Linguaggi Formali e Compilatori
Prof. Breveglieri e Crespi Reghizzi
Compitino[1] 16/12/004

COGNOME e NOME:.. Matricola:...................
Iscritto a

⋄ **Laurea Specialistica:** ⋄Milano, ⋄Como, ⋄Cremona

⋄**Altro:** Specificare

Espressioni e linguaggi regolari 25%

1. Progetto di espressioni regolari
   Il linguaggio di alfabeto $\{a, b, c\}$ è tale che:
   (il numero dei caratteri a è dispari) $\wedge$ (tra due b non può trovarsi nessuna c).
   Definire il linguaggio con una espressione regolare con i soli operatori di base $\cup, *, .$ e con la croce.

   Soluzione Il ling. è ottenuto mischiando i due ling. $a(aa)^*$ e $c^*b^*c^*$, corrispondenti alle due condizioni dell'enunciato. Il ling. si può scrivere come

$$
\begin{array}{cccl}
(c^*ac^*ac^*)^* & (b^*ab^*ab^*)^* & (c^*ac^*ac^*)^*ac^* & | \\
(c^*ac^*ac^*)^* & (b^*ab^*ab^*)^*ab^* & (c^*ac^*ac^*)^* & | \\
(c^*ac^*ac^*)^*ac^* & (b^*ab^*ab^*)^* & (c^*ac^*ac^*)^* & | \\
(c^*ac^*ac^*)^*ac^* & (b^*ab^*ab^*)^*bc^* & (c^*ac^*ac^*)^*ac^* &
\end{array}
$$

   Per illustrare, la prima riga produce frasi della forma $R_1R_2R_3$ in cui R_1 e R_2 contengono un numero pari di a e R_3 un numero dispari. Al contempo la sottoparola, formata dalle lettere b e c presenti nella stringa, ha la forma $c^*b^*c^*$.

[1]Tempo 2h. 30. Libri e appunti personali sono ammessi. È consentito scrivere a matita. Per la sufficienza è necessario dimostrare di conoscere tutte e tre le parti.

2. Analisi di espr. regolari
   Dire, spiegando le ragioni, se le seguenti sono delle identità:

$$(\neg a)b \stackrel{?}{=} b \mid b(a \mid b)^* b \tag{1}$$

$$((a \mid b \mid \varepsilon)^2)^+ \stackrel{?}{=} (a \mid b)^* \tag{2}$$

   Soluzione

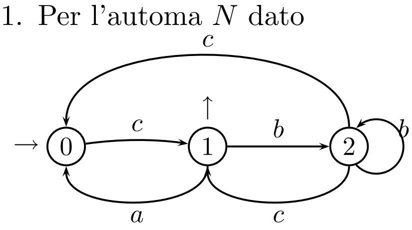
   - (1) No:
$$(\neg a)b = (\Sigma^* b \setminus \{ab\}) \ni aab \notin (b \mid b(a \mid b)^* b)$$
   - (2) Sì:
$$(a \mid b \mid \varepsilon)^2 = \{x \mid |x| \leq 2\} = \varepsilon \mid (a \mid b) \mid (a \mid b)^2$$
     Inoltre essendo
$$(a \mid b)^+ \supset ((a \mid b)^2)^+$$
     risulta
$$(\varepsilon \mid (a \mid b) \mid (a \mid b)^2)^+ = (\varepsilon \mid a \mid b)^+ = (a \mid b)^*$$

3. Verificare, motivando la risposta, se la seguente espr. reg. è ambigua:
$$(aa \mid ba)^* a \mid b(aa \mid ba)^*$$

   Soluzione L'espr. numerata
$$(a_1 a_2 \mid b_3 a_4)^* a_5 \mid b_6 (a_7 a_8 \mid b_9 a_{10})^*$$
   definisce le frasi $b_3 a_4 a_5$ e $b_6 a_7 a_8$ che si proiettano in modo ambiguo nella stessa frase baa.
   Ossia esistono due diverse implicazioni sinistre producenti la stessa stringa.

