

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 $e'_1, e'_2, \dots, e'_{k-1}$ tali che

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

$$e_1 \rightarrow^* e_2$$

indica che esiste $k \in \mathbb{N}$ per cui $e_1 \rightarrow_k^ e_2$.*

Qui \mathbb{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:

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

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

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:

$$\frac{}{12 \cdot 3 \rightarrow 12} A$$

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

$$\frac{}{4 \cdot 15 \rightarrow 15} A$$

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

Lucido 2: espressioni [1].

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

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

Caso base: $e \rightarrow_0^* n$ implica $e \cdot e \rightarrow^* n$.

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

Caso induttivo: se $e \rightarrow_k^* n$ implica $e \cdot e \rightarrow^* n$, allora $e' \rightarrow_{k+1}^* n$ implica $e' \cdot e' \rightarrow^* n$.

Se $e' \rightarrow_{k+1}^* n$ allora esiste e tale che $e' \rightarrow e$ ed $e \rightarrow_k^* n$. Quindi per l'ipotesi induttiva si ha

$e \cdot e \rightarrow^* n$ e dato che:

$$\frac{e' \rightarrow e}{e' \cdot e' \rightarrow e \cdot e'} B \quad \frac{e' \rightarrow e}{e_0 \cdot e' \rightarrow e \cdot e} C$$

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

Lucido 3: notazioni.

Appendice alla notazione \rightarrow^* .

Per la relazione \rightarrow_k^* si danno le regole

$$\frac{}{e \rightarrow_0^* e} A^* \quad \frac{e_1 \rightarrow e \quad e \rightarrow_k^* e_2}{e_1 \rightarrow_{k+1}^* e_2} B^*$$

\rightarrow^* è transitiva nel seguente senso:

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

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:

$$\frac{}{n_1 \cdot n_2 \rightarrow n} A \quad \text{se } n = n_1 - n_2$$
$$\frac{e_1 \rightarrow e_2}{e_1 \cdot e \rightarrow e_2 \cdot e} B \quad \frac{e_1 \rightarrow e_2}{e \cdot e_1 \rightarrow e \cdot e_2} C$$

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 $-2 \notin \mathbb{N}$ e quindi $12 \cdot -2 \cdot 0$ non sta in e ;

La seconda transizione si dimostra con

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

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 \mathbb{N} tale che

$$e_1 \rightarrow e_2 \text{ implichi } C(e_2) < C(e_1).$$

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 $e_1 \rightarrow e_2$.

Caso base: $C(e_2) < C(e_1)$ quando $e_1 \rightarrow e_2$ 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 $e_1 \rightarrow e_2$ è provato con k regole, allora $C(e_2') < C(e_1')$ quando $e_1' \rightarrow e_2'$ 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 $e_1' \rightarrow e_2'$ 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 $e_1 \rightarrow e_2$ 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, \rightarrow) dove " $e ::= n \mid e \cdot e$ " e " \rightarrow " è definita da:

$$\frac{}{n_1 \cdot n_2 \rightarrow n} A \quad \text{se } n = n_1 \div n_2$$
$$\frac{e_1 \rightarrow e_2}{e_1 \cdot e \rightarrow e_2 \cdot e} B \quad \frac{e_1 \rightarrow e_2}{e \cdot e_1 \rightarrow e \cdot e_2} C$$

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

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:

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

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

e si dimostrano con:

$$\frac{10 \cdot 5 \rightarrow 2}{24 \cdot 10 \cdot 5 \rightarrow 24 \cdot 2} \begin{matrix} A \\ C \\ B \end{matrix}$$
$$24 \cdot 10 \cdot 5 \cdot 20 \cdot 10 \rightarrow 24 \cdot 2 \cdot 20 \cdot 10$$

$$\frac{20 \cdot 10 \rightarrow 2}{24 \cdot 10 \cdot 5 \cdot 20 \cdot 10 \rightarrow 24 \cdot 10 \cdot 5 \cdot 2} \begin{matrix} A \\ C \end{matrix}$$

