COMPITO di LOGICA PER INFORMATICA (A)
23 giugno 2004

Nome: Matricola:

Esercizio 1.1
Sia $c : \gamma$ exp, $x : \alpha$ var e $a : \alpha$ exp.

1. Dimostrare che $\mathsf{FV}(c[x := a]) \subseteq (\mathsf{FV}(c) - \{x\}) \cup \mathsf{FV}(a)$.
2. Esibire un esempio in cui l'inclusione è propria.

Soluzione. La prova è per induzione sulla definizione ricorsiva dell'algoritmo di sostituzione.

- $(c \equiv x)$
$$\mathsf{FV}(x[x := a]) = \mathsf{FV}(a) \subseteq (\mathsf{FV}(x) - \{x\}) \cup \mathsf{FV}(a)$$
- $(c \equiv y)$
$$\mathsf{FV}(y[x := a]) = \mathsf{FV}(y) = \{y\} \subseteq (\mathsf{FV}(y) - \{x\}) \cup \mathsf{FV}(a)$$
- $(c \equiv c_1(c_2))$
$$\begin{array}{rcl} \mathsf{FV}(c_1(c_2)[x := a]) & = & \mathsf{FV}(c_1[x := a](c_2[x := a])) \\ & = & \mathsf{FV}(c_1[x := a]) \cup \mathsf{FV}(c_2[x := a]) \\ & \subseteq & ((\mathsf{FV}(c_1) - \{x\}) \cup \mathsf{FV}(a)) \cup ((\mathsf{FV}(c_1) - \{x\}) \cup \mathsf{FV}(a)) \\ & = & ((\mathsf{FV}(c_1(c_2)) - \{x\}) \cup \mathsf{FV}(a)) \end{array}$$
- $(c \equiv \lambda x.c_1)$
$$\mathsf{FV}((\lambda x.c_1)[x := a]) = \mathsf{FV}(\lambda x.c_1) = \mathsf{FV}(c_1) - \{x\} \subseteq (\mathsf{FV}(\lambda x.c_1) - \{x\}) \cup \mathsf{FV}(a)$$
- $(c \equiv \lambda y.c_1)$
$$\begin{array}{rcl} \mathsf{FV}((\lambda y.c_1)[x := a]) & = & \mathsf{FV}(\lambda y.c_1[x := a]) \\ & = & \mathsf{FV}(c_1[x := a]) - \{y\} \\ & \subseteq & ((\mathsf{FV}(c_1) - \{x\}) \cup \mathsf{FV}(a)) - \{y\} \\ & \subseteq & (\mathsf{FV}(\lambda y.c_1) - \{x\}) \cup \mathsf{FV}(a) \end{array}$$

Dalle precedenti inclusioni si nota che si può ottenere un semplice esempio di inclusione propria ponendo $c \equiv y : 0$, $x : 0$ e $a \equiv z : 0$. Infatti in tal caso si ha che $\mathsf{FV}(c[x := a]) \equiv \mathsf{FV}(y[x := z]) = \mathsf{FV}(y) = \{y\} \subseteq \{y, z\} = (\mathsf{FV}(y) - \{x\}) \cup \mathsf{FV}(z) \equiv (\mathsf{FV}(c) - \{x\}) \cup \mathsf{FV}(a)$.

Esercizio 1.2
Si consideri il seguente insieme di formule

$$\Phi \equiv \{\forall x.R(x,x), \exists x \exists y \exists z.\neg R(x,y) \wedge \neg R(x,z) \wedge \neg R(y,z)\}$$

Dimostrare che Φ è soddisfacibile ma non è soddisfacibile in alcuna struttura con meno di tre elementi.

Soluzione. Per rendere vere tutte le formule in Φ basta considerare una struttura con almeno tre elementi ed interpretare il simbolo R nella relazione di identità. Infatti in tal caso abbiamo una relazione riflessiva (prima formula) e tre elementi uno diverso dall'altro (seconda formula).
D'altra parte, se abbiamo meno di tre elementi e R deve essere interpretato in una relazione riflessiva allora non possiamo rendere vera la seconda formula perché se $\neg R(x,y)$ viene interpretato in vero allora l'interpretazione di x deve essere diversa dall'interpretazione di y.

Esercizio 1.3
Dimostrare che $\Phi \cup \{\alpha \to \beta\}$ è confutabile se e solo se $\Phi \cup \{\neg\alpha\}$ e $\Phi \cup \{\beta\}$ sono confutabili.

