Lucido 0: notazioni.

Dott. Ferruccio Guidi

`http://www.cs.unibo.it/~fguidi`

Esercizi sui sistemi di transizione.

Notazione $\rightarrow^*$.

$$e_1 \rightarrow^*_k e_2$$

indica che esiste una computazione da e_1 ad e_2 composta di k transizioni (o "passi").

ovvero esistono e'_1, e'_2, ... , e'_{k-1} tali che

$$\underbrace{e_1 \rightarrow e'_1 \rightarrow e'_2 \rightarrow \ldots \rightarrow e'_{k-1} \rightarrow e_2}_{k \text{ transizioni}}$$

$$e_1 \rightarrow^* e_2$$

indica che esiste $k \in \mathtt{N}$ per cui $e_1 \rightarrow^*_k e_2$.

Qui $\mathtt{N}$ indica l'insieme dei numeri naturali.

Lucido 1: espressioni [1].

Esercizio 1.

Si consideri il sistema di transizione $(e, \rightarrow)$ *dove* "$e ::= n \mid e \cdot e$" *e* "$\rightarrow$" *è definita da:*

$$\frac{}{n_1 \cdot n_2 \rightarrow n} A \qquad \text{se } n = \max(n_1, n_2)$$

$$\frac{e_1 \rightarrow e_2}{e_1 \cdot e \rightarrow e_2 \cdot e} B \qquad \frac{e_1 \rightarrow e_2}{e \cdot e_1 \rightarrow e \cdot e_2} C$$

Indicare le computazioni di $12 \cdot 3 \cdot 4 \cdot 15$ *dimostrandone ogni transizione; dimostrare poi che se* $e \rightarrow^* n$ *allora* $e \cdot e \rightarrow^* n$*.*

Le computazioni sono le seguenti:

$$12 \cdot 3 \cdot 4 \cdot 15 \rightarrow 12 \cdot 4 \cdot 15 \rightarrow 12 \cdot 15 \rightarrow 15$$
$$12 \cdot 3 \cdot 4 \cdot 15 \rightarrow 12 \cdot 3 \cdot 15 \rightarrow 12 \cdot 15 \rightarrow 15$$

e le transizioni si provano con:

$$\frac{\dfrac{}{12 \cdot 3 \rightarrow 12} A}{12 \cdot 3 \cdot 4 \cdot 15 \rightarrow 12 \cdot 4 \cdot 15} B$$

$$\frac{\dfrac{}{4 \cdot 15 \rightarrow 15} A}{12 \cdot 3 \cdot 4 \cdot 15 \rightarrow 12 \cdot 3 \cdot 15} C$$

Lucido 2: espressioni [1].

$$\dfrac{\dfrac{}{12 \cdot 4 \to 12}\,A}{12 \cdot 4 \cdot 15 \to 12 \cdot 15}\,B \qquad \dfrac{\dfrac{}{3 \cdot 15 \to 15}\,A}{12 \cdot 3 \cdot 15 \to 12 \cdot 15}\,C$$

$$\dfrac{}{12 \cdot 15 \to 15}\,A$$

Per dimostrare che $e \to^* n$ implica $e \cdot e \to^* n$ ragioniamo per induzione sulla lunghezza della computazione $e \to^* n$.

Caso base: $e \to^*_0 n$ implica $e \cdot e \to^* n$.

In questo caso $e = n$ e A prova $n \cdot n \to n$.

Caso induttivo: se $e \to^*_k n$ implica $e \cdot e \to^* n$, allora $e' \to^*_{k+1} n$ implica $e' \cdot e' \to^* n$.

Se $e' \to^*_{k+1} n$ allora esiste e tale che $e' \to e$ ed $e \to^*_k n$. Quindi per l'ipotesi induttiva si ha $e \cdot e \to^* n$ e dato che:

$$\dfrac{e' \to e}{e' \cdot e' \to e \cdot e'}\,B \qquad \dfrac{e' \to e}{e_0 \cdot e' \to e \cdot e}\,C$$

si trova $e' \cdot e' \to^* e \cdot e \to^* n$ come voluto.

Lucido 3: notazioni.

Appendice alla notazione $\to^*$.

Per la relazione $\to^*_k$ *si danno le regole*

$$\frac{}{e \to^*_0 e}\,A^* \qquad \frac{e_1 \to e \quad e \to^*_k e_2}{e_1 \to^*_{k+1} e_2}\,B^*$$

$\to^*$ *è transitiva nel seguente senso:*

da una prova di $e_1 \to^* e$ *e da una di* $e \to^* e_2$ *si ricava una prova di* $e_1 \to^* e_2$.

Lucido 4: espressioni [2].

Esercizio 2.

