Logica per l'Informatica

Deduzione Naturale 5: Compiti per casa

28/11/2023

Ispirati dalle soluzioni del laboratorio scorso, che trovi su Virtuale, per fare i seguenti esercizi.

Esercizio 1. [Commutazione dei quantificatori]

1. Dimostrare in DN intuizionista che, dato P un predicato binario, vale l'enunciato:

$$\forall x.\forall y.P(x, y) \vdash \forall y.\forall x.P(x, y)$$

2. Dimostrare in DN intuizionista che, dato P un predicato binario, vale l'enunciato:

$$\exists x.\exists y.P(x, y) \vdash \exists y.\exists x.P(x, y)$$

Osservazione: tutto questo esercizio vale, con una dimostrazione analoga, anche prendendo una formula A qualsiasi con al più 2 variabili libere x, y, invece che una formula della forma $P(x, y)$, per P un predicato binario.

Inoltre, sempre con una dimostrazione analoga, puoi anche dimostrare gli enunciati inversi (ovvero, da destra a sinistra del $\vdash$).

Esercizio 2. Sia P un predicato unario del linguaggio. Dimostrare in DN (e dire se le dimostrazioni fornite sono classiche o intuizioniste) i seguenti enunciati:

1. $$\neg\exists x.P(x) \vdash \forall x.\neg P(x)$$

2. $$\forall x.\neg P(x) \vdash \neg\exists x.P(x)$$

3. $$\forall x.P(x) \vdash \neg\exists x.\neg P(x)$$

4.

$$\neg\exists x.\neg P(x) \vdash \forall x.P(x)$$

[Suggerimento per tutti: Ispirarsi dell'esercizio 2 o 3 del laboratorio scorso.]
Osserva che, oltre agli enunciato dell'esercizio 2 del laboratorio scorso, gli enunciati simili: $\exists x.P(x) \vdash \neg\forall x.\neg P(x)$ e $\neg\forall x.\neg P(x) \vdash \exists x.P(x)$ li hai gia dimostrati nell'esercizio 3 del laboratorio scorso, nel quale usavamo la formula $Q(x) := x \in a$, per a un termine fissato, invece di un predicato unario P, ma la dimostrazione usava solo il fatto che Q avesse una variabile libera e "si comportasse" come un predicato unario.

Esercizio 3. Sia P un predicato unario del linguaggio ed A una formula nella quale la variabile x non occorre libera.

1. Dimostrare in DN intuizionista l'enunciato:

$$\exists x.(P(x) \to A) \vdash (\forall z.P(z)) \to A$$

2. Dimostrare in DN l'enunciato inverso:

$$(\forall z.P(z)) \to A \vdash \exists x.(P(x) \to A)$$

Osserva che questo enunciato generalizza l'esercizio 5 del laboratorio scorso (l'enunciato "sullo studente bocciato"): infatti prendendo $A := \forall z.P(z)$ si ottiene $\forall z.P(z) \to \forall z.P(z) \vdash \exists x.(P(x) \to \forall z.P(z)$, e siccome la formula $\forall z.P(z) \to \forall z.P(z)$ è sempre dimostrabile, l'enunciato che otteniamo è equivalente a $\vdash \exists x.(P(x) \to \forall z.P(z))$, che è precisamente quello dell'esercizio 5 dello scorso laboratorio. Siccome sappiamo che l'enunciato "sullo studente bocciato" si può dimostrare solo classicamente, abbiamo una indicazione che anche questo esercizio lo si farà classicamente.

[Suggerimento: Fate una dimostrazione in DN analoga a quella dell'esercizio 5 del lab scorso, usando subito il terzo escluso. Nella dimostrazione, potete servirvi delle formule NonPerogniEsisteNon $:= \neg\forall x.Q(x) \to \exists x.\neg Q(x)$ *e* Contronominale $:= (F \to F') \to \neg F' \to \neg F$*, che abbiamo visto negli esercizi 2 e 3 del laboratorio scorso essere dimostrabili in DN (rispettivamente, classica ed intuizionista), per qualsiasi formula F, F' e qualsiasi predicato unario Q.]*

Esercizio 4. [Teoria degli insiemi formale ed Aritmetica formale]

1. Ci diamo, nel linguaggio, un predicato binario $\in$, un simbolo per funzione unaria $\mathscr{P}$ ed una costante $\emptyset$. Per t, t' termini del linguaggio, scriviamo "$t \in t'$" come zucchero sintattico per la formula $\in(t, t')$, scriviamo "$t \notin t'$" come zucchero sintattico per la formula $\neg(t \in t')$.

   Dato un termine t, scriviamo "empty_t" come zucchero sintattico per la formula $\forall x.x \notin t$. Osserva che la scrittura "empty" da sola non è né un predicato del linguaggio (anche se potremmo formalizzarlo in quel modo) né una formula: è semplicemente una notazione che abbiamo stipulato per convenzione in questo esercizio perché rende le cose più chiare e perché siamo pigri e non abbiamo voglia di scrivere le formule $\forall x.x \notin t$ per esteso ogni volta. Per esempio, $\text{empty}_\emptyset$ abbrevia la formula $\forall x.x \notin \emptyset$.

