COMPITO di LOGICA PER INFORMATICA (A)
7 luglio 2004

Nome: Matricola:

Esercizio 1.1 Siano $z : 0 \to (0 \to 0)$ *var*, $x : 0$ *var*, $y : 0$ *var*, $a : 0$ *exp*, $b : 0$ *exp*. Si considerino le espressioni $e_1 \equiv (z)\ z(a)(b) \equiv \lambda z.z(a)(b) : (0 \to (0 \to 0)) \to 0$ e $e_2 \equiv (x)(y)\ x \equiv \lambda x.\lambda y.x : 0 \to (0 \to 0)$ *exp*, dove $\mathsf{FV}(a) = \mathsf{FV}(b) = \emptyset$ e $\mathsf{nf}(a) \equiv a$.

- Si determinino: $\mathsf{FV}(e_1)$ e $\mathsf{FV}(e_2)$;
- Si riduca in forma normale l'espressione $e_1(e_2) : 0$ *exp*

Siano inoltre $d : \beta$ *exp*, $b : \alpha$ *exp*, $f : \alpha$ *exp*, $y : \alpha$ *var*. Dimostrare che

- $d[y := b][y := f] = d[y := b[y := f]]$.

Soluzione.

- $\mathsf{FV}(e_1) = \mathsf{FV}(\lambda z.z(a)(b)) \equiv \mathsf{FV}(z(a)(b)) \setminus \{z\} = (\mathsf{FV}(z) \cup \mathsf{FV}(a) \cup \mathsf{FV}(b)) \setminus \{z\} = \emptyset$,
  $\mathsf{FV}(e_2) = \mathsf{FV}(\lambda x.\lambda y.x) = \mathsf{FV}(\lambda y.x) \setminus \{x\} = (\mathsf{FV}(x) \setminus \{y\}) \setminus \{x\} = \emptyset$.

- $e_1(e_2) \Rightarrow_\beta z(a)(b)[z := \lambda x.\lambda y.x] \equiv \mathrm{S}OST \equiv (\lambda x.\lambda y.x)(a)(b) \Rightarrow_\beta (\lambda y.x[x := a])(b) \equiv \mathrm{S}OST \equiv (\lambda y.a)(b) \Rightarrow_\beta a[y := b] \equiv a$, poiché per ipotesi $\mathsf{FV}(a) = \emptyset$.

  Fin qui abbiamo dimostrato che $e_1(e_2) = a$. Da ciò segue subito che $\mathsf{nf}(e_1(e_2)) = \mathsf{nf}(a)$ e quindi, in base all'ipotesi che $\mathsf{nf}(a) = a$, otteniamo per transitività $\mathsf{nf}(e_1(e_2)) = a$.

- Per la β-uguaglianza $d[y := b] = (\lambda y.d)(b) : \beta$. Quindi

  $$d[y := b][y := f] = (\lambda y.d)(b)[y := f]$$

  Ora, per definizione di sostituzione in un'applicazione, risulta

  $$(\lambda y.d)(b)[y := f] \equiv (\lambda y.d)[y := f](b[y := f])$$

  e, per le regole di sostituzione in una astrazione e la β-uguaglianza, otteniamo

  $$(\lambda y.d)[y := f](b[y := f]) \equiv (\lambda y.d)(b[y := f]) = d[y := b[y := f]]$$

  e quindi per transitività si ottiene $d[y := b][y := f] = d[y := b[y := f]]$.

Esercizio 1.2
Si considerino le seguenti formule

$$\begin{aligned}\Phi_1 &\equiv \forall x.\exists y.R(x,y) \land \neg R(y,x),\\ \Phi_2 &\equiv \forall x.\forall y.\forall z.R(x,y) \to (R(y,z) \to R(x,z)),\\ \Phi_3 &\equiv \forall x.\forall y.\forall z.(R(x,y) \land R(x,z) \land y \neq z) \to (R(y,z) \lor R(z,y))\end{aligned}$$

Esibire due strutture tali che la prima soddisfi Φ_1, Φ_2 e Φ_3 e la seconda soddisfi Φ_1, Φ_2 ma falsifichi Φ_3.

