COMPITO di LOGICA PER INFORMATICA
1 settembre 2004

Nome: Matricola:

Esercizio 1

- Siano a e b due espressioni di arietà 0. Determinare un tipo per l'espressione $\lambda z.z(a)(b)$.
- Si definisca la coppia $\langle a,b\rangle$ ponendo $\langle a,b\rangle \equiv \lambda z.z(a)(b)$. Costruire un programma Π_1 tale che $\Pi_1(\langle a,b\rangle) = a$ dimostrando che $\Pi_1(\langle a,b\rangle)$ si riduce ad a.
- Determinare il tipo di Π_1.

Soluzione.

- L'espressione $\lambda z.z(a)(b)$ è una astrazione quindi il suo tipo sarà della forma $\alpha \to \beta$ per una opportuna scelta di α e β. Ora α deve essere il tipo della variabile z che deve poter essere applicata ad a e b che per ipotesi sono due espressioni di arietà 0 mentre β deve essere il tipo del risultato di tale applicazione. Quindi α dovrà essere uguale a $0 \to (0 \to \beta)$. La soluzione più semplice è quindi porre $\beta = 0$ ed ottenere perciò che un possibile tipo per $\lambda z.z(a)(b)$ è $(0 \to (0 \to 0)) \to 0$.
- Un programma che risolve il problema è porre

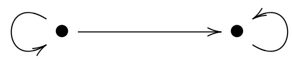
  $$\Pi_1 \equiv \lambda w.w(\lambda x.\lambda y.x)$$

  Infatti se applichiamo Π_1 alla coppia $\langle a,b\rangle$ otteniamo

  $$\begin{array}{rcl} \Pi_1(\langle a,b\rangle) & \equiv & (\lambda w.w(\lambda x.\lambda y.x))(\lambda z.z(a)(b)) \\ & =_\beta & (\lambda z.z(a)(b))(\lambda x.\lambda y.x) \\ & =_\beta & (\lambda x.\lambda y.x)(a)(b) \\ & =_\beta & (\lambda y.a)(b) \\ & =_\beta & a \end{array}$$

- Visto che abbiamo scelto $(0 \to (0 \to 0)) \to 0$ come tipo per la coppia $\langle a,b\rangle$ ed il tipo di a, vale a dire del risultato di applicare Π_1 a $\langle a,b\rangle$, è 0 ne segue immediatamente che il tipo di Π_1 è $((0 \to (0 \to 0)) \to 0) \to 0$.

Esercizio 2
Utilizzando un linguaggio del primo ordine contenente un predicato R a due posti, si determini un insieme S di formule tale che S è soddisfatto in tutte e sole le strutture di due elementi linearmente ordinati.



Soluzione. Condizione sufficiente affinchè la struttura che soddisfa S abbia esattamente due elementi è che S contenga la formula

$$(\exists x.\exists y.\neg(x = y)) \wedge (\forall x.\forall y.\forall z.(x = y) \vee (x = z) \vee (y = z))$$

Inoltre, affinchè la relazione tra gli elementi della struttura sia d'ordine bisogna che sia riflessiva, antisimmetrica e transitiva e quindi S deve contenere le formule

$$\begin{gathered}\forall x.R(x,x)\\ \forall x.\forall y.(R(x,y) \wedge R(y,x)) \rightarrow x = y\\ \forall x.\forall y.\forall z.(R(x,y) \wedge R(y,z)) \rightarrow R(x,z)\end{gathered}$$

Infine, affinchè l'ordine sia lineare basta che S contenga la formula

$$(\exists x.\exists y.R(x,y) \wedge \neg(x = y)) \wedge (\exists x.\forall y.R(x,y) \rightarrow (x = y))$$

Esercizio 3
Dimostrare, utilizzando il metodo degli alberi predicativi, che la seguente formula è una verità logica

$$((\exists x.A(x,a)) \to (\forall x.(\forall y.A(x,y)) \to B(x))) \to ((\forall y.A(b,y)) \to B(b))$$

Verificare poi tale risultato usando direttamente la definizione di interpretazione.

Soluzione. Costruiamo dapprima l'albero di confutazione per

$$\neg[((\exists x.A(x,a)) \to (\forall x.(\forall y.A(x,y)) \to B(x))) \to ((\forall y.A(b,y)) \to B(b))]$$

$$\neg[((\exists x.A(x,a)) \to (\forall x.(\forall y.A(x,y)) \to B(x))) \to ((\forall y.A(b,y)) \to B(b))]$$

$$((\exists x.A(x,a)) \to (\forall x.(\forall y.A(x,y)) \to B(x))), \neg[((\forall y.A(b,y)) \to B(b))]$$

$\neg(\exists x.A(x,a))$	$\forall x.(\forall y.A(x,y)) \to B(x))$
$\forall y.A(b,y), \neg B(b)$	$(\forall y.A(x,y)) \to B(x))[x := b]$
$A(b,y)[y := a]$	X
$\neg A(x,a)[x := b]$	
X	