   Inoltre, dati due termini t, t', scriviamo "$t \subseteq t'$" come zucchero sintattico per la formula $\forall x.(x \in t \to x \in t')$.

   Infine, sia powerset la formula:

   $$\text{powerset} := \forall z.\forall y.(y \subseteq z \leftrightarrow y \in \mathscr{P}(z))$$

   Dimostra in DN intuizionista il seguente enunciato, che esprime il fatto che per ogni insieme ne esiste uno che è non vuoto e contiene tutti i sottoinsiemi del primo:

   $$\forall x.\emptyset \subseteq x,\ \text{powerset},\ \text{empty}_\emptyset \vdash \forall a.\exists b.(\neg\text{empty}_b \wedge \forall c.(c \subseteq a \to c \in b)).$$

   [Suggerimento: Pensala prima in stile "matematichese": dato un insieme a, chi sarà mai questo misterioso insieme b (che sia esprimibile nel linguaggio formale che ci siamo dati) ? Nella dimostrazione, puoi servirti della formula $\text{EsisteNonPerogniNon} := \forall x.(\exists y.y \in x \to \neg\forall y.y \notin x)$*, che hai dimostrato nell'esercizio 3 del laboratorio scorso.]*

2. Ci diamo, nel linguaggio, un predicato binario $=$, un simbolo di funzione binaria $+$, ed un simbolo per funzione unaria succ.

   Per t, t' termini del linguaggio, scriviamo "$t = t'$" come zucchero sintattico per la formula $=(t, t')$ e "$t + t'$" come zucchero sintattico per il termine $+(t, t')$.

   Sia axiom la formula:

   $$\text{axiom} := \forall n.\forall m.\neg(\text{succ}(n) + m = n).$$

Infine, dati due termini t, t', scriviamo "$t \leq t'$" come zucchero sintattico per la formula $\exists k.(t + k = t')$.

Dimostra in DN intuizionista il seguente enunciato, che esprime il fatto che non esiste un numero naturale massimo:

$$\text{axiom} \vdash \neg\exists n.\forall m.m \leq n$$

[Suggerimento: Pensala prima in stile "matematichese": supponi che esiste un tale n e cerca una contraddizione (nota che non *stiamo usando il ragionamento per assurdo!). Che numero naturale puoi esibire che non è più piccolo di n (e che sia esprimibile nel linguaggio formale che ci siamo dati) ?]*

Esercizio 5. [Teoria degli insiemi formale]

In tutto l'esercizio ci diamo, nel linguaggio, un predicato binario $\in$.

Per t, t' termini del linguaggio, scriviamo "$t \in t'$" come zucchero sintattico per la formula $\in (t, t')$, scriviamo "$t \notin t'$" come zucchero sintattico per la formula $\neg(t \in t')$, scriviamo "HaElem_t" come zucchero sintattico per la formula $\exists x.x \in t$ e scriviamo "$t \subseteq t'$" come zucchero sintattico per la formula $\forall x.(x \in t \to x \in t')$.

Fissiamo ora un termine a del linguaggio.

1. Dimostra in DN intuizionista il seguente enunciato:

$$\vdash (\exists b.(b \subseteq a \wedge \text{HaElem}_b)) \to \text{HaElem}_a.$$

2. Ci diamo ora, nel linguaggio, un nuovo predicato binario $=$ ed un nuovo simbolo per funzione unaria singl. Per t un termine del linguaggio, scriviamo $t = t'$ come zucchero sintattico per la formula $= (t, t')$ e scriviamo $\{t\}$ come zucchero sintattico per il termine $\text{singl}(t)$. Siano ax_singl, identita, uguaglianza le seguenti formule:

$$\text{identita} := \forall a.a = a$$

$$\text{uguaglianza} := \forall a.\forall b.\forall c.(a = b \to b \in c \to a \in c)$$

$$\text{ax\_singl} := \forall a.\forall b.(b \in \{a\} \leftrightarrow b = a).$$

Dimostra in DN intuizionista il seguente enunciato:

$$\text{ax\_singl}, \text{identita}, \text{uguaglianza} \vdash \text{HaElem}_a \to \exists b.(b \subseteq a \wedge \text{HaElem}_b)$$

[Suggerimento: Pensala prima in stile "matematichese": dato l'insieme a, chi sarà mai questo misterioso insieme b (che sia esprimibile nel linguaggio formale che ci siamo dati) ?]

Osserva che, alla fine, avendo a disposizione il simbolo per funzione singl e gli assiomi del punto 2, otteniamo l'enunciato:

$$\vdash \forall a.(\mathrm{HaElem}_a \leftrightarrow \exists b.(b \subseteq a \wedge \mathrm{HaElem}_b))$$

che esprime il fatto che pe ogni insieme, avere almeno un elemento è equivalente ad ammettere un sottoinsieme con almeno un elemento.