu = \mu + \mu = \mu$. Abbiamo quindi provato che $\mathcal{L}_\omega$ ha cardinalità $\mu$, da cui segue, per l'Esercizio 5.17, che $|TER(\mathcal{L}_\omega)| = \mu$. □

**Teorema 5.30 (di Löweneim-Skolem "verso il basso")**
*Sia* $\mathcal{L}$ *un linguaggio di cardinalità* $\mu$. *Se* $\Gamma$ *è un insieme di fbf consistente, allora* $\Gamma$ *ha un modello di cardinalità* $\leq \mu$.

*Dimostrazione.* Segue dal Teorema del modello e dalla proposizione precedente. □

Si noti in particolare che un linguaggio del primo ordine su di un alfabeto numerabile ammette *sempre* un modello di cardinalità numerabile. Il caso è differente se si considerano linguaggi con uguaglianza, e se si richiede che il predicato sintattico di uguaglianza sia interpretato come *identità* sul modello. In questo caso (e solo in questo caso), determinati insiemi di formule ben formate possono ammettere solo modelli *finiti*. Se, tuttavia, si ammette che l'uguaglianza sia interpretata come una arbitraria congruenza (come imposto dalla sua assiomatizzazione), allora si avranno nuovamente, per necessità, modelli numerabili. Nel seguito, supporremo sempre che il predicato di uguaglianza sia interpretato come identità. Anche per linguaggi con uguaglianza vale tuttavia il risultato di Löweneim-Skolem:

**Teorema 5.31** *Sia $\mathcal{L}$ un linguaggio del primo ordine con uguaglianza di cardinalità $\mu$. Se $\Gamma$ è un insieme di fbf consistente, allora $\Gamma$ ha un modello di cardinalità $\leq \mu$.*

*Dimostrazione.* Segue dal Teorema del modello e dalla proposizione precedente, essendo l'insieme delle classi di equivalenza della relazione $\sim$ sui termini di un linguaggio di cardinalità $\mu$, sicuramente $\leq \mu$. $\square$

Dunque, ogni linguaggio numerabile con uguaglianza ammette modelli finiti o numerabili. Questo risultato può portare a conseguenze apparentemente paradossali. Consideriamo l'assiomatizzazione $ZF$ di Zermelo-Fraenkel della Teoria degli Insiemi (definita su di un alfabeto finito). Per il teorema precedente, $ZF$ ha un modello numerabile (per l'assioma dell'infinito, $ZF$ non può avere modelli finiti). In tale modello si ha una quantità numerabile di individui e dunque una quantità numerabile di "insiemi".
Tuttavia, il Teorema di Cantor, che afferma che esistono insiemi più che numerabili (ad esempio l'insieme $\mathcal{P}(\mathcal{N})$ dei sottoinsiemi di $\mathcal{N}$) è dimostrabile in $ZF$. Dunque tale teorema dovrà essere vero in ogni modello di $ZF$, in particolare in un modello numerabile la cui esistenza è stata affermata in precedenza. Non è strano che in un modello numerabile vi sia un elemento più che numerabile? Questa "stranezza" costituisce quello che è noto come *Paradosso di Skolem* (1992-23).
In realtà, non esiste una vera contraddizione. L'insieme di tutti i sottoinsiemi di $\mathcal{N}$ nel modello è in effetti numerabile, e dunque esiste una funzione da esso nell'insieme dei numeri naturali. Tuttavia questa funzione *non appartiene* al modello e dunque non contraddice il teorema che afferma che una tale funzione non può esistere (ed in effetti tale funzione non esiste *nel modello*). In altri termini, la visione del mondo all'interno del modello è differente da quella che si ha dall'esterno. Osserviamo inoltre che, benchè si possa dimostrare in un linguaggio logico di natura finitista come $ZF$ che esiste una infinità non numerabile di sottoinsiemi di un dato insieme, si è tuttavia solo in grado di *definire* una infinità al più *numerabile* di questi. La stessa cosa accade per i numeri reali: utilizzando un linguaggio numerabile, si può *definire*, ovviamente, solo una quantità numerabile di numeri reali. Non deve dunque soprendere che si possano costruire modelli numerabili di teorie che parlano di entità non numerabili.

**Teorema 5.32 (Löweneim-Skolem "verso l'alto")**
*Sia $\Gamma$ un insieme di fbf su un linguaggio $\mathcal{L}$ del primo ordine con uguaglianza. Se $\Gamma$ ha dei modelli infiniti, allora ha dei modelli non numerabili.*

*Dimostrazione.* Sia $(\mathcal{M}, \xi^{\mathcal{M}})$ un modello infinito di $\Gamma$ e $\mathcal{C}$ un insieme non numerabile di costanti tale che $\mathcal{C} \bigcap \mathcal{L} = \emptyset$. Sia $\mathcal{L}' = \mathcal{L} \cup \mathcal{C}$. Consideriamo l'insieme $\Gamma' = \Gamma \cup \{\neg(c_i = c_j) \mid i \neq j \ c_i, c_j \in \mathcal{C}\}$ e proviamo, utilizzando il Teorema di compattezza, che è soddisfacibile. È necessario dimostrare che ogni sottoinsieme finito $\Delta$ di $\Gamma'$ è soddisfacibile. Siccome $\Delta$ è finito conterrà solo un sottoinsieme finito di assiomi in $\{\neg(c_i = c_j) \mid i \neq j \ c_i, c_j \in \mathcal{C}\}$; in particolare, siano

$c_1, \ldots, c_n$ le costanti (necessariamente finite) che occorrono nelle fbf di $\Delta$. Essendo $(\mathcal{M}, \xi^{\mathcal{M}})$ infinito, è possibile fissare $n$ elementi distinti $a_1, \ldots, a_n$ in $D_{\mathcal{M}}$. Definiamo l'interpretazione $(\mathcal{M}_\Delta, \xi^{\mathcal{M}_\Delta})$ nel seguente modo: $D_{\mathcal{M}_\Delta} = D_{\mathcal{M}}$, per ogni simbolo di costante $c \in \mathcal{L}$, $c^{\mathcal{M}_\Delta} = c^{\mathcal{M}}$, per ogni simbolo di predicato $P$, $P^{\mathcal{M}_\Delta} = P^{\mathcal{M}}$, per ogni simbolo di funzione $f$, $f^{\mathcal{M}_\Delta} = f^{\mathcal{M}}$, $c_i^{\mathcal{M}_\Delta} = a_i$ per $1 \leq i \leq n$ e $c^{\mathcal{M}_\Delta} = a_1$ per ogni $c \in \mathcal{C}$ diversa da $c_1, \ldots, c_n$.
$(\mathcal{M}_\Delta, \xi^{\mathcal{M}_\Delta}) \models \Gamma$ in quanto $(\mathcal{M}, \xi^{\mathcal{M}}) \models \Gamma$ e $(\mathcal{M}, \xi^{\mathcal{M}})$ coincide con $(\mathcal{M}_\Delta, \xi^{\mathcal{M}_\Delta})$ su $\mathcal{L}$; poiché, inoltre, $a_1, \ldots, a_n$ sono elementi distinti di $D_{\mathcal{M}}$, $(\mathcal{M}_\Delta, \xi^{\mathcal{M}_\Delta}) \models \Delta$. Allora, per il Teorema di compattezza esiste un'interpretazione $(\mathcal{M}', \xi^{\mathcal{M}'})$ che è un modello per $\Gamma'$, e dunque per $\Gamma$. Poiché le costanti $c \in \mathcal{C}$ sono tutte distinte deve risultare $c_i^{\mathcal{M}'} \neq c_j^{\mathcal{M}'}$ per ogni $i \neq j$, e quindi $D_{\mathcal{M}'}$ deve avere almeno $|\mathcal{C}|$ elementi. Essendo $\mathcal{C}$ non numerabile si ha l'asserto, cioè esiste un modello per $\Gamma$ che non è numerabile. □

**Teorema 5.33 (di Löweneim, Skolem e Tarski)**
*Sia $\Gamma$ un insieme di fbf su un linguaggio $\mathcal{L}$ del primo ordine con uguaglianza. Se $\Gamma$ ha dei modelli infiniti allora ha dei modelli di cardinalità $\mathcal{K}$, con $\mathcal{K} \geq |\mathcal{L}|$.*

*Dimostrazione.* Sia $\mathcal{C}$ un insieme di costanti di cardinalità $\mathcal{K}$ tale che $\mathcal{C} \bigcap \mathcal{L} = \emptyset$. Definiamo l'insieme $\Gamma' = \Gamma \cup \{\neg(c_i = c_j) \mid i \neq j \; c_i, c_j \in \mathcal{C}\}$. Con considerazioni analoghe a quelle effettuate nella dimostrazione del precedente teorema, possiamo concludere che $\Gamma'$ è soddisfacibile. Per il Teorema di Löweneim-Skolem ha un modello $\mathcal{A}$ di cardinalitá $\leq \mathcal{K}$. Ma, per come è definito $\Gamma'$, ogni suo modello deve avere cardinalità $\geq \mathcal{K}$; da cui segue che $\mathcal{A}$ ha cardinalità $\mathcal{K}$. □

Vediamo qui di seguito alcune possibili applicazioni di tale teorema.

### 5.4.3 Modelli finiti ed infiniti ⋆

È possibile chiedersi se in un linguaggio del primo ordine si possano caratterizzare le interpretazioni numerabili mediante una fbf (o un insieme di fbf), i.e. se esiste una fbf, chiamiamola $P$, vera esattamente nelle interpretazioni numerabili; più formalmente, esiste $P \in FBF$ tale che $\{(\mathcal{A}, \xi^{\mathcal{A}}) \mid \mathcal{A} \models P\} = \{(\mathcal{A}, \xi^{\mathcal{A}}) \mid |D_{\mathcal{A}}|$ è numerabile $\}$?
Dal Teorema di Löweneim-Skolem segue immediatamente che tale formula non può esistere e dunque l'espressione "avere un insieme numerabile di elementi" non può essere formulata in un linguaggio del primo ordine, in altri termini, "numerabile" non è una proprietà del primo ordine. Dall'altro lato, "avere infiniti elementi" è invece una proprietà del primo ordine in quanto esistono formule vere esattamente nelle interpretazioni infinite. Infatti, se si considera ad esempio la fbf

$$\forall xy(f(x) = f(y) \rightarrow x = y) \wedge \neg\forall x \exists y f(y) = x$$

questa ha solo modelli infiniti in quanto non può esistere una funzione su insiemi finiti che sia allo stesso tempo iniettiva ma non suriettiva.
È possibile esprimere la proprietà "avere un numero finito di elementi"?
Dall'Esercizio 5.16 segue che esistono insiemi $\Gamma_1, \Gamma_2, \ldots$ di fbf tali che, $\forall n \in \mathcal{N}$,

$\Gamma_n$ è soddisfacibile in un'interpretazione se e solo se questa ha un numero di elementi $\geq n$; quindi le espressioni "avere almeno un elemento", "avere almeno 2 elementi", ... possono essere formulate in un linguaggio del primo ordine. Tuttavia non esiste una fbf (o un insieme di fbf) soddisfacibile in tutte e sole le interpretazioni finite.

Nel seguito utilizzeremo la seguente notazione:
sia $\Gamma$ un insieme di fbf, con $Mod(\Gamma)$ si indicherà l'insieme[10] delle interpretazioni che sono dei modelli per $\Gamma$, i.e.

$$Mod(\Gamma) = \{(\mathcal{A}, \xi^{\mathcal{A}}) \mid \mathcal{A} \models P, \quad \forall P \in \Gamma\}.$$

Viceversa, sia $\mathcal{K}$ un insieme di interpretazioni, si indicherà con $Teo(\mathcal{K})$ la teoria di $\mathcal{K}$, ossia

$$Teo(\mathcal{K}) = \{P \mid \mathcal{A} \models P, \quad \forall \mathcal{A} \in \mathcal{K}\}.$$

**Proposizione 5.34** *Sia $\Gamma$ un insieme di fbf. Se $\Gamma$ ha modelli finiti arbitrariamente grandi, allora ha un modello infinito.*

*Dimostrazione.* Sia $\Gamma' = \Gamma \cup \{\lambda_n \mid n > 1\}$ dove $\lambda_n$ esprime la proprietà "avere almeno $n$ elementi" (Esercizio 5.16). Proveremo, utilizzando il Teorema di compattezza, che $\Gamma'$ è soddisfacibile. Infatti, sia $\Delta \subseteq \Gamma'$ finito e sia $\lambda_m = max\ \{\lambda_n \mid n > 1 \wedge \lambda_n \in \Delta\}$. Poiché $\Gamma$ ha modelli finiti arbitrariamente grandi, avrà un modello $(\mathcal{A}, \xi^{\mathcal{A}})$ con almeno $m$ elementi e quindi $\mathcal{A} \models \Gamma \cup \{\lambda_m\}$; essendo $Mod(\Gamma \cup \{\lambda_m\}) \subset Mod(\Delta)$ (Esercizio 5.18), $\mathcal{A} \models \Delta$ e quindi $\Delta$ è soddisfacibile. Per il Teorema di compattezza $\Gamma'$ lo è e, per come sono definite le fbf $\lambda_n$, $\Gamma'$ ha un modello infinito, da cui $\Gamma$ ha un modello infinito. □

Dalla proposizione precedente segue che non può esistere alcuna fbf (o insieme di fbf) $P$ t.c. $Mod(P) = \{(\mathcal{A}, \xi^{\mathcal{A}}) \mid \mathcal{A} \models P\} = \{(\mathcal{A}, \xi^{\mathcal{A}}) \mid |D_{\mathcal{A}}| \text{ è finito}\}$. In conclusione, il potere espressivo della logica del primo ordine è limitato: "essere numerabile" o "essere finito" sono proprietà che non si possono formulare in essa.
È interessante osservare che tali proprietà sono invece esprimibili nella logica del secondo ordine (v. p.111). Infatti, sia Inf la seguente formula del secondo ordine:

$$\exists f[\forall xy(f(x) = f(y) \rightarrow x = y) \wedge \neg\forall x\exists z f(x) = z]$$

informalmente, questa è vera in un'interpretazione se e solo se esiste una funzione iniettiva ma non suriettiva, se e solo se l'interpretazione è infinita. La proprietà "essere finito" si esprime quindi al secondo ordine come $\neg Inf$.
Analogamente, sia $Num$ la seguente formula del secondo ordine:

$$\exists z f \forall A[A(z) \wedge \forall x(A(x) \rightarrow A(f(x))) \rightarrow \forall x A(x)]$$

questa è vera in un'interpretazione se e solo se è finita o numerabile.
Infatti, sia $\mathcal{M}$ un'interpretazione nella logica del secondo ordine avente come

[10] in realtà è una classe

dominio $D$. Intuitivamente $Num$ è vera in $\mathcal{M}$ se e solo se esistono $d \in D$ e $g : D \to D$ tali che, per tutti i $V \subseteq D$ se

1. $d \in V$ e
2. per tutti gli $x \in D$, se $x \in V$, allora $g(x) \in V$, implica che per tutti gli $x \in D$, $x \in V$.

($\Rightarrow$) Supponiamo che $\mathcal{M} \models Num$. Posto $V' = \{d, g(d), g(g(d)), \dots\}$, $V'$ soddisfa 1. e 2. Allora $D \subseteq V'$ e quindi $D$ è finito o numerabile.
($\Leftarrow$) Viceversa, supponiamo che $D$ sia finito o numerabile, allora $Num$ è vera in ogni interpretazione con dominio $D$.
Quindi la proprietà "essere numerabile" si può esprimere al secondo ordine come $Inf \wedge Num$.

***Osservazione*** I teoremi di compattezza, Löweneim-Skolem e molte altre proprietà della logica del primo ordine non sono vere in quella del secondo ordine.

### 5.4.4 Categoricità $\star$

Un'importante nozione della teoria dei modelli, che discuteremo qui di seguito, è quella della categoricità.

**Definizione 5.35** *Sia $\mathcal{L}$ un linguaggio del primo ordine e $(\mathcal{A}, \xi^{\mathcal{A}}), (\mathcal{A}', \xi^{\mathcal{A}'})$ due sue interpretazioni; queste sono* isomorfe *se esiste una corrispondenza biunivoca tra $D_{\mathcal{A}}$ e $D_{\mathcal{A}'}$ tale che:*

- *per ogni simbolo di funzione $f$ ad $n$ argomenti*

$$g(f^{\mathcal{A}}(b_1, \dots, b_n)) = f^{\mathcal{A}'}(g(b_1), \dots, g(b_n))$$

- *per ogni simbolo di predicato $P$ a $k$ argomenti*

$$g(P^{\mathcal{A}}(b_1, \dots, b_k)) = P^{\mathcal{A}'}(g(b_1), \dots, g(b_k))$$

Si noti che se $\mathcal{A}$ e $\mathcal{A}'$ sono isomorfe, i loro domini debbono avere la stessa cardinalità.

**Definizione 5.36** *Una* teoria *$T$ è un insieme di fbf chiuso per derivabilità, cioè $T \vdash P \Rightarrow P \in T$.*

**Definizione 5.37** *Una teoria $T$ definita su un linguaggio $\mathcal{L}$, si dice* completa *se è consistente e per ogni fbf* chiusa *$P \in \mathcal{L}$, $P \in T$ oppure $\neg P \in T$.*

Questa accezione del termine *completezza* non deve essere confusa con quella del Teorema di Completezza. Una teoria è completa se la nozione di "negazione" coincide con quella di "non derivabilità" (per formule chiuse). Infatti, se $\Gamma$ è completa $\Gamma \vdash \neg P$ se e solo se $\Gamma \nvdash P$. In generale, come sappiamo, la negazione

di $P$ esprime semplicemente la contraddittorietà di $P$, che è una cosa molto più debole: $\Gamma \vdash \neg P$ se e solo se $\Gamma, P \vdash \bot$. Osserviamo in particolare che ogni teoria completa le cui formule sono enumerabili in modo effettivo è anche decidibile rispetto alla validità, in quanto la non validità di $P$ si riduce al problema della derivabilità di $\neg P$.
Una caratterizzazione alternativa della completezza è la seguente:

**Proposizione 5.38** *Una teoria $T$ è completa se e solo se è consistente ed ogni formula chiusa $P \in T$ che è vera in un modello di $T$, lo è in ogni altro modello di $T$.*

*Dimostrazione.* ($\Rightarrow$) Sia $T$ una teoria completa e $P$ una formula chiusa di $T$ vera nel modello $(\mathcal{A}, \xi^{\mathcal{A}})$ di $T$. Proviamo che $P$ è vera in ogni altro modello di $T$. Supponiamo, per assurdo, che ciò non si verifichi, ossia che esiste un modello $(\mathcal{A}', \xi^{\mathcal{A}'})$ di $T$ tale che $(\mathcal{A}', \xi^{\mathcal{A}'}) \not\models P$; per il Teorema di correttezza $T \nvdash P$ da cui segue, essendo $T$ completa, che $T \vdash \neg P$ ma è assurdo.
($\Leftarrow$) Sia $T$ una teoria consistente e $P \in \mathcal{L}$. Per il Teorema 5.13 esiste un modello per $T$; in esso, è vera $P$ oppure $\neg P$. Per ipotesi $P$ o $\neg P$ sono vere in ogni modello di $T$, da cui segue, per il Teorema di completezza che $T \vdash P$ oppure $T \vdash \neg P$; essendo $T$ una teoria, $P \in T$ oppure $\neg P \in T$. $\square$

**Definizione 5.39** *Si dice* categorica *una teoria che ha un solo modello a meno di isomorfismi.*

Ovviamente una teoria categorica caratterizza in modo univoco (a meno di isomorfismi) una fissata interpretazione. Dal Teorema di Löweneim, Skolem e Tarski segue immediatamente che una teoria che ha modelli infiniti non può essere categorica. Dunque nessuna teoria potrà mai caratterizzare univocamente un'interpretazione infinita.
Ciò non è vero per interpretazioni finite, dove vale il seguente risultato (di cui non riportiamo la dimostrazione):

**Proposizione 5.40** *Data una qualunque interpretazione finita $\mathcal{A}$, allora $Teo(\mathcal{A})$ è categorica.*

Vale il seguente risultato:

**Teorema 5.41** *Ogni teoria categorica è completa.*

*Dimostrazione.* Sia $T$ una teoria categorica e $\mathcal{L}$ il linguaggio del primo ordine sul quale è definita $T$. Sia $P \in \mathcal{L}$; se $P \notin T$, allora $T \cup \{\neg P\}$ è consistente e quindi, per il Teorema 5.13, ha un modello. Essendo $T$ categorica, $T \models \neg P$ e quindi per il Teorema di completezza $T \vdash \neg P$; essendo $T$ una teoria, segue che $\neg P \in T$. $\square$

Esiste inoltre una nozione più ristretta di categoricità.

**Definizione 5.42** *Una teoria si dice* $\alpha$-categorica *se i suoi modelli di cardinalità $\alpha$ sono tra loro isomorfi.*

Per quanto concerne l'$\alpha$-categoricità di teorie di interpretazioni, la situazione è più articolata: vi sono teorie $\alpha$-categoriche per ogni cardinale infinito $\alpha$, teorie $\alpha$-categoriche per ogni[11] $\alpha$ non numerabile, teorie che lo sono solo per $\alpha = \aleph_0$ o teorie mai $\alpha$-categoriche. Esempi di teorie siffatte si possono trovare in [BM77]. La rilevanza della nozione di $\alpha$-categoricità per la completezza è mostrata dal seguente teorema

**Teorema 5.43** *(***di Łos-Vaught***)*
*Sia $T$ una teoria consistente che non ha modelli finiti ed è $\alpha$-categorica per un dato cardinale infinito $\alpha$, allora $T$ è completa.*

*Dimostrazione.* Supponiamo per assurdo che non lo sia, allora esiste una fbf $P$ tale che $P \notin T$ e $\neg P \notin T$, da cui segue che $T \cup \{P\}$ e $T \cup \{\neg P\}$ sono entrambi consistenti e quindi, per il Teorema del modello, hanno dei modelli che, per ipotesi, sono infiniti. Per il Teorema 5.33, hanno modelli di cardinalità $\alpha$. Essendo $P$ vera in uno dei due modelli ma non nell'altro, $T$ non è $\alpha$-categorica. Poiché l'unica assunzione fatta è che $T$ non è completa, segue che $T$ è una teoria completa. □

Nel paragrafo che segue mostreremo che l'aritmetica di Peano non è $\aleph_0$- categorica.