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

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

$$24 \cdot 2 \cdot 20 \cdot 10 \rightarrow 24 \cdot 2 \cdot 2$$

e si dimostrano con:

$$\frac{24 \cdot 2 \rightarrow 12}{24 \cdot 2 \cdot 20 \cdot 10 \rightarrow 12 \cdot 20 \cdot 10} \begin{matrix} A \\ B \end{matrix}$$

$$\frac{20 \cdot 10 \rightarrow 2}{24 \cdot 2 \cdot 20 \cdot 10 \rightarrow 24 \cdot 2 \cdot 2} \begin{matrix} A \\ C \end{matrix}$$

Lucido 8: espressioni [3].

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

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

$$\frac{\frac{20 \cdot 10 \rightarrow 2}{A}}{12 \cdot 20 \cdot 10 \rightarrow 12 \cdot 2} C \quad \frac{12 \cdot 2 \rightarrow 6}{A}$$

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

$$\begin{aligned} 24 \cdot 2 \cdot 2 &\rightarrow 12 \cdot 2 \rightarrow 6 \\ 24 \cdot 2 \cdot 2 &\rightarrow 24 \cdot 1 \rightarrow 24 \end{aligned}$$

$$\frac{\frac{24 \cdot 2 \rightarrow 12}{A}}{24 \cdot 2 \cdot 2 \rightarrow 12 \cdot 2} B \quad \frac{12 \cdot 2 \rightarrow 6}{A}$$

$$\frac{\frac{2 \cdot 2 \rightarrow 1}{A}}{24 \cdot 2 \cdot 2 \rightarrow 24 \cdot 1} C \quad \frac{24 \cdot 1 \rightarrow 24}{A}$$

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

$$24 \cdot 10 \cdot 5 \cdot 2 \rightarrow 24 \cdot 2 \cdot 2$$

$$\frac{\frac{10 \cdot 5 \rightarrow 2}{A}}{\frac{24 \cdot 10 \cdot 5 \rightarrow 24 \cdot 2}{C}} B$$
$$24 \cdot 10 \cdot 5 \cdot 2 \rightarrow 24 \cdot 2 \cdot 2$$

Lucido 9: espressioni [3-4].

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

$e \rightarrow^* 1$ si espande in

$$e \rightarrow e_1 \rightarrow e_2 \rightarrow \dots \rightarrow e_{k-1} \rightarrow 1$$

e posto $e_0 \equiv e$, $e_k \equiv 1$ e $0 \leq i < k$ si ha:

$$\frac{e_i \rightarrow e_{i+1}}{1 \cdot e_i \rightarrow 1 \cdot e_{i+1}} C$$

$$\frac{1 \cdot e_i \rightarrow 1 \cdot e_{i+1}}{1 \cdot e_i \cdot 1 \rightarrow 1 \cdot e_{i+1} \cdot 1} B$$

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

$$\frac{\frac{1 \cdot 1 \rightarrow 1}{1 \cdot 1 \cdot 1 \rightarrow 1 \cdot 1} A}{1 \cdot 1 \rightarrow 1} B$$

$$\frac{1 \cdot 1 \rightarrow 1}{1 \cdot 1 \rightarrow 1} A$$

Esercizio 4.

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

$$\frac{}{1 \cdot 0 \rightarrow 0 \cdot 1} A$$

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

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

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 $1 \cdot 0 \cdot e \cdot 0 \rightarrow^* 0 \cdot 1 \cdot e' \cdot 0$ quando $e \rightarrow^* e'$.