Si consideri il sistema di transizione $(e, \rightarrow)$ *dove "*$e ::= n \mid e \cdot e$*" e "*$\rightarrow$*" è definita da:*

$$\frac{}{n_1 \cdot n_2 \rightarrow n}\,A \qquad \textit{se } n = n_1 - n_2$$

$$\frac{e_1 \rightarrow e_2}{e_1 \cdot e \rightarrow e_2 \cdot e}\,B \qquad \frac{e_1 \rightarrow e_2}{e \cdot e_1 \rightarrow e \cdot e_2}\,C$$

Provare o confutare le transizioni

$$12 \cdot 5 \cdot 7 \cdot 0 \rightarrow 12 \cdot -2 \cdot 0$$
$$3 \cdot 15 \cdot 6 \cdot 11 \rightarrow 3 \cdot 9 \cdot 11$$
$$15 \cdot 15 \cdot 7 \cdot 11 \cdot 6 \rightarrow 0 \cdot 7 \cdot 5$$

e dimostrare la terminazione di ogni computazione di $(e, \rightarrow)$*.*

La prima transizione non può avvenire perché $-2 \notin \mathtt{N}$ e quindi $12 \cdot -2 \cdot 0$ non sta in e;

La seconda transizione si dimostra con

$$\cfrac{\cfrac{\cfrac{}{15 \cdot 6 \rightarrow 9}\,A}{15 \cdot 6 \cdot 11 \rightarrow 9 \cdot 11}\,B}{3 \cdot 15 \cdot 6 \cdot 11 \rightarrow 3 \cdot 9 \cdot 11}\,C$$

La terza transizione non può avvenire perché essa non è conclusione di A, B o C.

Lucido 5: espressioni [2].

Per provare la terminazione definiamo una funzione complessità C da e ad $\mathtt{N}$ tale che $e_1 \to e_2$ implichi $C(e_2) < C(e_1)$.
Si noti l'importanza di avere $<$ e non $\leq$.

Qui scegliamo come $C(e)$ la quantità di numeri presenti in e ovvero poniamo:

$$C(n) \equiv 1; \quad C(e_1 \cdot e_2) \equiv C(e_1) + C(e_2)$$

e ragioniamo per induzione sul numero di regole che compongono la prova di $e_1 \to e_2$.

Caso base: $C(e_2) < C(e_1)$ quando $e_1 \to e_2$ è provato con una regola.

In questo caso la regola deve essere A e quindi $e_1 = n_1 \cdot n_2$ ed $e_2 = n$ per certi n, n_1, n_2 da cui si ricava appunto: $C(e_2) = 1 < 2 = C(e_1)$.

Caso induttivo: se $C(e_2) < C(e_1)$ quando $e_1 \to e_2$ è provato con k regole, allora $C({e_2}') < C({e_1}')$ quando ${e_1}' \to {e_2}'$ è provato con $k + 1$ regole.

Importante: è sbagliatissimo enunciare il caso induttivo con ${e_1}' = e_1$ o ${e_2}' = e_2$.

Lucido 6: espressioni [2-3].

Assumendo che ${e_1}' \to {e_2}'$ sia provato con $k+1$ regole, l'ultima di queste deve essere B o C; nel primo caso si ha ${e_1}' = e_1 \cdot e$ ed ${e_2}' = e_2 \cdot e$ dove $e_1 \to e_2$ è provato con k regole per cui vale $C(e_2) < C(e_1)$; da ciò ricaviamo allora:

$$C({e_2}') = C(e_2) + C(e) < C(e_1) + C(e) = C({e_1}').$$

Il secondo caso è trattato analogamente.

Esercizio 3.

Si consideri il sistema di transizione $(e, \to)$ *dove* "$e ::= n \mid e \cdot e$" *e* "$\to$" *è definita da:*

$$\frac{}{n_1 \cdot n_2 \to n}\,A \qquad \textit{se } n = n_1 \div n_2$$

$$\frac{e_1 \to e_2}{e_1 \cdot e \to e_2 \cdot e}\,B \qquad \frac{e_1 \to e_2}{e \cdot e_1 \to e \cdot e_2}\,C$$

Scrivere tutte le computazioni dello stato $24 \cdot 10 \cdot 5 \cdot 20 \cdot 10$ *dimostrando ogni transizione e dimostrare che* $1 \cdot e \cdot 1 \to^* 1$ *quando* $e \to^* 1$.

Qui $n_1 \div n_2$ è la divisione intera esatta, che è definita quando n_1 è multiplo di $n_2 \neq 0$.

Lucido 7: espressioni [3].

Le transizioni da $24 \cdot 10 \cdot 5 \cdot 20 \cdot 10$ sono:

$$24 \cdot 10 \cdot 5 \cdot 20 \cdot 10 \to 24 \cdot 2 \cdot 20 \cdot 10$$
$$24 \cdot 10 \cdot 5 \cdot 20 \cdot 10 \to 24 \cdot 10 \cdot 5 \cdot 2$$

e si dimostrano con:

$$\cfrac{\cfrac{\cfrac{}{10 \cdot 5 \to 2}\,A}{24 \cdot 10 \cdot 5 \to 24 \cdot 2}\,C}{24 \cdot 10 \cdot 5 \cdot 20 \cdot 10 \to 24 \cdot 2 \cdot 20 \cdot 10}\,B$$

$$\cfrac{\cfrac{}{20 \cdot 10 \to 2}\,A}{24 \cdot 10 \cdot 5 \cdot 20 \cdot 10 \to 24 \cdot 10 \cdot 5 \cdot 2}\,C$$