### 5.4.5 Modelli non standard dell'aritmetica ⋆

Consideriamo la formalizzazione dell'aritmetica descritta a p.116. L'usuale struttura dei naturali $\mathcal{N} = (\mathcal{N}, \xi^{\mathcal{N}})$ dove:

- $D_{\mathcal{N}}$ := insieme dei numeri naturali
- $=^{\mathcal{N}}$:= predicato di uguaglianza
- $0^{\mathcal{N}}$ := costante 0
- $s^{\mathcal{N}}$ := funzione successore
- $+^{\mathcal{N}}$ := funzione somma
- $*^{\mathcal{N}}$ := funzione moltiplicazione

è banalmente un modello per essa; $\mathcal{N}$ viene detto *modello standard* dell'aritmetica.

**Definizione 5.44** *Si definisce* modello non standard *dell'aritmetica un modello di PA non isomorfo ad $\mathcal{N}$.*

[11] Se una teoria è $\alpha$-categorica per un qualche $\alpha$ *non* numerabile, allora lo è per tutti gli $\alpha$ non numerabili.

Dal Teorema di Löweneim-Skolem segue che l'aritmetica di Peano possiede dei modelli non standard in quanto, essendo non numerabili, non possono essere isomorfi ad $\mathcal{N}$.
Esistono inoltre dei modelli non standard dell'aritmetica che, pur essendo numerabili, non sono isomorfi ad $\mathcal{N}$. Infatti:

**Proposizione 5.45** *Esiste un modello non standard dell'aritmetica che è numerabile.*

*Dimostrazione.* Sia $T$ l'insieme dei teoremi dell'aritmetica, ossia $T = \{P \mid PA \vdash P\}$ e $d$ una costante diversa da 0. Sia

$$\overline{n} \stackrel{def}{=} \underbrace{s(s(\dots s}_{n\ volte}(0)\dots))$$

Sia $T' = T \cup \{\neg(d = \overline{n}) \mid n \in \mathcal{N}\}$. Mostreremo, utilizzando il Teorema di compattezza, che $T'$, e di conseguenza $T$, ha un modello numerabile che non è isomorfo a $\mathcal{N}$. Infatti, sia $\Delta$ un sottoinsieme finito di $T'$, allora esiste $k \in \mathcal{N}$ tale che $\Delta \subseteq T \cup \{\neg(d = \overline{n}) \mid n < k\}$; $\Delta$ è soddisfacibile in quanto $(\mathcal{N}, \xi^{\mathcal{N}})$, posto $d^{\mathcal{N}} = n$, è un modello per esso. Per il Teorema di compattezza, $T'$ è soddisfacibile. Essendo il linguaggio di $T'$ numerabile, in quanto $\mathcal{L}(T') = \mathcal{L}(T) \cup \{d\}$, per il Teorema di Löweneim-Skolem avrà un modello numerabile $(\mathcal{N}^*, \xi^{\mathcal{N}^*})$; ma poiché tale interpretazione non è isomorfa ad $(\mathcal{N}, \xi^{\mathcal{N}})$ (il lettore lo dimostri per esercizio), si ricava l'asserto. □

Da cui segue immediatamente che:

**Corollario 5.46** *L'aritmetica non è $\aleph_0$-categorica.*

## 5.5 I Teoremi di Incompletezza di Gödel ⋆

Se una teoria è consistente, ammette un modello. Se, inoltre, è categorica il modello è unico (a meno di isomorfismi) e tutto ciò che è vero nel modello è anche dimostrabile.
Per la maggior parte dei logici degli anni '20, raggruppati attorno al cosiddetto "programma di Hilbert", la non contradditorietà di una teoria era già di per sé una condizione sufficiente a garantire l'esistenza "effettiva" degli enti di cui essa tratta. Per usare le parole di Hilbert:

> Se gli assiomi *arbitrariamente* scelti non contraddicono se stessi attraverso le loro conseguenze, allora essi sono *veri*, allora gli oggetti definiti da quegli assiomi *esistono*. Questo, per me, è il criterio di *verità* e di *esistenza*.

Il programma di Hilbert consisteva essenzialmente nel dimostrare con metodi finitistici la consistenza di sistemi logici sufficientemente espressivi da consentire una trattazione adeguata della matematica. Tale programma era rivolto sia

a contrastare l'emergente pensiero intuizionista/costruttivista (propugnato da L.E.J.Brower e H.Weyl), che a svincolare il delicato problema fondazionale della logica da ogni ricorso alla semantica, ed in particolare a strutture infinite. La consistenza della teoria (una nozione sintattica, dunque, e completamente interna al sistema logico) è condizione sufficiente a garantire l'esistenza degli enti di cui tratta. Non importa che la logica contenga al suo interno dei principi (terzo escluso, doppia negazione, etc.) di dubbia evidenza: la possibilità di dimostrare in modo finitistico, cioè mediante semplici argomenti combinatori o algoritmici la consistenza della teoria è anche condizione sufficiente a mostrare la natura del tutto "effettiva" delle strutture in oggetto.
Per citare nuovamente Hilbert,

> Invece dello sciocco "ignorabimus", noi ci rifacciamo al nostro motto:
> noi *dobbiamo* sapere, noi *vogliamo* sapere.

Affermazione che naturalmente sottindende, al punto da non meritare neppure di essere menzionata, l'ovvia presunzione che noi *possiamo* sapere.

Nel 1930, Gödel annunciò due risultati, noti sotto il nome di Teoremi di incompletezza, che distruggevano alle fondamenta l'ambizioso programma di Hilbert.

In termini estremamente informali, essi affermano quanto segue:

**Teorema 5.47 (Primo Teorema di Incompletezza)** *Sia $\mathcal{T}$ una qualunque teoria assiomatizzabile e sufficientemente espressiva da contenere al suo interno l'aritmetica. Allora esiste un sentenza*[12] *$\phi$ tale che, sotto opportune ipotesi di consistenza di $\mathcal{T}$, nè $\phi$ nè $\neg\phi$ è dimostrabile in $\mathcal{T}$.*

**Teorema 5.48 (Secondo Teorema di Incompletezza)** *Sia $\mathcal{T}$ una qualunque teoria assiomatizzabile e sufficientemente espressiva da contenere al suo interno l'aritmetica. Sia $Cons_{\mathcal{T}}$ una sentenza che asserisce la consistenza di $\mathcal{T}$. Allora, se $\mathcal{T}$ è consistente,*

$$\mathcal{T} \nvdash Cons_{\mathcal{T}}$$

Il secondo teorema asserisce dunque che la consistenza della teoria non è dimostrabile all'interno della teoria stessa (nè, in effetti, all'interno di nessun altra Teoria che soddisfi le ipotesi del teorema), distruggendo completamente il programma di Hilbert. Il primo teorema, inoltre, dimostra che ogni teoria sufficientemente espressiva è anche incompleta, creando un varco incolmabile tra la nozione semantica di validità e quella sintattica di derivabilità (ovvero, esistono formule vere in un qualche modello della teoria, ed in particolare in quello "inteso", e che non sono dimostrabili). Prima di dare alcuni cenni sulla dimostrazione dei Teoremi di incompletezza, discutiamo brevemente le loro ipotesi. La richiesta che si tratti di una teoria "sufficientemente espressiva da contenere l'aritmetica" è un eufemismo per evitare di esplicitare le ipotesi effettivamente richieste alla teoria che, come vedremo, sono essenzialmente ipotesi di codificabilità (o meglio rappresentabilità): in un certo senso, la teoria deve

[12] Ricordiamo che una sentenza è una formula chiusa del linguaggio.

essere sufficientemente espressiva da "poter parlare di se stessa". Ad esempio, sia l'Aritmetica di Peano (PA) che la Teoria Assiomatica degli Insiemi (ZF) soddisfano le ipotesi di Gödel.
L'assunzione che la teoria sia "assiomatizzabile" significa invece che esiste un numero finito di assiomi o di schemi di assiomi tali che $\mathcal{T}$ coincida con l'insieme delle conseguenze logiche di questi. Questa ipotesi è essenziale, altrimenti potremmo semplicemente considerare una certa struttura (ad esempio l'insieme dei numeri naturali) e prendere come Teoria l'insieme di tutte le formule vere su di questa (che per ovvie ragioni è sia consistente che completa). Dal punto di visto finitistico, l'ipotesi di assiomatizzabilità garantisce la natura effettiva del procedimento logico inferenziale, e svincola la fondazione del discorso logico dal ricorso ad una sottostante struttura semantica di dubbia evidenza.

Veniamo ora a considerare le idee principali per la dimostrazione dei suddetti teoremi.

La sentenza $\phi$ utilizzata da Gödel per la dimostrazione del suo primo Teorema è una formula che essenzialmente asserisce la propria indimostrabilità in $\mathcal{T}$. In altre parole

$$\phi \equiv \text{"io non sono dimostrabile in } \mathcal{T}''$$

La formalizzazione di questa idea si basa sui presupposti che:

- la teoria $\mathcal{T}$ deve essere sufficientemente espressiva da poter "parlare di se stessa", ovvero da poter essere utilizzata, entro certi limiti, come metalinguaggio di se stessa.

- deve essere possibile il fenomeno dell'autoriferimento, che fa sì che una formula asserisca qualche cosa di "se stessa" (per cui si userà una tecnica nota come diagonalizzazione).

Analizziamo separatamente i due aspetti.

Affinchè $\mathcal{T}$ possa "parlare di se stessa" è necessario *codificare* ampia parte della meta-teoria di $\mathcal{T}$ all'interno di $\mathcal{T}$ stessa. L'idea è quella di associare ad ogni termine $t$ e ad ogni formula $\phi$ del linguaggio un numero intero $\ulcorner t \urcorner$ e $\ulcorner \phi \urcorner$, detto numero di Gödel dell'oggetto in questione. Che questo sia possibile non dovrebbe essere particolarmente sorprendente per un informatico[13]. Basta ricordare che ogni tipo di dato di un linguaggio di programmazione (che sono sicuramente abbastanza espressivi da permettere di rappresentare termini, proposizioni, etc. ) è in ultima istanza rappresentato in memoria attraverso una qualche sequenza di *bit*, e questa sequenza può ovviamente essere letta come un intero.

Siccome $\mathcal{T}$ contiene l'aritmetica, per ogni numero intero $n$ esiste necessariamente un certo termine chiuso (numerale) $\underline{n}$ che lo rappresenta. Possiamo a questo punto definire opportune relazioni nel linguaggio di $\mathcal{T}$ per esprimere proprietà sugli oggetti sintattici. Ad esempio, è possibile definire una funzione

[13] Anzi, a dire il vero, è proprio da queste idee di Gödel, che costituiscono le principali basi teoriche per la manipolazione *simbolica* (e non solo numerica) dell'informazione, che nasce l'Informatica.

*Subs* tale che per ogni formula $\phi(x)$ che contiene un'unica variabile libera $x$, ed ogni intero $m$

$$\mathcal{T} \vdash Subs(\underline{\ulcorner \phi(x) \urcorner}, \underline{m}) = \underline{\ulcorner \phi(\underline{m}) \urcorner}$$

ovvero, $Subs(\underline{\ulcorner \phi(x) \urcorner}, \underline{m})$ è (dimostrabilmente) uguale al codice della formula ottenuta da $\phi(x)$ rimpiazzando la variabile $x$ con il numerale $\underline{m}$.
In modo analogo possiamo definire una codifica $\ulcorner D \urcorner$ per ogni dimostrazione $D$, e definire un predicato binario $Prov(x, y)$ tale che

$$\mathcal{T} \vdash Prov(\underline{m}, \underline{n})$$

se esistono $D$ e $\phi$ tali che $m = \ulcorner D \urcorner$, $n = \ulcorner \phi \urcorner$, e $D$ è una dimostrazione di $\phi$,

$$\mathcal{T} \vdash \neg Prov(\underline{m}, \underline{n})$$

in caso contrario. Ecco apparire la metateoria all'interno della teoria stessa (naturalmente, la nostra introduzione è *estremamente* informale: una trattazione dettagliata della codifica della metateoria all'interno della teoria richiederebbe, da sola, una ventina di pagine).

Poniamo ora

$$Teor(x) := \exists y Prov(y, x)$$

Le relazioni tra $\mathcal{T} \vdash \phi$ e $\mathcal{T} \vdash Teor(\underline{\ulcorner \phi \urcorner})$ sono riassunte dalle seguenti *Condizioni di Derivabilità*, la cui dimostrazione è omessa.

**Proposizione 5.49**

1. *Se* $\mathcal{T} \vdash \phi$ *allora* $\mathcal{T} \vdash Teor(\underline{\ulcorner \phi \urcorner})$.
2. $\mathcal{T} \vdash Teor(\underline{\ulcorner \phi \urcorner}) \rightarrow Teor(\underline{\ulcorner Teor(\underline{\ulcorner \phi \urcorner}) \urcorner})$
3. $\mathcal{T} \vdash Teor(\underline{\ulcorner \phi \urcorner}) \wedge Teor(\underline{\ulcorner \phi \rightarrow \psi \urcorner}) \rightarrow (\underline{\ulcorner \psi \urcorner})$

Veniamo ora al problema della "sincretizzazione". Il risultato è del tutto generale, e si basa su di una tecnica nota come *diagonalizzazione*.

**Teorema 5.50 (Diagonalizzazione)** *Sia* $\phi(x)$ *una formula che contiene* $x$ *come unica variabile libera. Allora esiste una formula chiusa* $\psi$ *tale che*

$$\mathcal{T} \vdash \psi \leftrightarrow \phi(\underline{\ulcorner \psi \urcorner}).$$

*Dimostrazione.* Sia $m = \ulcorner \phi(Subs(x, x)) \urcorner$, e prendiamo $\psi = \phi(Subs(\underline{m}, \underline{m}))$. Abbiamo allora:

$$\begin{array}{lll}
\psi & \leftrightarrow & \phi(Subs(\underline{m}, \underline{m})) \\
& \leftrightarrow & \phi(\underline{Subs(\underline{\ulcorner \phi(Subs(x, x)) \urcorner}, \underline{m})}) \quad \text{per definizione di } m \\
& \leftrightarrow & \phi(\underline{\ulcorner \phi(Subs(\underline{m}, \underline{m})) \urcorner}) \quad \text{per definizione di } Subs \\
& \leftrightarrow & \phi(\underline{\ulcorner \psi \urcorner})
\end{array}$$

□

### 5.5.1 Primo Teorema di Incompletezza

Dobbiamo ancora introdurre la nozione di $\omega$-consistenza, utilizzata da Gödel per enunciare il suo primo teorema. Negli enunciati che seguono supporremo che $\mathcal{T}$ sia una teoria sufficientemente espressiva da contenere l'aritmetica.

**Definizione 5.51** *$T$ è* $\omega$-consistente *se e solo se per ogni formula $\phi(x)$, il fatto che $\mathcal{T} \vdash \phi(\underline{n})$ per ogni intero $n$ implica che $\mathcal{T} \nvdash \exists x \neg\phi(x)$.*

**Proposizione 5.52** *Se $\mathcal{T}$ è $\omega$-consistente allora $\mathcal{T}$ è consistente.*

*Dimostrazione.* $\mathcal{T} \vdash (\underline{n} = \underline{n})$ per ogni numerale $\underline{n}$. Quindi $\mathcal{T} \nvdash \exists x(x \neq x)$ e dunque $\mathcal{T}$ è consistente perché esiste una formula che non è dimostrabile in $\mathcal{T}$. □

**Proposizione 5.53** *Sia $\mathcal{N}$ l'insieme dei numeri naturali (cioè il modello* standard *dell'aritmetica). Se $\mathcal{N} \models \mathcal{T}$, allora $\mathcal{T}$ è $\omega$-consistente.*

*Dimostrazione.* Sia $\phi$ una generica fbf, e si supponga che, per ogni numerale $\underline{n}$, $\mathcal{T} \vdash \phi(\underline{n})$. Per il Teorema di correttezza, per ogni numerale $\underline{n}$, $\mathcal{N} \models \phi(\underline{n})$. Supponiamo per assurdo che $\mathcal{T} \vdash \exists x(\neg\phi(x))$. Allora $\mathcal{N} \models \exists x(\neg\phi(x))$, cioè esisterebbe un numerale $\underline{n}$ tale che $\mathcal{N} \models \neg\phi(\underline{n})$ che conduce ad una contraddizione. □

**Proposizione 5.54** *Sia $\mathcal{T}$ $\omega$-consistente. Se $\mathcal{T} \vdash Teor(\underline{\ulcorner\phi\urcorner})$ allora $\mathcal{T} \vdash \phi$.*

*Dimostrazione.* Supponiamo che $\mathcal{T} \nvdash \phi$. Dunque, per ogni $n$, $\mathcal{T} \vdash \neg Prov(\underline{n}, \underline{\ulcorner\phi\urcorner})$ (altrimenti $n$ sarebbe il codice di una dimostrazione di $\phi$). Per $\omega$-consistenza si ha allora $\mathcal{T} \nvdash \exists x(\neg\neg Prov(x, \underline{\ulcorner\phi\urcorner}))$ ovvero $\mathcal{T} \nvdash \exists x(Prov(x, \underline{\ulcorner\phi\urcorner}))$, e dunque $\mathcal{T} \nvdash Teor(\underline{\ulcorner\phi\urcorner})$. □

**Teorema 5.55 (I Teorema di incompletezza di Gödel)** *Data una teoria $\mathcal{T}$ assiomatizzabile e sufficientemente espressiva da contenere l'aritmetica, è possibile definire una sentenza $G$ tale che*

1. *se $\mathcal{T}$ è consistente, $\mathcal{T} \nvdash G$;*
2. *se $\mathcal{T}$ è $\omega$-consistente, $\mathcal{T} \nvdash \neg G$.*

*Dimostrazione.* Consideriamo la formula $\neg Teor(x)$. Per il Teorema di diagonalizzazione esiste $G$ tale che $G \leftrightarrow \neg Teor(\underline{\ulcorner G\urcorner})$. Dunque:

$$(*) \qquad \mathcal{T} \vdash G \Leftrightarrow \mathcal{T} \vdash \neg Teor(\underline{\ulcorner G\urcorner})$$

1. Se $\mathcal{T} \vdash G$, allora per la Proposizione 5.49.1, $\mathcal{T} \vdash Teor(\underline{\ulcorner G\urcorner})$, e dunque per $(*)$ $\mathcal{T} \vdash \neg G$. Questo contraddirrebbe l'ipotesi di consistenza di T.

2. Se $\mathcal{T} \vdash \neg G$; allora per $(*)$ $\mathcal{T} \vdash Teor(\underline{\ulcorner G\urcorner})$. Dalla Proposizione 5.54 abbiamo anche $\mathcal{T} \vdash G$, dunque $\mathcal{T}$ sarebbe inconsistente (contraddicendo l'ipotesi di $\omega$-consistenza e a fortiori quella di consistenza).

□

Dunque, nessuna teoria logica assiomatizzabile e sufficientemente espressiva da ambire a rappresentare la matematica può essere completa. In particolare, la nozione di *verità aritmetica* differisce inevitabilmente dalla nozione di *dimostrabilità* all'interno della teoria. Osserviamo in particolare che non è possibile ovviare al problema di Gödel semplicemente aggiungendo $G$ agli assiomi di $\mathcal{T}$. Infatti, la teoria così ottenuta soddisfarrebbe ancora le ipotesi del teorema, e dunque potremmo trovare una *nuova* formula $G'$ che mostrerebbe l'incompletezza della teoria estesa.

### 5.5.2 Secondo Teorema di Incompletezza

Tra le formule che testimoniamo l'incompletezza di $\mathcal{T}$, esiste anche quella che afferma la consistenza di $\mathcal{T}$ stessa. Il primo problema è ovviamente quello di formalizzare la nozione di consistenza. Ma questo è semplice: basta affermare che, presa una arbitraria formula contraddittoria (ad esempio $\bot$), questa non è un teorema di $\mathcal{T}$.

**Definizione 5.56** $Cons \equiv \neg Teor(\underline{\ulcorner \bot \urcorner})$

Per dimostrare che $\mathcal{T} \nvdash Cons$, possiamo "formalizzare" il primo Teorema di Gödel. L'affermazione "se $\mathcal{T}$ è consistente, $\mathcal{T} \nvdash G$" può infatti essere codificata come $Cons \rightarrow \neg Teor(\underline{\ulcorner G \urcorner})$. Ma per definizione $G \equiv \neg Teor(\underline{\ulcorner G \urcorner})$ e dunque, il nostro asserto è: $Cons \rightarrow G$. Se ora riusciamo a dimostrare questa formula in $\mathcal{T}$, cioè mostrare che $\mathcal{T} \vdash Cons \rightarrow G$, otteniamo il risultato voluto: $\mathcal{T} \nvdash Cons$, poiché altrimenti, per modus ponens, si avrebbe anche $\mathcal{T} \vdash G$, contraddicendo il primo Teorema di incompletezza. In effetti, è addirittura possibile dimostrare che le due formule $Cons$ e $G$ sono dimostrabilmente *equivalenti* in $\mathcal{T}$.

**Teorema 5.57 (II Teorema di incompletezza)** *Se $\mathcal{T}$ è consistente, $\mathcal{T} \nvdash Cons$.*

*Dimostrazione.* Sia $G$ la formula di Gödel che asserisce la propria indimostrabilità. Ricordiamo che

$$(**) \qquad \mathcal{T} \vdash G \leftrightarrow \neg Teor(\underline{\ulcorner G \urcorner})$$

Dimostriamo che

$$\mathcal{T} \vdash G \leftrightarrow Cons$$

La tesi segue allora dal primo teorema. Dimostriamo separatamente i due versi dell'implicazione.

- $\mathcal{T} \vdash G \rightarrow Cons$.
  $\bot \rightarrow G$ è una tautologia, dunque è dimostrabile in $\mathcal{T}$, e dunque $\mathcal{T} \vdash Teor(\underline{\ulcorner \bot \rightarrow G \urcorner})$. Per semplici manipolazioni logiche, utilizzando le proprietà di $Teor$ enunciate nella Proposizione 5.49, si ottiene

$$\mathcal{T} \vdash Teor(\underline{\ulcorner \bot \urcorner}) \rightarrow Teor(\underline{\ulcorner G \urcorner})$$

e dunque, per contrapposizione,

$$\mathcal{T} \vdash \neg Teor(\underline{\ulcorner G \urcorner}) \rightarrow \neg Teor(\underline{\ulcorner \bot \urcorner})$$

Per $(**)$, si ha infine

$$\mathcal{T} \vdash G \rightarrow \neg Teor(\underline{\ulcorner \bot \urcorner})$$

- $\mathcal{T} \vdash Cons \rightarrow G$.

