Prova scritta di

`Compilatori` `Linguaggi e Compilatori`

del 23 gennaio 2012

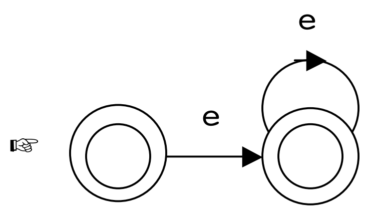
ESERCIZIO 1. È noto che, date una qualsiasi espressione **e** dell'algebra di Kleene, sono equivalenti le due espressioni

E = **e***   **E′** = ε | **ee***

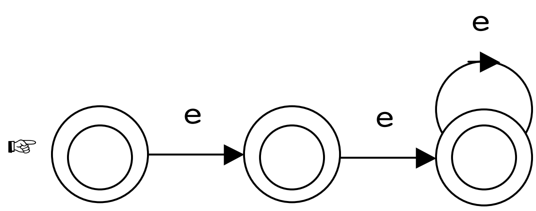
Per verificare l'equivalenza di **E** ed **E′**, provare l'equivalenza degli automi **A** e **A′** di Thompson associati rispettivamente alle due espressioni **E** ed **E′** prendendo per "alfabeto" l'insieme {**e**}.

Soluzione

L'automa finito deterministico equivalente all'automa **A** di Thompson associato all'espressione **E** è



e l'automa finito deterministico equivalente all'automa **A′** di Thompson associato all'espressioni **E′** è



Essendo gli stati dei due automi tutti di accettazione, i due automi sono equivalenti.

ESERCIZIO 2. Si consideri la grammatica acontestuale **G** con produzioni

$S \to Aa \mid b$
$A \to Ac \mid Sd \mid B$
$B \to e \mid A \mid \varepsilon$

2a) Trovare una grammatica **C** in forma normale di Chomsky ed equivalente a **G**. Applicare poi l'algoritmo CYK per decidere se la stringa ada è o meno una frase di **C** (e quindi di **G**) e, se lo è, se è una frase ambigua.

Soluzione

Tenuto conto che le variabili annullabili sono A e B, l'eliminazione delle produzioni nulle porta alla grammatica

$S \to Aa \mid a \mid b \qquad A \to Ac \mid c \mid Sd \mid B \qquad B \to e \mid A$

L'eliminazione delle produzioni unitarie $A \to B$ e $B \to A$ porta alla grammatica

$S \to Aa \mid a \mid b \qquad A \to Ac \mid c \mid Sd \mid e \qquad B \to Ac \mid c \mid Sd \mid e$

L'eliminazione della variabile improduttiva di II specie B porta alla grammatica **G**′

$S \to Aa \mid a \mid b \qquad A \to Ac \mid c \mid Sd \mid e$

e una grammatica **C** in forma normale di Chomsky ed equivalente a **G** è la seguente

$S \to AX \mid a \mid b \qquad A \to AY \mid c \mid SZ \mid e \quad X \to a \quad Y \to c \quad Z \to d$

La stringa ada è una frase non ambigua di **C** (e quindi di **G**).

a	ad	ada
d	da	
a		

$X \to a$ $S \to a$	$A \to SZ$	$S \to AX$
$Z \to d$	–	
$X \to a$ $S \to a$		

2b) Trovare una grammatica **H** priva di produzioni ricorsive a sinistra ed equivalente a **G**. Applicare l'algoritmo a discesa ricorsiva per decidere se la stringa `ada` è o meno una frase di **H**.

Soluzione

A partire dalla grammatica **G**′

$S \to Aa \mid a \mid b$
$A \to Ac \mid Sd \mid c \mid e$

ordiniamo i simboli nonterminali: primo S, secondo A. Allora sostituiamo la produzione $A \to Sd$ con le produzioni $A \to Aad \mid ad \mid bd$ e otteniamo

$S \to Aa \mid a \mid b \qquad A \to Ac \mid Aad \mid ad \mid bd \mid c \mid e$

A questo punto, eliminiamo le due produzioni mancine $A \to Ac \mid Aad$ ed otteniamo la grammatica **H**:

$S \to Aa \mid a \mid b \qquad A \to adB \mid bdB \mid cB \mid eB \qquad B \to cB \mid adB \mid \varepsilon$

A questo punto calcoliamo i valori delle funzioni I e J; in particolare, abbiamo

$I(S) = I(A) = \{a,\ b,\ c,\ e\}$, $I(B) = \{a,\ c,\ \varepsilon\}$,

$J(S) = \{\#\}$, $J(A) = J(B) = \{a\}$.

Così, la tabella di controllo è

	a	b	c	d	e	#
S	$S \to Aa$ $S \to a$	$S \to Aa$ $S \to b$	$S \to Aa$		$S \to Aa$	
A	$A \to adB$	$A \to bdB$	$A \to cB$		$A \to eB$	
B	$B \to adB$ $B \to \varepsilon$		$B \to cB$			

Analisi ricorsiva in discesa della stringa `ada` con backtracking:

etichetta del nodo di controllo	simbolo corrente	produzioni applicabili
S	a	<u>$S \to Aa$</u> $S \to a$
A	a	<u>$A \to adB$</u>

a	a		
d	d		
B	a	$B \rightarrow adB$	$\underline{B \rightarrow \varepsilon}$