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

$$1 \cdot 0 \cdot 0 \cdot 1 \rightarrow 0 \cdot 1 \cdot 0 \cdot 1 \rightarrow 0 \cdot 0 \cdot 1 \cdot 1$$

$$\begin{array}{l} \frac{}{10 \rightarrow 01} A \\ \frac{}{1001 \rightarrow 0101} B \end{array} \quad \begin{array}{l} \frac{}{10 \rightarrow 01} A \\ \frac{}{101 \rightarrow 011} B \\ \frac{}{0101 \rightarrow 0011} C \end{array}$$

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

$$\begin{array}{l} 10101 \rightarrow 01101 \rightarrow 01011 \rightarrow 00111 \\ 10101 \rightarrow 10011 \rightarrow 01011 \rightarrow 00111 \end{array}$$

$$\begin{array}{l} \frac{}{10 \rightarrow 01} A \\ \frac{}{10101 \rightarrow 01101} B \end{array} \quad \begin{array}{l} \frac{}{10 \rightarrow 01} A \\ \frac{}{101 \rightarrow 011} B \\ \frac{}{01101 \rightarrow 01011} C \end{array}$$

$$\begin{array}{l} \frac{}{10 \rightarrow 01} A \\ \frac{}{101 \rightarrow 011} B \\ \frac{}{10101 \rightarrow 10011} C \end{array} \quad \begin{array}{l} \frac{}{10 \rightarrow 01} A \\ \frac{}{10011 \rightarrow 01011} B \end{array}$$

Lucido 11: espressioni [4].

$$\frac{\frac{\frac{10 \rightarrow 01}{A}}{B}}{C}$$

Per dimostrare che

$1 \cdot 0 \cdot e \cdot 0 \rightarrow^* 0 \cdot 1 \cdot e' \cdot 0$ quando $e \rightarrow^* e'$,
usiamo la tecnica dell'esercizio precedente:

$e \rightarrow^* e'$ si espande in

$$e \rightarrow e_1 \rightarrow e_2 \rightarrow \dots \rightarrow e_{k-1} \rightarrow e'$$

e posto $e_0 \equiv e$, $e_k \equiv e'$ e $0 \leq i < k$ si ha:

$$\frac{\frac{\frac{e_i \rightarrow e_{i+1}}{C}}{B}}{1 \cdot 0 \cdot e_i \cdot 0 \rightarrow 1 \cdot 0 \cdot e_{i+1} \cdot 0}$$

da cui si ricava

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

$$\frac{\frac{1 \cdot 0 \rightarrow 0 \cdot 1}{A}}{B}$$

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 $\text{add } m$ aggiunge l'intero m al valore del registro, $\text{wnz } P$ 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:

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

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

Lucido 13: espressioni [5].

La relazione "→" è definita dalle regole:

$$\frac{}{\langle \text{add } m, n \rangle \rightarrow \langle n + m \rangle} A$$

$$\frac{}{\langle \text{wnz } P, 0 \rangle \rightarrow \langle 0 \rangle} B$$

$$\frac{}{\langle \text{wnz } P, n \rangle \rightarrow \langle P ; \text{wnz } P, n \rangle} C \quad (n \neq 0)$$

$$\frac{\langle P_1, n \rangle \rightarrow \langle P_1', n' \rangle}{\langle P_1 ; P_2, n \rangle \rightarrow \langle P_1' ; P_2, n' \rangle} D$$

$$\frac{\langle P_1, n \rangle \rightarrow \langle n' \rangle}{\langle P_1 ; P_2, n \rangle \rightarrow \langle P_2, n' \rangle} E$$

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

wnz add 0

non termina, infatti si hanno le transizioni:

$$\langle \text{wnz add } 0, n \rangle \rightarrow \langle \text{add } 0 ; \text{wnz add } 0, n \rangle$$

$$\langle \text{add } 0 ; \text{wnz add } 0, n \rangle \rightarrow \langle \text{wnz add } 0, n \rangle$$

La prima deriva direttamente dalla regola C:

$$\frac{}{\langle \text{wnz add } 0, n \rangle \rightarrow \langle \text{add } 0 ; \text{wnz add } 0, n \rangle} C$$

Lucido 14: espressioni [5].