Le transizioni da $24 \cdot 2 \cdot 20 \cdot 10$ sono:

$$24 \cdot 2 \cdot 20 \cdot 10 \to 12 \cdot 20 \cdot 10$$
$$24 \cdot 2 \cdot 20 \cdot 10 \to 24 \cdot 2 \cdot 2$$

e si dimostrano con:

$$\cfrac{\cfrac{}{24 \cdot 2 \to 12}\,A}{24 \cdot 2 \cdot 20 \cdot 10 \to 12 \cdot 20 \cdot 10}\,B$$

$$\cfrac{\cfrac{}{20 \cdot 10 \to 2}\,A}{24 \cdot 2 \cdot 20 \cdot 10 \to 24 \cdot 2 \cdot 2}\,C$$

Lucido 8: espressioni [3].

La computazione da $12 \cdot 20 \cdot 10$ è:

$$12 \cdot 20 \cdot 10 \to 12 \cdot 2 \to 6$$

$$\dfrac{\dfrac{}{20 \cdot 10 \to 2}\,A}{12 \cdot 20 \cdot 10 \to 12 \cdot 2}\,C \qquad \dfrac{}{12 \cdot 2 \to 6}\,A$$

Le computazioni da $24 \cdot 2 \cdot 2$ sono:

$$24 \cdot 2 \cdot 2 \to 12 \cdot 2 \to 6$$
$$24 \cdot 2 \cdot 2 \to 24 \cdot 1 \to 24$$

$$\dfrac{\dfrac{}{24 \cdot 2 \to 12}\,A}{24 \cdot 2 \cdot 2 \to 12 \cdot 2}\,B \qquad \dfrac{}{12 \cdot 2 \to 6}\,A$$

$$\dfrac{\dfrac{}{2 \cdot 2 \to 1}\,A}{24 \cdot 2 \cdot 2 \to 24 \cdot 1}\,C \qquad \dfrac{}{24 \cdot 1 \to 24}\,A$$

Le transizione da $24 \cdot 10 \cdot 5 \cdot 2$ è:

$$24 \cdot 10 \cdot 5 \cdot 2 \to 24 \cdot 2 \cdot 2$$

$$\dfrac{\dfrac{\dfrac{}{10 \cdot 5 \to 2}\,A}{24 \cdot 10 \cdot 5 \to 24 \cdot 2}\,C}{24 \cdot 10 \cdot 5 \cdot 2 \to 24 \cdot 2 \cdot 2}\,B$$

Lucido 9: espressioni [3-4].

Per dimostrare che $1 \cdot e \cdot 1 \to^* 1$ quando $e \to^* 1$ si può usare, oltre all'induzione sulla struttura di $e \to^* 1$, la seguente tecnica:

$e \to^* 1$ si espande in
$$e \to e_1 \to e_2 \to \ldots \to e_{k-1} \to 1$$
e posto $e_0 \equiv e$, $e_k \equiv 1$ e $0 \leq i < k$ si ha:

$$\cfrac{\cfrac{e_i \to e_{i+1}}{1 \cdot e_i \to 1 \cdot e_{i+1}}\,C}{1 \cdot e_i \cdot 1 \to 1 \cdot e_{i+1} \cdot 1}\,B$$

da cui $1 \cdot e \cdot 1 \to^* 1 \cdot 1 \cdot 1 \to 1 \cdot 1 \to 1$ infatti

$$\cfrac{\cfrac{}{1 \cdot 1 \to 1}\,A}{1 \cdot 1 \cdot 1 \to 1 \cdot 1}\,B \qquad \cfrac{}{1 \cdot 1 \to 1}\,A$$

Esercizio 4.

Si consideri il sistema di transizione $(e, \to)$ *dove "*$e ::= 0 \mid 1 \mid e \cdot e$*" e "*$\to$*" è definita da:*

$$\cfrac{}{1 \cdot 0 \to 0 \cdot 1}\,A$$

$$\cfrac{e_1 \to e_2}{e_1 \cdot e \to e_2 \cdot e}\,B \qquad \cfrac{e_1 \to e_2}{e \cdot e_1 \to e \cdot e_2}\,C$$

Lucido 10: espressioni [4].

Indicare, dimostrando ogni transizione, le computazioni di $1 \cdot 0 \cdot 0 \cdot 1$ *e di* $1 \cdot 0 \cdot 1 \cdot 0 \cdot 1$ *e dimostrare anche che* $1 \cdot 0 \cdot e \cdot 0 \to^* 0 \cdot 1 \cdot e' \cdot 0$ *quando* $e \to^* e'$*.*

La computazione di $1 \cdot 0 \cdot 0 \cdot 1$ è:

$$1 \cdot 0 \cdot 0 \cdot 1 \to 0 \cdot 1 \cdot 0 \cdot 1 \to 0 \cdot 0 \cdot 1 \cdot 1$$

$$\dfrac{\dfrac{}{10 \to 01}\,A}{1001 \to 0101}\,B \qquad \dfrac{\dfrac{\dfrac{}{10 \to 01}\,A}{101 \to 011}\,B}{0101 \to 0011}\,C$$

