10

Linguaggi Formali e Compilatori: soluzioni della Prova scritta 25/02/2005

AVVERTENZA: L'esame è diviso in 5 parti:
1 Espr. regolari e automi finiti
2 Grammatiche
3 Esercitazioni Flex Bison (fascicolo separato)
4 Grammatiche e analisi sintattica
5 Traduzione e semantica
Per superare la prova, l'allievo deve dimostrare la conoscenza di tutte e cinque le parti.

10.1 Espressioni regolari e automi finiti 20%

1. Dato il linguaggio di alfabeto $\{a, b, c\}$

$$L = \Big((b \mid c)(ab^*ab^*)^*\Big)^+ \;-\; \Big(c(a \mid b \mid c)^* \mid (a \mid b \mid c)^*aa(a \mid b \mid c)^*\Big)$$

   a) Trovare la (o le) stringa più breve che appartiene al linguaggio L.
   b) Scrivere una espr. reg. di L con i soli operatori $\{. \mid {}^* {}^+\}$
   c) Costruire, descrivendo il procedimento applicato, l'automa riconoscitore deterministico di L.

Soluzione

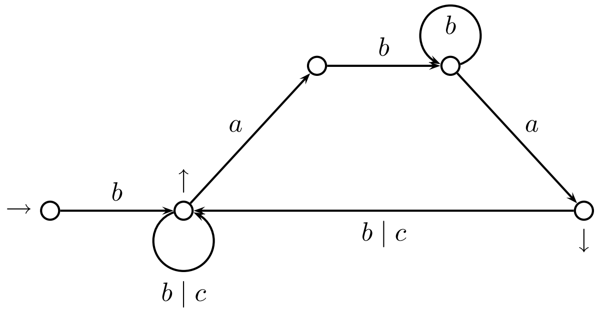
a)
$$L = L_1 - L_2 = L_1 \cap \neg L_3$$
$$L_3 = \{x \mid x \text{ inizia con } c \vee x \text{ contiene la sottostringa } aa\}$$

Ne segue che la stringa più breve di L è b.

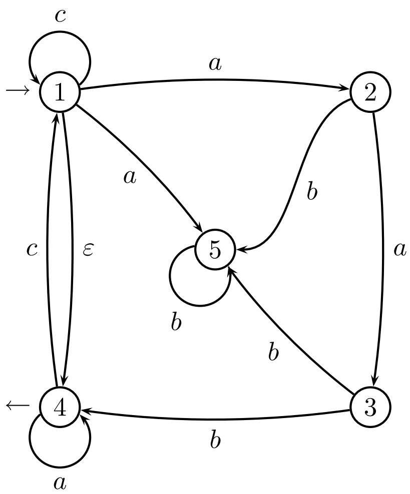
b) Seguendo la precedente descrizione di L si ha

$$L = b\left(\left(ab^+a \mid \varepsilon\right)(b \mid c)\right)^*\left(ab^+a \mid \varepsilon\right)$$

c) Costruzione semiintuitiva del riconoscitore deterministico di L, seguendo la struttura dell'espressione regolare

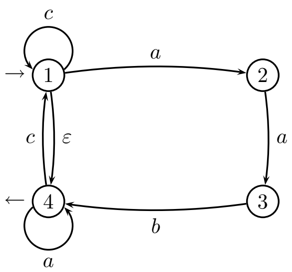


2. Determinizzare e poi minimizzare l'automa seguente.

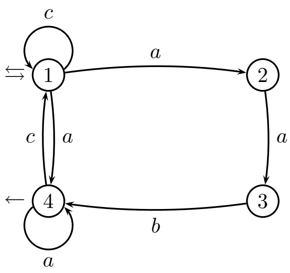


Soluzione

L'automa non è pulito, lo stato 5 si può eliminare:

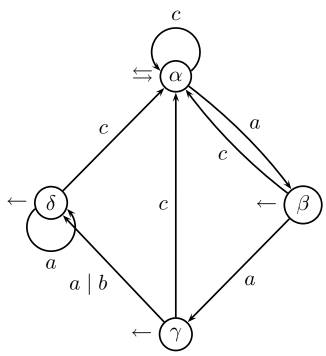


Eliminando la mossa spontanea si ottiene un automa indeterministico nello stato 1



Si calcola la seguente tabella delle transizioni:

	a	b	c	
1	$2,4$	$-$	1	α
$2,4$	$3,4$	$-$	1	β
$3,4$	4	4	1	γ
4	4	$-$	1	δ
$-$	$-$	$-$	$-$	



L'automa è minimo. Infatti, dalla colonna b si vede che γ non è equivalente a niente, dunque dalla colonna a si vede che β non è equivalente a niente e infine sempre dalla colonna a si vede che α non è equivalente a niente.

