Linguaggi

19: Deduzione naturale per la logica del prim'ordine

Claudio Sacerdoti Coen

<sacerdot@cs.unibo.it>

Universitá di Bologna

15/04/2011

Outline

Semantica

Wikipedia: "$\ldots$"

Deduzione naturaler per la logica del prim'ordine

Abbiamo già dato le regole per la deduzione naturale nel caso proposizionale indicando per ogni connettivo le regole di introduzione e di eliminazione.

Inoltre, per essere completi rispetto alla semantica classica, abbiamo aggiunto la regola di riduzione ad assurdo.

La logica del prim'ordine estende le proposizioni della logica proposizionale in due modi:

- Il caso P^0 è generalizzato a $P^n(t_1, \ldots, t_n)$: nessuna nuova regola necessaria
- Vengono aggiunti i quantificatori universale ed esistenziale: è sufficiente introdurre le apposite regole di introduzione ed eliminazione

Deduzione naturale: $\forall$

Regole di introduzione:

$$(\forall_i) \quad \frac{\begin{array}{c}\vdots \\ P[y/x]\end{array}}{\forall x.P} \quad y \notin FV(Foglie(\vdots))$$

dove $Foglie(\vdots)$ sono le foglie non (ancora!) cancellate nel sotto-albero $\vdots$

Attenzione: una foglia può non essere ancora cancellata in $\vdots$ ma sembrare cancellata in quanto cancellata successivamente da una regola applicata successivamente.

Idea: per concludere che P vale per un x qualunque basta dimostrare che P vale per un x (qui ridenominato in y) che sia veramente qualunque ovvero sul quale non vi siano altre ipotesi ovvero tale per cui y non compaia in nessuna ipotesi.

Deduzione naturale: ∀

Regole di introduzione:

$$(\forall_i) \quad \cfrac{\begin{array}{c}\vdots\\ P[y/x]\end{array}}{\forall x.P} \quad y \notin FV(Foglie(\vdots))$$

Esempi errati:

$$\cfrac{\begin{array}{c}y > 0\\ \vdots\\ y^2 > 0\end{array}}{\forall y.y^2 > 0}\ (\forall_i) \qquad \cfrac{\cfrac{\begin{array}{c}[y > 0]\\ \vdots\\ y^2 > 0\end{array}}{\forall y.y^2 > 0}\ (\forall_i)}{y > 0 \Rightarrow \forall y.y^2 > 0}\ (\Rightarrow_i)$$

Nota: cambiare il nome della variabile (da y a x) serve appunto per evitare le variabili sulle quali ci sono già ipotesi durante la ricerca top-down.

Deduzione naturale: $\forall$

Regole di introduzione:

$$(\forall_i)\quad \frac{\begin{array}{c}\vdots\\ P[y/x]\end{array}}{\forall x.P}\quad y \notin FV(Foglie(\vdots))$$

Lettura bottom-up: se P vale per una y qualunque, allora per ogni x vale P.

Lettura top-down: per dimostrare $\forall x.P$ è sufficiente scegliere una y sulla quale non sappiamo nulla e poi dimostrare $P[y/x]$. sia A che B. Ovviamente è sufficiente scegliere una y ancora mai usata.

Deduzione naturale: $\forall$

Regole di introduzione:

$$(\forall_i)\quad \frac{\begin{array}{c}\vdots\\ P[y/x]\end{array}}{\forall x.P}\quad y \notin FV(Foglie(\vdots))$$

Scrittura informale:
sia x una variabile arbitraria ma fissata
... e quindi P
e quindi $\forall x.P$

Molto raramente viene scelta una y al posto della x.

Deduzione naturale: $\forall$

Regole di introduzione:

$$(\forall_i)\quad \frac{\begin{matrix}\vdots\\ P[y/x]\end{matrix}}{\forall x.P}\quad y \notin FV(Foglie(\vdots))$$

Correttezza intuizionista: $P[y/x] \Vdash \forall x.P$ in quanto se $p \in [\![P[y/x]]\!]$ e p non chiama funzioni di libreria "che usano y" allora la funzione $f(x) = p[x/y]$ è una funzione polimorfa che ad ogni x restituisce un $p[x/y] \in [\![P[y/x][x/y]]\!] = [\![P]\!]$ e quindi $f \in [\![\forall x.P]\!]$.

Invertibilità intuizionista: se $f \in [\![\forall x.P]\!]$ allora f è una funzione polimorfa tale che $f(x) \in [\![P]\!]$ per ogni x. In particolare $f(y) \in [\![P[y/x]]\!]$.