L'altra transizione si dimostra con:

$$\frac{\langle \text{add } 0, n \rangle \rightarrow \langle n \rangle}{\langle \text{add } 0 ; \text{wnz add } 0, n \rangle \rightarrow \langle \text{wnz add } 0, n \rangle} \begin{array}{l} A \\ E \end{array}$$

Allora la computazione di $\langle \text{wnz add } 0, n \rangle$ 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: $\langle \text{wnz add } 1, 0 \rangle \rightarrow^* \langle 0 \rangle$.

deriva direttamente dalla regola *B*:

$$\frac{}{\langle \text{wnz add } 1, 0 \rangle \rightarrow \langle 0 \rangle} B$$

Caso induttivo: se $\langle \text{wnz add } 1, n \rangle \rightarrow^* \langle 0 \rangle$
allora $\langle \text{wnz add } 1, n - 1 \rangle \rightarrow^* \langle 0 \rangle$

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

Lucido 15: espressioni [5].

Dimostriamo allora:

$$\langle \text{wnz add } 1, n - 1 \rangle \rightarrow^* \langle \text{wnz add } 1, n \rangle$$

e concludiamo invocando la transitività di \rightarrow^* .

$$\frac{\langle \text{wnz add } 1, n - 1 \rangle \rightarrow}{\langle \text{add } 1; \text{wnz add } 1, n - 1 \rangle} C$$

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

$$\frac{\frac{\langle \text{add } 1, n - 1 \rangle \rightarrow \langle n \rangle}{\langle \text{add } 1; \text{wnz add } 1, n - 1 \rangle \rightarrow} A}{\langle \text{wnz add } 1, n \rangle} E$$

Lucido 16: espressioni [6].

Esercizio 6.

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

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

$$\frac{e_1 \rightarrow e_2}{e_1 \cdot b \rightarrow e_2 \cdot b} B \quad \frac{e_1 \rightarrow e_2}{b \cdot e_1 \rightarrow b \cdot e_2} C$$

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

Le computazioni di 11000 sono le seguenti:

$$11000 \rightarrow 1000 \rightarrow 100 \rightarrow 10$$

$$11000 \rightarrow 1100 \rightarrow 100 \rightarrow 10$$

$$11000 \rightarrow 1100 \rightarrow 110 \rightarrow 10$$

e le transizioni si provano con:

$11 \rightarrow 1$	A	$00 \rightarrow 0$	A
$110 \rightarrow 10$	B	$000 \rightarrow 00$	C
$1100 \rightarrow 100$	B	$1000 \rightarrow 100$	C
$11000 \rightarrow 1000$	B	$11000 \rightarrow 1100$	C

Lucido 17: espressioni [6].

$$\begin{array}{l}
 \hline 00 \rightarrow 0 \\
 \hline 000 \rightarrow 00 \\
 \hline 1000 \rightarrow 100
 \end{array}
 \begin{array}{l}
 A \\
 C \\
 C
 \end{array}
 \quad
 \begin{array}{l}
 \hline 11 \rightarrow 1 \\
 \hline 110 \rightarrow 10 \\
 \hline 1100 \rightarrow 100
 \end{array}
 \begin{array}{l}
 A \\
 B \\
 B
 \end{array}
 \quad
 \begin{array}{l}
 \hline 00 \rightarrow 0 \\
 \hline 100 \rightarrow 10 \\
 \hline 1100 \rightarrow 110
 \end{array}
 \begin{array}{l}
 A \\
 C \\
 C
 \end{array}$$

$$\begin{array}{l}
 \hline 00 \rightarrow 0 \\
 \hline 100 \rightarrow 10
 \end{array}
 \begin{array}{l}
 A \\
 C
 \end{array}
 \quad
 \begin{array}{l}
 \hline 11 \rightarrow 1 \\
 \hline 110 \rightarrow 10
 \end{array}
 \begin{array}{l}
 A \\
 B
 \end{array}$$

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

$e \rightarrow^* 0 \cdot 1$ si espande in

$$e \rightarrow e_1 \rightarrow e_2 \rightarrow \dots \rightarrow e_{k-1} \rightarrow 0 \cdot 1$$

e posto $e_0 \equiv e$, $e_k \equiv 0 \cdot 1$ e $0 \leq i < k$ si ha:

$$\begin{array}{l}
 e_i \rightarrow e_{i+1} \\
 \hline e_i \cdot 1 \rightarrow e_{i+1} \cdot 1 \\
 \hline 0 \cdot e_i \cdot 1 \rightarrow 0 \cdot e_{i+1} \cdot 1
 \end{array}
 \begin{array}{l}
 B \\
 C \\
 C
 \end{array}$$