10.2 Grammatiche 20%

1. Progettare la grammatica G_1 del sottolinguaggio di Dyck di alfabeto $\Sigma =$ { '[', ']', '(', ')' } tale che ogni coppia di parentesi quadre [ ] contenga un numero pari di coppie di parentesi (qualsiasi).
   Esempi: [([ ])], ([()()])
   Controesempio: [(())()]

   Soluzione

   $$\begin{array}{l} S \to P \mid D \\ P \to [P]D \mid (P)D \mid (D)P \mid \varepsilon \\ D \to [P]P \mid (P)P \mid (D)D \end{array}$$

   P genera solo nidi pari, D solo nidi dispari. Siccome si ha solo $[P]$, ma non $[D]$, il vincolo è rispettato.

2. Il ling. L da definire sono le formule del calcolo dei predicati del primo ordine (CPPO). I simboli che possono comparire in una formula sono:
   connettivi logici: $\wedge, \vee, \Rightarrow$, elencati in ordine di precedenza;
   quantificatori: $\forall, \exists$;
   parentesi tonde;
   virgola;
   predicati: denotati da $p1, p2, \ldots$, cioè da p seguito da un intero;
   variabili individuali: denotate da $x1, x2, \ldots$, cioè da x seguito da un intero.

   Esempi:
$$\forall x1 \forall x2 \Big(p5(x1, x2) \Rightarrow (p1(x1) \wedge p3(x2)) \Big)$$
$$\forall x9 \Big(p7(x9) \wedge p2(x9) \wedge \exists x10 \big(p4(x10) \vee p5(x9, x10) \big) \Big)$$

   Fate riferimento alla vostra conoscenza del CPPO, per individuare le formule da definire con la grammatica.
   a) Progettare una grammatica G EBNF non ambigua per il ling. L
   b) (Facoltativo) Discutere se le frasi di $L(G)$ soddisfano le condizioni per essere delle formule ben formate del CPPO.

   Soluzione

   a) Grammatica:
$$\begin{array}{l} S \to Q^* \, '(' I ')' \\ Q \to (\forall \mid \exists) V \\ I \to O (\Rightarrow O)^* \\ O \to A (\vee A)^* \\ A \to T (\wedge T)^* \\ T \to P \mid S \\ V \to x[1..9][0..9]^* \\ P \to p[1..9][0..9]^* \, '(' V (',' V)^* ')' \end{array}$$
   b) Variabili quantificate ma non usate; variabili quantificate più volte nello stesso campo; formule aperte (cioè dove non tutte le variabili sono quantificate); predicati con grado variabile.

3. (facoltativo) Per la grammatica G_2 seguente :

$$\begin{array}{l} S \to SA \mid Bb \mid a \\ A \to aS \mid \varepsilon \\ B \to bB \mid b \end{array}$$

a) Dimostrare che la grammatica G_2 è ambigua.
b) Trovare una grammatica G_3 non ambigua tale che $L(G_3) = L(G_2)$.

Soluzione

a) Basta osservare le derivazioni:

$$S \Rightarrow a \quad S \Rightarrow SA \Rightarrow aA \Rightarrow a\varepsilon = a$$

Inoltre la grammatica è ricorsiva bilaterale:

$$S \Rightarrow SA \Rightarrow SaS$$

e circolare:

$$S \Rightarrow SA \Rightarrow S\varepsilon = S$$

b) L_2 è regolare! Infatti

$$L(B) = b^+$$

e sostituendo B ed A nelle regole si ottiene

$$S \to SaS \mid S \mid b^+b \mid a$$

Eliminata la regola circolare, si vede che il ling. è una lista avente come separatore a e come elemento una stringa di $(a \mid bb^+)$

$$L_2 = (a \mid bb^+)(a(a \mid bb^+))^*$$

ed è facile trovare una gramm. lineare a destra non ambigua.

Si ricorda che in generale il problema se un linguaggio libero sia regolare è indecidibile; in questo caso però si riesce a deciderlo facilmente.

10.3 Domanda relativa alle esercitazioni

Vedi fascicolo separato.

10.4 Grammatiche e analisi sintattica 20%

1. È data la seguente grammatica:

$$S \to CBA \qquad \mathcal{G} =$$

$$S \to ABC \qquad \mathcal{G} =$$

$$A \to aA \qquad \mathcal{G} =$$

$$A \to c \qquad \mathcal{G} =$$

$$B \to BS \qquad \mathcal{G} =$$

$$B \to b \qquad \mathcal{G} =$$

$$C \to AS \qquad \mathcal{G} =$$

$$C \to \varepsilon \qquad \mathcal{G} =$$

$$C \to B \qquad \mathcal{G} =$$

   Calcolarne gli insiemi guida (scrivere a lato).