Automi finiti 25%

1. Per l'automa N dato

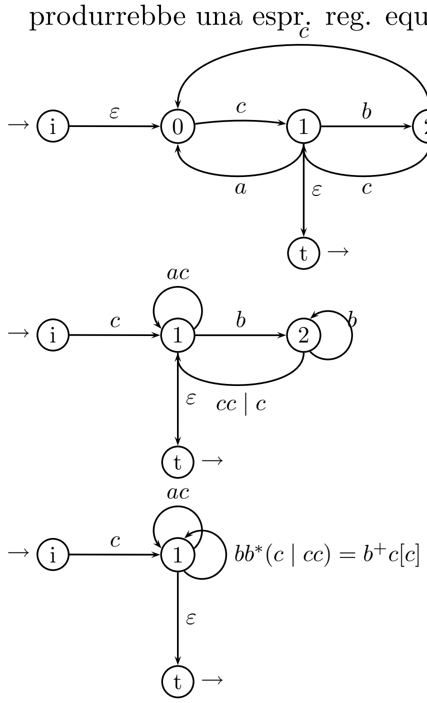


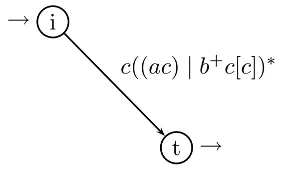
   (a) Calcolare, mostrando i passaggi, l'espressione regolare del linguaggio riconosciuto da N.

   (b) Costruire l'automa deterministico equivalente a N.

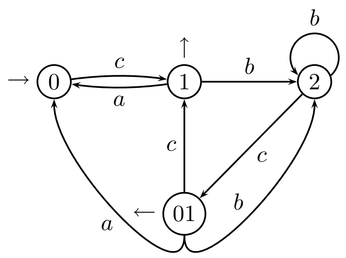
Soluzione

(a) Normalizziamo l'automa, per poi applicare il metodo di eliminazione di Brzozowsky e Mc Cluskey. (Un altro metodo sarebbe quello di scrivere la gramm. lineare a destra e risolvere il sistema di eq. insiemistiche.). Nell'applicare il metodo scegliamo di eliminare i nodi del grafo nell'ordine 0,1,2. (In effetti l'ordine 2,0,1 produrrebbe una espr. reg. equivalente più semplice.)





(b) Automa deterministico :
non essendovi archi epsilon, si applica direttamente la costruzione dell'insieme delle parti finite.



2. È data la espressione regolare

$$((a \mid \varepsilon)b^+ \mid a^*b)^+$$

(a) Costruire, mostrando i passaggi, l'automa deterministico del linguaggio

(b) Verificare se l'automa costruito è minimo, e minimizzarlo se necessario.

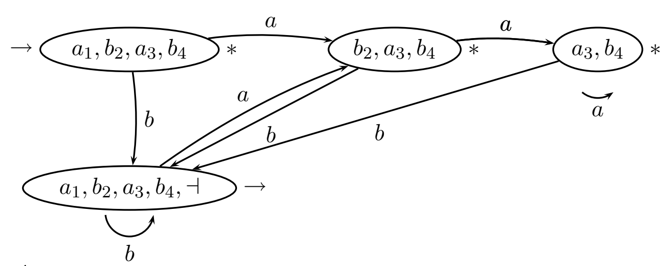
Soluzione

(a) Automa deterministico del linguaggio. Applichiamo il metodo di McNaughton-Yamada, costruendo direttamente il riconoscitore deterministico con l'algoritmo di Berry-Sethi.
Espressione numerata:

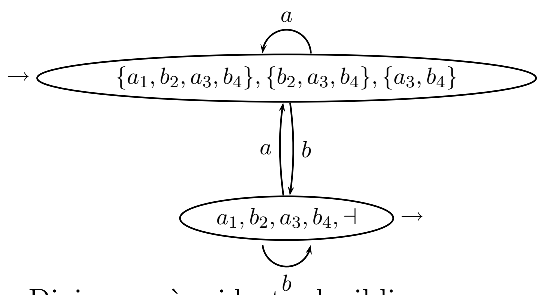
$$((a_1 \mid \varepsilon)b_2^+ \mid a_3^*b_4)^+ \dashv$$

Inizi: a_1, b_2, a_3, b_4
Fini: b_2, b_4
Seguiti:

	Séguiti
a_1	b_2
b_2	$a_1, b_2, a_3, b_4, \dashv$
a_3	a_3, b_4
b_4	$a_1, b_2, a_3, b_4, \dashv$



(b) Automa costruito minimo.
Gli stati asteriscati sono indistinguibili e vanno fusi insieme.



Diviene così evidente che il ling. comprende tutte le stringhe che terminano per b, ossia è definito dalla espr. reg.

$$(a^*b)^+$$

Come verifica finale osserviamo che la espr. reg. data nell'enunciato equivale a quest'ultima.

3. Costruire, spiegando il ragionamento, un riconoscitore, non importa se indeterministico, con il minor numero di stati possibile, per il linguaggio di alfabeto $\{a, b\}$ così definito:

   - la penultima lettera è b
     $\wedge$
   - la seconda lettera è b.

   Soluzione

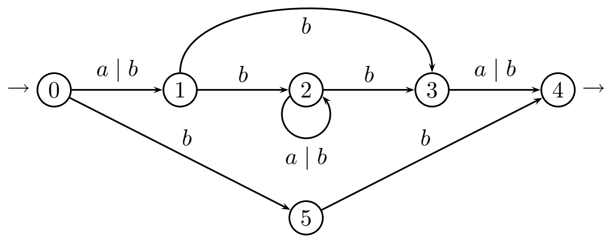
   - Poiché il ling. è formulato come congiunzione di due condizioni, esso è l'intersezione dei due corrispondenti ling.

     $$L_1 = (a \mid b)^* b (a \mid b) \qquad L_2 = (a \mid b) b (a \mid b)^*$$

     e un riconoscitore può essere costruito come il prodotto cartesiano dei due riconoscitori, ognuno con 3 stati. La macchina prodotto ha 9 stati di cui 3 irraggiungibili.
   - Il riconoscitore si può costruire direttamente osservando che il ling. è l'unione di tre casi:

     $$L = (a \mid b) b (a \mid b) \qquad \mid \qquad (a \mid b) b (a \mid b)^* b (a \mid b) \qquad \mid \qquad bb$$

     da cui si ricava l'automa



Grammatiche e linguaggi liberi 50%

1. Data la espr. reg.

$$(a(bc)^+b^* \mid (bc)^*a)^*$$

costruire una gramm. libera, non EBNF, del linguaggio.

Soluzione Fatta l'analisi sintattica della espr. reg. R nelle sue sottoespressioni,

$$(\underbrace{\underbrace{a\,\underbrace{(bc)^+}_{D}\,\underbrace{b^*}_{E}}_{B} \mid \underbrace{\underbrace{(bc)^*}_{F}\,a}_{C}}_{A})^*$$

si scrivono le regole:

$$R \to AR \mid \varepsilon \qquad A \to B \mid C \qquad B \to aDE \qquad C \to Fa$$

$$D \to bcD \mid bc \qquad E \to bE \mid \varepsilon \qquad F \to bcF \mid \varepsilon$$

Osservando che $(bc)^+ = bc(bc)^*$, si semplifica la grammatica:

$$R \to AR \mid \varepsilon \qquad A \to B \mid C \qquad B \to abcFE \qquad C \to Fa$$

$$E \to bE \mid \varepsilon \qquad F \to bcF \mid \varepsilon$$

2. Progettare una grammatica libera, BNF o EBNF, per il linguaggio (metalinguaggio) che rappresenta l'insieme di regole delle grammatiche EBNF, con le precisazioni seguenti:

   - L'alfabeto terminale delle grammatiche è a, b
   - Gli operatori sono:
     `\ast` per la stella
     `\cup` per l'unione
     concatenamento (non si scrive il puntino)
   - la freccia è scritta `\rightarrow`
   - la precedenza tra gli operatori è quella solita: stella, concatenamento, e infine unione
   - Le regole di una gramm. sono separate da `\\`
   - I simboli nonterminali sono del tipo `S301`, cioè iniziano per `S` seguito da un numero.
   - Esempi di regole:

     ```
     S301 \rightarrow (a S301 b)\ast \cup ab \\
     S3 \rightarrow (a S9 b\cup  a\ast )\ast
     ```

   - Per ogni aspetto non definito è lasciata libertà di scelta.
   - La grammatica non deve essere ambigua.

   Disegnare un albero sintattico abbastanza rappresentativo.

Soluzione Lista non vuota di regole R:

$$S \to R(\texttt{\\}R)^* \qquad R \to N\texttt{\rightarrow}D$$

La parte sinistra è un simbolo nonterminale:

$$N \to' S'(1 \dots 9)(0 \dots 9)^* \mid 'S'0$$

La parte destra delle regole è una lista di espressioni E separate dal segno di unione:

$$D \to E(\texttt{\cup}E)^*$$

Una espr. E è il concatenamento di fattori F eventualmente seguiti dal simbolo della stella:

$$E \to (F[\texttt{\ast}])^+$$

ognuno dei quali è uno dei seguenti casi:

$$F \to N \mid a \mid b \mid \ '(' \ D \ ')'$$

Sebbene l'enunciato non consideri la presenza della stringa vuota nelle parti destre, essa si può ammettere, aggiungendo la regola:

$$F \to \texttt{\textbackslash epsilon}$$

(non si può usare il carattere ε perché fa parte del metaalfabeto).

3. Progettare una grammatica, BNF o EBNF, per il seguente linguaggio di alfabeto $\{a, b, c\}$. Una frase x è così definita:

   - $x = x_1 c^+ x_2 c^+ \ldots c^+ x_n$ dove $n \geq 2$,
     ogni sottostringa x_i ha la forma $a^+ b^+$
     ma le stringhe di posto dispari e pari sono definite diversamente:

     $$x_{2j+1} = a^h b^k, h \geq k \geq 1 \qquad x_{2j} = a^k b^h, h \geq k \geq 1$$

   - Esempi: *aabcabb*, *aaabcccabcab*
     Controesempi: *abbcab*, *aababbcab*

   Soluzione Lista di elementi Dc^+P, separati da c^+, eventualmente chiusa dal termine c^+D:

   $$S \rightarrow \underbrace{Dc^+P}_{elem.}(c^+ \underbrace{Dc^+P}_{elem.})^*[c^+D]$$

   $$D \rightarrow a^*E \qquad P \rightarrow Eb^* \qquad E \rightarrow aEb \mid ab$$

4. Per la grammatica G:

$$S \rightarrow ScSdS \mid bS \mid B \qquad B \rightarrow bB \mid \varepsilon$$

Mostrare un es. di ambiguità e costruire una grammatica (non EBNF) equivalente e non ambigua.

Soluzione Due sono le cause di ambiguità presenti:

Ricorsione bilaterale: La prima regola, essendo ricorsiva a sin. e destra, causa l'ambiguità di associazione.