Deduzione naturale: $\forall$

Regole di introduzione:

$$(\forall_i)\quad \frac{\begin{array}{c}\vdots\\ P[y/x]\end{array}}{\forall x.P}\quad y \notin FV(Foglie(\vdots))$$

Correttezza classica: Dimostriamo che per ogni Γ si ha che se $\Gamma \Vdash P[y/x]$ dove $y \notin FV(\Gamma)$ allora $\Gamma \Vdash \forall x.P$. Infatti sia (A, I) un mondo e ξ un environment tali per cui $[\![F]\!]^{(A,I),\xi} = 1$ per ogni $F \in \Gamma$. Per definizione di conseguenza logica si ha $[\![P[y/x]]\!]^{(A,I),\xi} = 1$. Per induzione strutturale su P si dimostra che $[\![P[y/x]]\!]^{(A,I),\xi} = [\![P]\!]^{(A,I),\xi[x\mapsto\xi(y)]} = 1$. Tale risultato deve valere non solo per ξ, ma anche per $\xi[y \mapsto \alpha]$ per ogni $\alpha \in A$ dal momento che se $[\![F]\!]^{(A,I),\xi} = 1$ allora anche $[\![F]\!]^{(A,I),\xi[y\mapsto\alpha} = 1$ per ogni $F \in \Gamma$ in quanto $y \notin FV(\Gamma)$. Si ha pertanto $\min\{[\![P]\!]^{(A,I),\xi[y\mapsto\alpha][x\mapsto\xi(y)]} \mid \alpha \in A\} = \min\{[\![P]\!]^{(A,I),\xi[x\mapsto\alpha]} \mid \alpha \in A\} = \min\{1 \mid \alpha \in A\} = 1.$

Deduzione naturale: $\forall$

Regole di introduzione:

$$(\forall_i) \quad \frac{\begin{array}{c}\vdots \\ P[y/x]\end{array}}{\forall x.P} \quad y \notin FV(Foglie(\vdots))$$

Invertibilità classica: $\forall x.P \Vdash P[y/x]$. Infatti se $[\![\forall x.P]\!]^{(A,I),\xi} = \min\{[\![P]\!]^{(A,I),\xi[x\mapsto\alpha]} \mid \alpha \in A\} = 1$ allora si ha $[\![P]\!]^{(A,I),\xi[x\mapsto\xi(y)]} = [\![P[y/x]]\!]^{(A,I),\xi} = 1$.

Deduzione naturale: $\forall$

Regole di eliminazione:

$$(\forall_e) \quad \frac{\forall x.P}{P[t/x]}$$

Lettura bottom-up: se P vale per tutti gli x, allora vale in particolare per t.

Lettura top-down: per dimostrare $P[t/x]$ è sufficiente dimostrare il teorema generalizzato $\forall x.P$.

Scrittura informale:
…e quindi $\forall x.P$
e quindi $P[t/x]$

Deduzione naturale: $\forall$

Regole di eliminazione:

$$(\forall_e) \quad \frac{\forall x.P}{P[t/x]}$$

Correttezza intuizionista: $\forall x.P \Vdash P[t/x] \Vdash$ in quanto se $f \in [\![\forall x.P]\!]$ allora $f(t) \in [\![P[t/x]]\!]$.

Correttezza classica: $\forall x.P \Vdash P[t/x]$ in quanto per ogni mondo (A, I) ed environment ξ tali che $[\![\forall x.P]\!]^{(A,I),\xi} = \max\{[\![P]\!]^{(A,I),\xi[x\mapsto\alpha]} = 1$ si ha $[\![P[t/x]]\!]^{(A,I),\xi} = [\![P]\!]^{(A,I),\xi[x\mapsto[\![t]\!]^{(A,I),\xi}]} = 1$ per induzione strutturale su P.

La regola è chiaramente non invertibile: per esempio, $Pari(2)$ ma non si ha $\forall x.Pari(x)$.

Deduzione naturale: $\exists$

Regole di introduzione:

$$(\exists_i) \quad \frac{P[t/x]}{\exists x.P}$$

Lettura bottom-up: se P vale per t allora esiste un x per cui P vale.

Lettura top-down: per dimostrare $\exists x.P$ bisogna scegliere un t per il quale $P[t/x]$ valga e dimostrarlo.