   Soluzione

$$
\begin{array}{ll}
S \to CBA & \mathcal{G} = a, b, c \\
S \to ABC & \mathcal{G} = a, c \\
A \to aA & \mathcal{G} = a \\
A \to c & \mathcal{G} = c \\
B \to BS & \mathcal{G} = b \\
B \to b & \mathcal{G} = b \\
C \to AS & \mathcal{G} = a, c \\
C \to \varepsilon & \mathcal{G} = a, b, c, \dashv \\
C \to B & \mathcal{G} = b
\end{array}
$$

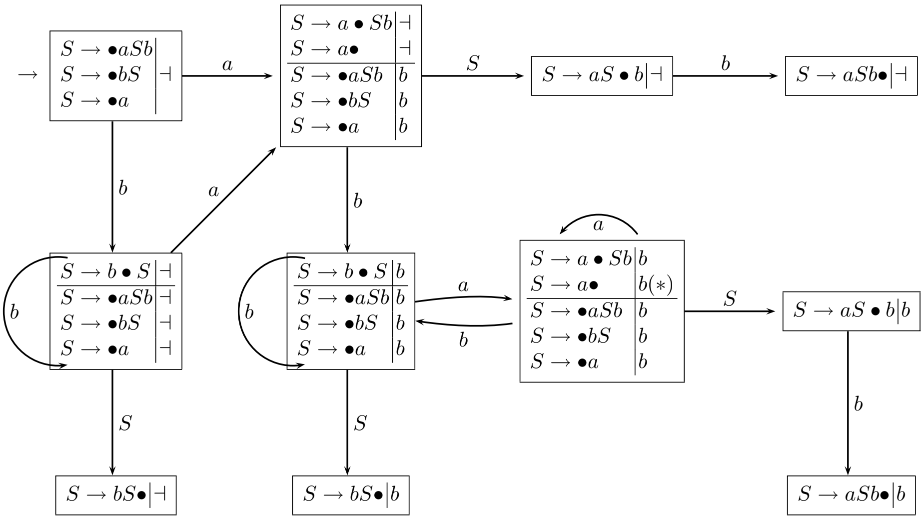
G è ricorsiva a sin., quindi non è LL(k).

2. È data la seguente grammatica:

$$S \to aSb \qquad S \to bS \qquad S \to a$$

Costruire il riconoscitore dei prefissi ascendenti LR(1) e stabilire in quali stati la gramamtica è LR(0), LALR, LR(1).

Soluzione



Conflitto LR(1) nello stato (*). C'è una candidata di riduzione con prospezione b ma lo stato presenta anche un arco uscente con etichetta b: conflitto riduzione-spostamento.

10.5 Traduzione e semantica 20%

1. Data la traduzione seguente, dove $u \in \{a, b\}^*$:

$$\tau(u) = u^R \qquad \text{se } |u| \text{ è pari}$$

$$\tau(u) = u \qquad \text{se } |u| \text{ è dispari}$$

a) Scrivere lo schema di traduzione puramente sintattico, ossia la grammatica di traduzione, che realizza la traduzione.
b) Esiste un trasduttore a pila deterministico che realizza la trasduzione τ? (motivare la risposta)
c) Definire la traduzione inversa di τ, sempre attraverso uno schema di traduzione sintattico.

Soluzione

a) Schema di traduzione puramente sintattico:

Sorgente	Pozzo
$S \to P$	$S \to P$
$S \to D$	$S \to D$
$P \to aP_1$	$P \to P_1a$
$P_1 \to aP$	$P_1 \to Pa$
$P \to bP_1$	$P \to P_1b$
$P_1 \to bP$	$P_1 \to Pb$
$P \to \varepsilon$	$P \to \varepsilon$
$D \to aD_1$	$D \to aD_1$
$D_1 \to aD$	$D_1 \to aD$
$D \to bD_1$	$D \to bD_1$
$D_1 \to bD$	$D_1 \to bD$
$D \to a$	$D \to a$
$D \to b$	$D \to b$

b) Non esiste un trasduttore a pila deterministico che realizza la trasduzione. Infatti l'automa soltanto alla fine della lettura di u può sapere se deve emettere u stessa o la riflessa; ma tale momento è troppo tardi.
c) La traduzione inversa coincide con quella diretta!