Soluzione. Se $\Phi \cup \{\neg\alpha\}$ e $\Phi \cup \{\beta\}$ sono confutabili allora possiamo costruire un albero di confutazione per $\Phi \cup \{\alpha \to \beta\}$ semplicemente attaccando gli alberi di confutazione per $\Phi \cup \{\neg\alpha\}$ e $\Phi \cup \{\beta\}$ alle foglie dell'albero che si ottiene espandendo la formula $\alpha \to \beta$. D'altra parte se $\Phi \cup \{\alpha \to \beta\}$ è confutabile allora esso non è soddisfacibile (teorema di correttezza) e quindi o Φ non è soddisfacibile, e quindi anche $\Phi \cup \{\neg\alpha\}$ e $\Phi \cup \{\beta\}$ non lo sono e perciò sono confutabili (teorema di completezza), oppure $\alpha \to \beta$ non è soddisfacibile e quindi una qualsiasi valutazione rende vero α e falso β e quindi non soddisfa ne $\Phi \cup \{\neg\alpha\}$ ne $\Phi \cup \{\beta\}$ che sono quindi confutabili (teorema di completezza).

Esercizio 1.4
Dimostrare che

$$\forall x((\alpha \to \beta) \land \exists x.\alpha) \vdash \exists x.\beta$$

Soluzione. Per definizione $\forall x((\alpha \to \beta) \land \exists x.\alpha) \vdash \exists x.\beta$ vale se e solo se

$$\{\forall x((\alpha \to \beta) \land \exists x.\alpha), \neg\exists x.\beta\}$$

è confutabile. Ecco un possibile albero di confutazione:

$$\{\forall x((\alpha \to \beta) \land \exists x.\alpha), \neg\exists x.\beta\}$$

$$(\alpha \to \beta) \land \exists x.\alpha$$

$$\{\alpha \to \beta, \exists x.\alpha\}$$

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

$$(\alpha \to \beta)[x := y] \land \exists x.\alpha$$

$$\{\alpha[x := y] \to \beta[x := y], \exists x.\alpha\}$$

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

Esercizio 1.5
Costruire una macchina di Turing che, agendo su un nastro contenente un input costituito da un numero $n \geq 0$ del solo simbolo S_1 e che contiene altrove solamente S_0, si arresta se e solo se tale input è costituito da esattamente un S_1. Fornire quindi la formula del primo ordine che ne rappresenta l'arresto. (Nota bene: non verranno considerate risposte che non spieghino il funzionamento della macchina)

Soluzione. La macchina che cerchiamo si può realizzare con il seguente programma

$$\begin{array}{cccc} q_0 & S_0 & S_0 & q_0 \\ q_0 & S_1 & R & q_1 \\ q_1 & S_1 & S_1 & q_1 \end{array}$$

che entra in loop immediatamente se il nastro è vuoto e altrimenti controlla che ci sia un unico S_1 entrando in un loop nel caso ne trovi più di uno.
La formula che ne esprime l'arresto è quindi

$$H \equiv \exists t \exists x.(t\, Q_1\, x \wedge t\, S_0\, x)$$

Esercizio 1.6
Si supponga che l'insieme $\Phi \equiv \{\alpha_1 \to \beta_1, \ldots, \alpha_n \to \beta_n, \ldots\}$ sia insoddisfacibile. Dimostrare che esiste un numero naturale k tale che la formula $\beta_1 \wedge \ldots \wedge \beta_k$ è confutabile.

Soluzione. Per il teorema di compattezza, $\Phi \equiv \{\alpha_1 \to \beta_1, \ldots, \alpha_n \to \beta_n, \ldots\}$ 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 $\alpha_k \to \beta_k$ appartiene a tale sottoinsieme finito e quindi anche l'insieme $\{\alpha_1 \to \beta_1, \ldots, \alpha_k \to \beta_k\}$ è insoddisfacibile in quanto sovrainsieme di un insieme insoddisfacibile. Ma allora per qualsiasi valutazione non può accadere che tutte le formule $\beta_1, \ldots, \beta_k$ siano interpretate in vero e quindi la formula $\beta_1 \wedge \ldots \wedge \beta_k$ deve essere interpretata in falso; pertanto la formula $\beta_1 \wedge \ldots \wedge \beta_k$ è insoddisfacibile e quindi confutabile per il teorema di completezza.

COMPITO di LOGICA PER INFORMATICA (B)
23 giugno 2004

Nome: Matricola:

Esercizio 2.1
Sia e una espressione di arietà α.