Soluzione. Per soddisfare Φ_1, Φ_2 e Φ_3 si può considerare un qualsiasi insieme infinito su cui si possa definere una relazione d'ordine lineare senza massimo (ad esempio si possono considerare i numeri naturali con l'ordine stretto) ed interpretare R in tale relazione d'ordine. Per soddisfare Φ_1 e Φ_2 e falsificare Φ_3 basta considerare un qualsiasi insieme infinito su cui si possa definire una relazione d'ordine senza massimo (ad esempio, l'insieme degli alberi binari infiniti) ed interpretare R in tale relazione ordine.

Esercizio 1.3
Dimostrare che $\Phi \cup \{\alpha\} \vdash \beta$ se e solo se $\Phi \cup \{\neg\beta\} \vdash \neg\alpha$.

Soluzione. Per definizione sappiamo che $\Phi \cup \{\alpha\} \vdash \beta$ vale se e solo se $\Phi \cup \{\alpha\} \cup \{\neg\beta\}$ è confutabile e che $\Phi \cup \{\neg\beta\} \vdash \neg\alpha$ vale se e solo se $\Phi \cup \{\neg\beta\} \cup \{\neg\neg\alpha\}$ è confutabile.
Ora se assumiamo che $\Phi \cup \{\alpha\} \cup \{\neg\beta\}$ sia confutabile possiamo ottenere un albero di confutazione per $\Phi \cup \{\neg\beta\} \cup \{\neg\neg\alpha\}$ semplicemente incollando l'albero di confutazione per $\Phi \cup \{\alpha\} \cup \{\neg\beta\}$ all'albero ottenuto espanendo la formula $\neg\neg\alpha$.
Viceversa, se l'insieme di formule $\Phi \cup \{\neg\beta\} \cup \{\neg\neg\alpha\}$ è confutabile allora esso non è soddisfacibile (teorema di correttezza) e quindi, in una qualsiasi valutazione σ, almeno una delle sue formule viene valutata in falso; se si tratta di una formula in Φ oppure di $\neg\beta$ allora σ non soddisfa neppure $\Phi \cup \{\alpha\} \cup \{\neg\beta\}$; d'altra parte se si tratta di $\neg\neg\alpha$ allora σ rende falsa anche α e quindi anche in questo caso non soddisfa l'insieme $\Phi \cup \{\alpha\} \cup \{\neg\beta\}$.
Quindi l'insieme $\Phi \cup \{\alpha\} \cup \{\neg\beta\}$ non è soddisfacibile e quindi è confutabile per il teorema di completezza.

Esercizio 1.4
Dimostrare, utilizzando il metodo degli alberi predicativi, che la seguente formula è una verità logica

$$\exists x.((\exists x.\alpha) \rightarrow \alpha)$$

Verificare poi tale risultato usando direttamente la definizione di interpretazione.

Soluzione. Costruiamo dapprima l'albero di confutazione per $\neg\exists x.((\exists x.\alpha) \rightarrow \alpha)$.

$$\neg\exists x.((\exists x.\alpha) \rightarrow \alpha)$$

$$\neg((\exists x.\alpha) \rightarrow \alpha)[x := x]$$

$$(\exists x.\alpha), \neg\alpha[x := x]$$

$$\alpha[x := y]$$

$$\neg((\exists x.\alpha) \rightarrow \alpha)[x := y]$$

$$(\exists x.\alpha), \neg\alpha[x := y]$$

$$X$$

Vediamo ora che si tratta di una formula vera in ogni interpretazione σ.