Scrittura informale:
... e quindi $P[t/x]$
e quindi $\exists x.P$

Deduzione naturale: $\exists$

Regole di introduzione:

$$(\exists_i) \quad \frac{P[t/x]}{\exists x.P}$$

Correttezza intuizionista: $P[t/x] \Vdash \exists x.P$ in quanto se $p \in [\![P[t/x]]\!]$ allora $\langle t, p\rangle \in [\![\exists x.P]\!]$.

Correttezza classica: $P[t/x] \Vdash \exists x.P$ in quanto per ogni mondo (A, I) ed environment ξ tali che $[\![P[t/x]]\!]^{(A,I),\xi} = 1$ si ha $[\![P[t/x]]\!]^{(A,I),\xi} = [\![P]\!]^{(A,I),\xi[x\mapsto [\![t]\!]^{(A,I),\xi}]} = 1$ per induzione strutturale su P. Pertanto $\max\{[\![P]\!]^{(A,I),\xi[x\mapsto\alpha]} \mid \alpha \in A\} = 1$.

La regola è chiaramente non invertibile: per esempio, $\exists x.Pari(x)$ ma non si ha $Pari(3)$.

Deduzione naturale: $\exists$

Regole di eliminazione:

$$(\exists_e) \quad \frac{\exists x.P \qquad \begin{array}{c}[P[y/x]] \\ \vdots \\ C\end{array}}{C} \qquad y \notin FV(C) \cup FV(Foglie(\vdots))$$

Lettura bottom-up: se $\exists x.P$ e se dimostro C sotto l'ipotesi che P valga per un generico y, allora C vale.

Lettura top-down: per dimostrare un qualche C sotto l'ipotesi $\exists x.P$ è sufficiente dimostrare C assumendo P per una qualche variabile generica y.

Deduzione naturale: $\exists$

Regole di eliminazione:

$$(\exists_e)\quad \frac{\exists x.P \qquad \begin{matrix}[P[y/x]]\\ \vdots \\ C\end{matrix}}{C} \qquad y \notin FV(C) \cup FV(Foglie(\vdots))$$

Scrittura informale:

…e quindi $\exists x.P$

[supponiamo che valga P]

…e quindi C

[e quindi C]

Molto raramente viene scelta una y al posto della x. Per tale motivo il passo in cui viene assunto $P[y/x]$ è normalmente sottomesso.

Deduzione naturale: ∀

Regole di eliminazione:

$$(\exists_e)\quad \dfrac{\exists x.P \qquad \begin{array}{c}[P[y/x]]\\ \vdots \\ C\end{array}}{C} \qquad y \notin FV(C) \cup FV(Foglie(\vdots))$$

Correttezza intuizionista:

Correttezza classica:

La regola è chiaramente non invertibile: per esempio, quando C è $\top$ e P è $\bot$. Tuttavia la regola diventa chiaramente invertibile quando $\Vdash \exists x.P$.

Deduzione naturale: pragmatica

Le regole $\forall_i$ ed $\exists_e$ sono invertibili ed è bene applicarle quanto prima.

In particolare, la regola $\exists_e$ deve essere anticipata il prima possibile perchè suggerisce quale sia quel testimone che molto probabilmente deve poi essere utilizzato successivamente in regole $\exists_i$.

La regola $\exists_i$ deve essere posticipata fino al momento in cui è chiaro quale sia il testimone.

La regola $\forall_e$ viene normalmente utilizzata solamente in modalità bottom-up, quando la ricerca top-down della prova esaurisce il suo impulso.

Deduzione naturale: pragmatica

Esercizi:

- $\vdash (\exists x.P(x)) \Rightarrow \exists y.P(y)$
- $\forall x.(P(x) \Rightarrow Q(x)) \vdash (\exists x.P(x)) \Rightarrow \exists y.Q(y)$
- $\exists x.(C(x) \Rightarrow \forall z.C(z))$ (paradosso dell'uomo con il cappello)

Deduzione naturale: completezza intuizionista

Teorema di completezza per la deduzione naturale per la logica intuizionista del prim'ordine: per ogni Γ e F, se $\Gamma \Vdash F$ in logica intuizionista allora $\Gamma \vdash F$ in deduzione naturale senza usare il principio della RAA.

Idea: il teorema vale a patto che il mio linguaggio di programmazione contenga anche funzioni polimorfe di un certo tipo.

Deduzione naturale: completezza intuizionista

Teorema: dati Γ e F, l'esistenza di un albero di derivazione per F con foglie non cancellate in Γ è solo semi-decidibile.

Teorema di completezza debole per la deduzione naturale per la logica classica del prim'ordine: per ogni Γ e F con Γ finito, se *Ter* è enumerabile e se $\Gamma \Vdash F$ in logica classica allora $\Gamma \vdash F$ in deduzione naturale (compreso il principio di RAA).