1. Sapendo che, per ogni $c : \gamma$ exp, ogni $x : \alpha$ var e ogni $a : \alpha$ exp, $\mathsf{FV}(c[x := a]) \subseteq (\mathsf{FV}(c) - \{x\}) \cup \mathsf{FV}(a)$, dimostrare che $\mathsf{FV}(\mathsf{nf}(e)) \subseteq \mathsf{FV}(e)$.
2. Esibire un esempio in cui l'inclusione è propria.

Soluzione. La prova è per induzione sulla definizione ricorsiva dell'algoritmo di normalizzazione.

- ($\alpha \neq 0$ e x nuova)

$$\begin{array}{rcl} \mathsf{FV}(\mathsf{nf}(e)) & = & \mathsf{FV}(\lambda x.\mathsf{nf}(e(x))) \\ & = & \mathsf{FV}(\mathsf{nf}(e(x)) - \{x\} \\ & \subseteq & \mathsf{FV}(e(x)) - \{x\} \\ & = & \mathsf{FV}(e) \end{array}$$

- ($\alpha = 0$ e $e \equiv x(a_1)\ldots(a_n)$)

$$\begin{array}{rcl} \mathsf{FV}(\mathsf{nf}(e)) & = & \mathsf{FV}(\mathsf{nf}(x(a_1)\ldots(a_n))) \\ & = & \mathsf{FV}(x(\mathsf{nf}(a_1))\ldots(\mathsf{nf}(a_n))) \\ & = & \{x\} \cup \mathsf{FV}(\mathsf{nf}(a_1)) \cup \ldots \cup \mathsf{FV}(\mathsf{nf}(a_n)) \\ & \subseteq & \{x\} \cup \mathsf{FV}(a_1) \cup \ldots \cup \mathsf{FV}(a_n) \\ & = & \mathsf{FV}(x(a_1)\ldots(a_n)) \end{array}$$

- ($\alpha = 0$ e $e \equiv (\lambda x.c)(a)(a_1)\ldots(a_n)$)

$$\begin{array}{rcl} \mathsf{FV}(\mathsf{nf}(e)) & = & \mathsf{FV}(\mathsf{nf}((\lambda x.c)(a)(a_1)\ldots(a_n))) \\ & = & \mathsf{FV}(\mathsf{nf}(c[x := a](a_1)\ldots(a_n))) \\ & \subseteq & \mathsf{FV}(c[x := a](a_1)\ldots(a_n)) \\ & = & \mathsf{FV}(c[x := a]) \cup \mathsf{FV}(a_1) \cup \ldots \cup \mathsf{FV}(a_n) \\ & \subseteq & (\mathsf{FV}(c) - \{x\}) \cup \mathsf{FV}(a) \cup \mathsf{FV}(a_1) \cup \ldots \cup \mathsf{FV}(a_n) \\ & = & \mathsf{FV}((\lambda x.c)(a)(a_1)\ldots(a_n)) \end{array}$$

Dalle precedenti inclusioni si nota che si può ottenere un semplice esempio di inclusione propria ponendo $e \equiv (\lambda y.x)(z) : 0$ supponendo che $y : 0$, $x : 0$ e $z : 0$. Infatti in tal caso si ha che $\mathsf{FV}(\mathsf{nf}(e)) = \mathsf{FV}(\mathsf{nf}((\lambda y.x)(z))) = \mathsf{FV}(z) = \{z\} \subseteq \{x, z\} = (\mathsf{FV}(x) - \{y\}) \cup \mathsf{FV}(z) \equiv \mathsf{FV}((\lambda y.x)(z)) = \mathsf{FV}(e)$.

Esercizio 2.2
Si consideri il seguente insieme di formule

$$\Phi \equiv \{\forall x.R(x,x), \exists x((\forall y.R(x,y)) \to (\exists x.\neg R(x,x)))\}$$

Dimostrare che Φ è soddisfacibile ma non è soddisfacibile in alcuna struttura con meno di due elementi.

Soluzione. Per rendere vere tutte le formule in Φ basta considerare una struttura con almeno due elementi ed interpretare il simbolo R nella relazione di identità. Infatti in tal caso abbiamo una relazione riflessiva (prima formula) e quindi tale che $\exists x.\neg R(x,x)$ è interpretato in falso per cui la seconda formula viene interpretata in vero.
D'altra parte, se abbiamo un solo elemento e R deve essere interpretato in una relazione riflessiva per soddisfare la prima formula allora non possiamo rendere vera la seconda formula perché se $\neg R(x,y)$ viene interpretato in vero allora l'interpretazione di x deve essere diversa dall'interpretazione di y.

