Esercizi di Fondamenti di Informatica per la sicurezza

Stefano Ferrari

23 dicembre 2003

Argomento 1

Grammatiche e linguaggi

Esercizi

Es. 1.1

- Definiti i linguaggi:
  - $L_1 = \{aa, ab, bc, c\}$
  - $L_2 = \{1, 22, 31\}$

  descrivere i linguaggi:
  - L_1^*
  - $L_1 L_2$
  - $L_1 L_2^*$
  - $L_1 \cup L_2$

Es. 1.2

- Data la grammatica $G = \langle T, V, P, S \rangle$:
  - $T = \{a, b\}$
  - $V = \{S, A, B\}$
  - $P = \{S \to B, S \to a, B \to bA, B \to aA, A \to aB, A \to b\}$

  mostrare la sequenza di regole da applicare per generare le seguenti frasi:
  - $aababb$
  - bb
  - $aababababb$

Es. 1.3

- Data la grammatica $G = \langle T, V, P, S \rangle$:
  - $T = \{a, b, c\}$
  - $V = \{S, A, B\}$
  - $P = \{S ::= A|aA, A ::= aAb|b|AB, B ::= b|ABa|c\}$

  mostrare la sequenza di regole da applicare per generare le seguenti frasi:
  - $aabbcbb$
  - $aabbb$
  - $aabbbca$

Soluzioni

Sol. 1.1

- $L_1^* = \{\epsilon, aa, ab, bc, c, aaaa, aaab, aabc, aac, abaa, abab, \dots\}$
- $L_1 L_2 = \{aa1, aa22, aa31, ab1, ab22, ab31, bc1, bc22, bc31, c1, c22, c31\}$
- $L_1 L_2^* = \{\dots, aa11131, \dots, ab22333, \dots, bc3111221, \dots, c31221122, \dots\}$
- $L_1 \cup L_2 = \{aa, ab, bc, c, 1, 22, 31\}$

Nota Nei casi con cardinalità infinita, i linguagi sono stati descritti riportando solo alcune delle parole che li compongono.

Sol. 1.2

$aababb$	
	S
$S \to B$	B
$B \to aA$	aA
$A \to aB$	aaB
$B \to bA$	$aabA$
$A \to aB$	$aabaB$
$B \to bA$	$aababA$
$A \to b$	$aababb$

bb	
	S
$S \to B$	B
$B \to bA$	bA
$A \to b$	bb

$aababababb$	
	S
$S \to B$	B
$B \to aA$	aA
$A \to aB$	aaB
$B \to bA$	$aabA$
$A \to aB$	$aabaB$
$B \to bA$	$aababA$
$A \to aB$	$aababaB$
$B \to bA$	$aababababA$
$A \to b$	$aababababb$

Sol. 1.3

$aabbcbb$	
	S
$S ::= A$	A
$A ::= aAb$	aAb
$A ::= aAb$	$aaAbb$
$A ::= AB$	$aaABbb$
$B ::= c$	$aaAcbb$
$A ::= AB$	$aaABcbb$
$A ::= b$	$aabBcbb$
$B ::= b$	$aabbcbb$

$aabbb$	
	S
$S ::= A$	A
$A ::= aAb$	aAb
$A ::= aAb$	$aaAbb$
$A ::= b$	$aabbb$

$aabbbca$	
	S
$S ::= A$	A
$A ::= AB$	aAB
$B ::= ABa$	$aAABa$
$B ::= c$	$aAAca$
$A ::= b$	$aAbca$
$A ::= aAb$	$aaAbbca$
$A ::= b$	$aabbbca$

Nota La successione delle regole di produzione da applicare per generare le strighe richieste non è necessariamente unica.

Argomento 2

Automi a stati finiti

Un automa a stati finiti deterministico (DFA) è una quintupla $\langle Q, \Sigma, \delta, q_0, F \rangle$ dove:

- Q è un insieme finito di stati
- Σ è un alfabeto (alfabeto di input)
- $\delta : Q \times \Sigma \to Q$ è la funzione di transizione
- q_0 è lo stato iniziale
- $F \subset Q$ è l'insieme degli stati finali

Esercizi

Es. 2.1

Costruire un DFA su $\Sigma = \{0, 1\}$ che accetti l'insieme di tutte le stringhe aventi tre 0 consecutivi.

Es. 2.2

Costruire un DFA su $\Sigma = \{0, 1\}$ che accetti l'insieme di tutte le stringhe aventi uno 0 come penultimo simbolo.

Es. 2.3

Costruire un DFA su $\Sigma = \{0, 1\}$ che accetti l'insieme di tutte le stringhe che se interpretate in notazione binaria risultino divisibili per 2.

Es. 2.4

Costruire un DFA su $\Sigma = \{0, 1\}$ che accetti l'insieme di tutte le stringhe che se interpretate in notazione binaria risultino divisibili per 3.