Dimostrazione: l'idea, come nel caso della dimostrazione per la logica proposizionale, è quella di dare un algoritmo che cerchi la prova e di dimostrare che, quando $\Gamma \Vdash F$ allora l'algoritmo termina con successo. A differenza del caso proposizionale, se $\Gamma \nVdash F$ l'algoritmo, invece di terminare con insuccesso, può divergere (per via dell'indecidibilità).

Dimostrazione (continua):

A differenza del caso proposizionale, non possiamo costruire la prova per casi su $A \vee \neg A$, $B \vee \neg B$, ... in quanto le disgiunzioni $P^n(t_1, \ldots, t_n) \vee \neg P^n(t_1, \ldots, t_n)$ sono ora infinite.

Seguiamo quindi una nuova idea: usiamo la RAA per ridurci a dimostrare $\Gamma, \neg F \vdash \bot$ e poi costruiamo ricorsivamente una dimostrazione in cui, ad ogni passo dell'algoritmo, tutti i sotto-alberi ancora da dimostrare concludono $\bot$.

L'algoritmo prende pertanto in input un contesto $\Delta = F_1, \ldots, F_n$ e lavora sempre su F_1, chiamandosi ricorsivamente su nuovi contesti.

L'algoritmo termina con successo una chiamata ricorsiva quando in Δ occorre $\neg F_1$. La prova generata è $\dfrac{\neg F_1 \quad F_1}{\bot}$.

Dimostrazione (continua):

Caso $\top$: chiamata ricorsiva su $F_2, \ldots, F_n$.

Caso $\bot$: l'algoritmo termina con successo trovando l'albero $\bot$.

Caso $G_1 \wedge G_2$: chiamata ricorsiva su $G_1, G_2, F_2, \ldots, F_n$ dopo aver costruito l'albero

$$\dfrac{G_1 \wedge G_2 \qquad \begin{array}{c}[G_1][G_2]\\ \vdots \\ \bot\end{array}}{\bot}$$

Dimostrazione (continua):

Caso $G_1 \vee G_2$: chiamata ricorsiva su $G_1, F_2, \ldots, F_n$ e su $G_2, F_2, \ldots, F_n$ dopo aver costruito l'albero

$$\dfrac{G_1 \vee G_2 \qquad \begin{array}{c}[G_1]\\ \vdots\\ \bot\end{array} \qquad \begin{array}{c}[G_2]\\ \vdots\\ \bot\end{array}}{\bot}$$

Caso $G_1 \Rightarrow G_2$: usando la prova per $G_1 \Rightarrow G_2 \Vdash \neg G_1 \vee G_2$ ci si riduce al caso $\neg G_1 \vee G_2, F_2, \ldots, F_n$

Dimostrazione (continua):

Caso $\exists x.G$: chiamata ricorsiva su $G[y], F_2, \ldots, F_n$ per $y \notin FV(F_2, \ldots, F_n)$ dopo aver costruito l'albero

$$\dfrac{\exists x.G \qquad \begin{array}{c}[G[y]] \\ \vdots \\ \bot\end{array}}{\bot}$$

Caso $P^n(t_1, \ldots, t_n)$: chiamata ricorsiva su $F_2, \ldots, F_n, P^n(t_1, \ldots, t_n)$ senza avanzare nella costruzione dell'albero di deduzione, a patto che in $F_2, \ldots, F_n$ vi sia almeno una formula che non sia un predicato o un predicato negato. In tal caso l'algoritmo termina con un insuccesso.

Dimostrazione (continua):

Fino a questo momento per tutti i connettivi considerati avevamo a disposizione una regola invertibile. Pertanto potevamo rimpiazzare l'ipotesi F_1 con altre.

Per quanto riguarda il $\forall$, possiamo trovare una regola invertibile, ma solo a patto di non cancellare l'ipotesi F_1 e sotto l'ipotesi aggiuntiva che *Ter* (l'insieme dei termini) sia enumerabile.

Caso $\forall x.P$: chiamata ricorsiva su $P[t], F_2, \ldots, F_n, \forall x.P$ ove l'ipotesi $P[t]$ non è stata ancora generata considerando il caso $\forall x.P$. La costruzione dell'albero prosegue come segue:

$$\begin{array}{c} \dfrac{\forall x.G}{P[t]} \\ \vdots \\ \bot \end{array}$$

Dimostrazione (continua):

Problema: dobbiamo ancora trattare il caso $\neg G$. Non possiamo però applicare la regola $\neg_e$ perchè essa ha come premessa G e non $\bot$. Per ridurre una dimostrazione di G a una dimostrazione di $\bot$ potremmo usare la RAA, ma questo vorrebbe dire dimostrare $\bot$ sotto l'ipotesi $\neg G$ e ci saremmo quindi ridotti al problema iniziale senza fare progressi.