Da $(**)$ si ottiene immediatamente che $\mathcal{T} \vdash \neg G \leftrightarrow Teor(\underline{\ulcorner G \urcorner})$. In particolare $\mathcal{T} \vdash Teor(\underline{\ulcorner G \urcorner}) \rightarrow \neg G$, dunque

$$\mathcal{T} \vdash Teor(\underline{\ulcorner Teor(\underline{\ulcorner G \urcorner}) \rightarrow \neg G \urcorner})$$

Per semplici manipolazioni logiche, utilizzando le proprietà della Proposizione 5.49, abbiamo anche

$$\mathcal{T} \vdash Teor(\underline{\ulcorner Teor(\underline{\ulcorner G \urcorner}) \urcorner}) \rightarrow Teor(\underline{\ulcorner \neg G \urcorner})$$

Poichè inoltre, per ogni $\phi$, $\mathcal{T} \vdash Teor(\underline{\ulcorner \phi \urcorner}) \rightarrow Teor(\underline{\ulcorner Teor(\underline{\ulcorner \phi \urcorner}) \urcorner})$, si ottiene

$$\mathcal{T} \vdash Teor(\underline{\ulcorner G \urcorner}) \rightarrow Teor(\underline{\ulcorner \neg G \urcorner})$$

Sempre utilizzando le proprietà di $Teor$ è facile ricavare che

$$\mathcal{T} \vdash Teor(\underline{\ulcorner G \urcorner}) \rightarrow Teor(\underline{\ulcorner G \wedge \neg G \urcorner})$$

Ma $G \wedge \neg G \rightarrow \bot$ e internalizzando questa prova otteniamo

$$\mathcal{T} \vdash Teor(\underline{\ulcorner G \urcorner}) \rightarrow Teor(\underline{\ulcorner \bot \urcorner})$$

Per contrapposizione

$$\mathcal{T} \vdash Teor(\underline{\ulcorner \bot \urcorner}) \rightarrow \neg Teor(\underline{\ulcorner G \urcorner})$$

L'asserto segue da $(**)$.

□

### 5.5.3 Teoremi di Tarski e Church

Abbiamo visto che se una teoria è sufficientemente espressiva, la nozione di "dimostrabilità" è, (debolmente) internalizzabile. In particolare, esiste una formula $Teor(x)$ tale che $\mathcal{T} \vdash \phi \Rightarrow \mathcal{T} \vdash Teor(\underline{\ulcorner \phi \urcorner})$, per ogni sentenza $\phi$.
Che dire della nozione di verità? Quello che vorremmo è un predicato $Vero(x)$, tale che, per ogni formula $\phi$, $Vero(\underline{\ulcorner \phi \urcorner})$ sia valido in un generico modello se e solo se $\phi$ lo è. Per completezza, questo equivale a richiedere che $Vero(\underline{\ulcorner \phi \urcorner})$ sia dimostrabilmente equivalente a $\phi$ all'interno della teoria.

**Definizione 5.58** *Una* nozione di verità *per una teoria $\mathcal{T}$ è una formula $Vero(x)$ dove $x$ è la sola variabile libera, tale che, per ogni sentenza $\phi$ di $\mathcal{T}$,*

$$\mathcal{T} \vdash Vero(\underline{\ulcorner \phi \urcorner}) \leftrightarrow \phi$$

**Teorema 5.59 (di Tarski)** *Se $\mathcal{T}$ è consistente non ammette nozioni di verità.*

*Dimostrazione.* Supponiamo per assurdo che esista una formula $Vero(x)$, tale che $\mathcal{T} \vdash Vero(\underline{\ulcorner \phi \urcorner}) \leftrightarrow \phi$ (e dunque $\mathcal{T} \vdash \neg Vero(\underline{\ulcorner \phi \urcorner}) \leftrightarrow \neg\phi$).
Consideriamo la formula $\neg Vero(x)$. Per il Teorema di diagonalizzazione esiste una sentenza $\psi$ tale che $\mathcal{T} \vdash \psi \leftrightarrow \neg Vero(\underline{\ulcorner \psi \urcorner})$. Dunque avremmo $\mathcal{T} \vdash \psi \leftrightarrow \neg Vero(\underline{\ulcorner \psi \urcorner}) \leftrightarrow \neg\psi$, che dimostra la contraddittorietà di $\mathcal{T}$. $\square$

Si noti che l'idea della dimostrazione di Tarski è esattamente quella del celebre paradosso del mentitore, in una delle sue tante possibili forme: "io sto mentendo", "questa frase è falsa", etc.

Osserviamo inoltre che nell'enunciato del Teorema di Tarski *non si richiede* che $\mathcal{T}$ sia assiomatizzabile. Dunque, una volta fissato un certo linguaggio $\mathcal{L}$, possiamo definire $\mathcal{T}$ come l'insieme di tutte le formule di $\mathcal{L}$ vere di un determinato modello, ed in particolare nel modello standard $\mathcal{N}$ dell'aritmetica. Come conseguenza del Teorema di Tarski abbiamo che non esiste nessuna formula $Vero(x)$ in $\mathcal{L}$ tale che per ogni sentenza $\phi$

$$\mathcal{N} \models \phi \Leftrightarrow \mathcal{N} \models Vero(\phi)$$

Questo corollario viene abitualmente enunciato dicendo che *la nozione di verità aritmetica non è aritmetica.*

Un altro esempio interessante di nozione non internalizzabile è quella di *non-dimostrabilità.* In questo caso vorremmo avere una formula $NonTeor(x)$ tale che

$$\mathcal{T} \nvdash \phi \Rightarrow \mathcal{T} \vdash NonTeor(\underline{\ulcorner \phi \urcorner})$$

per ogni sentenza $\phi$.
Supponiamo che tale sentenza esista. Applicando la solita tecnica di diagonalizzazione avremmo, per una certa sentenza $\psi$,

$$\mathcal{T} \nvdash \psi \Rightarrow \mathcal{T} \vdash NonTeor(\underline{\ulcorner \psi \urcorner}) \Leftrightarrow \mathcal{T} \vdash \psi$$

che è una ovvia contraddizione.
In effetti, questa è anche la strada per dimostrare il più volte enunciato Teorema di Curch, in quanto è possibile dimostrare che ogni proprietà semi-decidibile di formule è debolmente internalizzabile nel senso suesposto. Dunque, la non-dimostrabilità non è semi-decidibile, e per la discussione del paragrafo 5.4.1, la dimostrabilità non è decidibile.

## 5.6 Cenni storici e bibliografici

Il Teorema di completezza (per linguaggi numerabili) è dovuto a Gödel ([God30]). La dimostrazione presentata in 5.1.1 è quella di Henkin [Hen49], semplificata da Hasenjaeger ([Has53]); questa può essere facilmente estesa a linguaggi di qualunque cardinalità. Altre dimostrazioni del Teorema di completezza sono state realizzate, tra gli altri, da Rasiowa-Sikorski ([RS51]) e Beth ([Bet51]) facendo

uso, rispettivamente, di metodi algebrici (booleani) e topologici. Il Teorema di compattezza per linguaggi numerabili è una conseguenza diretta del teorema di completezza. La sua estensione a linguaggi non numerabili è dovuta a Malcev [Mal36]. Esistono numerosi testi che si occupano di teoria dei modelli. I risultati discussi nel paragrafo 5.4 si possono trovare nei libri di livello intermedio [VDa80, **?**, **?**]. Libri avanzati sull'argomento sono il "classico" Chang e Keisler [CK73], Bell e Machover [BM77] e Shoenfield [Sho67].
Per una panoramica sulla teoria dei cardinali si veda [Kun80]. Sui teoremi di incompletezza di Gödel si vedano [Smo77, Smu92].

## Esercizi

**5.1** Dimostrare in un sistema deduttivo che:

1. $\vdash \exists x P(x) \to \exists y P(y)$
2. $\vdash \forall x P(x) \to \neg\exists x \neg P(x)$
3. $\vdash \neg\exists x \neg P(x) \to \forall x P(x)$
4. $\vdash \neg\forall x P(x) \to \exists x \neg P(x)$
5. $\vdash \exists x \neg P(x) \to \neg\forall x P(x)$
6. $\vdash \neg\exists x P(x) \to \forall x \neg P(x)$
7. $\vdash \forall x \neg P(x) \neg\exists x P(x)$
8. $\vdash \forall x (P(x) \vee Q) \to \forall x P(x) \vee Q$ se $x \notin FV(Q)$
9. $\vdash \exists x (P \to Q(x)) \to (P \to \exists x Q(x))$ se $x \notin FV(Q)$
10. $\vdash (P \to \exists x Q(x)) \to \exists x (P \to Q(x))$ se $x \notin FV(Q)$
11. $\exists x y P(x) \to \exists y x P(x)$
12. $\exists y x P(x) \to \exists x y P(x)$

**5.2** Definire l'interpretazione canonica per i seguenti insiemi di fbf:

1. $\Gamma_1 = \{\forall x A(x), \exists x C(x), A(a)\}$
2. $\Gamma_2 = \{\forall x A(x), \exists x C(x), A(a), C(b), A(a) \to B(x)\}$
3. $\Gamma_3 = \{\forall x A(x), \exists x C(x), A(a), C(b), A(a) \to B(x), B(a)\}$

Per quali di questi è un modello?

**5.3** Sia $\Gamma = \{\neg\forall x P(x), P(x_1), P(x_2), \dots\}$. $\Gamma$ è consistente? È soddisfacibile?

**5.4** Sia $\Gamma$ un insieme consistente massimale e di Henkin. Provare che $\exists x P(x) \in \Gamma$ se e solo se esiste un termine $t$ tale che $P[^t/_x] \in \Gamma$.

**5.5** Sia $\Phi$ un insieme di fbf. Dimostrare che $\Phi$ è inconsistente:

1. se esiste $P$ tale che $\neg\neg P \in \Phi$ e $\Phi, P$ è inconsistente;
2. se esistono $P, x$ e $t$ tali che $\forall x P \in \Phi$ e $\Phi, P[^t/_x]$ è inconsistente;
3. se esistono $P, x$ ed $y$ tali che $\neg\forall x P \in \Phi$, $y$ non è libera in $\Phi$ e $\Phi, \neg P[^y/_x]$ è inconsistente.

**5.6** Sia $(\mathcal{A}_\Gamma, \xi^{\mathcal{A}_\Gamma})$ l'interpretazione canonica definita in 5.5, provare che per ogni termine $t$ di $\Gamma$, $[\![t]\!]_\xi^{\mathcal{A}_\Gamma} = t$. Ciò risulta vero anche in un linguaggio del primo ordine con uguaglianza? Motivare la risposta.

**5.7** Sia $T$ una teoria (cfr. Definizione 5.36) e $T_\omega$ l'insieme definito in 5.10; provare che $T_\omega$ è una teoria.

**5.8** Siano $\{T_i \mid i \in I\}$ un insieme di teorie totalmente ordinato per inclusione, i.e. $\forall i, j \in I \quad T_i \subseteq T_j$ oppure $T_j \subseteq T_i$, e $T = \bigcup_{i \in I} T_i$; Provare che se $T_i$ è consistente $\forall i \in I$, allora $T$ lo è.

**5.9** Sia $\Gamma$ un insieme finito di fbf; dimostrare che se $\Gamma \vdash P$, allora $\Gamma[^y/_c] \vdash P[^x/_c]$ dove $x$ è una variabile che non occorre in $\Gamma$ o in $P$.
(Suggerimento: si utilizzi l'induzione sulla derivazione).

**5.10** Provare che ogni insieme di fbf soddisfacibile è consistente.

**5.11** Siano $T$ e $T'$ due teorie definite rispettivamente sui linguaggi $\mathcal{L}$ e $\mathcal{L}'$; si dice che $T'$ è un'*estensione conservativa* di $T$ se $T \subseteq T'$ e $T' \bigcap \mathcal{L} = \emptyset$, cioè tutti i teoremi di $T'$ nel linguaggio $\mathcal{L}$ sono teoremi di $T$.
Sia $T$ una teoria e $T^*$ la teoria ottenuta applicando la Definizione 5.8, provare che $T^*$ è conservativa su $T$.

**5.12** Per ognuno dei seguenti sequenti esibire una prova oppure fornire un'interpretazione che lo falsifica:

1. $\forall x A(x) \vdash \exists x A(x)$
2. $\forall x(A(x) \wedge B(x)) \vdash \forall x A(x) \wedge \forall x B(x)$
3. $\neg\exists x A(x) \vdash \exists x \neg A(x)$
4. $\forall x y C(x, y) \vdash \forall y x C(x, y)$
5. $\exists x \forall y C(x, y) \vdash \forall y \exists x C(x, y)$
6. $\forall x y(C(x, y) \rightarrow \neg C(y, x)) \vdash \forall x \neg C(x, x)$
7. $\forall x((\exists y C(x, y)) \rightarrow A(x)) \vdash \forall x \exists y(C(x, y) \rightarrow A(x))$
8. $\forall x((\forall y C(x, y)) \rightarrow A(x)) \vdash \forall x \exists y(C(x, y) \rightarrow A(x))$
9. $\forall x((\exists y C(x, y)) \rightarrow A(x)) \vdash \forall x \forall y(C(x, y) \rightarrow A(x))$

10. $\forall x((\forall y C(x,y)) \rightarrow A(x)) \vdash \forall x \forall y(C(x,y) \rightarrow A(x))$

**5.13** Mostrare che $\{A(a), \neg A(b)\}$ è soddisfacibile in un'interpretazione $(\mathcal{A}, \xi^{\mathcal{A}})$ se e solo se $D_{\mathcal{A}}$ contiene almeno due elementi.

**5.14** Trovare un insieme di fbf soddisfacibile in un'interpretazione $(\mathcal{A}, \xi^{\mathcal{A}})$ se e solo se $D_{\mathcal{A}}$ contiene almeno tre elementi.

**5.15** Si consideri la formula $\forall x A(x,x) \wedge \forall xyz[(A(x,y) \wedge A(y,z)) \rightarrow A(x,z)] \wedge \forall xy[A(x,y) \vee A(y,x)] \rightarrow \exists y \forall x A(y,x)$

1. Mostrare che ogni interpretazione che ha un dominio finito è un modello per essa.

2. Trovare un'interpretazione che non ne è un modello.

**5.16** Siano $P_n$ e $Q_n$, con $n > 1$, le seguenti fbf:

$$P_n = \exists x_1 \ldots x_n \bigwedge_{i \neq j} x_i \neq x_j$$

$$Q_n = \forall x_1 \ldots x_n \bigvee_{i \neq j} x_i = x_j$$

Provare che:

1. $Mod(P_n) = \{(\mathcal{A}, \xi^{\mathcal{A}}) \mid |D_{\mathcal{A}}| \geq n\}$
2. $Mod(Q_n) = \{(\mathcal{A}, \xi^{\mathcal{A}}) \mid |D_{\mathcal{A}}| \leq n\}$
3. $Mod(P_n \wedge Q_n) = \{(\mathcal{A}, \xi^{\mathcal{A}}) \mid |D_{\mathcal{A}}| = n\}$
4. $Mod(\{P_n \mid n > 1\}) = \{(\mathcal{A}, \xi^{\mathcal{A}}) \mid |D_{\mathcal{A}}|$ è infinito $\}$

**5.17** Sia $\mu$ un cardinale infinito; provare che se $|\mathcal{L}| = \mu$ allora $|Ter(\mathcal{L})| = \mu$.

**5.18** Dimostrare che:

1. $\Delta \subseteq \Gamma \Rightarrow Mod(\Gamma) \subseteq Mod(\Delta)$
2. $\mathcal{K}_1 \subseteq \mathcal{K}_2 \Rightarrow Teo(\mathcal{K}_2) \subseteq Teo(\mathcal{K}_1)$
3. $Mod(\Delta \cup \Gamma) = Mod(\Gamma) \bigcap Mod(\Delta)$
4. $Teo(\mathcal{K}_1 \cup \mathcal{K}_2) = Teo(\mathcal{K}_2) \bigcap Teo(\mathcal{K}_1)$

**5.19** Siano $(\mathcal{A}, \xi^{\mathcal{A}})$ e $(\mathcal{A}', \xi^{\mathcal{A}'})$ due interpretazioni isomorfe. Verificare che, per ogni fbf $P$, $\mathcal{A} \models P$ se e solo se $\mathcal{A}' \models P$.

**5.20** Sia $(\mathcal{A}, \xi^{\mathcal{A}})$ un'interpretazione e $D$ un insieme avente la stessa cardinalità di $D_{\mathcal{A}}$; mostrare che è possibile definire un'interpretazione avente $D$ come dominio ed isomorfa ad $(\mathcal{A}, \xi^{\mathcal{A}})$.

**5.21** Sia $(\mathcal{A}, \xi^{\mathcal{A}})$ un'interpretazione, provare che $Teo(\mathcal{A})$ è completa.

# Capitolo 6

# Metodo di Risoluzione

Nel capitolo 5 è stato provato che l'insieme delle formule valide della logica del primo ordine coincide con quello delle formule dimostrabili in uno dei calcoli discussi in precedenza: Deduzione Naturale, Sistemi Assiomatici e Calcolo dei Sequenti. Si è inoltre presentato un algoritmo in grado di trovare una prova nel Calcolo dei Sequenti per ogni formula valida.
In questo capitolo discuteremo due metodi alternativi per provare la validità di una formula del primo ordine, che risultano particolarmente semplici ed intuitivi: il Teorema di Herbrand ed il Metodo di Risoluzione. Il primo, di natura semantica, ha una grande importanza storica, mentre il secondo, di natura sintattica, costituisce la base della programmazione logica e del Prolog. Entrambi procedono in modo indiretto (per *refutazione*), vale a dire che per provare la validità di una formula dimostrano che la sua negazione è insoddisfacibile. Tali metodi, così come l'algoritmo mostrato in 5.3.2, consentono di stabilire se una data formula è valida ma *non* se non lo è; ricordiamo che (paragrafo 5.4.1), ciò non dipende dai particolari metodi considerati, ma è un limite intrinseco della logica del primo ordine.

## 6.1 Teoria di Herbrand

Come già accennato nell'introduzione, per dimostrare che una formula $P$ è valida, è sufficiente provare che $\neg P$ è insoddisfacibile, vale a dire che $\neg P$ è falsa in ogni interpretazione. In logica proposizionale, essendo finito il numero di possibili interpretazioni di una formula, la verifica di ciò non costituisce un problema. Nella logica del primo ordine, invece, avendo a che fare con strutture che ammettono insiemi arbitrari come possibili domini, per stabilire se una formula è insoddisfacibile, è impossibile procedere considerando tutte le interpretazioni su ogni possibile dominio. L'idea, discussa nella presente sezione, è di cercare un qualche dominio "canonico" tale che una qualunque formula risulta insoddisfacibile se e solo se è falsa in tutte le interpretazioni su questo dominio. Un tale dominio esiste ed è chiamato *universo di Herbrand*.

Nel seguito considereremo formule della logica del primo ordine chiuse ed in forma di Skolem. Ricordiamo che ogni formula può essere trasformata in un'altra avente tali caratteristiche che risulta soddisfacibile se e solo se lo è la formula di partenza (Proposizione 4.41).

**Definizione 6.1** *L'* universo di Herbrand *di un linguaggio $\mathcal{L}$ del primo ordine, indicato con $H(\mathcal{L})$, è l'insieme dei termini contenente:*

1. *tutte le costanti che compaiono in $\mathcal{L}$; se $\mathcal{L}$ non ha costanti, allora scelto un simbolo di costante $a$, si pone $a \in H(\mathcal{L})$;*

2. *tutti i termini costruiti applicando i simboli di funzione che occorrono in $\mathcal{L}$ agli elementi di $H(\mathcal{L})$.*

Mostreremo nel seguito che questo è il dominio "canonico" cercato.

***Osservazione*** Se $\mathcal{L}$ contiene almeno un simbolo di funzione, allora $H(\mathcal{L})$ è infinito.

**Esempio 6.1** Sia $\mathcal{L}$ il linguaggio contenente i simboli di costanti $a$ e $c$, un simbolo di funzione unaria $f$, un simbolo di funzione binaria $g$ ed i predicati $A$ binario e $B$ ternario; l'universo di Herbrand per $\mathcal{L}$ è

$$H(\mathcal{L}) = \{a, c, f(a), f(c), g(a,a), g(c,c), f(f(a)), f(g(a,c)), g(a, f(a)), \dots\}$$

**Definizione 6.2** *Una formula che non contiene variabili viene detta* formula ground. *Una sostituzione che elimina le variabili si dice* sostituzione ground. *Una formula ottenuta mediante una sostituzione ground si dice* istanza ground *della formula di partenza.*

**Definizione 6.3** *Si definisce* base di Herbrand *di un linguaggio $\mathcal{L}$ del primo ordine, e si indica con $B(\mathcal{L})$, l'insieme di tutte le formule atomiche ground che si possono costruire utilizzando i simboli di predicato che compaiono in $\mathcal{L}$, aventi come argomenti gli elementi dell'universo di Herbrand $H(\mathcal{L})$.*

**Esempio 6.2** Sia $\mathcal{L}$ il linguaggio dell'esempio precedente, la base di Herbrand per $\mathcal{L}$ è $B(\mathcal{L}) = \{(A(a,a), A(a, f(a)), A(g(a,a), c), B(a,a,a), \dots\}$.

**Definizione 6.4** *Sia $\mathcal{L}$ un linguaggio del primo ordine; $\mathcal{H} = (D_{\mathcal{H}}, I_{\mathcal{H}})$[1] è un'* interpretazione di Herbrand *per $\mathcal{L}$ se sono verificate le seguenti condizioni:*

1. $D_{\mathcal{H}} = H(\mathcal{L})$

2. *Ogni simbolo di costante $c \in D_{\mathcal{H}}$ viene interpretato in se stesso, cioè*

$$c^{\mathcal{H}} = c$$

[1] Poiché si considerano solo formule chiuse, non è necessario esplicitare la funzione ambiente $\xi^{\mathcal{H}}$.