Soluzioni

Sol. 2.1

- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{0, 1\}$

- $\delta : Q \times \Sigma \to Q$

	0	1
q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_3	q_0
q_3	q_3	q_3

- q_0 è lo stato iniziale
- $F = \{q_3\}$

Nota Lo stato q_i codifica il fatto che si siano già letti i "0" consecutivi. Lo stato q_3, inoltre assume lo speciale ruolo di stato "pozzo" che cattura tutti i simboli seguenti ai primi tre "0" consecutivi.

Sol. 2.2

Consideriamo inizialmente quattro stati, identificati con le stringhe binarie di lunghezza due, {00, 01, 10, 11}, ai quali attribuiamo il seguente significato: se l'automa si trova in uno di questi stati è perché gli ultimi due caratteri letti sono la stringa binaria associato allo stato stesso. Grazie a questo concetto, siamo in grado di scrivere la seguente tabella delle trasizioni:

	0	1
00	00	01
01	10	11
10	00	01
11	10	11

Ad esempio, se l'automa si trovasse nello stato 01 e leggesse il simbolo "1", si porterebbe nello stato 11 in quanto gli ultimi due simboli letti formerebbero la stringa bianria 11.

Le specifiche del problema richiedono che gli stati finali siano 00 e 01, tutti e soli quelli che hanno il simbolo "0" come penultimo simbolo.

La scelta dello stato 11 come stato iniziale consente di non accettare le stringhe con meno di due caratteri.

Quindi, più formalmente, l'automa a stati finiti richiesto dal problema è:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{0, 1\}$
- $\delta : Q \times \Sigma \to Q$

	0	1
q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_0	q_1
q_3	q_2	q_3

- q_3 è lo stato iniziale
- $F = \{q_0, q_1\}$

dove lo stato q_1 denota che gli ultimi due simboli letti sono interpretati come la rappresentazione binaria di i.

Sol. 2.3

I numeri binari divisibili per 2 hanno la caratteristica di terminare per "0". Pertanto, questo esercizio è una versione semplificata dell'esercizio 2.2. Con ragionamenti analoghi, si può costruire il seguente automa:

- $Q = \{q_0, q_1\}$
- $\Sigma = \{0, 1\}$
- $\delta : Q \times \Sigma \to Q$

	0	1
q_0	q_0	q_1
q_1	q_0	q_1

- q_1 è lo stato iniziale
- $F = \{q_0\}$

Sol. 2.4

La stringa di simboli di input viene utilizzata per codificare un numero naturale e l'accettazione o meno della stringa dipende dalle caratteristiche di questo numero. Per risolvere questo esercizio bisogna quindi rifarsi alle proprietà dei numeri naturali ed alla loro rappresentazione binaria.

Supponiamo che $x_n x_{n-1} \dots x_1 x_0$ sia la stringa letta fino ad un certo momento. Essa viene interpretata secondo la nota regola come il numero x tale che $x = \sum_i x_i \cdot 2^i$. Se il successivo simbolo fosse y, la stringa $x_n x_{n-1} \dots x_1 x_0 y$ rappresenterebbe il numero $x_n x_{n-1} \dots x_1 x_0 0 + y$. In altri termini, il numero rappresentato risulterebbe sommando la cifra y al doppio del numero precedentemente rappresentato: $2x + y$.

Quello che bisogna fare, quindi è vedere come varia la divisibilità per 3 al verificarsi della trasformazione di cui sopra (raddoppiamento e somma di na cifra binaria).

Un numero naturale x è divisibile per 3 se esiste un numero naturale k tale che valga $x = 3k$. Se ipotizziamo che ad un certo istante l'automa abbia letto la stringa che rappresenta il numero x divisibile per 3 e legga il simbolo y, possono darsi due casi:

- $y = 0$: quindi $2x + y = 2 \cdot 3k + 0 = 3(2k)$ (divisibile per 3)
- $y = 1$: quindi $2x + y = 2 \cdot 3k + 1 = 3(2k) + 1$ (resto pari a 1)

Pertanto, se usiamo lo stato q_0 per indicare che l'automa ha letto una stringa divisibile per 3 e lo stato q_1 per indicare che l'automa ha letto una stringa che rappresenta un numero x che diviso per 3 dà resto 1 (x = 3 k + 1), possiamo abbozzare la seguente tabella delle transizioni:

	0	1
q_0	q_0	q_1

Proseguendo il ragionamento ed usando lo stato q_2 per indicare che l'automa ha letto una stringa che rappresenta un numero x che diviso per 3 dà resto 2 ($x = 3k + 2$), quando l'automa si troverà nello stato q_1 ($x = 3k + 1$) saranno possibili due situazioni:

- $y = 0$: quindi $2x + y = 2 \cdot (3k + 1) + 0 = 6k + 2 = 3(2k) + 2$ (resto di 2)
- $y = 1$: quindi $2x + y = 2 \cdot (3k + 1) + 1 = 6k + 2 + 1 = 6k + 3 = 3(2k + 1)$ (divisibile per 3)

mentre, quando l'automa si troverà nello stato q_2 ($x = 3k + 2$) saranno possibili le due seguenti situazioni:

- $y = 0$: quindi $2x + y = 2 \cdot (3k + 2) + 0 = 6k + 4 = 3(2k + 1) + 1$ (resto di 1)
- $y = 1$: quindi $2x + y = 2 \cdot (3k + 2) + 1 = 6k + 4 + 1 = 6k + 5 = 3(2k + 1) + 2$ (resto di 2)