$$\begin{array}{llll}
(\exists x.((\exists x.\alpha) \rightarrow \alpha))^\sigma = \top & \text{sse} & \text{esiste } u \in U \text{ tale che} & ((\exists x.\alpha) \rightarrow \alpha)^{\sigma(x/u)} = \top \\
& \text{sse} & \text{esiste } u \in U \text{ tale che} & (\exists x.\alpha)^{\sigma(x/u)} = \bot \\
& & & \text{oppure } \alpha^{\sigma(x/u)} = \top \\
& \text{sse} & \text{esiste } u \in U \text{ tale che} & \text{per ogni } v \in U \ \alpha^{\sigma(x/u)(x/v)} = \bot \\
& & & \text{oppure } \alpha^{\sigma(x/u)} = \top \\
& \text{sse} & \text{esiste } u \in U \text{ tale che} & \text{per ogni } v \in U \ \alpha^{\sigma(x/v)} = \bot \\
& & & \text{oppure } \alpha^{\sigma(x/u)} = \top
\end{array}$$

Esercizio 1.5
Costruire una macchina di Turing che ricevendo un input costituito da un numero $n \geq 0$ di S_1, sia tale che:

- se $n = 0$ non si ferma mai;
- se $n > 0$ aggiunge un S_1 e si ferma.

Scrivere quindi la formula del primo ordine che ne rappresenta l'arresto. (Si utilizzi la convenzione sulla configurazione iniziale di una macchina di Turing)

Soluzione. Una possibile macchina di Turing che operi come richiesto si può realizzare col seguente programma:

$$\begin{array}{cccc} Q_1 & S_0 & S_0 & Q_1 \\ Q_1 & S_1 & L & Q_2 \\ Q_2 & S_0 & S_1 & Q_2 \end{array}$$

Una tale macchina non si aresta mai se l'input è costituito da soli S_0 (cioè da un numero $n = 0$ di S_1), mentre se trova almeno un S_1 sul nastro, nella configurazione iniziale standard, si sposta a sinistra e passa nello stato Q_2 dove legge S_0 scrive S_1 e rimane nello stato Q_2, e se legge S_1 si arresta (perché non sa cosa fare).
La formula che ne esprima l'arresto è quindi:

$$\exists\, t \exists\, x (tQ_2x \wedge tS_1x).$$

Esercizio 1.6
Sia $\mathsf{Th}(\mathsf{Z})$ una qualsiasi teoria del primo ordine per i numeri interi in un linguaggio che contenga una costante 0, un simbolo di funzione unario s, un simbolo di funzione unario p e un simbolo predicativo binario $<$ che saranno interpretati rispettivamente nel numero naturale zero, nella funzione successore, nella funzione predecessore e nella relazione di ordine stretto. Dimostrare che $\mathsf{Th}(\mathsf{Z})$ è valida in una struttura che contiene elementi minori di ogni numero intero. (Sugg.: si utilizzi il teorema di compattezza dopo aver definito un insieme di formule che per essere soddisfatte richiedano l'esistenza di elementi minori di qualunque numero intero)

Soluzione. Si consideri il seguente insieme di formule

$$\Sigma = \{x < 0, x < \mathsf{p}(0), x < \mathsf{p}(\mathsf{p}(0)), \ldots, x < \mathsf{p}^n(0), \ldots\}$$

Allora ogni sottoinsieme finito di $\mathsf{Th}(\mathsf{Z}) \cup \Sigma$ è soddisfacibile nella struttura dei numeri interi a patto di interpretare la variabile x in un numero intero minore del più grande indice della formula $x\mathsf{p}^n(0)$ che appare nel sottoinsieme finito di Σ che stiamo considerando. Quindi, per il teorema di compattezza, l'insieme di formule $\mathsf{Th}(\mathsf{Z}) \cup \Sigma$ è soddisfacibile ma allora la struttura che lo soddisfa deve contenere almeno un elemento minore di tutti i numeri interi per permettere di interpretare la variabile x.

COMPITO di LOGICA PER INFORMATICA (B)
7 luglio 2004

Nome: Matricola:

Esercizio 2.1 Siano $z : 0 \to (0 \to 0)\ var$, $x : 0\ var$, $y : 0\ var$, $a : 0\ exp$, $b : 0\ exp$. Si considerino le espressioni $e_1 \equiv (z)\ z(a)(b) \equiv \lambda z.z(a)(b) : (0 \to (0 \to 0)) \to 0$ e $e_2 \equiv (x)(y)\ x \equiv \lambda x.\lambda y.y : 0 \to (0 \to 0)\ exp$, dove $\mathsf{FV}(a) = \mathsf{FV}(b) = \emptyset$ e $\mathsf{nf}(b) \equiv b$.