*3. Per ogni simbolo di funzione $f$, di arità $k$, con $k > 0$, che occorre in $\mathcal{L}$, e per tutti i termini $t_1, \ldots, t_k \in D_{\mathcal{H}}$*

$$f^{\mathcal{H}}(t_1, \ldots, t_k) = f(t_1, \ldots, t_k)$$

Notiamo che non si impone alcuna restrizione sull'interpretazione dei simboli di predicato. Pertanto, due diverse interpretazioni di Herbrand differiscono solo per il modo in cui vengono interpretati i predicati.
Dunque se $A$ è un simbolo di predicato n-ario, $I_{\mathcal{H}}$ gli associa un sottoinsieme $\mathcal{I}(A)$ di atomi della base di Herbrand;

$$A^{\mathcal{H}}(t_1, \ldots, t_n) = \begin{cases} 1 & \text{se } A(t_1, \ldots, t_n) \in \mathcal{I}(A) \subseteq B(\mathcal{L}) \\ 0 & \text{altrimenti} \end{cases}$$

**Esempio 6.3** Sia $\mathcal{L}$ il linguaggio dell'Esempio 6.1, un'interpretazione di Herbrand per $\mathcal{L}$ è:
$D_{\mathcal{H}} = H(\mathcal{L})$
$f^{\mathcal{H}}(a) = f(a)$
$f^{\mathcal{H}}(f(a)) = f(f(a))$
$g^{\mathcal{H}}(a, a) = g(a, a)$
$\vdots$
$A^{\mathcal{H}}(t_1, t_2) = \{(t_1, t_2) \mid t_1, t_2 \in D_{\mathcal{H}} \text{ e } t_1 = t_2\}$
$B^{\mathcal{H}}(t_1, t_2, t_3) = \{(t_1, t_2, t_3) \mid t_1, t_2, t_3 \in D_{\mathcal{H}} \text{ e } t_1 = g(t_2, t_3)\}$

**Definizione 6.5** *Data una fbf $P$, un* modello di Herbrand *per $P$ è un'interpretazione di Herbrand che la soddisfa.*

Osserviamo che i modelli di Herbrand, così come le interpretazioni canoniche (cfr.5.5) sono costruiti a partire dalla *sintassi* delle formule stesse.

**Esempio 6.4** Consideriamo l'interpretazione di Herbrand dell'esempio precedente, questa non è un modello per la formula ben formata $P$ data da $\forall xyB(x, y, g(x, y))$ in quanto non è vero che $\forall \xi,\ v^{(\mathcal{H}, \xi^{\mathcal{H}})}(P) = 1$ , infatti, $v^{(\mathcal{H}, \xi^{\mathcal{H}}[{}^{a}/_{x}, {}^{a}/_{y}])}(P) = 0$ essendo $a \neq g(a, g(a, a))$). Mentre scegliendo
$B^{\mathcal{H}}(t_1, t_2, t_3) = \{(t_1, t_2, t_3) \mid t_1, t_2, t_3 \in D_{\mathcal{H}} \text{ e } t_3 = g(t_1, t_2)\}$ la nuova interpretazione di Herbrand è un modello per $P$.

**Teorema 6.6** *Sia $P$ una formula ben formata chiusa ed in forma di Skolem; $P$ è soddisfacibile se e solo se ha un modello di Herbrand.*

*Dimostrazione.* Se $P$ ha un modello di Herbrand allora è soddisfacibile. Viceversa, supponiamo che $P$ sia soddisfacibile. Sia $(\mathcal{A}, \xi^{\mathcal{A}})$ un modello di $P$. Se il linguaggio $\mathcal{L}$ sul quale è definita $P$ non contiene costanti, si prende un nuovo simbolo di costante $c$ e si pone

$$c^{\mathcal{A}} = m$$

dove $m$ è un arbitrario elemento di $D_{\mathcal{A}}$.
Definiamo quindi l'interpretazione di Herbrand $(D_{\mathcal{H}}, I_{\mathcal{H}})$ nel seguente modo[2]: per ogni predicato n-ario $A \in \mathcal{L}$,

$$A^{\mathcal{H}}(t_1, \ldots, t_n) = \{(t_1, \ldots, t_n) \mid t_1, \ldots, t_n \in D_{\mathcal{H}} \text{ e } v^{(\mathcal{A}, \xi^{\mathcal{A}})}([\![t_1]\!]^{\mathcal{A}}_{\xi}, \ldots, [\![t_n]\!]^{\mathcal{A}}_{\xi}) = 1\}$$

$\mathcal{H}$ è un modello per $P$. Infatti, proviamo che $\mathcal{A} \models P$ se e solo se $\mathcal{H} \models P$. Per induzione sulla struttura di $P$.
(*caso base*) Se $P$ è un atomo, allora, per definizione di $\mathcal{H}$, $v^{(\mathcal{A}, \xi^{\mathcal{A}})}(P) = v^{(\mathcal{H}, \xi^{\mathcal{H}})}(P)$ e quindi $\mathcal{A} \models P$ se e solo se $\mathcal{H} \models P$.
Veniamo al caso induttivo.

- Se $P$ ha la forma $\neg P_1$, allora $\mathcal{A} \models P$ se e solo se, per il Lemma 4.28, $\mathcal{A} \not\models P_1$, ma, per ipotesi induttiva $\mathcal{H} \not\models P_1$ e quindi $\mathcal{H} \models P$.

- Se $P$ ha la forma $P_1 \wedge P_2$, allora $\mathcal{A} \models P$ se e solo se $\mathcal{A} \models P_1$ e $\mathcal{A} \models P_2$; ma, per ipotesi induttiva $\mathcal{H} \models P_1$ e $\mathcal{H} \models P_2$ e quindi $\mathcal{H} \models P$.

- Se $P$ ha la forma $P_1 \vee P_2$ o $P_1 \rightarrow P_2$, la dimostrazione è simile al caso precedente.

- Se $P$ ha la forma $\forall x P_1$, allora $\mathcal{A} \models P$ se e solo se $\forall a \in D_{\mathcal{A}}$, e per ogni $\xi^{\mathcal{A}}$, $v^{(\mathcal{A}, \xi^{\mathcal{A}}[a/x])}(P_1) = 1$; in particolare, per tutti gli $a \in D_{\mathcal{A}}$ nella forma $a = [\![t]\!]^{\mathcal{A}}_{\xi}$ per un qualche $t \in D_{\mathcal{H}}$, risulta $v^{(\mathcal{A}, \xi^{\mathcal{A}}[[\![t]\!]^{\mathcal{A}}_{\xi}/x])}(P_1) = 1$ e che è uguale, per l'Esercizio 4.6, a $v^{(\mathcal{A}, \xi^{\mathcal{A}})}(P_1[t/x])$. Per ipotesi induttiva, $v^{(\mathcal{H}, \xi^{\mathcal{H}})}(P_1[t/x]) = 1$ per ogni $t \in D_{\mathcal{H}}$. Di nuovo, risulta che $\forall t \in D_{\mathcal{H}}$, $v^{(\mathcal{H}, \xi^{\mathcal{H}}[t/x])}(P_1) = v^{(\mathcal{H}, \xi^{\mathcal{H}})}(P_1[t/x]) = 1$ da cui $v^{(\mathcal{H}, \xi^{\mathcal{H}})}(\forall x P_1) = v^{(\mathcal{H}, \xi^{\mathcal{H}})}(P) = 1$, cioè $\mathcal{H} \models P$.

□

Osserviamo che quando $P$ ha la forma $\forall x P_1$, non è possibile applicare direttamente l'ipotesi induttiva a $P_1$ in quanto non è detto che questa sia chiusa ($x$ potrebbe occorrere libera in $P_1$).

Abbiamo quindi trovato un particolare dominio, l'universo di Herbrand, tale che una formula è soddisfacibile se e solo se ha un modello costruito su di esso. Dunque è possibile restringere la ricerca dei potenziali modelli di una formula soddisfacibile ai modelli di Herbrand.
Il teorema precedente può essere facilmente generalizzato ad insiemi di fbf. Infatti:

**Corollario 6.7** *Sia $S$ un insieme di formule ben formate chiuse ed in forma di Skolem; $S$ è soddisfacibile se e solo se ha un modello di Herbrand.*

[2]Ricordiamo che, per definizione di interpretazione di Herbrand è necessario specificare solo l'interpretazione dei simboli di predicato.

È possibile riformulare il Teorema 6.6 nel seguente modo:

**Teorema 6.8** *Sia $P$ una formula ben formata chiusa ed in forma di Skolem; $P$ è insoddisfacibile se e solo se non ha un modello di Herbrand.*

**Corollario 6.9** *Sia $S$ un insieme di formule ben formate chiuse ed in forma di Skolem; $S$ è insoddisfacibile se e solo se non ha un modello di Herbrand.*

***Osservazione*** Se si considerano formule non in forma di Skolem, il Teorema 6.6 non è più valido; in altri termini, se $S$ è un insieme di arbitrarie formule chiuse, non è possibile mostrarne l'insoddisfacibilità limitandosi a considerare le interpretazioni di Herbrand. Infatti, preso ad esempio l'insieme $S = \{R(c), \exists x \neg R(x)\}$, questo è soddisfacibile ma non ha un modello di Herbrand in quanto, consideriamo, per fissare le idee, il linguaggio $\mathcal{L}$ avente come elementi la costante $c$ ed il predicato unario $R$, allora $B(\mathcal{L}) = \{R(c)\}$; i possibili sottoinsiemi di $B(\mathcal{L})$ sono: $\emptyset$ (che soddisfa $\exists x \neg R(x)$ ma non $R(c)$) e $\{R(c)\}$ (che soddisfa $R(c)$ ma non $\exists x \neg R(x)$).

### 6.1.1 Teorema di Herbrand

In questa sezione presenteremo un importante teorema, al quale faremo riferimento come "teorema di Herbrand"[3], che deriva dai risultati dimostrati da Herbrand in [Her30]. Questo è alla base della maggior parte delle attuali procedure di dimostrazione automatica dei teoremi.
Abbiamo in precedenza stabilito che una formula chiusa ed in forma di Skolem è insoddisfacibile se e solo se è falsa in tutte le interpretazioni costruite sull'universo di Herbrand. Poiché, però, in generale, esistono infinite interpretazioni siffatte, per provare che una formula è insoddisfacibile, non è possibile considerarle tutte; procediamo quindi nel seguente modo:

**Definizione 6.10** *Sia $P = \forall x_1 x_2, \ldots x_n P'$ una formula chiusa in forma di Skolem. Definiamo* espansione di Herbrand *di $P$, indicata con $\mathcal{E}(P)$, l'insieme delle formule ground ottenute sostituendo i termini dell'universo di Herbrand per il linguaggio $\mathcal{L}$, sul quale definita $P$, alle variabili di $P$, in tutti i modi possibili; formalmente:*

$$\mathcal{E}(P) = \{P'[^{t_1,t_2,\ldots,t_n}/_{x_1,x_2,\ldots,x_n}] \mid t_1, t_2, \ldots, t_n \in H(\mathcal{L})\}$$

**Esempio 6.5** Sia $P$ la fbf $\forall xy((A(a) \vee A(f(x))) \wedge \neg A(y))$. Dunque, la base di Herbrand è $H(\mathcal{L}) = \{a, f(a), f(f(a)), \ldots\}$, e l'espansione di Herbrand di $P$, vale a dire $\mathcal{E}(P)$, è:

[3] La prima pubblicazione del "teorema di Herbrand" in tale forma è dovuta a Quine ([Qui55]); la sua formulazione originaria verrà discussa a p.176.

| $x$ | $y$ | $(A(a) \vee A(f(x))) \wedge \neg A(y)$ |
|---|---|---|
| $a$ | $a$ | $(A(a) \vee A(f(a))) \wedge \neg A(a)$ |
| $a$ | $f(a)$ | $(A(a) \vee A(f(a))) \wedge \neg A(f(a))$ |
| $f(a)$ | $a$ | $(A(a) \vee A(f(f(a)))) \wedge \neg A(a)$ |
| $f(a)$ | $f(a)$ | $(A(a) \vee A(f(f(a)))) \wedge \neg A(f(a))$ |
| $a$ | $f(f(a))$ | $(A(a) \vee A(f(a))) \wedge \neg A(f(f(a)))$ |
| $\vdots$ | $\vdots$ | $\vdots$ |

Le formule dell'espansione di Herbrand non avendo variabili, e di conseguenza quantificatori, si possono considerare come formule della logica proposizionale. Quindi per definire un'interpretazione di $\mathcal{E}(P)$, è sufficiente assegnare un valore di verità ad ogni formula atomica che vi compare.

**Esempio 6.6** Nell'esempio precedente $A(a), A(f(a)), A(f(f(a))), \ldots$ sono le formule atomiche in $\mathcal{E}(P)$.

**Teorema 6.11** *Per ogni formula chiusa $P$ in forma di Skolem, $P$ è soddisfacibile se e solo se $\mathcal{E}(P)$[4] lo è.*

*Dimostrazione.* Per il Teorema 6.6 è sufficiente provare che $P$ ha un modello di Herbrand se e solo se $\mathcal{E}(P)$ è soddisfacibile. Supponiamo, per fissare le idee, che $P$ abbia la forma $\forall x_1 x_2 \ldots x_n P'$, allora $\mathcal{H}$ è un modello per $P$ se e solo se $\forall t_1 t_2 \ldots t_n \in D_{\mathcal{H}}$ $v^{(\mathcal{H},\xi^{\mathcal{H}}[t_1,\ldots,t_n/x_1,\ldots,x_n])}(P') = 1$ se e solo se $\forall t_1 t_2 \ldots t_n \in D_{\mathcal{H}}$ $v^{(\mathcal{H},\xi^{\mathcal{H}})}(P'[t_1,\ldots,t_n/x_1,\ldots,x_n]) = 1$ se e solo se, per definizione di espansione di Herbrand, per ogni formula $Q \in \mathcal{E}(P)$, $v^{(\mathcal{H},\xi^{\mathcal{H}})}(Q) = 1$ se e solo se $\mathcal{H}$ è un modello per $\mathcal{E}(P)$. □

Tale teorema asserisce la possibilità di "approssimare" la soddisfacibilità di una formula della logica del primo ordine con insiemi sempre più grandi di formule della logica proposizionale. È evidente che in tempo finito non si è in grado di stabilire se una formula è soddisfacibile. Ciò non è vero per l'insoddisfacibilità; infatti:

**Teorema 6.12 (di Herbrand)**
*Una formula $P$ chiusa ed in forma di Skolem è insoddisfacibile se e solo se esiste un sottoinsieme finito di $\mathcal{E}(P)$[5] che lo è.*

*Dimostrazione.* Segue dal Teorema di compattezza e dal Teorema 6.11. □

Il teorema di Herbrand suggerisce una procedura di refutazione; infatti da esso deriva la correttezza del seguente algoritmo che consente di *semidecidere* se una formula della logica del primo ordine è insoddisfacibile.

**Notazione**
Sia $P$ una fbf chiusa ed in forma di Skolem. Fissata un'enumerazione per gli

[4] Inteso come insieme di formule della logica proposizionale.
[5] Inteso come insieme di formule della logica proposizionale

elementi di $\mathcal{E}(P)$, siano questi $P_1, P_2, \ldots$, allora:

**Algoritmo**

1. $n := 0$
2. Ripeti:
   (a) $n := n + 1$

   Finché $(P_1 \wedge \cdots \wedge P_n)$ è insoddisfacibile
3. Output "$P$ è insoddisfacibile"

Per verificare che $(P_1 \wedge \cdots \wedge P_n)$ è insoddisfacibile, ricordando che gli elementi dell'espansione di Herbrand sono a tutti gli effetti formule della logica proposizionale, si può utilizzare il metodo delle tabelle di verità.
Osserviamo che, se la formula in input è insoddisfacibile, l'algoritmo termina dopo un numero finito di passi, non termina altrimenti (ad ogni esecuzione di 2. viene generato un nuovo elemento di $\mathcal{E}(P)$ nel tentativo di trovare un sottoinsieme di questa che è insoddisfacibile). Dunque tale algoritmo consente di *semidecidere* l'insoddisfacibilità di una formula della logica del primo ordine nel senso che risponde "insoddisfacibile" se la formula lo è, non risponde (ripete all'infinito il ciclo 2.), altrimenti.

**Esempio 6.7** Sia $P$ la fbf dell'Esempio 6.5, se consideriamo il sottoinsieme finito di $\mathcal{E}(P)$ costituito dalle formule $((A(a) \vee A(f(a))) \wedge \neg A(a))$ e $((A(a) \vee A(f(a))) \wedge \neg A(f(a)))$, questo è insoddisfacibile. Infatti, la sua tabella di verità è la seguente:

| $A(a)$ | $A(f(a))$ | $(A(a) \vee A(f(a)) \wedge \neg A(a)) \wedge ((A(a) \vee A(f(a))) \wedge \neg A(f(a)))$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

E quindi, per il teorema di Herbrand, $P$ è insoddisfacibile.

La procedura di refutazione basata sul teorema di Herbrand risulta notevolmente inefficiente in quanto per provare l'insoddisfacibilità di una formula, necessita di generare un gran numero di elementi della sua espansione di Herbrand.
Per convincersi di ciò, verificare, utilizzando l'algoritmo di p.175, che l'insieme delle seguenti formule

$$A(x, e, x),$$
$$\neg A(y, z, v) \vee \neg A(y, v, w) \vee A(e, z, w),$$
$$A(a, f(u, v), e),$$
$$\neg A(e, f(f(b, c), a), a)$$

è insoddisfacibile.

***Osservazione*** È possibile riformulare i risultati presentati in questa sezione in forma "duale", cioè semidecidere (in modo diretto) la validità di una formula della logica del primo ordine anziché l'insoddisfacibilità (cfr. Esercizio 6.4).

**Considerazioni sul teorema di Herbrand**

Poiché vi sono numerosi enunciati noti in letteratura come "teorema di Herbrand", si ritiene utile concludere la presente sezione illustrando brevemente i principali risultati realmente contenuti nel lavoro di Herbrand. Egli ha introdotto un sistema deduttivo per la logica del primo ordine, che chiameremo $\mathcal{F}_H$, i cui assiomi sono tutte le tautologie prive di quantificatori, e le cui regole di inferenza sono: introduzione dei quantificatori (esistenziale ed universale), spostamento dei quantificatori all'interno della formula, ridenominazione delle variabili legate e contrazione ($P \vee P$ viene rimpiazzata da $P$). Osserviamo che il modus ponens non è una regola di inferenza in tale sistema. Il teorema di Herbrand, nella sua formulazione originaria, è il seguente: Sia $\mathcal{F}$ un sistema alla Hilbert per la logica del primo ordine (insieme di assiomi + modus ponens), ed $\mathcal{E}(P)$ l'espansione di Herbrand[6] di $P$; allora:

1. Se $\vdash_{\mathcal{F}_H} P$ allora $\vdash_{\mathcal{F}} P$.

2. Se $\vdash_{\mathcal{F}} P$ allora esiste un sottoinsieme di $\mathcal{E}(P)$ che è valido[7].

3. Se esiste un sottoinsieme di $\mathcal{E}(P)$ che è valido allora $\vdash_{\mathcal{F}_H} P$.

Di conseguenza, le seguenti affermazioni risultano equivalenti:
(a) $\vdash_{\mathcal{F}_H} P$
(b) $\vdash_{\mathcal{F}} P$
(c) Esiste un sottoinsieme di $\mathcal{E}(P)$ che è valido.

Se ad 1-3 si aggiungono la correttezza e la completezza di $\mathcal{F}$, segue che (d) $P$ è valida.
è equivalente alle affermazioni (a)-(c).
In letteratura, le varie versioni del teorema che asserisce l'equivalenza di (c) e (d) vengono chiamate "teorema di Herbrand" o "teorema di Gödel-Herbrand-Skolem[8]". Tuttavia rimarchiamo che la versione originaria di tale teorema è un risultato puramente *sintattico*. Osserviamo che il teorema di Herbrand implica che il *modus ponens* è una regola derivata nel sistema $\mathcal{F}_H$; tale risultato è strettamente legato al *teorema di eliminazione del taglio* di Gentzen ([Gen34]).

[6] Si veda l'osservazione precdente.

[7] La dimostrazione di tale enunciato, realizzata da Herbrand, contiene un errore ([AAD63]), successivamente corretto da Dreben e Denton ([DD66]).

[8] In quanto i risultati dimostrati da Gödel semplificano notevolmente la dimostrazione originaria di Herbrand, ed il riferimento a Skolem è dovuto al ruolo di primaria importanza assunto dalle funzioni di Skolem nell'eliminazione dei quantificatori nelle formule.

## 6.2 Metodo di Risoluzione

In questa sezione presenteremo un metodo alternativo di dimostrazione automatica di teoremi introdotto da J. Robinson nel 1965 ([Rob65]): il metodo di risoluzione. Questo risulta molto più efficiente della procedura esaminata nella sezione 6.1.1, ed è alla base della programmazione logica e del *Prolog*[9]. Tale metodo si basa su un sistema formale, privo di assiomi, che ha un'unica regola di inferenza (la risoluzione) che si applica a formule in una particolare forma detta *forma a clausole*. Esso utilizza una strategia di refutazione. La sua attrattiva principale è la semplicità di impiego; infatti, avendo una sola regola, l'unica scelta che si presenta è quella relativa a quali formule applicarla. Tale metodo verrà prima presentato per formule della logica proposizionale, quindi esteso a quelle della logica del primo ordine.

### 6.2.1 Risoluzione nella logica proposizionale

Come già accennato in precedenza, il metodo di risoluzione è basato su un sistema formale, particolarmente semplice, che ha un'unica regola di inferenza: la risoluzione, che si applica a formule in forma a clausole.

**Definizione 6.13** *Una* clausola *è una disgiunzione finita di zero o più letterali. Una clausola che non contiene letterali è detta* clausola vuota *e si indica con il simbolo* $\square$.

Ovviamente, una clausola è soddisfatta se e solo se lo è almeno uno dei suoi letterali; poiché la clausola vuota non contiene letterali che possono essere soddisfatti da una qualche interpretazione, questa risulta sempre falsa.