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

$(((\exists x.A(x,a)) \to (\forall x.(\forall y.A(x,y)) \to B(x))) \to ((\forall y.A(b,y)) \to B(b)))^{\sigma} = \top$
sse
$(((\exists x.A(x,a)) \to (\forall x.(\forall y.A(x,y)) \to B(x)))^{\sigma} = \bot$ oppure $((\forall y.A(b,y)) \to B(b)))^{\sigma} = \top$
sse
$(\exists x.A(x,a))^{\sigma} = \top$ e $(\forall x.(\forall y.A(x,y)) \to B(x))^{\sigma} = \bot$ oppure $(\forall y.A(b,y))^{\sigma} = \bot$ oppure $(B(b))^{\sigma} = \top$
sse
esiste $u \in U$ tale che $(A(x,a))^{\sigma(x/u)} = \top$ e esiste $v \in U$ tale che $(\forall y.A(x,y)) \to B(x))^{\sigma(x/v)} = \bot$
oppure esiste $w \in U$ tale che $(A(b,y))^{\sigma(y/w)} = \bot$ oppure $(B(b))^{\sigma} = \top$
sse
esiste $u \in U$ tale che $(A(x,a))^{\sigma(x/u)} = \top$ e esiste $v \in U$ tale che $(\forall y.A(x,y))^{\sigma(x/v)} = \top$ e $(B(x))^{\sigma(x/v)} = \bot$
oppure esiste $w \in U$ tale che $(A(b,y))^{\sigma(y/w)} = \bot$ oppure $(B(b))^{\sigma} = \top$
sse
esiste $u \in U$ tale che $(A(x,a))^{\sigma(x/u)} = \top$ e esiste $v \in U$ tale che, per ogni $z \in U$, $(A(x,y))^{\sigma(x/v)(y/z)} = \top$ e $(B(x))^{\sigma(x/v)} = \bot$
oppure esiste $w \in U$ tale che $(A(b,y))^{\sigma(y/w)} = \bot$ oppure $(B(b))^{\sigma} = \top$

Esercizio 4

Costruire una macchina di Turing che ricevendo in input un numero naturale in rappresentazione binaria (vale a dire scritto utilizzando solamente 0 e 1 ed utilizzando il simbolo $\flat$ per rappresentare una casella vuota del nastro) si arresta se e solo se tale numero è divisibile per 4. 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. Si osservi prima di tutto che un numero scritto in notazione binaria è divisibile per 4 se e solo se è il numero 0 oppure le ultime due cifre meno significative sono 0. Una possibile macchina di Turing che operi come richiesto si può quindi realizzare col seguente programma:

$$\begin{array}{cccc} q_1 & 0 & R & q_2 \\ q_1 & 1 & 1 & q_1 \\ q_1 & \flat & \flat & q_1 \\ q_2 & 1 & 1 & q_2 \end{array}$$

Infatti questo programma non si arresta comunque in q_1 mentre si arresta in q_2 qualora arrivi allo stato q_2, e quindi se la cifra meno significativa del numero in input è 0, e in q_2 trova un $\flat$, e quindi il numero in input è 0, oppure ancora uno 0 come seconda cifra meno significativa.

La formula che ne esprime l'arresto è quindi:

$$\exists\, t \exists\, x (t Q_2 x \wedge t S_0 x) \vee \exists\, t \exists\, x (t Q_2 x \wedge t S_\flat x)$$

Esercizio 5

Si dimostri che se l'insieme $S = \{\alpha_1 \wedge \neg\alpha_2, \alpha_2 \wedge \neg\alpha_3, \ldots, \alpha_n \wedge \neg\alpha_{n+1}, \ldots\}$ non è soddisfacibile allora esiste un numero naturale k tale che $(\alpha_1 \to \alpha_2) \vee \ldots \vee (\alpha_k \to \alpha_{k+1})$ è una tautologia. (Sugg.: si utilizzi il teorema di compattezza)

Soluzione. Per il teorema di compattezza, S è insoddisfacibile se e solo se esiste un suo sottoinsieme finito che è insoddisfacibile. Ma allora esiste un massimo indice k tale che la formula $\alpha_k \wedge \neg\alpha_{k+1}$ appartiene a tale sottoinsieme finito e quindi anche l'insieme $\{\alpha_1 \wedge \neg\alpha_2, \ldots, \alpha_k \wedge \neg\alpha_{k+1}\}$ è insoddisfacibile in quanto sovrainsieme di un insieme insoddisfacibile. Ma allora per una qualsiasi valutazione σ deve esistere almeno una formula $\alpha_h \wedge \neg\alpha_{h+1}$, per h compreso tra 1 e k, che viene interpretata in falso da σ, ma allora o $\sigma(\alpha_h) = \bot$ o $\sigma(\alpha_{h+1}) = \top$ e quindi in ogni caso $\sigma(\alpha_h \to \alpha_{h+1}) = \top$ e quindi la formula $(\alpha_1 \to \alpha_2) \vee \ldots \vee (\alpha_k \to \alpha_{k+1})$ viene interpretata da σ in vero. Ma questo vale per ogni valutazione e quindi $(\alpha_1 \to \alpha_2) \vee \ldots \vee (\alpha_k \to \alpha_{k+1})$ è una tautologia.

Soluzione veloce. Esiste tuttavia una soluzione molto più semplice e veloce di questo esercizio se ci si accorge che per rendere vera la conclusione basta scegliere $k = 2$ visto che $(\alpha_1 \to \alpha_2) \vee (\alpha_2 \to \alpha_3)$ è già una tautologia come si può facilmente verificare utilizzando le tabelline di verità.