Le computazioni di $1 \cdot 0 \cdot 1 \cdot 0 \cdot 1$ sono:

$$\begin{array}{c} 10101 \to 01101 \to 01011 \to 00111 \\ 10101 \to 10011 \to 01011 \to 00111 \end{array}$$

$$\dfrac{\dfrac{}{10 \to 01}\,A}{10101 \to 01101}\,B \qquad \dfrac{\dfrac{\dfrac{}{10 \to 01}\,A}{101 \to 011}\,B}{01101 \to 01011}\,C$$

$$\dfrac{\dfrac{\dfrac{}{10 \to 01}\,A}{101 \to 011}\,B}{10101 \to 10011}\,C \qquad \dfrac{\dfrac{}{10 \to 01}\,A}{10011 \to 01011}\,B$$

Lucido 11: espressioni [4].

$$\cfrac{\cfrac{\overline{10 \to 01}\,A}{1011 \to 0111}\,B}{01011 \to 00111}\,C$$

Per dimostrare che
$1 \cdot 0 \cdot e \cdot 0 \to^* 0 \cdot 1 \cdot e' \cdot 0$ quando $e \to^* e'$,
usiamo la tecnica dell'esercizio precedente:

$e \to^* e'$ si espande in
$e \to e_1 \to e_2 \to \ldots \to e_{k-1} \to e'$
e posto $e_0 \equiv e$, $e_k \equiv e'$ e $0 \leq i < k$ si ha:

$$\cfrac{\cfrac{e_i \to e_{i+1}}{1 \cdot 0 \cdot e_i \to 1 \cdot 0 \cdot e_{i+1}}\,C}{1 \cdot 0 \cdot e_i \cdot 0 \to 1 \cdot 0 \cdot e_{i+1} \cdot 0}\,B$$

da cui si ricava

$$1 \cdot 0 \cdot e \cdot 0 \to^* 1 \cdot 0 \cdot e' \cdot 0 \to 0 \cdot 1 \cdot e' \cdot 0$$

$$\cfrac{\overline{1 \cdot 0 \to 0 \cdot 1}\,A}{1 \cdot 0 \cdot e' \cdot 0 \to 0 \cdot 1 \cdot e' \cdot 0}\,B$$

Lucido 12: espressioni [5].

Esercizio 5.

Una macchina $\mathcal{M}$ con un registro a valori interi può eseguire questo insieme di comandi:

$$P ::= \text{add } m \mid \text{wnz } P \mid P \,;\, P$$

dove add m aggiunge l'intero m al valore del registro, wnz P itera il comando P finché il valore del registro è diverso da 0 e termina nell'altro caso, $P_1 \,;\, P_2$ compone sequenzialmente i comandi P_1 e P_2.

Definire un sistema di transizione per i programmi di $\mathcal{M}$; scrivere un programma che non termina e uno che azzera il valore del registro quando questo è negativo; dimostrare la correttezza di entrambi i programmi.

L'insieme Γ delle configurazioni di $\mathcal{M}$ (vedi: l'insieme "e" negli esercizi precedenti) è:

$$(P \times \mathtt{Z}) \cup \mathtt{Z}$$

dove $\mathtt{Z}$ è l'insieme dei numeri interi.

Lucido 13: espressioni [5].

La relazione "$\rightarrow$" è definita dalle regole:

$$\frac{}{<add\ m\ ,n> \rightarrow <n+m>}\ A$$

$$\frac{}{<\text{wnz}\ P\ ,0> \rightarrow <0>}\ B$$

$$\frac{}{<\text{wnz}\ P\ ,n> \rightarrow <P\ ;\ \text{wnz}\ P\ ,n>}\ C\ \ (n \neq 0)$$

$$\frac{<P_1\ ,n> \rightarrow <P_1{}'\ ,n'>}{<P_1\ ;\ P_2\ ,n> \rightarrow <P_1{}'\ ;\ P_2\ ,n'>}\ D$$

$$\frac{<P_1\ ,n> \rightarrow <n'>}{<P_1\ ;\ P_2\ ,n> \rightarrow <P_2\ ,n'>}\ E$$

dimostriamo ora che per $n \neq 0$ il programma:

$$wnz\ add\ 0$$

non termina, infatti si hanno le transizioni:

$$<wnz\ add\ 0\ ,n> \rightarrow <add\ 0\ ;\ wnz\ add\ 0\ ,n>$$

$$<add\ 0\ ;\ wnz\ add\ 0\ ,n> \rightarrow <wnz\ add\ 0\ ,n>$$

La prima deriva direttamente dalla regola C:

$$\frac{}{<wnz\ add\ 0\ ,n> \rightarrow <add\ 0\ ;\ wnz\ add\ 0\ ,n>}\ C$$

Lucido 14: espressioni [5].

L'altra transizione si dimostra con:

$$\cfrac{\overline{<\!add\ 0\,,n\!> \,\to\, <\!n\!>}\ A}{<\!add\ 0\,;\,wnz\ add\ 0\,,n\!> \,\to\, <\!wnz\ add\ 0\,,n\!>}\ E$$

Allora la computazione di $<\!wnz\ add\ 0\,,n\!>$ è ciclica e ciò è sufficiente (ma non necessario) per provare la non terminazione.