Due modi di produrre le b: Ad es. la frase b è ambigua:

$$S \Rightarrow bS \Rightarrow bB \Rightarrow b \qquad S \Rightarrow B \Rightarrow bB \Rightarrow b$$

Il ling. generato è una variante di quello di Dyck di alfabeto $\{c, d\}$, definito dalla gram. non ambigua $S \rightarrow cSdS \mid \varepsilon$. La variante è ottenuta inserendo le b in qualsiasi posizione. Una grammatica equivalente non ambigua è:

$$S \rightarrow BcSdS \mid B \qquad B \rightarrow bB \mid \varepsilon$$

o anche

$$S \rightarrow cSdS \mid bS \mid \varepsilon$$

5. Dati l'alfabeto $\Sigma = \{'\{','\}','[',']','(',')'\}$ e la grammatica G

$$S \to \{S\}S \mid [S]S \mid (S)S \mid \varepsilon$$

(a) Scrivere la grammatica (non EBNF) del ling. in cui nessuna coppia di graffe può contenere direttamente una coppia di tonde. Vedere gli esempi proibiti: $\{(\ldots)\}$, $\{\{\ldots\}(\ldots)[\ldots]\}$ e gli esempi ammessi $\{[(\ldots)]\}$, $\{[\{\ldots\}(\ldots)][\ldots]\}$

(b) (Facoltativo) Dato il ling. regolare

$$R = \neg(\Sigma^* \, '\{' \, '(' \, \Sigma^*)$$

scrivere la grammatica (non EBNF) del ling. $L(G) \cap R$

Soluzione

(a) Differenziamo il simbolo nonterminale all'interno delle graffe, in modo da evitare la scelta delle parentesi tonde:

$$S \to \{X\}S \mid [S]S \mid (S)S \mid \varepsilon$$

$$X \to \{X\}X \mid [S]X \mid \varepsilon$$

(b) Questo linguaggio include strettamente il precedente, ad es. esso contiene il secondo es. proibito. La grammatica è dunque più permissiva, e consente le parentesi tonde in posizioni prima proibite:

$$S \to \{X\}S \mid [S]S \mid (S)S \mid \varepsilon$$

$$X \to \{X\}S \mid [S]S \mid \varepsilon$$

6. (Facoltativo) Dato il linguaggio

$$L = \{a^m b^n c^m \mid m \geq n \geq 0\}$$

(a) Scrivere la grammatica contestuale (tipo 1) di L

(b) Giustificare che L non è un ling. libero

(c) Il complemento $\neg L$ è libero?

Soluzione

(a) Una grammatica contestuale è

$$S \to aSBC \mid \varepsilon \qquad CB \to BC \mid C$$

La prima regola genera ricorsivamente le forme di frase

$$a^m (BC)^m$$

La regola $CB \to BC$ raccoglie tutte le B prima di tutte le C, producendo le stringhe:

$$a^m B^m C^m$$

oppure, se si sceglie l'alternativa $CB \to C$, le stringhe:

$$a^m B^n C^m, \qquad m \geq n$$

Le seguenti regole sostituiscono i caratteri terminali al posto dei simboli nonterminali.

$$bB \to bb \qquad aB \to ab \qquad C \to c$$

(b) Data una stringa sorgente $a^r b^s c^t$, un automa a pila, dopo il necessario controllo che sia $r \geq s$, non può tenere memoria dei valori di r e s, ma soltanto della loro differenza. Dunque esso non può controllare l'eguaglianza di r e di t.

(c) Il complemento $\neg L$ è essenzialmente l'unione di due sottolinguaggi:

- Il ling. delle stringhe che non hanno la forma $a^* b^* c^*$, ossia il ling. $\neg(a^* b^* c^*)$ che è regolare.

- Il ling. delle stringhe $a^r b^s c^t$ che violano la condizione

  $$r \geq s \wedge r = t$$

  Ossia le stringhe soddisfano una delle tre condizioni

  $$r \neq s \vee r < s \vee t < s$$

  Esso è dunque l'unione di tre ling.:

  $$\{a^r b^s c^t \mid r \neq s\} \cup \{a^r b^s c^t \mid r < s\} \cup \{a^r b^s c^t \mid t < s\}$$

  ognuno dei quali è facilmente libero. In conclusione il complemento $\neg L$ è l'unione di un ling. regolare e di tre linguaggi liberi ossia è un ling. libero.