**Notazione**
Se $l$ è un letterale, indicheremo con $\bar{l}$ la sua negazione.

***Osservazione*** Una clausola è una tautologia se contiene un letterale $l$ e la sua negazione $\bar{l}$.

È conveniente rappresentare le clausole come *insiemi* di letterali; in tal modo le proprietà commutativa, associativa e di idempotenza della disgiunzione sono automaticamente "fornite" dalla notazione insiemistica.
Ad esempio

$$A \vee B \vee C$$

$$A \vee (B \vee C)$$

$$B \vee C \vee A$$

$$A \vee B \vee A \vee C$$

hanno la stessa rappresentazione

$$\{A, B, C\}.$$

[9] PROgramming in LOGic.

Ogni formula ben formata si può rappresentare in forma a clausole. Infatti, data una fbf, per il Teorema 1.33 è possibile trasformarla in una forma normale congiuntiva equivalente: tale forma sarà, in generale, del tipo

$$C_1 \wedge \cdots \wedge C_n$$

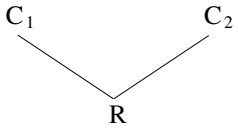
dove le lettere proposizionali $C_i$ rappresentano generiche clausole. Con considerazioni analoghe a quelle effettuate in precedenza si può essere più coincisi sul significato della formula data e dire che essa consiste nell'insieme $\{C_1, \ldots, C_n\}$. Osserviamo che la virgola che separa due letterali in una clausola, rappresentata in notazione insiemistica, è da intendersi come $\vee$, mentre quella che separa due clausole corrisponde ad $\wedge$.
Dunque ogni fbf è equivalente ad un insieme di clausole.

***Osservazione*** La clausola vuota ($\Box$) è diversa dall'insieme vuoto di clausole ($\emptyset$). Ricordiamo, infatti, che una formula in forma normale congiuntiva è vera se e solo se tutte le sue clausole lo sono e dunque $\emptyset$, non avendo elementi falsi, risulta sempre vero.

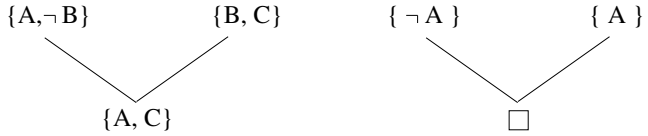
Vediamo ora la regola di inferenza, detta *risoluzione*, sulla quale si basa il metodo omonimo.

**Definizione 6.14** *Siano $C_1, C_2$ ed $R$ clausole. Si dice che $R$ è una* risolvente *di $C_1$ e $C_2$ se esiste un letterale $l \in C_1$ tale che $\bar{l} \in C_2$ ed $R$ ha la forma $(C_1 - \{l\}) \cup (C_2 - \{\bar{l}\})$.*

Graficamente possiamo denotare la situazione con il seguente diagramma:




**Esempio 6.8** Esempi di applicazione della regola di risoluzione sono:




***Osservazione*** La regola di risoluzione si può vedere come una caso particolare della regola di taglio. Infatti, consideriamo la regola di taglio:

$$(taglio) \qquad \text{se } \Gamma \vdash A \text{ e } \Gamma, A \vdash \Delta \text{ allora } \Gamma \vdash \Delta$$

riscritta a livello di formule diventa:

$$\frac{\Gamma \to A \qquad \Gamma \wedge A \to \Delta}{\Gamma \to \Delta}$$

che è equivalente a: da $\neg\Gamma \vee A$ e $\neg\Gamma \vee \neg A \vee \Delta$ si inferisce $\neg\Gamma \vee \Delta$ che, nel caso particolare in cui $\Gamma$ e $\Delta$ sono clausole ed $A$ è un letterale, è un'applicazione della regola di risoluzione.

**Definizione 6.15** *Sia $S$ un insieme di clausole, una* derivazione (o prova) per risoluzione *di $S_1$ da $S$ è una sequenza $C_1, \ldots, C_m$ di clausole tali che:*

- $C_m = S_1$
- $C_i$, $\forall i = 1, \ldots, m$ *è una clausola in $S$ oppure una risolvente di due clausole $C_j, C_k$ con $j, k < i$.*

**Notazione**
Se esiste una derivazione per risoluzione di $S_1$ da $S$, scriveremo $S \vdash_R S_1$.

**Definizione 6.16** *Una derivazione per risoluzione di $\Box$ da $S$ si dice* refutazione *di $S$.*

Proviamo ora che la risoluzione è una regola di inferenza che preserva l'equivalenza.

**Lemma 6.17** *Siano $C_1$ e $C_2$ clausole, ed $R$ una loro risolvente, allora $R$ è conseguenza logica di $C_1$ e $C_2$.*

*Dimostrazione.* Sia $v$ un'interpretazione che è un modello per $C_1$ e $C_2$. Proviamo che $v(R) = 1$. Assumiamo, per fissare le idee, che $R$ abbia la forma: $R = (C_1 - \{l\}) \cup (C_2 - \{\bar{l}\})$ dove $l \in C_1$ mentre $\bar{l} \in C_2$. Vi sono due possibilità:

1. $v(l) = 1$. Allora da $v(C_2) = 1$ e $v(\bar{l}) = 0$ segue $v(C_2 - \{\bar{l}\}) = 1$ e quindi $v(R) = 1$.
2. $v(l) = 0$. Allora da $v(C_1) = 1$ segue $v(C_1 - \{l\}) = 1$; dunque $v(R) = 1$.

$\Box$

Il metodo di risoluzione, dato un insieme di clausole $S$, tenta di costruire da questo una derivazione per risoluzione della clausola vuota; se ciò si verifica $S$ è insoddisfacibile.
È possibile provare che il calcolo che utilizza la risoluzione è corretto e completo rispetto all'insoddisfacibilità (ossia per refutazione), dove per *correttezza* si intende che ogni formula che si dimostra insoddisfacibile, lo è effettivamente, mentre *completezza* significa che per ogni formula insoddisfacibile esiste una prova di ciò utilizzando la risoluzione. Infatti:

**Teorema 6.18 (di Risoluzione)**
*Un insieme di clausole $S$ è insoddisfacibile se e solo se $S \vdash_R \Box$.*

*Dimostrazione.* Assumiamo, senza perdere di generalità, che $S$ sia finito, in quanto, in caso contrario, per il teorema di compattezza, potremmo comunque operare su un suo sottoinsieme finito.

(*Correttezza*) Vogliamo provare che se esiste una prova di $\Box$ da $S$, allora $S$ è insoddisfacibile. Infatti, supponiamo per assurdo che $S$ sia soddisfacibile, allora esiste un'interpretazione $v$ che è un modello per tutte le clausole in esso contenute. Sia $C_1, \ldots, C_k$ la refutazione di $S$; dal lemma precedente segue che $v(C_i) = 1 \ \ \forall i = 1, \ldots, k$, in particolare $v(\Box) = 1$ che è assurdo. Dunque $S$ è insoddisfacibile.
(*Completezza*) Supponiamo che $S$ sia insoddisfacibile e proviamo che $S \vdash_R \Box$. Per induzione sul numero $n$ di differenti proposizioni atomiche che compaiono in $S$.
Caso base: se $n = 0$ allora dev'essere $S = \{\Box\}$ e quindi $S \vdash_R \Box$. Assumiamo, per ipotesi induttiva che per ogni insieme di clausole $S'$ insoddisfacibile, contenente al più i letterali distinti[10] $l_1, \ldots, l_n$, $S' \vdash \Box$. Sia $S$ un insieme di clausole contenente i letterali $l_1, \ldots, l_n, l_{n+1}$. Supponiamo che nessuna clausola in $S$ contenga sia $l_{n+1}$ che $\bar{l}_{n+1}$ (in caso contrario potremmo eliminarla senza influire sull'insoddisfacibilità di $S$, si veda a tal proposito l'Esercizio 6.10). Da $S$ possiamo ottenere due nuovi insiemi di clausole $S^{l_{n+1}}$ ed $S^{\bar{l}_{n+1}}$ nel seguente modo: $S^{l_{n+1}}$ risulta da $S$ cancellando ogni occorrenza di $l_{n+1}$ nelle clausole e togliendo le clausole che contengono $\bar{l}_{n+1}$ (corrisponde a porre $v(l_{n+1}) = 0$ in un'interpretazione $v$), cioè $S^{l_{n+1}} = \{C - \{l_{n+1}\} \mid C \in S \wedge \ \bar{l}_{n+1} \notin C\}$; $S^{\bar{l}_{n+1}}$ si ottiene in modo analogo scambiando, però, i ruoli di $l_{n+1}$ ed $\bar{l}_{n+1}$. $S^{l_{n+1}}$ ed $S^{\bar{l}_{n+1}}$ sono insoddisfacibili. Infatti, supponiamo per assurdo che $S^{l_{n+1}}$ sia soddisfacibile, allora esiste un'interpretazione $v$ che è un modello per $S^{l_{n+1}}$; definiamo un'interpretazione $v'$ nel seguente modo

$$v'(B) = \begin{cases} v(B) & \text{se } B \in \{l_1, \ldots, l_n\} \\ 0 & \text{se } B = l_{n+1} \end{cases}$$

$v'(S) = 1$ in quanto ogni clausola $C \in S$ è tale che o $\bar{l}_{n+1} \in C$, ed in tal caso $v'(C) = 1$, oppure $\bar{l}_{n+1} \notin C$, ma ancora $v'(C) = 1$ essendo $S^{l_{n+1}}$ soddisfacibile; ma è assurdo avendo supposto che $S$ è insoddisfacibile. In modo analogo si dimostra che $S^{\bar{l}_{n+1}}$ è insoddisfacibile.
Applicando l'ipotesi induttiva risulta che $S^{l_{n+1}} \vdash_R \Box$ e $S^{\bar{l}_{n+1}} \vdash_R \Box$; ciò significa che in $S^{l_{n+1}}$ esiste una sequenza di clausole $C_1, C_2, \ldots, C_m$ tale che

- $C_m = \Box$
- $\forall i = 1, \ldots, m, \ C_i \in S^{l_{n+1}}$ oppure $C_i$ è una risolvente di due clausole $C_j, C_k$ con $j, k < i$.

In $S^{\bar{l}_{n+1}}$ esiste una sequenza analoga $C'_1, \ldots, C'_r$. Supponiamo che alcune delle clausole $C_i \in S^{l_{n+1}}$ siano state ottenute cancellando il letterale $l_{n+1}$ (in caso contrario l'asserto risulta già verificato); ripristinando le clausole originarie, cioè ponendo $C_i \cup \{l_{n+1}\}$, e trasportando $l_{n+1}$ lungo i passi di risoluzione, da $C_1, \ldots, C_m$ otteniamo una nuova sequenza di clausole per $S$ tale che $S \vdash_R \{l_{n+1}\}$. In modo analogo, ripristinando $\bar{l}_{n+1}$ in $C'_1, \ldots, C'_r$, si ottiene $S \vdash_R \bar{l}_{n+1}$. Da un ulteriore passo di risoluzione è possibile derivare la clausola vuota e dunque $S \vdash_R \Box$. $\Box$

[10] In tale contesto, un generico letterale $\overline{l_i}$, non si considera distinto da $l_i$.

Il calcolo che utilizza la risoluzione è un calcolo per derivare la clausola vuota da un insieme di fbf quando questo è insoddisfacibile, ed è completo rispetto a tale problema. Esso non è però utilizzabile per derivare le conseguenze logiche di un insieme di fbf. Infatti il metodo di risoluzione è corretto ma non completo rispetto alla deduzione, cioè se esiste una prova per risoluzione di $P$ da $S$, per il Lemma 6.17, $S \models P$, ma non è vero il viceversa; un semplice controesempio è il seguente: $A \models A \vee B$ ma non è possibile applicare la regola di risoluzione ad $\{A\}$ per ottenere $\{A, B\}$. Tuttavia:

**Proposizione 6.19** *$S \models P$ se e solo se $S \cup \{\neg P\} \vdash_R \square$.*

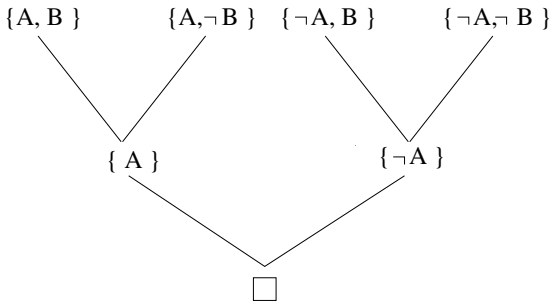
*Dimostrazione.* Segue dal teorema di risoluzione e dal Lemma 1.13. $\square$

Dunque per verificare che $S \models P$ si dimostra che $S^c \cup \{(\neg P)^c\} \vdash_R \square$.
Osserviamo che in tal modo non si ottiene alcuna prova diretta del fatto che $P$ è conseguenza logica di $S$; ciò equivale ad effettuare una dimostrazione *per assurdo*.

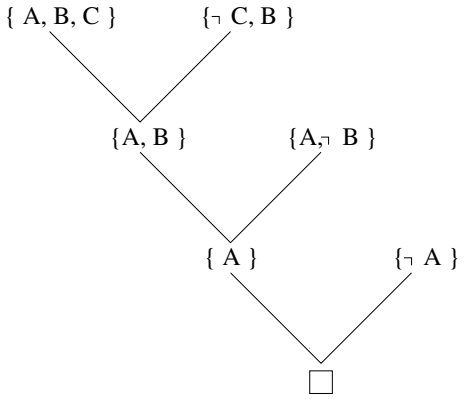
**Esempio 6.9** Siano $S = \{(\neg A \to B) \wedge (A \to B), \neg A \to \neg B\}$ e
$P = \{A \wedge B\}$, vogliamo provare che $S \models P$. La trasformazione in clausole di $S$ e $\neg P$ produce: $S^c = \{\{A, B\}, \{\neg A, B\}, \{A, \neg B\}\}$ e $(\neg P)^c = \{\{\neg A, \neg B\}\}$. Allora $S^c \cup (\neg P)^c$ è insoddisfacibile, infatti, $\exists C_1, \ldots, C_7$ tali che:

| | |
|---|---|
| $C_1 = \{A, B\}$ | (clausola in $S^c$) |
| $C_2 = \{A, \neg B\}$ | (clausola in $S^c$) |
| $C_3 = \{A\}$ | (risolvente di $C_1$ e $C_2$) |
| $C_4 = \{\neg A, B\}$ | (clausola in $S^c$) |
| $C_5 = \{\neg A, \neg B\}$ | (clausola in $(\neg P)^c$) |
| $C_6 = \{\neg A\}$ | (risolvente di $C_4$ e $C_5$) |
| $C_7 = \square$ | (risolvente di $C_3$ e $C_6$) |

Graficamente:




**Esempio 6.10** Sia $S = \{\{A, B, C\}, \{\neg C, B\}, \{A, \neg B\}\}$ proviamo che $S \models A$. Infatti




Il metodo di risoluzione, nella sua formulazione originaria, per provare che una formula $P$ è valida, cioè trovare una refutazione di $(\neg P)^c$, procede esaustivamente generando i risolventi per tutte le coppie di clausole dell'insieme di partenza $(\neg P)^c$; i risolventi sono quindi aggiunti all'insieme iniziale ed il procedimento viene iterato fino a derivare, se è possibile, la clausola vuota.

**Definizione 6.20** *Sia $S$ un insieme di clausole, l'* insieme dei risolventi *di $S$, che si indica con $Ris(S)$ è definito nel seguente modo:*

$$Ris(S) = S \cup \{C_{i,j} \mid C_{i,j} \text{ è la risolvente di } \mathrm{C_i}, \mathrm{C_j} \in \mathrm{S}\}.$$

*Definiamo inoltre*

$$Ris^0(S) = S$$

$$Ris^{n+1}(S) = Ris(Ris^n(S)) \text{ per } n \geq 0 \qquad \text{e}$$

$$Ris^*(S) = \bigcup_{n \geq 0} Ris^n(S)$$

Osserviamo che, cfr. Esercizio 6.6, per ogni insieme di clausole $S$, $\exists k \in \mathcal{N}$ tale che $Ris^k(S) = Ris^*(S)$.
Il teorema di risoluzione si può riformulare nel seguente modo:

**Teorema 6.21** *Un insieme di clausole $S$ è insoddisfacibile se e solo se $\square \in Ris^*(S)$.*

*Dimostrazione.* È lasciata al lettore come esercizio. □

Dal teorema precedente deriva la correttezza del seguente algoritmo , che consente di verificare se una formula $P$ è valida o meno.

1. Negare $P$.
2. Trasformare $\neg P$ in forma a clausole.
3. $S := \{(\neg P)^c\}$

4. Ripetere:

   (a) $F := S$

   (b) $S := Ris(S)$

   Finché ($\square \in S$) o ($S = F$)

5. Se $\square \in S$ allora output "$P$ è valida", altrimenti "$P$ non è valida".

Osserviamo che l'algoritmo sopra scritto termina sempre; vedremo che ciò non avviene nella logica del primo ordine.

***Osservazione*** Nell'algoritmo di p.175, per verificare che $(P_1 \wedge \cdots \wedge P_n)$ è insoddisfacibile, poiché gli elementi di $\mathcal{E}(P)$ sono a tutti gli effetti formule della logica proposizionale, è possibile utilizzare, anziché il metodo delle tabelle di verità, quello di risoluzione (previa trasformazione dei $P_i$ in forma a clausole).

***Osservazione*** Il metodo di risoluzione è in molti casi notevolmente più "efficiente" di quello delle tabelle di verità (si veda a tal proposito il paragrafo 6.2.4). Tuttavia esistono insiemi di clausole insoddisfacibili tali che ogni derivazione di $\square$ necessita di un numero di passi esponenziale ([Urq87]); per questi, il metodo di risoluzione si comporta come quello delle tabelle di verità. Essendo UNSAT un problema co-$\mathcal{NP}$, per tali insiemi, non sembrano esistere algoritmi significativamente più veloci.

### 6.2.2 Unificazione

Come già accennato in precedenza, l'algoritmo di p.175 per provare l'insoddisfacibilità di una formula, in molti casi, necessita di generare un numero proibitivo di elementi della sua espansione di Herbrand.
Il metodo di risoluzione per formule della logica del primo ordine risulta notevolmente più efficiente. L'idea alla base di tale metodo è di non effettuare prima le sostituzioni ground e poi applicare la risoluzione alle formule proposizionali ottenute, ma generalizzare quest'ultima alla logica del primo ordine facendo il "minimo indispensabile" di sostituzioni. L'approccio generale al problema di *quali* sostituzioni effettuare, conduce al concetto di unificazione.
L'unificazione viene universalmente impiegata nelle dimostrazioni automatiche dei teoremi, sia nei sistemi che utilizzano la risoluzione che in quelli che non la utilizzano (si veda a tal proposito l'algoritmo presentato nel paragrafo 5.3.2), ed in molte altre aree dell'informatica come la programmazione logica ed i sistemi di riscrittura.
Vediamo quindi il concetto di sostituzione, già discusso in 4.2:

**Definizione 6.22** *Una* sostituzione $\theta$ *è un insieme finito, eventualmente vuoto, della forma* $\{{}^{t_1}/_{x_1}, \ldots, {}^{t_n}/_{x_n}\}$ *dove*

- *ogni* $x_i$ *è una variabile distinta, i.e.* $x_1 \neq x_j$ *per* $i \neq j$
- *ogni* $t_i$ *è un termine diverso da* $x_i$

**Notazione**
Indicheremo con $\epsilon$ la sostituzione vuota (vale a dire { }).

Nel seguito considereremo sostituzioni applicate ad espressioni, che sono termini o letterali.

**Definizione 6.23** *Sia* $\sigma = \{^{t_1}/_{x_1}, \ldots, ^{t_n}/_{x_n}\}$ *una sostituzione, ed* $E$ *un'espressione;* $E\sigma$ *è l'*applicazione *di* $\sigma$ *ad* $E$ *ottenuta rimpiazzando ogni occorrenza di* $x_i$*, in* $E$*, con* $t_i$*, per* $i = 1, \ldots, n$*.*

**Esempio 6.11** Siano $\sigma = \{^{c}/_{x}, ^{g(b)}/_{y}, ^{a}/_{z}\}$ ed $E = A(x, f(y), z)$, allora $E\sigma = A(c, f(g(b)), a)$.

**Definizione 6.24** *Siano* $\sigma = \{^{v_1}/_{x_1}, \ldots, ^{v_n}/_{x_n}\}$ *e* $\theta = \{^{u_1}/_{y_1}, \ldots, ^{u_m}/_{y_m}\}$ *due sostituzioni; la* composizione *di* $\sigma$ *e* $\theta$*, indicata con* $\sigma \circ \theta$*, si ottiene a partire dall'insieme* $\{^{v_1\theta}/_{x_1}, \ldots, ^{v_n\theta}/_{x_n}, ^{u_1}/_{y_1}, \ldots, ^{u_m}/_{y_m}\}$ *cancellando tutti gli elementi*

- $^{u_j}/_{y_j}$ *per i quali* $y_j \in \{x_1, \ldots, x_n\}$
- $^{v_k\theta}/_{x_k}$ *per i quali* $v_k\theta = x_k$

**Esempio 6.12** Siano $\sigma = \{^{f(u)}/_{x}, ^{b}/_{y}, ^{y}/_{z}\}$ e $\theta = \{^{c}/_{u}, ^{a}/_{y}, ^{b}/_{z}\}$, allora $\sigma \circ \theta = \{^{f(c)}/_{x}, ^{b}/_{y}, ^{a}/_{z}\}$.

**Teorema 6.25** *Per tutte le sostituzioni* $\sigma_1, \sigma_2, \sigma_3$ *ed ogni espressione* $E$*, valgono le seguenti proprietà:*

1. $\sigma_1 \circ \epsilon = \epsilon \circ \sigma_1 = \sigma_1$
2. $(E\sigma_1)\sigma_2 = E(\sigma_1 \circ \sigma_2)$
3. $(\sigma_1 \circ \sigma_2) \circ \sigma_3 = \sigma_1 \circ (\sigma_2 \circ \sigma_3)$

*Dimostrazione.*

1. Segue dalla definizione di $\epsilon$.

2. È sufficiente provare l'affermazione quando $E$ è una variabile $x$. Siano $\sigma_1 = \{^{u_1}/_{x_1}, \ldots, ^{u_n}/_{x_n}\}$ e $\sigma_2 = \{^{v_1}/_{y_1}, \ldots, ^{v_m}/_{y_m}\}$. Vi sono tre possibilità:
   - $x \notin \{x_1, \ldots, x_n\} \cup \{y_1, \ldots, y_m\}$, allora $(x\sigma_1)\sigma_2 = x(\sigma_1 \circ \sigma_2) = x$.
   - $x = x_i$ con $i \in \{1, \ldots, n\}$, allora $(x\sigma_1)\sigma_2 = u_i\sigma_2 = x(\sigma_1 \circ \sigma_2)$.
   - $x = y_j$ con $j \in \{1, \ldots, m\}$, allora $(x\sigma_1)\sigma_2 = v_j = x(\sigma_1 \circ \sigma_2)$.