Un programma che azzera il valore del registro quando questo è negativo, è invece:

$$wnz\ add\ 1$$

Se $n \leq 0$ è il valore iniziale del registro, la correttezza si dimostra per induzione su $|n|$.

Caso base: $<\!wnz\ add\ 1\,,0\!> \,\to^*\, <\!0\!>$.

deriva direttamente dalla regola B:

$$\frac{}{<\!wnz\ add\ 1\,,0\!> \,\to\, <\!0\!>}\ B$$

Caso induttivo: se $<\!wnz\ add\ 1\,,n\!> \,\to^*\, <\!0\!>$ allora $<\!wnz\ add\ 1\,,n-1\!> \,\to^*\, <\!0\!>$

Si noti che $n \leq 0$ implica $|n| + 1 = |n - 1|$.

Lucido 15: espressioni [5].

Dimostriamo allora:

$<\!\mathit{wnz}\ \mathit{add}\ 1\,, n-1\!> \to^* <\!\mathit{wnz}\ \mathit{add}\ 1\,, n\!>$

e concludiamo invocando la transitività si $\to^*$.

$$\frac{}{\begin{array}{c}<\!\mathit{wnz}\ \mathit{add}\ 1\,, n-1\!> \to \\ <\!\mathit{add}\ 1\,;\ \mathit{wnz}\ \mathit{add}\ 1\,, n-1\!>\end{array}}\ C$$

Si noti che $n \leq 0$ implica $n-1 \neq 0$.

$$\dfrac{\dfrac{}{<\!\mathit{add}\ 1\,, n-1\!> \to <\!n\!>}\ A}{\begin{array}{c}<\!\mathit{add}\ 1\,;\ \mathit{wnz}\ \mathit{add}\ 1\,, n-1\!> \to \\ <\!\mathit{wnz}\ \mathit{add}\ 1\,, n\!>\end{array}}\ E$$

Lucido 16: espressioni [6].

Esercizio 6.

Si consideri il sistema di transizione $(e, \to)$ *dove* ”$e ::= b \mid e \cdot e$”, $b \in \mathtt{B} = \{0, 1\}$ *e* ”$\to$” *è definita da:*

$$\frac{}{b \cdot b \to b} A$$

$$\frac{e_1 \to e_2}{e_1 \cdot b \to e_2 \cdot b} B \qquad \frac{e_1 \to e_2}{b \cdot e_1 \to b \cdot e_2} C$$

Indicare le computazioni di $1 \cdot 1 \cdot 0 \cdot 0 \cdot 0$ *dimostrandone ogni transizione; dimostrare poi che se* $e \to^* 0 \cdot 1$ *allora* $0 \cdot e \cdot 1 \to^* 0 \cdot 1$.

Le computazioni di 11000 sono le seguenti:

$$11000 \to 1000 \to 100 \to 10$$
$$11000 \to 1100 \to 100 \to 10$$
$$11000 \to 1100 \to 110 \to 10$$

e le transizioni si provano con:

$$\cfrac{\cfrac{\cfrac{\cfrac{}{11 \to 1} A}{110 \to 10} B}{1100 \to 100} B}{11000 \to 1000} B \qquad \cfrac{\cfrac{\cfrac{\cfrac{}{00 \to 0} A}{000 \to 00} C}{1000 \to 100} C}{11000 \to 1100} C$$

Lucido 17: espressioni [6].

$$\dfrac{\dfrac{\dfrac{}{00 \to 0}A}{000 \to 00}C}{1000 \to 100}C \qquad \dfrac{\dfrac{\dfrac{}{11 \to 1}A}{110 \to 10}B}{1100 \to 100}B \qquad \dfrac{\dfrac{\dfrac{}{00 \to 0}A}{100 \to 10}C}{1100 \to 110}C$$

$$\dfrac{\dfrac{}{00 \to 0}A}{100 \to 10}C \qquad \dfrac{\dfrac{}{11 \to 1}A}{110 \to 10}B$$

Vediamo che se $e \to^* 0 \cdot 1$ allora $0 \cdot e \cdot 1 \to^* 0 \cdot 1$ con la tecnica della volta scorsa:

$e \to^* 0 \cdot 1$ si espande in
$e \to e_1 \to e_2 \to \ldots \to e_{k-1} \to 0 \cdot 1$
e posto $e_0 \equiv e$, $e_k \equiv 0 \cdot 1$ e $0 \leq i < k$ si ha:

$$\dfrac{\dfrac{e_i \to e_{i+1}}{e_i \cdot 1 \to e_{i+1} \cdot 1}B}{0 \cdot e_i \cdot 1 \to 0 \cdot e_{i+1} \cdot 1}C$$

da cui $0 \cdot e \cdot 1 \to^* 0 \cdot 0 \cdot 1 \cdot 1 \to^* 0 \cdot 1$ infatti

$$\dfrac{\dfrac{\dfrac{}{0 \cdot 0 \to 0}A}{0 \cdot 0 \cdot 1 \to 0 \cdot 1}B}{0 \cdot 0 \cdot 1 \cdot 1 \to 0 \cdot 1 \cdot 1}B \qquad \dfrac{\dfrac{}{1 \cdot 1 \to 1}A}{0 \cdot 1 \cdot 1 \to 0 \cdot 1}C$$

Lucido 18: espressioni [7].