Esercizio 2.3
Dimostrare che $\Phi \cup \{\neg(\alpha \to \beta)\}$ è confutabile se e solo se $\Phi \cup \{\alpha, \neg\beta\}$ è confutabile.

Soluzione. Se $\Phi \cup \{\alpha, \neg\beta\}$ è confutabile allora possiamo costruire un albero di confutazione per $\Phi \cup \{\neg(\alpha \to \beta)\}$ semplicemente attaccando l'albero di confutazione per $\Phi \cup \{\alpha, \neg\beta\}$ alla foglia dell'albero che si ottiene espandendo la formula $\neg(\alpha \to \beta)$.
D'altra parte se $\Phi \cup \{\neg(\alpha \to \beta)\}$ è confutabile allora esso non è soddisfacibile (teorema di correttezza) e quindi o Φ non è soddisfacibile, e quindi anche $\Phi \cup \{\alpha, \neg\beta\}$ non lo è e perciò è confutabile (teorema di completezza), oppure $\neg(\alpha \to \beta)$ non è soddisfacibile e quindi una qualsiasi valutazione o rende falso α o rende vero β e quindi non soddisfa $\Phi \cup \{\alpha, \neg\beta\}$ che è quindi confutabile (teorema di completezza).

Esercizio 2.4
Dimostrare che

$$\forall x((\alpha \to \beta) \wedge \exists x.\neg\beta) \vdash \exists x.\neg\alpha$$

Soluzione. Per definizione $\forall x((\alpha \to \beta) \wedge \exists x.\neg\beta) \vdash \exists x.\neg\alpha$ vale se e solo se

$$\{\forall x((\alpha \to \beta) \wedge \exists x.\neg\beta), \neg\exists x.\neg\alpha\}$$

è confutabile. Ecco un possibile albero di confutazione:

$$\{\forall x((\alpha \to \beta) \wedge \exists x.\neg\beta), \neg\exists x.\neg\alpha\}$$

$$(\alpha \to \beta) \wedge \exists x.\neg\beta$$

$$\{\alpha \to \beta, \exists x.\neg\beta\}$$

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

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

$$\{\alpha[x := y] \to \beta[x := y], \exists x.\neg\beta\}$$

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

Esercizio 2.5

Costruire una macchina di Turing che agendo su un nastro contenente un input costituito da una sequenza di lunghezza $n \geq 0$ del simbolo S_1 e contiene altrove solamente S_0 si arresta se e solo se n è pari. Fornire quindi la formula del primo ordine che ne rappresenta l'arresto. (Nota bene: non verranno considerate risposte che non spieghino il funzionamento della macchina)

Soluzione. La macchina che cerchiamo si può realizzare con il seguente programma

$$\begin{array}{cccc} q_0 & S_1 & R & q_1 \\ q_1 & S_0 & S_0 & q_1 \\ q_1 & S_1 & R & q_0 \end{array}$$

che si arresta nello stato q_0 dopo aver scandito un numero pari di S_1 oppure entra in loop nello stato q_1 nel caso il numero di S_1 sia dispari.

La formula che esprime l'arresto è quindi

$$H \equiv \exists t \exists x.(t\, Q_0\, x \wedge t\, S_0\, x)$$

Esercizio 2.6

Si supponga che l'insieme $\Phi \equiv \{\alpha_1 \to \beta_1, \dots, \alpha_n \to \beta_n, \dots\}$ sia insoddisfacibile. Dimostrare che esiste un numero naturale k tale che la formula $\alpha_1 \vee \dots \vee \alpha_k$ è una tautologia.

Soluzione. Per il teorema di compattezza, $\Phi \equiv \{\alpha_1 \to \beta_1, \dots, \alpha_n \to \beta_n, \dots\}$ 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 $\alpha_k \to \beta_k$ appartiene a tale sottoinsieme finito e quindi anche l'insieme $\{\alpha_1 \to \beta_1, \dots, \alpha_k \to \beta_k\}$ è insoddisfacibile in quanto sovrainsieme di un insieme insoddisfacibile. Ma allora per qualsiasi valutazione non può accadere che tutte le formule α_1, $\dots$, α_k siano interpretate in falso e quindi la formula $\alpha_1 \vee \dots \vee \alpha_k$ deve essere interpretata in vero. Quindi essa è una tautologia.