3. È lasciata al lettore come esercizio.

□

La composizione di sostituzioni non è commutativa.

Siamo interessati alle sostituzioni che unificano un insieme di espressioni, cioè che rendono ogni espressione dell'insieme sintatticamente identica.

**Definizione 6.26** *Una sostituzione $\sigma$ si dice* unificatore *di un insieme $E = \{E_1, \ldots, E_n\}$ di espressioni se $E_1\sigma = E_2\sigma = \cdots = E_n\sigma$; in tale situazione l'insieme $E$ viene detto* unificabile.

**Esempio 6.13** Sia $E = \{A(x, a), A(y, a)\}$; $E$ può essere unificato dalla sostituzione $\theta = \{{}^{b}/_{x}, {}^{b}/_{y}\}$ in quanto $A(x, a)\theta = A(y, a)\theta = A(b, a)$.
$\sigma = \{{}^{y}/_{x}\}$ è ancora un unificatore di $E$ poiché $A(x, a)\sigma = A(y, a)\sigma = A(y, a)$. Osserviamo che $\sigma$ è un unificatore "più generale" di $\theta$ in quanto dopo la sua applicazione è possibile effettuare ulteriori sostituzioni; in particolare $\theta$ si può ottenere da $\sigma$ componendo quest'ultima con la sostituzione $\{{}^{b}/_{y}\}$.

Formalizziamo tale proprietà con la seguente definizione:

**Definizione 6.27** *Un unificatore $\sigma$ per un insieme $\{E_1, \ldots, E_n\}$ di espressioni si dice* unificatore più generale *o* mgu *(most general unifier) se e solo se per ogni altro unificatore $\theta$ dello stesso insieme, esiste una sostituzione $\rho$ tale che $\theta = \sigma \circ \rho$.*

È possibile provare che ([Lov78]) se esiste un unificatore più generale di un insieme di espressioni, allora questo è *unico* a meno di ridenominazione delle variabili.

**Esempio 6.14** $\sigma = \{{}^{y}/_{x}\}$ e $\sigma' = \{{}^{x}/_{y}\}$, sono entrambi mgu per l'insieme $E$ del precedente esempio.

In [Rob65], Robinson ha presentato un algoritmo per calcolare un unificatore più generale, se esiste, di un insieme di espressioni.
L'idea intuitiva su cui si basa tale algoritmo è la seguente: supponiamo di voler unificare due espressioni. Immaginiamo di avere due puntatori, ognuno posizionato sul simbolo più a sinistra di ogni espressione; questi si spostano insieme verso destra finché non raggiungono simboli differenti, quindi si tenta di unificare le due sottoespressioni che iniziano con tali simboli effettuando una sostituzione. Se il tentativo ha successo, il procedimento continua sulle espressioni ottenute applicando tale sostituzione; altrimenti, le due espressioni non sono unificabili. Se i puntatori raggiungono la fine delle due espressioni, la composizione di tutte le sostituzioni effettuate è un mgu per esse.

**Definizione 6.28** *Sia $E$ un insieme finito di espressioni. L'* insieme di disaccordo *(disagreement set) di $E$, indicato con $D(E)$, è definito nel seguente modo: si localizzi la posizione più a sinistra in cui non tutte le espressioni di $E$ hanno lo stesso simbolo e si estragga da ogni espressione di $E$ la sottoespressione che comincia in quella posizione. L'insieme di tali sottoespressioni costituisce l'insieme di disaccordo.*

**Esempio 6.15** Sia $E = \{A(x, f(u)), A(g(y), f(h(z))\}$, allora $D(E) = \{x, g(y)\}$; posta $\sigma = \{^{g(y)}/_{x}\}$, allora $D(E\sigma) = \{u, h(z)\}$.

Osserviamo che un qualunque unificatore di $E$ unifica $D(E)$.
**Notazione**
Indichiamo con $|E\sigma|$ il numero di espressioni differenti nell'insieme $E$ al quale applichiamo la sostituzione $\sigma$.

**Algoritmo di unificazione**

1. $k := 0$ e $\sigma_k := \epsilon$
2. Se $|E\sigma_k| = 1$ allora output "$\sigma_k$" e STOP. Altrimenti, trovare l'insieme di disaccordo $D(E\sigma_k)$.
3. Se esistono $x$ e $t$ in $D(E\sigma_k)$ tali che $x$ è una variabile che non occorre in $t$, porre $\sigma_{k+1} = \sigma_k \circ \{^{t}/_{x}\}$, incrementare $k$ e tornare al passo 2. Altrimenti output "$E$ non è unificabile" e STOP.

***Osservazione*** L'algoritmo di unificazione appena presentato è non deterministico, nel senso che vi possono essere diverse scelte per $x$ e $t$ al passo 3. Tuttavia, l'applicazione di due diversi mgu prodotti dall'algoritmo porta ad espressioni che differiscono solo per i nomi delle variabili (Esercizio 6.12).

Osserviamo che è possibile effettuare la sostituzione $\{^{t}/_{x}\}$ se $x$ non compare in $t$; tale controllo, noto in letteratura come *occur check*, è necessario per garantire la terminazione dell'algoritmo; infatti, in assenza di tale condizione l'algoritmo di unificazione, nel tentativo di unificare, ad esempio, le espressioni $A(x)$ e $A(f(x))$ non terminerebbe.

**Teorema 6.29 (Unificazione)** *Ogni insieme di espressioni unificabile ha un unificatore più generale.*

*Dimostrazione.* La dimostrazione consiste nel provare che l'algoritmo di unificazione termina per tutti gli input e calcola l'unificatore più generale di un insieme di espressioni, se questo esiste.
Infatti, tale algoritmo termina sempre in quanto $E$ contiene un numero finito di variabili distinte, ed il numero di queste diminuisce di 1 ad ogni applicazione del passo 3; quindi le iterazioni dell'algoritmo saranno al più tante quante le variabili distinte che occorrono in $E$.
Sia $\theta$ un altro unificatore per $E$. Proviamo anzitutto che per $k \geq 0$, sia $\sigma_k$ la sostituzione calcolata alla k-esima iterazione, esiste una sostituzione $\gamma_k$ tale che $\theta = \sigma_k \circ \gamma_k$. Per induzione su $k$.
Caso base: $k = 0$, allora posto $\gamma_0 = \theta$, per il Teorema 6.25, l'asserto risulta verificato. Supponiamo, per ipotesi induttiva che $\exists\ \gamma_k$ tale che $\theta = \sigma_k \circ \gamma_k$. Se $|E\sigma_k| = 1$, l'algoritmo termina al passo 2 e $\sigma_k$ è l'unificatore più generale. Sia allora $|E\sigma_k| > 1$, vogliamo provare che l'algoritmo genera una sostituzione $\sigma_{k+1}$ ed esiste $\gamma_{k+1}$ tale che $\theta = \sigma_{k+1} \circ \gamma_{k+1}$. Essendo $|E\sigma_k| > 1$, l'algoritmo

determinerà l'insieme di disaccordo $D(E\sigma_k)$ di $E\sigma_k$ ed andrà al passo 3. Dal fatto che $\theta = \sigma_k \circ \gamma_k$ e $\theta$ unifica $E$, segue che $\gamma_k$ unifica $D(E\sigma_k)$.
Quindi $D(E\sigma_k)$ dovrà contenere una variabile $x$ ed un termine $t$ tale che $x$ non occorre in $t$. Allora l'algoritmo porrà $\sigma_{k+1} = \sigma_k \circ \{^t/_x\}$. Definiamo $\gamma_{k+1} = \gamma_k - \{^{t\gamma_k}/_x\}$. Poiché $x$ non occorre in $t$, risulta $t\gamma_{k+1} = t(\gamma_k - \{^{t\gamma_k}/_x\}) = t\gamma_k$. Quindi

$$\begin{aligned}\gamma_k &= \{^{t\gamma_k}/_x\} \cup \{\gamma_k - \{^{t\gamma_k}/_x\}\} \\ &= \{^{t\gamma_k}/_x\} \cup \gamma_{k+1} \\ &= \{^{t\gamma_{k+1}}/_x\} \cup \gamma_{k+1} \\ &= \{^t/_x\} \circ \gamma_{k+1}\end{aligned}$$

Da cui
$\theta = \sigma_k \circ \gamma_k = \sigma_k \circ \{^t/_x\} \circ \gamma_{k+1} =$ (per il Teorema 6.25) $= \sigma_{k+1} \circ \gamma_{k+1}$
Dunque $\forall k \geq 0$ esiste una sostituzione $\gamma_k$ tale che $\theta = \sigma_k \circ \gamma_k$. Quindi se $E$ è unificabile, supponiamo che l'algoritmo termini all'n-esima iterazione, allora $\theta = \sigma_n \circ \gamma_n$, quindi $\sigma_n$ è un unificatore più generale per $E$. □

Nella tabella che segue vengono presentate sinteticamente le regole che sono alla base dell'algoritmo di unificazione tra due espressioni.
Siano

| | |
|---|---|
| $C_{1,2}$ | costanti |
| $X_{1,2}$ | variabili |
| $S_{1,2}$ | applicazioni di funzioni o predicati |

| | $C_2$ | $X_2$ | $S_2$ |
|---|---|---|---|
| $C_1$ | se $C_1 = C_2$ | $\{^{C_1}/_{X_2}\}$ | NO |
| $X_1$ | $\{^{C_2}/_{X_1}\}$ | $\{^{X_2}/_{X_1}\}$ | $\{^{S_2}/_{X_1}\}$ previo *o.c.* |
| $S_1$ | NO | $\{^{S_1}/_{X_2}\}$ previo *o.c.* | $*$ |

nel caso $*$, $S_1$ ed $S_2$ possono essere unificati se:

- il primo simbolo di $S_1$ (nome della funzione o del predicato) ed il primo simbolo di $S_2$ sono uguali,
- quindi si procede, utilizzando la tabella, ricorsivamente sugli argomenti.

**Esempio 6.16**

- $t_1 = a$ e $t_2 = f(x, y)$ non sono unificabili.
- $t_1 = a$ e $t_2 = b$ non sono unificabili.
- $t_1 = f(x)$ e $t_2 = f(g(x))$ non sono unificabili.
- $t_1 = f(h(x))$ e $t_2 = f(h(P(y)))$ sono unificabili da $\{^{P(y)}/_x\}$.

**Esempio 6.17** Sia

$$E = \{A(f(y, g(v)), h(b)), A(f(h(w), g(a)), t), A(f(h(b), g(v)), t)\}$$

Applichiamo l'algoritmo di unificazione per trovare, se esiste, un unificatore più generale per $E$.

**Passo 1** : $|E\sigma_0| > 1$; $D(E\sigma_0) = \{y, h(w), h(b)\}$; $\sigma_1 = \{^{h(w)}/_{y}\}$ allora $E\sigma_1 = \{A(f(h(w), g(v)), h(b)), A(f(h(w), g(a)), t), A(f(h(b), g(v)), t)\}$.

**Passo 2** : $|E\sigma_1| > 1$; $D(E\sigma_1) = \{w, b\}$; $\sigma_2 = \sigma_1 \circ \{^{b}/_{w}\}$, allora $E\sigma_2 = \{A(f(h(b), g(v)), h(b)), A(f(h(b), g(a)), t), A(f(h(b), g(v)), t)\}$.

**Passo 3** : $|E\sigma_2| > 1$; $D(E\sigma_2) = \{v, a\}$; $\sigma_3 = \sigma_2 \circ \{^{a}/_{v}\}$, allora $E\sigma_3 = \{A(f(h(b), g(a)), h(b)), A(f(h(b), g(a)), t), A(f(h(b), g(a)), t)\}$.

**Passo 4** : $|E\sigma_3| > 1$; $D(E\sigma_3) = \{h(b), t\}$; $\sigma_4 = \sigma_3 \circ \{^{h(b)}/_{t}\}$, allora $E\sigma_4 = \{A(f(h(b), g(a)), h(b)), A(f(h(b), g(a)), h(b)), A(f(h(b), g(a)), h(b))\}$.

**Passo 5** : $|E\sigma_4| = 1$; $\sigma_4 = \{^{h(w)}/_{y}, ^{b}/_{w}, ^{a}/_{v}, ^{h(b)}/_{t}\}$ è un unificatore più generale per $E$.

***Osservazione*** L'algoritmo di unificazione descritto in precedenza è inefficiente. In alcuni casi il suo tempo di esecuzione è una funzione esponenziale della lunghezza dell'input. Consideriamo infatti il seguente esempio:

$$E = \{A(x_1, \ldots x_n), A(f(x_0, x_0), \ldots, f(x_{n-1}, x_{n-1}))\}$$

allora $D(E\sigma_0) = \{x_1, f(x_0, x_0)\}$; $\sigma_1 = \{^{f(x_0,x_0)}/_{x_1}\}$
$E\sigma_1 = \{A(f(x_0, x_0), x_2, \ldots, x_n), A(f(x_0, x_0), f(f(x_0, x_0), f(x_0, x_0)), \ldots$
$f(x_{n-1}, x_{n-1}))\}$
$D(E\sigma_1) = \{f(f(x_0, x_0), f(x_0, x_0)), x_2\}$; $\sigma_2 = \{^{f(f(x_0,x_0),f(x_0,x_0))}/_{x_2}\}$
$\vdots$
osserviamo che la seconda espressione in $E\sigma_n$ ha $2^{n-1}$ occorrenze di $f$, quindi l'esecuzione dell'*occur check* richiede tempo esponenziale. Algoritmi di unificazione più efficienti (quasi lineari) sono descritti in [PW78] e [MM82].

### 6.2.3 Risoluzione nella logica del primo ordine

Come nel caso proposizionale, il metodo di risoluzione consta di un'unica regola di inferenza: la risoluzione, che si applica a formule in forma a clausole. Ricordiamo che una formula della logica del primo ordine può essere trasformata in un'altra chiusa e priva di quantificatori esistenziali che è soddisfacibile se e solo se lo è la formula di partenza (Proposizione 4.41). La formula risultante sarà, in generale, del tipo

$$\forall x_1 x_2 \ldots x_n P$$

è possibile trasformarne la matrice in forma normale congiuntiva

$$\forall x_1 x_2 \ldots x_n (C_1 \wedge C_2 \wedge \cdots \wedge C_m)$$

dove le $C_i$, per $i = 1, \ldots, m$ sono clausole ed $x_j$ per $j = 1, \ldots, n$ le sole variabili che vi compaiono. Questa è equivalente a:

$$\forall x_1 x_2 \ldots x_n C_1 \wedge \cdots \wedge \forall x_1 x_2 \ldots x_n C_m$$

È quindi possibile eliminare i quantificatori ottenendo $C_1 \wedge \cdots \wedge C_m$ che, con considerazioni analoghe a quelle effettuate nel paragrafo 6.2.1 si può riscrivere come $\{C_1, \ldots, C_m\}$.
Vediamo in che modo integrare la regola di risoluzione per la logica proposizionale con l'unificazione, per ottenere la regola generale di risoluzione.

**Definizione 6.30** *Siano $C_1, C_2$ ed $R$ clausole. Si dice che $R$ è una* risolvente *di $C_1$ e $C_2$ se sono verificate le seguenti condizioni:*

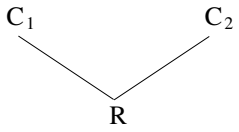
1. *esistono due sostituzioni $s_1$ ed $s_2$ che ridenominano opportunamente le variabili in modo tale che $C_1 s_1$ e $C_2 s_2$ non abbiano variabili in comune;*

2. *esiste un insieme di letterali $l_1, \ldots, l_m \in C_1 s_1$ (con $m \geq 1$) ed $l'_1, \ldots, l'_n \in C_2 s_2$ (con $n \geq 1$) tali che l'insieme $L = \{\bar{l}_1, \ldots, \bar{l}_m, l'_1, \ldots l'_n\}$ è unificabile. Sia $\sigma$ l'unificatore più generale per $L$;*

3. *$R$ ha la forma*

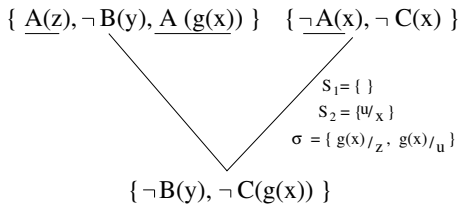
$$((C_1 s_1 - \{l_1, \ldots, l_m\} \cup (C_2 s_2 - \{l'_1, \ldots, l'_n\}))\sigma$$

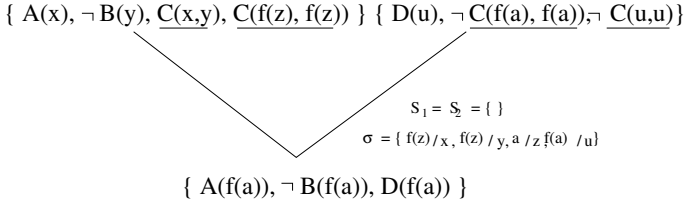
Graficamente possiamo denotare la situazione con il seguente diagramma:




Per migliorare la leggibilità, i letterali $l_1, \ldots, l_m, l'_1, \ldots, l'_n$, coinvolti nella generazione della risolvente di due clausole, verranno sottolineati.

**Esempio 6.18** Esempi di applicazione della regola di risoluzione sono:



***Osservazione*** È opportuno sottolineare che, nella Definizione 6.30 di risolvente di due clausole:

1. la ridenominazione delle variabili è necessaria; ad esempio, l'insieme

$$\{\{A(x)\}, \{\neg A(f(x))\}\}$$

è insoddisfacibile ma, a causa delle restrizioni fatte sull'algoritmo di unificazione, le due clausole non possono essere risolte se non si cambia nome alla variabile in una delle due;

2. non è possibile assumere che $m$ o $n$ siano uguali a 1, in quanto, nella logica del primo ordine accade di dover eliminare più letterali alla volta; tale operazione è nota in letteratura con il nome di *fattorizzazione*. Infatti, se consideriamo ad esempio l'insieme

$$S = \{\{A(x), A(y)\}, \{\neg A(x), \neg A(y)\}\}$$

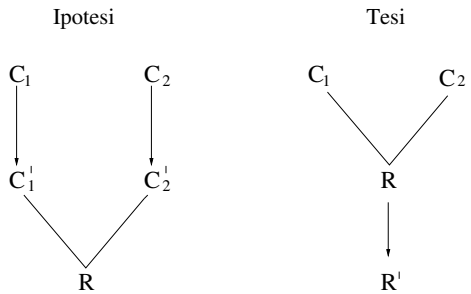
questo è chiaramente insoddisfacibile ma nessuna prova per risoluzione, che elimina un solo letterale ad ogni applicazione della regola, può produrre da $S$ la clausola vuota (il lettore lo verifichi per esercizio).

***Osservazione*** La regola di risoluzione della logica proposizionale è un caso particolare di quella del primo ordine, infatti, la prima è ottenibile ponendo $s_1 = s_2 = \epsilon$ ed $m = n = 1$ nella Definizione 6.30.

In analogia con quanto è stato fatto nel paragrafo 6.2.1, proviamo che il metodo di risoluzione della logica del primo ordine, è corretto e completo rispetto all'insoddisfacibilità; l'idea è quella di ricondursi al caso proposizionale. Per fare ciò vediamo un lemma, noto in letteratura come *lifting lemma* che collega la regola di risoluzione proposizionale e quella del primo ordine.

**Lemma 6.31** *Siano $C_1$ e $C_2$ clausole della logica del primo ordine e $C_1'$ e $C_2'$ due arbitrarie istanze ground di $C_1$ e $C_2$ risolvibili (mediante la regola di risoluzione della logica proposizionale). Sia $R'$ la risolvente di $C_1'$ e $C_2'$. Allora esiste una clausola $R$ che è la risolvente di $C_1$ e $C_2$ (mediante la regola generale di risoluzione) tale che $R'$ è un'istanza ground di $R$.*

Graficamente possiamo denotare la situazione nel seguente modo:



*Dimostrazione.* Siano $s_1$ ed $s_2$ due sostituzioni che ridenominano le variabili in modo tale che $C_1 s_1$ e $C_2 s_2$ non abbiano variabili in comune. Essendo $C_1'$ e $C_2'$ istanze ground rispettivamente di $C_1$ e $C_2$, lo sono anche di $C_1 s_1$ e $C_2 s_2$. Siano $\sigma_1$ e $\sigma_2$ le sostituzioni tali che $C_1' = C_1 s_1 \sigma_1$ e $C_2' = C_2 s_2 \sigma_2$. Poiché non vi sono variabili rimpiazzate sia in $\sigma_1$ che in $\sigma_2$, posta $\sigma = \sigma_1 \circ \sigma_2$ risulta $C_1' = C_1 s_1 \sigma$ e $C_2' = C_2 s_2 \sigma$. Per ipotesi, $R'$ è la risolvente di $C_1'$ e $C_2'$, quindi deve esistere un letterale $l \in C_1'$ tale che $\bar{l} \in C_2'$ ed $R' = (C_1' - \{l\}) \cup (C_2' - \{\bar{l}\})$. Ma $l$ ed $\bar{l}$ risultano dall'applicazione di $\sigma$ ad uno o più letterali, rispettivamente, di $C_1 s_1$ e di $C_2 s_2$; dunque esistono $l_1, \ldots, l_m \in C_1 s_1$ (con $m \geq 1$) ed $l_1', \ldots, l_n' \in C_2 s_2$ (con $n \geq 1$) tali che $l = l_1\sigma = \cdots = l_m\sigma$ e $\bar{l} = l_1'\sigma = \cdots = l_n'\sigma$; quindi $C_1$ e $C_2$ sono risolvibili essendo $\sigma$ un unificatore per l'insieme di letterali $L = \{l_1, \ldots, l_m, \bar{l}_1', \ldots, \bar{l}_n'\}$. Sia $\theta$ un mgu per $L$, allora $R = ((C_1 s_1 - \{l_1, \ldots, l_m\}) \cup (C_2 s_2 - \{l_1', \ldots, l_n'\}))\theta$ è una risolvente di $C_1$ e $C_2$; essendo $\theta$ un mgu e $\sigma$ un unificatore per $L$, esiste una sostituzione $s$ tale che $\theta \circ s = \sigma$. Allora:

$$\begin{aligned} R' &= (C_1' - \{l\}) \cup (C_2' - \{\bar{l}\}) \\ &= (C_1 s_1 \sigma - \{l\}) \cup (C_2 s_2 \sigma - \{\bar{l}\}) \\ &= ((C_1 s_1 - \{l_1, \ldots, l_m\}) \cup (C_2 s_2 - \{l_1', \ldots, l_n'\}))\sigma \\ &= ((C_1 s_1 - \{l_1, \ldots, l_m\}) \cup (C_2 s_2 - \{l_1', \ldots, l_n'\}))\theta \circ s \\ &= Rs \end{aligned}$$

Dunque $R'$ è un'istanza ground di $R$. □

Come nel caso proposizionale, proviamo che la risoluzione è una regola di inferenza che preserva l'equivalenza.