Esercizio 7.

Una macchina $\mathcal{M}$ con un registro fatto di un byte può eseguire questo insieme di comandi:

$$P ::= \mathit{reset} \mid \mathit{inc} \mid \mathit{lshift} \mid \mathit{wnz}\ P \mid P\,;P$$

dove reset azzera il registro, inc ne aumenta il valore di 1, *lshift sposta a sinistra di una posizione i bit del registro inserendo un bit* 0 *nella posizione meno significativa, wnz e ; sono quelli dell'esercizio 5.*

Definire un sistema di transizione per i programmi di $\mathcal{M}$; scrivere un programma che non termina e dimostrarne la correttezza; dimostrare poi che il programma:

wnz inc

termina per ogni valore iniziale del registro.

Visto che un byte può assumere solo valori interi compresi fra 0 e 255, l'insieme Γ delle configurazioni di $\mathcal{M}$ è contenuto in:

$$(P \times \mathtt{N}) \cup \mathtt{N}$$

Lucido 19: espressioni [7].

La relazione "$\to$" è definita dalle regole:

$$\frac{}{<\mathit{reset}\,,n> \to <0>}\,A$$

$$\frac{}{<\mathit{inc}\,,n> \to <(n+1) \bmod 256>}\,B$$

$$\frac{}{<\mathit{lshift}\,,n> \to <(2n) \bmod 256>}\,C$$

$$\frac{}{<\mathit{wnz}\ P\,,0> \to <0>}\,D$$

$$\frac{}{<\mathit{wnz}\ P\,,n> \to <P\,;\,\mathit{wnz}\ P\,,n>}\,E\ (n>0)$$

$$\frac{<P_1\,,n> \to <P_1{}'\,,n'>}{<P_1\,;\,P_2\,,n> \to <P_1{}'\,;\,P_2\,,n'>}\,F$$

$$\frac{<P_1\,,n> \to <n'>}{<P_1\,;\,P_2\,,n> \to <P_2\,,n'>}\,G$$

con la condizione generale: $0 \leq n, n' \leq 255$.

dimostriamo ora che per $n = 1$ il programma:

$$\mathit{wnz}\ (\mathit{reset}\,;\,\mathit{inc})$$

non termina, infatti $<\mathit{wnz}\ (\mathit{reset}\,;\,\mathit{inc})\,,1>$ transisce in $<(\mathit{reset}\,;\,\mathit{inc})\,;\,\mathit{wnz}\ (\mathit{reset}\,;\,\mathit{inc})\,,1>$ per via della regola E.

Lucido 20: espressioni [7].

ora $<\mathit{reset}\,;\mathit{inc}\,;\mathit{wnz}\ (\mathit{reset}\,;\mathit{inc})\,,1>$ transisce in $<\mathit{inc}\,;\mathit{wnz}\ (\mathit{reset}\,;\mathit{inc})\,,0>$ infatti

$$\cfrac{\cfrac{\cfrac{}{<\mathit{reset}\,,1> \to <0>}\,A}{<\mathit{reset}\,;\mathit{inc}\,,1> \to <\mathit{inc}\,,0>}\,G}{\begin{array}{c}<\mathit{reset}\,;\mathit{inc}\,;\mathit{wnz}\ (\mathit{reset}\,;\mathit{inc})\,,1> \to \\ <\mathit{inc}\,;\mathit{wnz}\ (\mathit{reset}\,;\mathit{inc})\,,0>\end{array}}\,F$$

in fine $<\mathit{inc}\,;\mathit{wnz}\ (\mathit{reset}\,;\mathit{inc})\,,0>$ transisce in $<\mathit{wnz}\ (\mathit{reset}\,;\mathit{inc})\,,1>$ infatti

$$\cfrac{\cfrac{}{<\mathit{inc}\,,0> \to <1>}\,B}{\begin{array}{c}<\mathit{inc}\,;\mathit{wnz}\ (\mathit{reset}\,;\mathit{inc})\,,0> \to \\ <\mathit{wnz}\ (\mathit{reset}\,;\mathit{inc})\,,1>\end{array}}\,G$$

Così la computazione di $<\mathit{wnz}\ (\mathit{reset}\,;\mathit{inc})\,,1>$ è ciclica e ciò è sufficiente (ma non necessario) per provare la non terminazione.

Dimostriamo ora che $<\mathit{wnz}\ \mathit{inc}\,,n> \to^* <0>$ per ogni $0 \leq n \leq 255$ e distinguiamo due casi:

Se $n = 0$ allora l'asserto vale per la regola D;

Lucido 21: espressioni [7].