- Si determinino: $\mathsf{FV}(e_1)$ e $\mathsf{FV}(e_2)$;
- Si riduca in forma normale l'espressione $e_1(e_2) : 0\ exp$

Siano inoltre $b : \beta\ exp$, $d : \alpha\ exp$, $c : \alpha\ exp$, $y : \alpha\ var$. Dimostrare che:

- $b[y := d][y := c] = b[y := d[y := c]]$.

Soluzione.

- $\mathsf{FV}(e_1) = \mathsf{FV}(\lambda z.z(a)(b)) \equiv \mathsf{FV}(z(a)(b)) \setminus \{z\} = (\mathsf{FV}(z) \cup \mathsf{FV}(a) \cup \mathsf{FV}(b)) \setminus \{z\} = \emptyset$,
  $\mathsf{FV}(e_2) = \mathsf{FV}(\lambda x.\lambda y.y) = \mathsf{FV}(\lambda y.y) \setminus \{x\} = (\mathsf{FV}(x) \setminus \{y\}) \setminus \{x\} = \emptyset$.
- $e_1(e_2) \Rightarrow_\beta z(a)(b)[z := \lambda x.\lambda y.y] \equiv \mathrm{S}OST \equiv (\lambda x.\lambda y.y)(a)(b) \Rightarrow_\beta (\lambda y.y[x := a])(b) \equiv \mathrm{S}OST \equiv (\lambda y.y)(b) \Rightarrow_\beta y[y := b] \equiv b$.

  Fin qui abbiamo dimostrato che $e_1(e_2) = b$. Da ciò segue subito che $\mathsf{nf}(e_1(e_2)) = \mathsf{nf}(b)$ e quindi, in base all'ipotesi che $\mathsf{nf}(b) = b$, otteniamo per transitività $\mathsf{nf}(e_1(e_2)) = b$.
- Per la β-uguaglianza $b[y := d] = (\lambda y.b)(d) : \beta$. Quindi

  $$b[y := d][y := c] = (\lambda y.b)(d)[y := c]$$

  Ora, per definizione di sostituzione in un'applicazione, risulta

  $$(\lambda y.b)(d)[y := c] \equiv (\lambda y.b)[y := c](d[y := c])$$

  e, per le regole di sostituzione in un'astrazione e la β-uguaglianza, otteniamo

  $$(\lambda y.b)[y := c](d[y := c]) \equiv (\lambda y.b)(d[y := c]) = b[y := d[y := c]]$$

  e quindi per transitività si ottiene $b[y := d][y := c] = b[y := d[y := c]]$.

Esercizio 2.2
Si considerino le seguenti formule

$$\begin{aligned}
\Phi_1 &\equiv \forall x.\exists y.R(x,y) \wedge \neg R(y,x),\\
\Phi_2 &\equiv \forall x.\forall y.\forall z.(R(x,y) \wedge R(y,z)) \rightarrow R(x,z),\\
\Phi_3 &\equiv \forall x.\forall y.\forall z.(R(x,y) \wedge R(x,z)) \rightarrow \exists t.(R(y,t) \wedge R(z,t))
\end{aligned}$$

Esibire due strutture tali che la prima soddisfi Φ_1, Φ_2 e Φ_3 e la seconda soddisfi Φ_1, Φ_2 ma falsifichi Φ_3.

Soluzione. Per soddisfare Φ_1, Φ_2 e Φ_3 basta considerare un qualsiasi insieme infinito su cui si possa definire una relazione d'ordine senza massimo (ad esempio, l'insieme dei numeri naturali e la relazione d'ordine stretto) ed interpretare R in tale relazione ordine mentre per soddisfare Φ_1, Φ_2 e falsificare Φ_3 si possono cosiderare due copie di un qualsiasi insieme infinito su cui si possa definere una relazione d'ordine lineare connesse da un elemento (ad esempio, due copie dei numeri naturali con l'ordine stretto connesse sullo 0) ed interpretare R in tale relazione d'ordine.