Soluzione: come nel caso del predicato n-ario applicato, anche la negazione del predicato n-ario applicato si può gestire semplicemente mettendo l'ipotesi in coda (a meno che tutte le ipotesi non siano predicati n-ari applicati in forma negata o meno); in tutti i casi $\neg G$ dove G non è un predicato n-ario ricorriamo alle leggi di De Morgan (che sappiamo dimostrare in deduzione naturale) per ridurci a un caso senza negazioni in testa.

Dimostrazione (continua):

Fino a questo momento abbiamo descritto un algoritmo che può terminare con successo (trovando una prova per il teorema di correttezza), terminare con insuccesso (dimostrando che la prova non esiste, in quanto tutte le regole usate erano invertibili) oppure può divergere.

Ci resta da dimostrare che se l'algoritmo diverge allora una prova non esiste, ovvero che se l'algoritmo diverge allora $\Gamma \not\Vdash \bot$ ovvero che se l'algoritmo diverge allora Γ è soddisfacibile.

Sia Δ l'insieme di tutte i predicati applicati (in forma negata o meno) incontrati nel ramo di prova infinito prodotto dall'algoritmo.

Dimostrazione (continua):

Consideriamo il seguente modello sintattico:

$A = Ter$
$I(f^n) = f^n$
$\xi(x) = x$
$I(P^n)(t_1, \ldots, t_n) = 1$ sse $P^n(t_1, \ldots, t_n) \in \Delta$

Idea: il modello sintattico usa la sintassi come semantica. In particolare, per ogni termine $t \in Ter$ si ha $[\![t]\!]^{(A,I),\xi} = t$!

OSSERVAZIONE IMPORTANTE (per l'ultima lezione): la cardinalità del modello sintattico (ovvero di A) è la cardinalità di Ter.

Resta da dimostrare che $(A, I), \xi$ è un modello per Γ.

Dimostrazione (continua):

Resta da dimostrare che $(A, I), \xi$ è un modello per Γ.

Supponiamo per assurdo che non lo sia, ovvero che vi sia in Γ una formula F t.c. $[\![F]\!]^{(A,I),\xi} = 0$.

Supponiamo, senza perdita di generalità, che F sia F_1.

Osservando le regole di costruzione dell'albero usate dall'algoritmo si dimostra che in ogni caso, se $[\![F_1]\!]^{(A,I),\xi}$ allora per una qualche formula G sulla quale l'algoritmo si chiama ricorsivamente si ha $[\![G]\!]^{(A,I),\xi} = 0$. Esempio: nella regola per il $\wedge$, we $[\![G_1 \vee G_2]\!]^{(A,I),\xi} = 0$ allora di sicuro o $[\![G_1]\!]^{(A,I),\xi} = 0$ oppure $[\![G_2]\!]^{(A,I),\xi} = 0$.

Dimostrazione (continua):

Il caso $\forall$ deve essere trattato in maniera similare: se $[\![\forall x.P]\!]^{(A,I),\xi} = 0$ allora vi è un t t.c. $[\![P[t/x]]\!]^{(A,I),\xi} = 0$ e $P[t/x]$ verrà generato prima o poi nella costruzione del ramo infinito.

Concludendo, se $[\![F_1]\!]^{(A,I),\xi} = 0$ allora prima o poi verrà trovata una formula atomica G (ovvero $P^n(t_1, \ldots, t_n)$ oppure $\neg P^n(t_1, \ldots, t_n)$) tale che $[\![G]\!]^{(A,I),\xi} = 0$. Questo perchè a ogni passo o la nuova formula falsa ha un connettivo in meno, oppure prima o poi se ne trova una con un quantificatore universale in meno e quindi la somma del numero di connettivi (esclusa la negazione) e quantificatori universali decresce fino a zero.

ASSURDO in quanto $[\![P^n(t_1, \ldots, t_n)]\!]^{(A,I),\xi} = 1$ per definizione di I e $[\![\neg P^n(t_1, \ldots, t_n)]\!]^{(A,I),\xi} = 1$ sempre per definizione di I in quanto, per costruzione dell'algoritmo, si ha $P^n(t_1, \ldots, t_n) \notin \Delta$.