Se $n > 0$ allora dobbiamo trovare una quantità che decresce ad ogni passo della computazione in modo da dimostrarne la terminazione. Considerato che *wnz inc* aumenta il valore del registro ad ogni ciclo, questa quantità è

$$255 - n$$

Usiamo allora l'induzione su questa quantità:

Caso base: $255 - n = 0$ implica $n = 255$ bisogna provare $<\mathit{wnz\ inc}\,, 255> \to^* <0>$

Infatti, essendo $255 > 0$, la regola E dà $<\mathit{wnz\ inc}\,, 255> \to <\mathit{inc}\,;\, \mathit{wnz\ inc}\,, 255>$ e poi la computazione continua con $<\mathit{inc}\,;\mathit{wnz\ inc}\,, 255> \to <\mathit{wnz\ inc}\,, 0> \to <0>$. Il secondo passo segue dalla regola D mentre il primo è dimostrato con

$$\dfrac{\dfrac{}{<\mathit{inc}\,, 255> \to <0>}\,B}{<\mathit{inc}\,;\, \mathit{wnz\ inc}\,, 255> \to <\mathit{wnz\ inc}\,, 0>}\,G$$

Si noti che

$$(255 + 1) \bmod 256 = (256 \bmod 256) = 0$$

Lucido 22: espressioni [7-8].

Caso induttivo: se $<\!wnz\ inc\,,n\!> \to^* <\!0\!>$ allora $<\!wnz\ inc\,,n-1\!> \to^* <\!0\!>$.

Si noti $(255-n)+1 = 255-(n-1)$.

Basta dimostrare

$$<\!wnz\ inc\,,n-1\!> \to^* <\!wnz\ inc\,,n\!>$$

e usare la transitività di $\to^*$. La regola E dà: $<\!wnz\ inc\,,n-1\!> \to^* <\!inc\,;\, wnz\ inc\,,n-1\!>$. Ora $255-(n-1)>0$ implica $n-1<255$ e quindi è corretto applicare B come segue:

$$\dfrac{\dfrac{}{<\!inc\,,n-1\!> \to <\!n\!>}\,B}{<\!inc\,;\, wnz\ inc\,,n-1\!> \to <\!wnz\ inc\,,n\!>}\,G$$

Esercizio 8.

Una macchina $\mathcal{M}$ con un registro a valori in $\mathtt{N}$ *può eseguire questo insieme di comandi:*

$$P ::= set\ m \mid add\ m \mid sub\ m \mid P\,;\,P$$

dove set m pone m nel registro, add m ne aumenta il valore di m, sub m ne diminuisce il valore di m, ; è quello di prima.

Lucido 23: espressioni [8].

Definire un sistema di transizione per i programmi di $\mathcal{M}$; definire una relazione di equivalenza fra programmi e discutere l'equivalenza delle seguenti coppie di comandi:

$P_1 = \textit{sub}\ 2\,; \text{add}\ 2$ *e* $P_1{}' = \text{add}\ 0$

$P_2 = \textit{sub}\ 2\,; \text{add}\ 2$ *e* $P_2{}' = \textit{sub}\ 1\,; \text{add}\ 1$

$P_3 = \text{add}\ 2$ *e* $P_3{}' = \text{add}\ 2\,; \textit{sub}\ 1\,; \text{add}\ 1$

Dimostrare poi che ogni comando termina.

Le configurazioni stanno in $\Gamma \equiv (P \times \mathtt{N}) \cup \mathtt{N}$.
La relazione "$\rightarrow$" è definita dalle regole:

$$\frac{}{<\textit{set}\ m\,,n> \,\rightarrow\, <m>}\ A$$

$$\frac{}{<\text{add}\ m\,,n> \,\rightarrow\, <n+m>}\ B$$

$$\frac{}{<\textit{sub}\ m\,,n> \,\rightarrow\, <n-m>}\ C\ (m \leq n)$$

$$\frac{}{<\textit{sub}\ m\,,n> \,\rightarrow\, <0>}\ D\ (m > n)$$

$$\frac{<P_1\,,n> \,\rightarrow\, <P_1{}'\,,n'>}{<P_1\,;P_2\,,n> \,\rightarrow\, <P_1{}'\,;P_2\,,n'>}\ E$$

$$\frac{<P_1\,,n> \,\rightarrow\, <n'>}{<P_1\,;P_2\,,n> \,\rightarrow\, <P_2\,,n'>}\ F$$

Lucido 24: espressioni [8].

Due programmi sono equivalenti quando, partendo dalla stessa configurazione di memoria, producono lo stesso risultato, ovvero $P \equiv P'$ quando per ogni $n \in \mathtt{N}$

$$<P\,,n> \to^* <n'> \text{ sse } <P'\,,n> \to^* <n'>$$

Si noti che non occorre quantificare su n'.

I programmi P_1 e $P_1{}'$ non sono equivalenti perché $<P_1\,,0> \to^* <2>$ e $<P_1{}'\,,0> \to^* <0>$.

La prima computazione si fa in due passi:

$$\dfrac{\dfrac{}{<sub\ 2\,,0> \to <0>}\ D\ (2>0)}{<sub\ 2\,;\, add\ 2\,,0> \to <add\ 2\,,0>}\ F$$

$$\dfrac{}{<add\ 2\,,0> \to <2>}\ B$$

La seconda computazione si fa in un passo:

$$\dfrac{}{<add\ 0\,,0> \to <0>}\ B$$

I programmi P_2 e $P_2{}'$ non sono equivalenti perché $<P_2\,,0> \to^* <2>$ e $<P_2{}'\,,0> \to^* <1>$.