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

$$\begin{array}{l}
 \hline 0 \cdot 0 \rightarrow 0 \\
 \hline 0 \cdot 0 \cdot 1 \rightarrow 0 \cdot 1 \\
 \hline 0 \cdot 0 \cdot 1 \cdot 1 \rightarrow 0 \cdot 1 \cdot 1
 \end{array}
 \begin{array}{l}
 A \\
 B \\
 B
 \end{array}
 \quad
 \begin{array}{l}
 \hline 1 \cdot 1 \rightarrow 1 \\
 \hline 0 \cdot 1 \cdot 1 \rightarrow 0 \cdot 1
 \end{array}
 \begin{array}{l}
 A \\
 C
 \end{array}$$

Lucido 18: espressioni [7].

Esercizio 7.

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

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

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:

$$\text{wnz inc}$$

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:

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

Lucido 19: espressioni [7].

La relazione " \rightarrow " è definita dalle regole:

$$\frac{}{\langle \text{reset}, n \rangle \rightarrow \langle 0 \rangle} A$$

$$\frac{}{\langle \text{inc}, n \rangle \rightarrow \langle (n + 1) \bmod 256 \rangle} B$$

$$\frac{}{\langle \text{lshift}, n \rangle \rightarrow \langle (2n) \bmod 256 \rangle} C$$

$$\frac{}{\langle \text{wnz } P, 0 \rangle \rightarrow \langle 0 \rangle} D$$

$$\frac{}{\langle \text{wnz } P, n \rangle \rightarrow \langle P ; \text{wnz } P, n \rangle} E \quad (n > 0)$$

$$\frac{\langle P_1, n \rangle \rightarrow \langle P_1', n' \rangle}{\langle P_1 ; P_2, n \rangle \rightarrow \langle P_1' ; P_2, n' \rangle} F$$

$$\frac{\langle P_1, n \rangle \rightarrow \langle n' \rangle}{\langle P_1 ; P_2, n \rangle \rightarrow \langle P_2, n' \rangle} G$$

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

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

$\text{wnz } (\text{reset} ; \text{inc})$

non termina, infatti $\langle \text{wnz } (\text{reset} ; \text{inc}), 1 \rangle$
transisce in $\langle (\text{reset} ; \text{inc}) ; \text{wnz } (\text{reset} ; \text{inc}), 1 \rangle$
per via della regola E.

Lucido 20: espressioni [7].

ora $\langle \text{reset} ; \text{inc} ; \text{wnz} (\text{reset} ; \text{inc}) , 1 \rangle$ transisce
in $\langle \text{inc} ; \text{wnz} (\text{reset} ; \text{inc}) , 0 \rangle$ infatti

$$\frac{}{\langle \text{reset} , 1 \rangle \rightarrow \langle 0 \rangle} \quad A$$

$$\frac{}{\langle \text{reset} ; \text{inc} , 1 \rangle \rightarrow \langle \text{inc} , 0 \rangle} \quad G$$

$$\frac{}{\langle \text{reset} ; \text{inc} ; \text{wnz} (\text{reset} ; \text{inc}) , 1 \rangle \rightarrow \langle \text{inc} ; \text{wnz} (\text{reset} ; \text{inc}) , 0 \rangle} \quad F$$

in fine $\langle \text{inc} ; \text{wnz} (\text{reset} ; \text{inc}) , 0 \rangle$ transisce in
 $\langle \text{wnz} (\text{reset} ; \text{inc}) , 1 \rangle$ infatti

$$\frac{}{\langle \text{inc} , 0 \rangle \rightarrow \langle 1 \rangle} \quad B$$

$$\frac{}{\langle \text{inc} ; \text{wnz} (\text{reset} ; \text{inc}) , 0 \rangle \rightarrow \langle \text{wnz} (\text{reset} ; \text{inc}) , 1 \rangle} \quad G$$