Esercizio 2.3
Dimostrare che $\Phi \cup \{\alpha\} \vdash \neg\beta$ se e solo se $\Phi \cup \{\beta\} \vdash \neg\alpha$.

Soluzione. Per definizione sappiamo che $\Phi \cup \{\alpha\} \vdash \neg\beta$ vale se e solo se $\Phi \cup \{\alpha\} \cup \{\neg\neg\beta\}$ è confutabile e che $\Phi \cup \{\beta\} \vdash \neg\alpha$ vale se e solo se $\Phi \cup \{\beta\} \cup \{\neg\neg\alpha\}$ è confutabile.
Ora se assumiamo che l'insieme di formule $\Phi \cup \{\alpha\} \cup \{\neg\neg\beta\}$ sia confutabile allora esso non è soddisfacibile (teorema di correttezza) e quindi presa una qualsiasi valutazione σ almeno una delle formule in tale insieme viene valutata da σ in falso. Ora, se si tratta di una formula di Φ allora anche $\Phi \cup \{\beta\} \cup \{\neg\neg\alpha\}$ non è soddisfacibile mentre se σ rende falsa α allora esse rende falsa anche $\neg\neg\alpha$ e quindi di nuovo $\Phi \cup \{\beta\} \cup \{\neg\neg\alpha\}$ non è soddisfacibile; infine se σ rende falsa $\neg\neg\beta$ allora esse rende falsa anche β e quindi anche in questo caso $\Phi \cup \{\beta\} \cup \{\neg\neg\alpha\}$ non è soddisfacibile. Quindi in ogni caso $\Phi \cup \{\beta\} \cup \{\neg\neg\alpha\}$ non è soddisfacibile ed è perciò confutabile per il teorema di completezza.
Viceversa, in modo completamente analogo si può dimostrare che se l'insieme di formule $\Phi \cup \{\beta\} \cup \{\neg\neg\alpha\}$ è confutabile allora anche $\Phi \cup \{\alpha\} \cup \{\neg\neg\beta\}$ è confutabile.

Esercizio 2.4
Dimostrare, utilizzando il metodo degli alberi predicativi, che la seguente formula è una verità logica

$$\neg\forall x.(\alpha \wedge (\exists x.\neg\alpha))$$

Verificare poi tale risultato usando direttamente la definizione di interpretazione.

Soluzione. Costruiamo dapprima l'albero di confutazione per $\neg\neg\forall x.(\alpha \wedge (\exists x.\neg\alpha))$.

$$\neg\neg\forall x.(\alpha \wedge (\exists x.\neg\alpha))$$

$$\forall x.(\alpha \wedge (\exists x.\neg\alpha))$$

$$(\alpha \wedge (\exists x.\neg\alpha))[x := x]$$

$$\alpha[x := x], \exists x.\neg\alpha$$

$$\neg\alpha[x := y]$$

$$(\alpha \wedge (\exists x.\neg\alpha))[x := y]$$

$$\alpha[x := y], \exists x.\neg\alpha$$

$$X$$

Vediamo ora che si tratta di una formula vera in ogni interpretazione σ.