La prima computazione ci è già nota perché $P_2 = P_1$.

Lucido 25: espressioni [8].

La seconda computazione si fa in due passi:

$$\dfrac{\dfrac{}{<\mathit{sub}\ 1\,,0> \;\to\; <0>}\; D\ (1 > 0)}{<\mathit{sub}\ 1\,;\mathit{add}\ 1\,,0> \;\to\; <\mathit{add}\ 1\,,0>}\; F$$

$$\dfrac{}{<\mathit{add}\ 1\,,0> \;\to\; <1>}\; B$$

I programmi P_3 e $P_3{}'$ sono equivalenti infatti la regola B dà $<\mathit{add}\ 2\,,n> \;\to\; <n+2>$ e si ha anche $<\mathit{add}\ 2\,;\mathit{sub}\ 1\,;\mathit{add}\ 1\,,n> \;\to^{*}\; <n+2>$. Questa computazione si fa in tre passi:

$$\dfrac{\dfrac{\dfrac{}{<\mathit{add}\ 2\,,n> \;\to\; <n+2>}\; B}{<\mathit{add}\ 2\,;\mathit{sub}\ 1\,,n> \;\to\; <\mathit{sub}\ 1\,,n+2>}\; F}{\begin{array}{c}<\mathit{add}\ 2\,;\mathit{sub}\ 1\,;\mathit{add}\ 1\,,n> \;\to\\ <\mathit{sub}\ 1\,;\mathit{add}\ 1\,,n+2>\end{array}}\; E$$

$$\dfrac{\dfrac{}{<\mathit{sub}\ 1\,,n+2> \;\to\; <n+1>}\; C}{<\mathit{sub}\ 1\,;\mathit{add}\ 1\,,n+2> \;\to\; <\mathit{add}\ 1\,,n+1>}\; F$$

$$\dfrac{}{<\mathit{add}\ 1\,,n+1> \;\to\; <n+2>}\; B$$

La regola C si applica perché $1 \leq n+2$. L'equivalenza si ha perché n è arbitrario.

Lucido 26: espressioni [8].

Dimostriamo ora la terminazione di ogni programma di $\mathcal{M}$ facendo vedere che ogni computazione di $\mathcal{M}$ è finita.

Per questo definiamo una funzione C da Γ ad $\mathtt{N}$ tale che se $\gamma_1, \gamma_2 \in \Gamma$ e se $\gamma_1 \to \gamma_2$ allora

$$C(\gamma_2) < C(\gamma_1)$$

Intuitivamente $C(<P\,,n>)$ è il numero di comandi semplici che compongono P ovvero:

$$C(<n>) = 0 \quad C(<set\ m\,,n>) = 1$$

$$C(<add\ m\,,n>) = 1 \quad C(<sub\ m\,,n>) = 1$$

$$C(<P_1\,;P_2\,,n>) = C(<P_1\,,n>) + C(<P_2\,,n>)$$

$C(\gamma_2) < C(\gamma_1)$ si dimostra per induzione sul numero di regole della prova di $\gamma_1 \to \gamma_2$.

Caso base: $\gamma_1 \to \gamma_2$ è provato con una regola.

In questo caso la regola non deve avere premesse e quindi deve essere A, B, C o D. Allora la regola ha la forma

$$\frac{}{<P\,,n> \to <n'>}$$

e si trova $C(<n'>) = 0 < 1 = C(<P\,,n>)$.

Lucido 27: espressioni [8].

Caso induttivo: se $C(\gamma_2) < C(\gamma_1)$ quando $\gamma_1 \to \gamma_2$ è provato con $k \geq 1$ regole allora $C({\gamma_2}') < C({\gamma_1}')$ quando ${\gamma_1}' \to {\gamma_2}'$ è provato con $k+1$ regole.

In questo caso ${\gamma_1}' \to {\gamma_2}'$ è provato con almeno due regole e quindi l'ultima regola della prova deve avere premesse per cui deve essere E oppure F che entrambe sono della forma

$$\frac{\gamma_1 \to \gamma_2}{{\gamma_1}' \to {\gamma_2}'}$$

dove $\gamma_1 \to \gamma_2$ è provato con k regole (tutte quelle che provano ${\gamma_1}' \to {\gamma_2}'$ tranne l'ultima) sicché l'ipotesi induttiva dà $C(\gamma_2) < C(\gamma_1)$.

Se l'ultima regola è E allora si ha:
$C(<{P_1}', n'>) < C(<P_1, n>)$ e quindi

$$C(<{P_1}' ; P_2, n'>) = C(<{P_1}', n'>) +$$
$$+ C(<P_2, n'>) < C(<P_1, n>) +$$
$$+ C(<P_2, n>) = C(<P_1 ; P_2, n>)$$

Si noti che $C(<P_2, n>) = C(<P_2, n'>)$.

Il caso della regola F si tratta analogamente.