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

Dimostriamo ora che $\langle \text{wnz} \text{inc} , n \rangle \rightarrow^* \langle 0 \rangle$
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 $\langle \text{wnz inc}, 255 \rangle \rightarrow^* \langle 0 \rangle$

Infatti, essendo $255 > 0$, la regola *E* dà
 $\langle \text{wnz inc}, 255 \rangle \rightarrow \langle \text{inc}; \text{wnz inc}, 255 \rangle$
e poi la computazione continua con
 $\langle \text{inc}; \text{wnz inc}, 255 \rangle \rightarrow \langle \text{wnz inc}, 0 \rangle \rightarrow \langle 0 \rangle$.

Il secondo passo segue dalla regola *D*
mentre il primo è dimostrato con

$$\frac{\langle \text{inc}, 255 \rangle \rightarrow \langle 0 \rangle}{\langle \text{inc}; \text{wnz inc}, 255 \rangle \rightarrow \langle \text{wnz inc}, 0 \rangle} \begin{array}{l} B \\ G \end{array}$$

Si noti che

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

Lucido 22: espressioni [7-8].

Caso induttivo: se $\langle wnz\ inc, n \rangle \rightarrow^* \langle 0 \rangle$
allora $\langle wnz\ inc, n - 1 \rangle \rightarrow^* \langle 0 \rangle$.

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

Basta dimostrare

$$\langle wnz\ inc, n - 1 \rangle \rightarrow^* \langle wnz\ inc, n \rangle$$

e usare la transitività di \rightarrow^* . La regola E dà:

$$\langle wnz\ inc, n - 1 \rangle \rightarrow^* \langle inc ; wnz\ inc, n - 1 \rangle.$$

Ora $255 - (n - 1) > 0$ implica $n - 1 < 255$
e quindi è corretto applicare B come segue:

$$\frac{\frac{\langle inc, n - 1 \rangle \rightarrow \langle n \rangle}{\langle inc ; wnz\ inc, n - 1 \rangle \rightarrow \langle wnz\ inc, n \rangle} B}{\langle inc ; wnz\ inc, n - 1 \rangle \rightarrow \langle wnz\ inc, n \rangle} G$$

Esercizio 8.

Una macchina \mathcal{M} con un registro a valori in \mathbb{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:

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

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

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

Dimostrare poi che ogni comando termina.

Le configurazioni stanno in $\Gamma \equiv (P \times \mathbb{N}) \cup \mathbb{N}$.

La relazione " \rightarrow " è definita dalle regole:

$$\frac{}{\langle \text{set } m, n \rangle \rightarrow \langle m \rangle} A$$

$$\frac{}{\langle \text{add } m, n \rangle \rightarrow \langle n + m \rangle} B$$

$$\frac{}{\langle \text{sub } m, n \rangle \rightarrow \langle n - m \rangle} C \quad (m \leq n)$$

$$\frac{}{\langle \text{sub } m, n \rangle \rightarrow \langle 0 \rangle} D \quad (m > n)$$

$$\frac{\langle P_1, n \rangle \rightarrow \langle P_1', n' \rangle}{\langle P_1 ; P_2, n \rangle \rightarrow \langle P_1' ; P_2, n' \rangle} E$$

$$\frac{\langle P_1, n \rangle \rightarrow \langle n' \rangle}{\langle P_1 ; P_2, n \rangle \rightarrow \langle P_2, n' \rangle} F$$

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 $n \in \mathbb{N}$

$$\langle P, n \rangle \rightarrow^* \langle n' \rangle \text{ sse } \langle P', n \rangle \rightarrow^* \langle n' \rangle$$

Si noti che non occorre quantificare su n' .

I programmi P_1 e P_1' non sono equivalenti perché $\langle P_1, 0 \rangle \rightarrow^* \langle 2 \rangle$ e $\langle P_1', 0 \rangle \rightarrow^* \langle 0 \rangle$.