**Lemma 6.32** *Siano $C_1$ e $C_2$ clausole della logica del primo ordine ed $R$ una loro risolvente; $R$ è conseguenza logica di $C_1$ e $C_2$.*

*Dimostrazione.* Sia $(\mathcal{A}, \xi^{\mathcal{A}})$ un'interpretazione che è un modello per $C_1$ e $C_2$,cioè $\mathcal{A} \models C_1$ e $\mathcal{A} \models C_2$. Assumiamo, per fissare le idee che $R$ abbia la forma

$$\begin{aligned} R &= ((C_1 s_1 - \{l_1, \ldots, l_m\}) \cup (C_2 s_2 - \{l_1', \ldots, l_n'\}))\theta \\ &= (C_1 s_1 \theta - \{l\}) \cup (C_2 s_2 \theta - \{\bar{l}\}) \end{aligned}$$

dove $\theta$ è un unificatore più generale dell'insieme $\{l_1, \ldots, l_m, \bar{l}_1', \ldots, \bar{l}_n'\}$ ed $l = l_1\theta = \cdots = l_m\theta = \bar{l}_1'\theta = \cdots = \bar{l}_n'\theta$. Proviamo che $\mathcal{A} \models R$. Vi sono due possibilità:

1. $v^{(\mathcal{A},\xi^{\mathcal{A}})}(l) = 1$. Allora da $v^{(\mathcal{A},\xi^{\mathcal{A}})}(C_2 s_2 \theta) = 1$ e da $v^{(\mathcal{A},\xi^{\mathcal{A}})}(\bar{l}) = 0$ segue $v^{(\mathcal{A},\xi^{\mathcal{A}})}(C_2 s_2 \theta - \bar{l}) = 1$ e quindi $\mathcal{A} \models R$.

2. $v^{(\mathcal{A},\xi^{\mathcal{A}})}(l) = 0$. Allora da $v^{(\mathcal{A},\xi^{\mathcal{A}})}(C_1 s_1 \theta) = 1$ segue $v^{(\mathcal{A},\xi^{\mathcal{A}})}(C_1 s_1 \theta - l) = 1$ e quindi $\mathcal{A} \models R$.

□

**Teorema 6.33 (di Risoluzione)**
*Un insieme di clausole $S$ è insoddisfacibile se e solo se $S \vdash_R \square$.*

*Dimostrazione.* Assumiamo, senza perdere di generalitá, che $S$ sia finito, in quanto, in caso contrario, per il teorema di compattezza potremmo comunque operare su un suo sottoinsieme finito.
(*Correttezza*) La dimostrazione è analoga al caso proposizionale.
(*Completezza*) Supponiamo che $S$ sia insoddisfacibile, proviamo che esiste una prova della clausola vuota da $S$. Per il teorema di Herbrand, esiste un insieme $S'$ di istanze ground delle clausole di $S$ che è insoddisfacibile; dal teorema di risoluzione per formule della logica proposizionale segue l'esistenza di una sequenza di clausole $C'_1, \ldots, C'_k$ tali che $C'_k = \Box$ e $C'_i$ è un elemento di $S'$ oppure la risolvente di due clausole $C'_a$ e $C'_b$ con $a, b < i$, per $i = 1, \ldots, k$. Per il Lemma 6.31, da $C'_1, \ldots, C'_k$ è possibile trovare una sequenza di clausole $C_1, \ldots, C_k$ tali che $C_k = \Box$ e $C_i$ è un elemento di $S$ oppure la risolvente di due clausole $C_a$ e $C_b$ con $a, b < i$, per $i = 1, \ldots, k$, che dimostra l'asserto. $\Box$

Come nel caso proposizionale, definiamo l'insieme dei risolventi di un insieme di clausole:

**Definizione 6.34**

$$Ris(S) = S \cup \{C_{i,j} \mid C_{i,j} \text{ è la risolvente di } \mathrm{C_i}, \mathrm{C_j} \in \mathrm{S}\}.$$

$$Ris^0(S) = S$$

$$Ris^{n+1}(S) = Ris(Ris^n(S)) \; per \; n \geq 0$$

$$Ris^*(S) = \bigcup_{n \geq 0} Ris^n(S)$$

Il teorema di risoluzione si può riformulare nel seguente modo:

**Teorema 6.35** *Un insieme di clausole $S$ è insoddisfacibile se e solo se $\Box \in Ris^*(S)$.*

Dal teorema precedente deriva la correttezza del seguente algoritmo che consente di verificare se una formula $P$ della logica del primo ordine è valida.

1. Negare $P$
2. Trasformare $\neg P$ in forma a clausole
3. $S := (\neg P)^c$
4. Ripetere:
   $S := Ris(S)$
   Finché $\Box \in S$
5. Output "$P$ è insoddisfacibile."

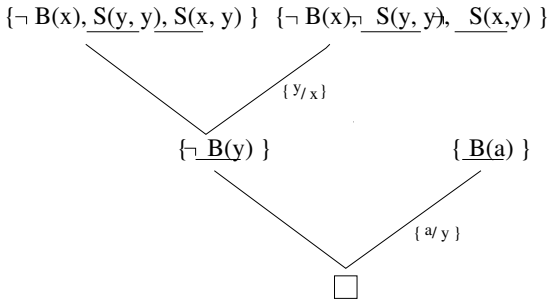
Osserviamo che, in accordo con quanto discusso nella sezione 5.4.1, tale algoritmo, se la formula di partenza è soddisfacibile, può non terminare. Infatti, nella logica del primo ordine, non è vero che per ogni insieme $S$ di clausole, $\exists k \in \mathcal{N}$ tale che $Ris^k(S) = Ris^*(S)$. Ad esempio, sia

$$S = \{A(0), \neg A(x) \vee A(s(x))\}$$

i suoi risolventi avranno la forma $A(s(0)), A(s(s(0))), A(s(s(s(0)))), \ldots$ [11]; $Ris^*(S)$ è infinito.

***Osservazione*** Nell'algoritmo precedente, al contrario di quanto avviene per quello di p.182, non è possibile invertire l'ordine delle istruzioni 1. e 2. Infatti, ricordiamo che, nella logica dei predicati, la trasformazione di una formula in forma a clausole non preserva l'equivalenza ma soltanto la soddisfacibilità (Proposizione 4.41).

**Esempio 6.19** Si considerino le seguenti affermazioni:
(a) Ogni barbiere rade tutti coloro che non si radono da soli.
(b) Nessun barbiere rade chi si rade da solo.
Dimostriamo, utilizzando il metodo di risoluzione che
(c) Non esistono barbieri
è conseguenza logica di (a) e (b). Siano:
$B(x)$ "$x$ è un barbiere" e
$S(x, y)$ "$x$ rade $y$ "
La forma a clausole delle precedenti affermazioni è:
(a) $\{\neg B(x), S(y, y), S(x, y)\}$
(b) $\{\neg B(x), \neg S(y, y), \neg S(x, y)\}$
(c) $\{B(a)\}$
Allora:




## 6.2.4 Raffinamenti

Benchè il metodo di risoluzione risulti notevolmente più efficiente dell'algoritmo suggerito dal teorema di Herbrand, può comunque generare numerose clausole irrilevanti e ridondanti. Infatti, consideriamo il seguente esempio:

[11] È una rappresentazione dei numeri naturali.

**Esempio 6.20** Sia $S = \{\{A, B\}, \{\neg A, B\}, \{A, \neg B\}, \{\neg A, \neg B\}\}$, vogliamo provare che $S$ è insoddisfacibile.

$\text{Ris}^0(S)$ :

| | | |
|---|---|---|
| (1) | $\{A, B\}$ | |
| (2) | $\{\neg A, B\}$ | |
| (3) | $\{A, \neg B\}$ | |
| (4) | $\{\neg A, \neg B\}$ | |

$\text{Ris}^1(S)$ :

| | | |
|---|---|---|
| (5) | $\{B\}$ | (risolvente di (1) e (2)) |
| (6) | $\{A\}$ | (risolvente di (1) e (3)) |
| (7) | $\{B, \neg B\}$ | (risolvente di (1) e (4)) |
| (8) | $\{A, \neg A\}$ | (risolvente di (1) e (4)) |
| (9) | $\{B, \neg B\}$ | (risolvente di (2) e (3)) |
| (10) | $\{A, \neg A\}$ | (risolvente di (2) e (3)) |
| (11) | $\{\neg A\}$ | (risolvente di (2) e (4)) |
| (12) | $\{\neg B\}$ | (risolvente di (3) e (4)) |

$\text{Ris}^2(S)$:

| | | |
|---|---|---|
| (13) | $\{A, B\}$ | (risolvente di (1) e (7)) |
| (14) | $\{A, B\}$ | (risolvente di (1) e (8)) |
| (15) | $\{A, B\}$ | (risolvente di (1) e (9)) |
| (16) | $\{A, B\}$ | (risolvente di (1) e (10)) |
| (17) | $\{B\}$ | (risolvente di (1) e (11)) |
| (18) | $\{A\}$ | (risolvente di (1) e (12)) |
| (19) | $\{B\}$ | (risolvente di (2) e (6)) |
| (20) | $\{\neg A, B\}$ | (risolvente di (2) e (7)) |
| (21) | $\{\neg A, B\}$ | (risolvente di (2) e (8)) |
| (22) | $\{\neg A, B\}$ | (risolvente di (2) e (9)) |
| (23) | $\{\neg A, B\}$ | (risolvente di (2) e (10)) |
| (24) | $\{\neg A\}$ | (risolvente di (2) e (12)) |
| (25) | $\{A\}$ | (risolvente di (3) e (5)) |
| (26) | $\{A, \neg B\}$ | (risolvente di (3) e (7)) |
| (27) | $\{A, \neg B\}$ | (risolvente di (3) e (8)) |
| (28) | $\{A, \neg B\}$ | (risolvente di (3) e (9)) |
| (29) | $\{A, \neg B\}$ | (risolvente di (3) e (10)) |
| (30) | $\{\neg B\}$ | (risolvente di (3) e (11)) |
| (31) | $\{\neg A\}$ | (risolvente di (4) e (5)) |
| (32) | $\{\neg B\}$ | (risolvente di (4) e (6)) |
| (33) | $\{\neg A, \neg B\}$ | (risolvente di (4) e (7)) |
| (34) | $\{\neg A, \neg B\}$ | (risolvente di (4) e (8)) |
| (35) | $\{\neg A, \neg B\}$ | (risolvente di (4) e (9)) |
| (36) | $\{\neg A, \neg B\}$ | (risolvente di (4) e (10)) |
| (37) | $\{B\}$ | (risolvente di (5) e (7)) |
| (38) | $\{B\}$ | (risolvente di (5) e (9)) |
| (39) | $\square$ | (risolvente di (5) e (12)) |

Per ridurre il numero di risolventi generati ad ogni passo, è possibile operare delle **semplificazioni** ([DRW86]). Esempi di semplificazioni sono:

- eliminazione delle clausole che sono tautologie (si veda a tal proposito l'Esercizio 6.10)
- eliminazione delle clausole già generate
- eliminazione delle clausole che sono "comprese" in altre[12];

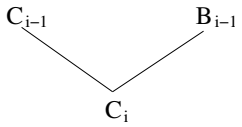
ad **Esempio**

(5) $\{B\}$
(13) $\{A, B\}$

(5) comprende già (13), ed in particolare se (5) è soddisfacibile allora anche (13) lo è.

Tuttavia, pur effettuando tali semplificazioni, in molti casi, il numero di risolventi generati può essere comunque troppo grande. Per tale motivo, per migliorare le prestazioni dei dimostratori automatici dei teoremi, si utilizzano delle **strategie** che scelgono in modo opportuno le clausole da cui generare una risolvente. Vi sono strategie *complete*, i.e. che garantiscono sempre la derivazione della clausola vuota da un insieme insoddisfacibile di clausole, ed *incomplete*. Ovviamente, quando si considerano dei raffinamenti al metodo di risoluzione, sarebbe preferibile che questi fossero completi, tuttavia, poiché l'efficienza assume un ruolo di primaria importanza nelle dimostrazioni automatiche dei teoremi, sono largamente utilizzate strategie efficienti ma incomplete, tali però da essere in grado di dimostrare un'ampia "classe" di teoremi. Descriveremo brevemente qui di seguito alcune di queste. Per approfondimenti si rimanda alla bibliografia consigliata nel paragrafo 6.3.

**Definizione 6.36** *Una prova per* risoluzione lineare *di $C$ da un insieme di clausole $S$, è una sequenza $C_1, \ldots, C_n$ tale che $C_1 \in S$, $C_n = C$ e $\forall i = 2, \ldots, n$ risulta*

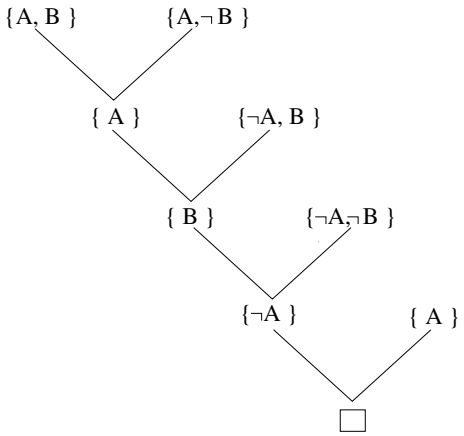



*dove la clausola $B_{i-1}$, detta* clausola laterale*, appartiene ad $S$ oppure $B_{i-1} = C_j$ con $j < i$.*

In altri termini, una prova per risoluzione lineare è una derivazione per risoluzione, nella quale si utilizza sempre la risolvente ottenuta al passo precedente.

[12] In generale, si dice che una clausola $C_1$ è compresa in una clausola $C_2$ se esiste una sostituzione $\theta$ tale che ogni letterale in $C_2\theta$ compare in $C_1$.

**Esempio 6.21** Sia $S = \{A \vee B, A \vee \neg B, \neg A \vee B, \neg A \vee \neg B\}$; la refutazione usuale è quella descritta nell'Esempio 6.9. Una refutazione per risoluzione lineare è:




**Teorema 6.37** *La risoluzione lineare è completa (per refutazione), i.e. se un insieme di clausole $S$ è insoddisfacibile, allora la clausola vuota è derivabile per risoluzione lineare.*

Osserviamo che anche nella risoluzione lineare è necessario "ricordare" tutte le risolventi calcolate in precedenza; volendo implementare tale metodo, ciò risulta oneroso sia per quanto riguarda l'occupazione di memoria che per il tempo necessario a scegliere le clausole da cui derivare una risolvente. Vediamo quindi una strategia più efficiente.

**Definizione 6.38** *Una prova per* risoluzione di input *di $C$ da un insieme di clausole $S$, è una sequenza $C_1, \ldots, C_n$ tali che $C_n = C$ e ad ogni passo una delle clausole risolventi è un (istanza di[13]) elemento di $S$.*

La risoluzione di input non è completa (per refutazione). Ad esempio, l'insieme $S = \{A \vee B, A \vee \neg B, \neg A \vee B, \neg A \vee \neg B\}$ è insoddisfacibile (Esempio 6.9), ma non esiste una prova di $\square$ per risoluzione di input da $S$.
Se si restringe l'insieme delle clausole alle quali applicare la risoluzione di input a clausole in una forma particolare detta *di Horn*, la risoluzione di input risulta completa (per refutazione).

**Definizione 6.39** *Una* clausola di Horn *è una clausola che contiene al più una formula atomica positiva.*

**Esempio 6.22** $A \vee \neg B$, $\neg A \vee \neg B \vee \neg C \vee D$, $A, \neg B$
sono clausole di Horn.

**Terminologia**
$A_0 \vee \neg A_1 \vee \cdots \vee \neg A_k$ è detta clausola di Horn definita, mentre $\neg A_0 \vee \neg A_1 \vee \cdots \vee \neg A_k$ è detta clausola goal.

[13] Se si considerano clausole della logica del primo ordine

**Teorema 6.40** *La risoluzione di input è completa (per refutazione) per clausole di Horn.*

**Definizione 6.41** *Una prova per* risoluzione SLD[14] *è una derivazione per risoluzione lineare di input tale che ogni risolvente laterale è una clausola di Horn definita mentre ogni altro risolvente è una clausola goal.*

**Teorema 6.42** *La risoluzione SLD è completa per clausole di Horn.*

Tale strategia è utilizzata dal linguaggio di programmazione *Prolog*.

## 6.3 Cenni storici e bibliografici

Uno dei problemi principali della logica matematica è stato da sempre quello di cercare una procedura in grado di stabilire se una formula è valida o meno (*Entscheidungsproblem*[15]). Tale ricerca ha infatti avuto inizio diversi secoli fa. Venne inizialmente tentata da Leibniz (1646-1716), successivamente fu ripresa da Peano e, attorno agli anni venti, dalla scuola di Hilbert. Soltanto con Church e Turing ([Chu36, Tur36]) fu possibile dimostrare la non esistenza di tale procedura; questi, infatti, provarono, indipendentemente l'uno dall'altro, che è possibile soltanto semidecidere la validità di una formula, cioè stabilire se questa è valida, ma non se non lo è. Per una trattazione formale del problema dell'indecidibilità della logica del primo ordine si vedano Kleene ([Kle67]), Bell e Machover ([BM77]), Boolos e Jeffrey ([BJ74]), Rogers ([Rog71]) e Mendelson ([Men64]).
Un approccio molto importante alla dimostrazione automatica dei teoremi venne sviluppato da Herbrand nel 1930 [Her30].
L'idea sottostante l'algoritmo di p.175 è presente in Skolem ([Sko28]). I primi tentativi di implementarla sono stati effettuati in modo indipendente da Gilmore ([Gil59]), Prawitz, Prawitz e Voghera ([PPV60]) e Wang ([Wan60]). Procedure più efficienti sono state sviluppate da Dunham, Fridsal e Sward ([DFS59]) e da Davis e Putnam ([DP60]); queste non erano però sufficientemente veloci da dimostrare in tempo ragionevole, utilizzando un calcolatore, tutte le formule valide. Un notevole passo avanti in tale direzione è stato ottenuto da Robinson con il metodo di risoluzione ([Rob65, Rob68, Rob79]); alcuni miglioramenti apportati a tale metodo, allo scopo di aumentarne l'efficienza, sono: la *risoluzione semantica*, *con blocco*, *lineare*, *con clausole unitarie*, la *theory resolution* e la *strategia dell'insieme di supporto*. Sull'argomento, si vedano [CL73, Lov78, NS93, Duf92, Lol91]. Per un approccio formale alla programmazione logica si rimanda a [Sch89, Llo87], quest'ultimo fornisce, tra le altre cose, un'ampia bibliografia sull'argomento. Tra i tanti testi di introduzione alla programmazione logica si segnalano [Bra86, CLM90].

[14] Risoluzione Lineare con funzione di Selezione per clausole Definite.

[15] "*Entscheidungsproblem must be termed the main problem of mathematical logic*" (Hilbert e Ackermann 1928). "*The central problem of the mathematical logic ... is the Entscheidungsproblem*" (Bernays e Schönfikel 1928).

## Esercizi

**6.1** Sia $\mathcal{L}$ il linguaggio del primo ordine contenente i simboli di funzione unari $f$ e $g$, il simbolo di funzione binario $h$ ed i predicati binari $A, B, C$, e sia $P$ la fbf $\forall x \forall y \forall z (A(f(y), g(y)) \wedge B(h(x, z), z))$; determinare:

1. $H(\mathcal{L})$
2. $B(\mathcal{L})$
3. $\mathcal{E}(P)$
4. Un modello di Herbrand per $P$.

**6.2** Sia $\mathcal{L}$ un linguaggio del primo ordine avente i simboli di costanti $a, b, c$ ed il predicato unario $A$. Quante sono le possibili interpretazioni di Herbrand per $\mathcal{L}$?

**6.3** Provare, utilizzando il teorema di Herbrand, che gli insiemi di formule:

1. $\{P(x), \neg P(f(a))\}$
2. $\{\neg P(x) \vee Q(f(x), x), P(g(b)), \neg Q(y, z)\}$
3. $\{P(a), \neg P(x) \vee Q(f(x)), \neg Q(f(a))\}$

sono insoddisfacibili.

**6.4** Mostrare che se $P$ è una fbf chiusa, skolemizzata rispetto ai quantificatori universali (Esercizio 4.13), allora il teorema di Herbrand si può formulare nel seguente modo:
Sia $\mathcal{E}(P) = \{P_1, P_2, \dots\}$; $P$ è valida se e solo se esiste un sottoinsieme finito dell'espansione di Herbrand $P_1 \vee \cdots \vee P_n$ che lo è.

**6.5** Dimostrare che gli insiemi di fbf della logica del primo ordine:

1. che non contengono variabili (e quindi quantificatori)
2. che non contengono simboli di funzione ed il quantificatore esistenziale
3. che non contengono simboli di funzione ed il quantificatore universale
4. che non contengono simboli di funzione ed hanno un numero finito di costanti
5. monadiche (si veda l'Esercizio 4.16)

sono decidibili.

**6.6** Provare che per ogni insieme *finito* di clausole $S$ della logica proposizionale, esiste un $k \in \mathcal{N}$ tale che

$$Ris^k(S) = Ris^{k+1}(S) = \cdots = Ris^*(S).$$

Se $S$ contiene $n$ clausole formate a partire dalle proposizioni atomiche $A_1, \ldots, A_n$; qual'è il numero massimo di elementi in $Ris^*(S)$?.