2. Considerate un quesito o *query* in un ling. simile a SQL, esemplificato da:

$$\mathsf{select}\ '*'\ \mathsf{where}\ (a_2 = 3)\ \mathsf{from}\ \underbrace{(1,3,5)(2,2,5)(2,3,2)(8,9,2)}_{\text{relazione contenente 4 tuple}}$$

Il comando seleziona le tuple che soddisfano il predicato $a_2 = 3$, ossia che hanno il valore 3 nel 2ndo campo. Il risultato è la relazione:

$$\text{ris of } S = \{(1,3,5)(2,3,2)\}$$

La sintassi del ling. è data:
$S \rightarrow \mathsf{select}\ '*'\ \mathsf{where}\ (\ \mathsf{name}\ =\ \mathsf{value}\ )\ \mathsf{from}\ R$
$R \rightarrow (T)R$
$R \rightarrow (T)$
$T \rightarrow\ \mathsf{value}\ , T$
$T \rightarrow\ \mathsf{value}$

a) Completare il progetto della gramm. ad attributi, che assegna all'attributo ris of S il risultato di un quesito. Per ipotesi tutte le tuple della relazione hanno lo stesso grado. Gli attributi sono così specificati:

ris of S	risultato del quesito: un insieme di tuple;
sel of R	risultato del quesito sulla parte della relazione avente radice R
ques of R	il quesito è un record con 2 info.: ordinale dell'attributo su cui si fa la selezione, valore di esso; nell'es. $\mathsf{record}(2,3)$
ord of name	numero ordinale dell'attributo presente nel predicato: nell'es. vale 2;
num of value	valore presente nel predicato; nell'es. vale 3
tupla of T	vettore contenente gli n interi della tupla; ad es.: $\langle 1,3,5 \rangle$

Gramm. da completare, specificando in pseudocodice le funzioni semantiche necessarie:

$S \to \mathsf{select}\ {}'*' \ \mathsf{where}\ (\ \mathsf{name}\ =\ \mathsf{value}\ )\ \mathsf{from}\ R$

ris of $S \leftarrow$ sel of R
ques of $R \leftarrow$ record(ord of name, num of value)

$R_0 \to (T)R_2$

ques of $R_2 \leftarrow \ldots$

sel of $R_0 \leftarrow \ldots$

$R \to (T)$

sel of $R \leftarrow \ldots$

$T_0 \to \ \mathsf{value}\ , T_1$

tupla of $T_0 \leftarrow \ldots$

$T \to \ \mathsf{value}$

tupla of $T \leftarrow \langle$num of $\mathsf{value}\rangle$

b) Esaminare se la condizione L è soddisfatta
c) Scrivere almeno una procedura semantica
d) (Facoltativo) Estendere il progetto della sintassi e della semantica in modo di poter scegliere su quale relazione del data-base si deve fare la selezione. Il data-base sarà fatto da più relazioni identificate dal loro nome. La clausola `from` conterrà anche il nome della relazione su cui operare.

Soluzione

a) Completare il progetto della gramm. ad attributi
ques of R è ereditato; tutti gli altri attributi sono sintetizzati.
Grammatica ad attributi:

$S \rightarrow \textsf{select}\ '*'\ \textsf{where}\ (\ \textsf{name}\ =\ \textsf{value}\ )\ \textsf{from}\ R$

ris of $S \leftarrow$ sel of R
ques of $R \leftarrow$ record(ord of $\textsf{name}$, num of $\textsf{value}$)

$R_0 \rightarrow (T)R_2$

ques of $R_2 \leftarrow$ ques of R_0

sel of R_0 $\leftarrow$ if (tupla of T[ques of R_0.ord] == ques of R_0.num) then tupla of T $\cup$ sel of R_2 else sel of R_2

(notazione C simile, supponendo che tupla of T sia un vettore di interi e ques of R una struct con campi ord e num, di tipo intero)

$R \rightarrow (T)$

sel of $R \leftarrow$ if (tupla of T[ques of R_0.ord] == ques of R.num) then tupla of T else $\emptyset$

$T_0 \rightarrow\ \textsf{value}\ , T_1$

tupla of $T_0 \leftarrow$ cat(num of $\textsf{value}$, tupla of T_1)

$T \rightarrow\ \textsf{value}$

tupla of $T \leftarrow \langle$num of $\textsf{value}\rangle$

b) La condizione L è soddisfatta (verifica tu regola per regola)
c) Piuttosto ovvio, per es. per la regola $R \rightarrow (T)R$; prova tu a scrivere la procedura
d) Ritoccare la sintassi in modo opportuno: mettere l'identificatore della relazione nella clausola select e dotare di identificatore anche la relazione. Aggiungere un attributo ID of R, e aggiungere a ques anche l'identificatore della relazione. Poi ritoccare le regole semantiche.