La prima computazione si fa in due passi:

$$\frac{\langle \text{sub } 2, 0 \rangle \rightarrow \langle 0 \rangle}{\langle \text{sub } 2 ; \text{add } 2, 0 \rangle \rightarrow \langle \text{add } 2, 0 \rangle} \begin{matrix} D \ (2 > 0) \\ F \\ B \end{matrix}$$

La seconda computazione si fa in un passo:

$$\frac{\langle \text{add } 0, 0 \rangle \rightarrow \langle 0 \rangle}{\langle \text{add } 0, 0 \rangle \rightarrow \langle 0 \rangle} B$$

I programmi P_2 e P_2' non sono equivalenti perché $\langle P_2, 0 \rangle \rightarrow^* \langle 2 \rangle$ e $\langle P_2', 0 \rangle \rightarrow^* \langle 1 \rangle$.

La prima computazione ci è già nota perché

$$P_2 = P_1.$$

Lucido 25: espressioni [8].

La seconda computazione si fa in due passi:

$$\frac{\frac{\frac{\langle \text{sub } 1, 0 \rangle \rightarrow \langle 0 \rangle}{\langle \text{sub } 1 ; \text{add } 1, 0 \rangle \rightarrow \langle \text{add } 1, 0 \rangle} D (1 > 0)}{F}}{\langle \text{add } 1, 0 \rangle \rightarrow \langle 1 \rangle} B$$

I programmi P_3 e P_3' sono equivalenti infatti la regola B dà $\langle \text{add } 2, n \rangle \rightarrow \langle n + 2 \rangle$ e si ha anche $\langle \text{add } 2 ; \text{sub } 1 ; \text{add } 1, n \rangle \rightarrow^* \langle n + 2 \rangle$.

Questa computazione si fa in tre passi:

$$\frac{\frac{\frac{\frac{\langle \text{add } 2, n \rangle \rightarrow \langle n + 2 \rangle}{\langle \text{add } 2 ; \text{sub } 1, n \rangle \rightarrow \langle \text{sub } 1, n + 2 \rangle} B}{E}}{\langle \text{add } 2 ; \text{sub } 1 ; \text{add } 1, n \rangle \rightarrow \langle \text{sub } 1 ; \text{add } 1, n + 2 \rangle} F}{\frac{\frac{\langle \text{sub } 1, n + 2 \rangle \rightarrow \langle n + 1 \rangle}{\langle \text{sub } 1 ; \text{add } 1, n + 2 \rangle \rightarrow \langle \text{add } 1, n + 1 \rangle} C}{\langle \text{add } 1, n + 1 \rangle \rightarrow \langle n + 2 \rangle} B}$$

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 \mathbb{N} tale che se $\gamma_1, \gamma_2 \in \Gamma$ e se $\gamma_1 \rightarrow \gamma_2$ allora

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

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

$$C(\langle n \rangle) = 0 \quad C(\langle \text{set } m, n \rangle) = 1$$

$$C(\langle \text{add } m, n \rangle) = 1 \quad C(\langle \text{sub } m, n \rangle) = 1$$

$$C(\langle P_1 ; P_2, n \rangle) = C(\langle P_1, n \rangle) + C(\langle P_2, n \rangle)$$

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

Caso base: $\gamma_1 \rightarrow \gamma_2$ 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

$$\frac{}{\langle P, n \rangle \rightarrow \langle n' \rangle}$$

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

Lucido 27: espressioni [8].

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

In questo caso $\gamma_1' \rightarrow \gamma_2'$ 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

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

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

Se l'ultima regola è E allora si ha:
 $C(\langle P_1', n' \rangle) < C(\langle P_1, n \rangle)$ e quindi

$$\begin{aligned} C(\langle P_1'; P_2, n' \rangle) &= C(\langle P_1', n' \rangle) + \\ &+ C(\langle P_2, n' \rangle) < C(\langle P_1, n \rangle) + \\ &+ C(\langle P_2, n \rangle) = C(\langle P_1; P_2, n \rangle) \end{aligned}$$

Si noti che $C(\langle P_2, n \rangle) = C(\langle P_2, n' \rangle)$.

Il caso della regola F si tratta analogamente.