**6.7** Mostrare che $A \wedge B \wedge C$ è una conseguenza logica dell'insieme di clausole $S = \{\{\neg B, C\}, \{\neg A, B\}, \{A, \neg C\}, \{A, B, C\}\}$.

**6.8** Provare che un insieme di clausole $S$ è soddisfacibile se e solo se $S^l = \{C - \{l\} \mid C \in S \wedge\ \bar{l} \notin C\}$ oppure $S^{\bar{l}} = \{C - \{\bar{l}\} \mid C \in S \wedge\ l \notin C\}$ lo è.

**6.9** Quali dei seguenti insiemi di clausole sono soddisfacibili?

1. $\{\{A, \neg B\}, \{\neg A, B\}\}$
2. $\{\{A, B\}, \{\neg A, \neg B\}, \{\neg A, B\}\}$
3. $\{\{\neg A\}, \{A, \neg B\}, \{B\}\}$
4. $\{\{\neg A, C\}, \{\neg B\}, \{B\}, \Box\}$
5. $\{\{A, \neg B\}, \{A, B\}, \{\neg A\}\}$

Per gli insiemi di clausole soddisfacibili trovare un'interpretazione che è un modello.

**6.10** Dimostrare che se si restringe l'applicazione della regola di risoluzione a clausole che non sono tautologie, il metodo di risoluzione risulta ancora completo (rispetto all'insoddisfacibilità).

**6.11** Supponiamo di poter effettuare le seguenti reazioni chimiche:

- $O \wedge H_2 \rightarrow H_2O$
- $C \wedge O_2 \rightarrow CO_2$
- $CO_2 \wedge H_2O \rightarrow H_2CO_3$

e di disporre di alcune quantità di $O, H_2, O_2$ e $C$. Mostrare che è possibile ottenere $H_2CO_3$.

**6.12** Siano $\sigma_1$ e $\sigma_2$ due sostituzioni tali che $\sigma_1 = \sigma_2 \circ \theta_1$ e $\sigma_2 = \sigma_1 \circ \theta_2$. Mostrare che esiste una sostituzione $\gamma$, il cui unico effetto è di cambiare nome alle variabili, tale che $\sigma_1 = \sigma_2 \circ \gamma$.

**6.13** Applicare l'algoritmo di unificazione a ciascuno dei seguenti insiemi per trovare un mgu o mostrare che non esiste:

1. $\{A(x, y), A(y, f(z))\}$

2. $\{B(a, y, f(y)), B(z, z, u)\}$
3. $\{A(x, g(x)), A(y, y)\}$
4. $\{B(x, g(x), y), B(z, u, g(a)), B(a, g(a), v)\}$
5. $\{A(g(x), y), A(y, y), A(y, f(u))\}$

**6.14** Trovare i risolventi delle seguenti clausole:

1. $\{A(x, y), A(y, z)\}, \{\neg A(u, f(u))\}$
2. $\{B(x, x), \neg C(x, f(x))\}, \{C(x, y), D(y, z)\}$
3. $\{A(x, y), \neg A(x, x)), B(x, z, f(x))\}, \{\neg B(f(x), z, x), A(x, z)\}$

**6.15** Provare che per ogni insieme $S$ di clausole, $S \equiv Ris(S)$.

**6.16** Provare, utilizzando la risoluzione, che da:

1. $\forall x \forall y \forall z(P(x, y) \wedge P(y, z) \rightarrow P(x, z))$
2. $\forall x \forall y(P(x, y) \rightarrow P(y, x))$

Segue $\forall x \forall y \forall z(P(x, y) \wedge P(z, y) \rightarrow P(x, z))$

**6.17** Provare, utilizzando la risoluzione che da:

- Gli studenti sono cittadini.

Segue

I voti degli studenti sono voti di cittadini.
(Suggerimento: si denotino con $A(x), B(x)$ e $C(x, y)$ rispettivamente "$x$ è uno studente", "$x$ è un cittadino" e "$x$ è un voto di $y$ ").

**6.18** Si considerino le seguenti affermazioni:

- I poliziotti hanno perquisito tutti coloro che sono scesi dall'aereo ad eccezione dei membri dell'equipaggio.
- Alcuni ladri sono scesi dall'aereo e sono stati perquisiti solo da ladri.
- Nessun ladro era un membro dell'equipaggio.

Stabilire se l'affermazione:
Alcuni poliziotti erano ladri.
è conseguenza logica delle prime tre.

# Bibliografia


[AAD63] S.Aanderaa, P.Andrews, B.Dreben. "False Lemmas in Herbrand". *Bull. Amer. Math. Soc.*, v. 69, pp. 699-706, 1963.

[AGM92] S.Abramsky, D.M.Gabbay, T.S.E.Maibaum eds. *Handbook of Logic in Computer Science*, Oxford Science Publications, 1992.

[AAAM79] M.Aiello, A.Albano, G.Attardi, U.Montanari. *Teoria della Computabilità, Logica, Teoria dei Linguaggi Formali.* ed. ETS, Pisa, 1979.

[AB75] A.R.Anderson, N.D.Belnap. *The Logic of Relevance and Necessity.* vol. 1, Princeton University Press, New Jersey, 1975.

[ABD92] A.R.Anderson, N.D.Belnap e J.M.Dunn. *The Logic of Relevance and Necessity.* vol. 2, Princeton University Press, New Jersey. 1992.

[AL91] A.Asperti, G.Longo. *Categories, Types, and Structures.* Foundations of Computing Series. M.I.T.Press, Cambridge, Massachusetts, 1991.

[Bar84] H.Barendregt. *The Lambda Calculus: its syntax and semantics.* North-Holland, 1984 (Seconda edizione).

[Bar77] J.Barwise. *Handbook of Mathematical Logic.* North-Holland. 1977.

[BM77] J.Bell, M.Machover. *A Course in Mathematical Logic.* North-Holland, 1977.

[Ber26] P.Bernays. "Axiomatische Untersuchung des Aussagen-Kalküls". *Mathematische Zeitschrift*, v.25, pp. 305-320, 1926.

[Bet51] E.Beth. "A Topological Proof of the Theorem of Löweneim-Skolem-Gödel". *Indag. Math.*, v.13, pp. 436-444, 1951.

[Bib82] W.Bibel. *Automated Theorem Proving.* Vieweg, Braunschweig, 1982.

[Bir40] G.Birkhoff. *Lattice Theory.* American Mathematical Society, 1940. (Seconda edizione del 1948).

[BJ74] G.Boolos, R.Jeffrey. *Computability and Logic.* Cambridge University Press, 1974.

[Bra86] I.Bratko. *Prolog Programming for Artificial Intelligence*. Addison-Wesley, 1986.

[CL73] C.L.Chang, R.C.T.Lee. *Symbolic Logic and Mechanical Theorem Proving*, Academic Press, New York, 1973.

[CK73] C.C.Chang, H.J.Keisler. *Model Theory*. North-Holland, 1973.

[Chu36] A.Church. "A note on the Entscheidungs problem". *J. of Symbolic Logic*, v.1, pp. 40-41, 1936.

[Chu41] A.Church. "The Calculi of Lambda-conversion". *Annals of Mathematical Studies*, v.6, Princeton N.J., 1941. (Seconda edizione del 1951).

[Chu56] A.Church. *Introduction to Mathematical Logic I*. Princeton University Press, 1956.

[CLM90] L.Console, E.Lamma, P.Mello. *Programmazione Logica e Prolog*. Utet, Torino, 1990.

[Coo71] S.A.Cook. "The Complexity of Theorem Proving Procedures". *Proc. third Annual ACM Symposium on the Theory of Computing*, pp. 233-246, 1971.

[Cur63] H.B.Curry. *Foundations of Mathematical Logic*. McGraw-Hill Series in Higher Mathematics, McGraw-Hill, 1963.

[CF58] H.B.Curry, R.Feys. *Combinatory Logic*. North-Holland, 1958.

[DP60] M.Davis, H.Putnam. "A computing Procedure for Quantification Theory". *J. Ass. Comp. Mach.*, v.7, pp. 201-215, 1960.

[DeS93] ?? De Swart. *Logic: Mathematics, Language, Computer Science and Philosophy*. vol.1, Peter Lang, 1993.

[DS93] K.Dosen, P.Schroeder-Heister. *Substructural Logics, studies in Logic and Computation*. D.M.Gabbay ed. Clarendon Press, Oxford, 1993.

[DRW86] R.D.Dowsing, V.J.Rayward-Smith, C.D.Walter. *A First Course in Formal Logic and its Applications in Computer Science*. Blackwell Scientific Pubblications, 1986.

[DD66] B.Dreben, J.Denton. "A Supplement to Herbrand". *J. of Symbolic Logic* v.31, pp. 393-398, 1966.

[Duf92] P.Duffy. *Principles of Automated Theorem Proving*. John Wiley and Sons, 1992.

[Dum77] M.A.E.Dummet. *Elements of Intuitionism*. Clarendon Press, Oxford, 1977.

[DFS59] B.Dunham, R.Fridsal, G.Sward. "A Non-heuristic Program for Proving Elementary Logical Theorems". Proceeding of the Intern. Conf. on Inform. Process. pp. 282-285, 1959.

[Dun86] J.Dunn. "Relevance Logic and Entailment". In [GG86], pp. 117-224, 1986.

[EFT94] H.D.Ebbinghaus, J.Flum, W.Thomas. *Mathematical Logic*. Springer-Verlag, 1994. (Seconda edizione).

[Gab81] D.M.Gabbay. *Semantical Investigation in Heyting's Intuitionistic Logic*. Reidel Publishing Company, 1981.

[GG86] D.Gabbay, F.Guenter eds. *Handbook of Philosophical Logic*, vol.3, Dordrecht: Reidel, 1986.

[GJ79] M.Garey, D.S.Johnson. *Computers and Intractability; A guide to the Theory of NP-completeness*. W.H.Freeman. S.Francisco. 1979.

[Gal86] J.Gallier. *Logic for Computer Science*. Harper & Row, 1986.

[Gal91] J.Gallier. *Constructive Logics. Part I: A tutorial on proof Systems and Typed $\lambda$-Calculi*. Technical Report of the Digital Equipment Corporation, Paris Research Center, Maggio 1991.

[Gal91b] J.Gallier. *Constructive Logics. Part II: Linear logic and Proof Nets*. Technical Report of the Digital Equipment Corporation, Paris Research Center, Maggio 1991.

[Gen34] G.Gentzen. "Untersuchungen über das logische Schliessen", *Mathematische Zeitschrift*, v.39, pp. 176-210, pp.405-431, 1934-35. Traduzione inglese in [Gen69].

[Gen55] G.Gentzen. *Recherches sur la Déduction Logique*. Traduzione francese a cura di R.Feys e J.Ladrière di [Gen34]. Presses Universitaires, Paris, 1955.

[Gen69] *The Collected Papers of Gerhard Gentzen*. M.E.Szabo ed, North-Holland, 1969.

[Gil59] P.C.Gilmore. "A Program for the Production from Axioms of Proofs for Theorems Derivable within the First Order Predicate Calculus". Proceeding of the Intern. Conf. on Inform. Process. pp. 265-273, 1959.

[Gil60] P.C.Gilmore. "A Proof Method for Quantification Theory: its Justification and Realization". *IBM J. Res. Develop.*, pp. 28-35, 1960.

[Gir72] J.Y.Girard. *Interpretation Functionelle et Elimination des Coupure dans l'Arithmetic d'Order Superieur*. These de doctorat d'Etat, Paris, 1972.

[Gir86] J.Y.Girard. "The system F of Variable Types, fifteen years later". *Theor. Comp. Science*, v.45, pp. 159-192, 1986.

[Gir87] J.Y.Girard. "Linear Logic". *Theor. Comp. Science*, v.50, pp. 1-102, 1987.

[Gir95] J.Y.Girard. "Linear Logic: its Syntax and Semantic". In *Advances in Linear Logic*. Cambridge University Press. 1995.

[GLT90] J.Y.Girard, Y.Lafont, P.Taylor. *Proofs and Types*. Cambridge Tracts in Theoretical Computer Science 7. Cambridge University Press. 1990.

[God30] K.Gödel. "Die Vollständigkeit der Axiome Logischen Funktionenkalülus". *Monatshefte für Mathematik und Physik*, v.37, pp. 349-360, 1930.

[Has53] G.Hasenjaeger. "Eine Bemerkung zu Henkin's Beweis für die Vollständigkeit des Prädikatenkalkülus der ersten stufe". *J. of Symbolic Logic*, v.18, pp. 42-48, 1953.

[Hen49] L.Henkin. "The Completeness of the First Order Functional Calculus". *J. of Symbolic Logic*, v. 14, pp. 158-166, 1949.

[Hen54] L.Henkin. "Boolean Representation Throught Propositional calculus". *Fundamenta Mathematicae*. v.41, pp. 89-96, 1954.

[Her30] J.Herbrand. "Recherches sur la théorie de la démonstration". *Travaux de la Société des Sciences et de Lettres de Varsovie*, Classe III, Sciences Mathématiques et Physiques , v.33, Varsavia, 1930. Riscritto e tradotto in [Her71].

[Her71] J.Herbrand. *Logical Writings*. ed. Warren D. Goldfarb. Harvard University Press, 1971.

[HB34] D.Hilbert, P.Bernays. *Grundlagen der Mathematik*, Springer, Berlin, vol.1, 1934, vol.2, 1939.

[How80] W.A.Howard. "The Formulae-as-types Notion of Construction". In *To H.B.Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, R.Hindley, J.Seldin eds. Academic Press. 1980.

[HS86] R.Hindley, J.Seldin. *Introduction to Combinators and Lambda-Calculus*, London Mathematical Society, 1986.

[HU79] J.E.Hopcroft, J.D.Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley, 1979.

[Jon86] ??. *Systematic Software Development Using VDM*. Prentice-Hall, Englewood Cliffs, New Jersey, 1986.

[Kal35] L.Kalmar. "Über die Axiomatisierbarkeit des Aussagenkalküls". *Acta Scientiarum Mathematicarum*, v.7, pp. 222-243, 1935.

[Kar72] R.M.Karp. *Reducibility Among Combinatorial Problems*. Complexity of Computer Computations, pp. 85-104, Plenum Press, N.Y., 1972

[KW84] J.Ketonen, R.Weyhrauch. "A Decidable Fragment of Predicate Calculus". *Theor. Comp. Science*, vol. 32, pp. 297-307, 1984.

[Kle67] S.C.Kleene. *Mathematical Logic*. John Wiley and Sons eds, Inc., New York, 1967.

[KK67] G.Kreisel, J.L.Krivine. *Elements of Mathematical Logic*. North-Holland, 1967.

[Kun80] K.Kunen. *Set Theory. An Introduction to Independence Proofs*. North-Holland, 1980.

[Kur65] A.G.Kurosh. *Lectures in General Algebra*. Pergamon Press, 1965.

[LS86] J.Lambek, P.J.Scott. *Introduction to Higher-Order Categorical Logic*. Cambridge University Press, 1980.

[Lam58] J.Lambek. "The Mathematics of Sentence Structure". *The Amer. Math. Montly*, vol. 65, pp. 154-170, 1958.

[Lev79] A.Levy. *Basic Set Theory*. Springer-Verlag, 1979.

[Llo87] J.W.Lloyd. *Foundations of Logic Programming*. Springer-Verlag. (Seconda edizione), 1987. Traduzione italiana *Fondamenti di Programmazione Logica*. Muzzio ed., Padova, 1986.

[Lol91] G.Lolli. *Introduzione alla Logica Formale*. Il Mulino, Bologna, 1991.

[Lov78] D.W. Loveland. *Automated Theorem Proving: A Logical Basis*. North-Holland, 1978.

[Luk20] J.Łukasiewicz. "O Logice Trójwartościowej". *Ruch Filozoficzny*, vol.5, pp. 170-171, 1920. Traduzione inglese *On Three-valued Logic*. In Jan Łukasiewicz Selected Works. pp. 87-88. North-Holland, 1970.

[Mal36] A.Malcev. "Untersuchungen aus dem Gebiete der Matematischen Logik". *Rec. Math. N.S.*, v.1, pp. 323-336, 1936.

[Man78] Z.Manna. *Teoria Matematica della Computazione*. Boringhieri, Torino. 1978.

[MM82] A.Martelli, U.Montanari. "An Efficient Unification Algorithm". A.C.M. Trans. on Programming Languages and Systems, v.4, pp. 258-282, 1982.

[Men64] E.Mendelson. *Introduction to Mathematical Logic*. ed. D. Van Nostrand Company. Princeton, New Yersey, 1964.

[Mun92] D.Mundici. "The Logic of Ulam's Game with Lies". In *Knowledge, Belief and Strategic Interaction*. Bicchieri C., Dalla Chiara M.L., eds. Cambridge University Press, Cambridge Studies in Probability, Induction and Decision Theory, pp. 275-284, 1992.

[NS93] A.Nerode, R.A.Shore. *Logic for Applications*. Springer-Verlag, 1993.

[Odi89] P.Odifreddi. *Classical Recursion Theory*. North-Holland, 1989.

[OK85] H.Ono, Y.Komori. "Logics without the Contraction Rule". *J. of Symbolic Logic*. vol. 50. pp. 169-201, 1985.

[PW78] M.S.Paterson, M.N.Wegman. "Linear Unification". *J. Comput. System Sci.*, v.16, pp. 158-167, 1978.

[Pau92] L.P.Paulson. *Designing a Theorem Prover*. In [AGM92], pp. ??, 1992.

[PeiCP] C.S.Peirce. *Collected Papers of Charles Sanders Peirce*. Vol.I-VIII, C.Hartshorne, P.Weiss e A.Burks eds., Cambridge Mass., 1931-58.

[Pos21] E.L.Post. "Introduction to a General Theory of Elementary Propositions". *Amer. Journal of Math*, v.43, pp. 163-185, 1921.

[Pos41] E.L.Post. *The Two Valued Iterative Systems of Mathematical Logic*, Princeton N.J., 1941.

[PPV60] D.Prawitz, H.Prawitz, N.Voghera. "A Mechanical Proof Procedure and its Realization in an Electronic Computer". *J. Ass. Comp. Mach.*, v.7, pp. 102-128, 1960.

[Pra65] D.Prawitz. *Natural Deduction*. Almqvist & Wiksell, Stockolm, 1965.

[Qui55] W.V.Quine. "A Proof Procedure for Quantification Theory". *J. of Symbolic Logic*, v.20, pp. 141-149, 1955.

[RS51] H.Rasiowa, R.Sikorski. "A Proof of the Completeness Theorem of Gödel". *Fundam. Math.*, v.37, pp. 193-200, 1951.

[Rey74] J.C.Reynolds. "Towards Theory of Type Structure". Paris Colloquium on Programming, LNCS 19, Springer-Verlag, 1974.

[Rob65] J.A.Robinson. "A Machine-oriented Logic based on the Resolution Principle". *J. of the ACM*, v.12, pp. 23-41, 1965.

[Rob68] J.A.Robinson. "The Generalized Resolution Principle". In *Machine intelligence*, vol.3, ed. D. Michie, American Elsevier, New York, pp. 77-94, 1968.

[Rob79] J.A.Robinson. *Logic: Form and Function*. Elsevier North-Holland, New York, 1979.

[Rog67] H.Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw Hill, 1967.

[Rog71] H.Rogers. *Mathematical Logic and Formalized Theories*. North-Holland, 1971.

[RT58] J.B.Rosser, A.R.Turquette. *Many Valued Logics*. North-Holland, 1958.

[Rus03] B.Russel. *The Principles of Mathematics*, Cambridge, England, 1903. Traduzione italiana di L.Geymonat, Longanesi, Milano, 1950.

[RW10] B.Russel, A.N.Whitehead. *Principia Mathematica*, 3 vol., Cambridge, England, 1910-1913. Seconda edizione del 1925-27.

[Sho67] J.R.Shoenfield. *Mathematical Logic*. Addison-Wesley, 1967.

[Sch89] U.Schöning. *Logic for Computer Scientists*. Birkhäuser, Boston, 1989.

[Sko28] T.Skolem. "Über Die Mathematische Logik". *Norsk Matematisk Tidskrift*, v.10, pp. 125-142, 1928.

[Smo77] C.Smorynski. "The Incompleteness Theorems". In [Bar77]. pp. 821-867, 1977.

[Smu92] R.M.Smullyan. *Gödel Incompleteness Theorems*. Oxford University Press, 1992.

[Sur73] S.J.Surma. *Studies in the History of Mathematical Logic*. ed. Surma. Polish academy of Sciences, Warsaw, 1973.

[Tak75] G.Takeuti. *Proof Theory*. Studies in Logic and the Foundations of Mathematics, vol.81, North-Holland Publishing Company, 1975.

[Tro73] A.S.Troelstra. "Intuitionistic Formal System. In *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*. A.S.Troelstra ed. PV. pp. 1-96, 1973.

[Tro92] A.S.Troelstra. *Lectures on Linear Logic*. CSLI-Lecture Notes 29 Center for the Study of Language and Information, Stanford, California, 1992.

[TS96] A.S.Troelstra, H.Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, 1996.

[Tur36] A.M.Turing. "On Computable Numbers with an Application to the Entscheidungsproblem". *Proc. London Math. Soc.*, v.42, pp. 230-265, 1936-1937.

[Tur84] R.Turner. *Logics for Artificial Intelligence*. Ellis Horwood, 1984.

[Urq86] A.Urquhart. *Many-valued Logic*. In [GG86], pp. 71-116, 1986.

[Urq87] A.Urquhart. "Hard Examples for Resolution". *J. of the Association of Comp. Mach.*, v.34, pp. 209-219, 1987.

[VDa80] D.Van Dalen. *Logic and Structure*. Springer Verlag, Berlin, 1980.

[VDa86] D.Van Dalen. "Intuitionistic Logic". In [GG86], pp. 225-339, 1986.

[VDD78] D.Van Dalen, H.C.Doets, H.C.M. De Swart *Sets: Naive, Axiomatic and Applied*. Pergamon Press, Oxford, 1978.

[VNe27] J.Von Neumann. "Zur Hilbertschen Beweistheorie". *Mathematische Zeitschrift*, v.26, 1927.

[Wan60] H.Wang. "Towards Mechanical Mathematics". *IBM J. Res. Dev.*, v.4, pp. 2-22, 1960.

# Indice analitico