$$\begin{array}{llll}
(\neg\forall x.(\alpha \wedge (\exists x.\neg\alpha)))^{\sigma} = \top & \text{sse} & (\forall x.(\alpha \wedge (\exists x.\neg\alpha)))^{\sigma} = \bot & \\
 & \text{sse} & \text{esiste } u \in U \text{ tale che} & (\alpha \wedge (\exists x.\neg\alpha))^{\sigma(x/u)} = \bot \\
 & \text{sse} & \text{esiste } u \in U \text{ tale che} & (\alpha)^{\sigma(x/u)} = \bot \text{ oppure} \\
 & & & (\exists x.\neg\alpha)^{\sigma(x/u)} = \bot \\
 & \text{sse} & \text{esiste } u \in U \text{ tale che} & (\alpha)^{\sigma(x/u)} = \bot \text{ oppure} \\
 & & & \text{per ogni } v \in U \ (\neg\alpha)^{\sigma(x/u)(x/v)} = \bot \\
 & \text{sse} & \text{esiste } u \in U \text{ tale che} & (\alpha)^{\sigma(x/u)} = \bot \text{ oppure} \\
 & & & \text{per ogni } v \in U \ (\alpha)^{\sigma(x/u)(x/v)} = \top \\
 & \text{sse} & \text{esiste } u \in U \text{ tale che} & (\alpha)^{\sigma(x/u)} = \bot \text{ oppure} \\
 & & & \text{per ogni } v \in U \ (\alpha)^{\sigma(x/v)} = \top
\end{array}$$

Esercizio 2.5
Costruire una macchina di Turing che ricevendo un input costituito da un numero $n \geq 0$ di S_1, sia tale che:

- se $n = 0$ scrive un S_1 e si ferma;
- se $n > 0$ cancella un S_1 e si ferma.

Scrivere quindi la formula del primo ordine che ne rappresenta l'arresto. (Si utilizzi la convenzione sulla configurazione iniziale di una macchina di Turing)

Soluzione. Una possibile macchina di Turing che operi come richiesto si può realizzare col seguente programma:

$$\begin{array}{cccc} Q_1 & S_0 & S_0 & Q_2 \\ Q_1 & S_1 & S_1 & Q_3 \\ Q_2 & S_0 & S_1 & Q_2 \\ Q_3 & S_1 & S_0 & Q_3 \end{array}$$

Una tale macchina si arresta sia che l'input sia costituito da soli S_0 (cioè da un numero $n = 0$ di S_1), sia che trovi almeno un S_1 sul nastro. Infatti se nello stato iniziale Q_1 la macchina legge S_0 allora passa nello stato Q_2 dove scrive un S_1 e si arresta, mentre se nello stato Q_1 legge S_1 allora passa nello stato Q_3 dove cancella un S_1 e si ferma.
La formula che ne esprima l'arresto è:

$$\exists\, t \exists\, x(tQ_2x \wedge tS_1x) \vee \exists\, t \exists\, x(tQ_3x \wedge tS_0x).$$

Esercizio 2.6
Sia $\mathsf{Th}(\mathsf{N})$ una qualsiasi teoria del primo ordine per i numeri naturali in un linguaggio che contenga una costante 0, un simbolo di funzione unario s e un simbolo predicativo binario $<$ che saranno interpretati rispettivamente nel numero naturale zero, nella funzione sucessore e nella relazione di ordine stretto. Dimostrare che $\mathsf{Th}(\mathsf{N})$ è valida in una struttura che contiene elementi maggiori di ogni numero naturale. (Sugg.: si utilizzi il teorema di compattezza dopo aver definito un insieme di formule che per essere soddisfatte richiedano l'esistenza di elementi maggiori di qualunque numero naturale)

Soluzione. Si consideri il seguente insieme di formule

$$\Sigma = \{0 < x, \mathsf{s}(0) < x, \mathsf{s}(\mathsf{s}(0)) < x, \ldots, \mathsf{s}^n(0) < x, \ldots\}$$

Allora ogni sottoinsieme finito di $\mathsf{Th}(\mathsf{N}) \cup \Sigma$ è soddisfacibile nella struttura dei numeri naturali a patto di interpretare la variabile x in un numero naturale maggiore del più grande indice della formula $\mathsf{s}^n(0) < x$ che appare nel sottoinsieme finito di Σ che stiamo considerando. Quindi, per il teorema di compattezza, l'insieme di formule $\mathsf{Th}(\mathsf{N}) \cup \Sigma$ è soddisfacibile ma allora la struttura che lo soddisfa deve contenere almeno un elemento maggiore di tutti i numeri naturali per permettere di interpretare la variabile x.