L'automa richiesto è quindi formalizzabile come segue:

- $Q = \{q_0, q_1, q_2\}$
- $\Sigma = \{0, 1\}$

- $\delta : Q \times \Sigma \to Q$

	0	1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

- q_0 è lo stato iniziale
- $F = \{q_0\}$

Argomento 3

Espressioni regolari

- Metodo alternativo per descrivere i linguaggi regolari
- Una *espressione regolari* su un alfabeto Σ denota un insieme di stringhe di Σ^*, definite ricorsivamente come segue:
  - $\emptyset$ è una espressione regolare che denota l'insieme vuoto
  - ϵ è una espressione regolare che denota l'insieme $\{\epsilon\}$
  - $\forall a \in \Sigma$, a è una espressione regolare che denota l'insieme $\{a\}$
  - se r e s sono espressioni regolari che denotano rispettivamente gli insiemi R e S:
    - $r + s$ denota $R \cup S$
    - rs denota RS
    - r^* denota R^*

Esercizi

Es. 3.1

Scrivere una espressione regolare per stringhe binarie che descriva l'insieme {0, 11, 101}.

Es. 3.2

Scrivere una espressione regolare per stringhe binarie che descriva l'insieme delle stringhe composte solo da simboli "0".

Es. 3.3

Scrivere una espressione regolare per tutte stringhe binarie.

Es. 3.4

Scrivere una espressione regolare per tutte stringhe binarie che cominciano e finiscono per "1".

Es. 3.5

Scrivere una espressione regolare per le stringhe binarie che contengono almeno tre "1" consecutivi.

Es. 3.6

Scrivere una espressione regolare per le stringhe binarie che contengono almeno tre "1".

Es. 3.7

Scrivere una espressione regolare per le stringhe binarie di lunghezza dispari.

Es. 3.8

Scrivere una espressione regolare per stringhe di testo che descriva le date in formato GG/MM/AAAA.

Es. 3.9

Scrivere una espressione regolare per stringhe di testo che descriva i numeri di telefono.

Soluzioni

Sol. 3.1

L'insieme composto dalla sola stringa 0, $\{0\}$, è denotato dall'espressione regolare $\mathbf{0}$, mentre l'insieme composto dalla sola stringa 1, $\{1\}$, è denotato dall'espressione regolare $\mathbf{1}$.

La stringa 11 è ottenuto come concatenazione della stringa 1 con se stessa. L'espressione regolare che denota l'insieme costituito dalla stringa 11, $\{11\}$, è quindi ottentuta concatenando l'espressione regolare per la stringa 1 con se stessa: $\mathbf{11}$.

Analogamente, l'insieme costituito dalla sola stringa 101, $\{101\}$ viene rappresentato dall'espressione regolare $\mathbf{101}$.

L'insieme $\{0, 11, 101\}$ è ottenibile come $\{0\} \cup \{11\} \cup \{101\}$, e quindi l'espressione che lo denota è $\mathbf{0} + \mathbf{11} + \mathbf{101}$.

Sol. 3.2

L'insieme delle stringhe binarie composte da tutti "0" è dato dalla chiusura dell'insieme $\mathbf{0}$. L'espressione regolare che lo denota è quindi $\mathbf{0}^*$.

Sol. 3.3

Una stringa binaria è ottenuta concatenando elementi dell'insieme 0, 1 L'espressione regolare che denota l'insieme di tutte le stringhe binarie è $(\mathbf{0} + \mathbf{1})^*$.

Sol. 3.4

Le stringhe che iniziano e terminano per "1" si ottengono anteponendo e postponendo (tramite l'operatore di concatenazione) un simbolo "1" alle stringhe binarie. L'espressione regolare che denota tale insieme è quindi $\mathbf{1}(\mathbf{0} + \mathbf{1})^*\mathbf{1}$.

Sol. 3.5

L'espressione regolare che denota la stringa con tre "1" consecutivi è $\mathbf{111}$.

L'insieme delle stringhe che contengono almeno tre "1" consecutivi si può ottenere anteponendo e postponendo una qualsiasi stringa binaria alla stringa 111. L'espressione regolare per denotare tale insieme è quindi $(\mathbf{0} + \mathbf{1})^*\mathbf{111}(\mathbf{0} + \mathbf{1})^*$.

Nota L'insieme chiusura contiene la stringa vuota. Quindi, l'espressione $(\mathbf{0}+\mathbf{1})^*\mathbf{111}(\mathbf{0}+\mathbf{1})^*$ descrive anche le stringhe che iniziano o terminano con per 111.

Sol. 3.6

Le stringhe che contengono almeno tre "1" si ottengono intercalando le stringhe binarie con tre simboli "1". L'espressione regolare che descrive queste stringhe è $(\mathbf{0} + \mathbf{1})^*\mathbf{1}(\mathbf{0} + \mathbf{1})^*\mathbf{1}(\mathbf{0} + \mathbf{1})^*\mathbf{1}(\mathbf{0} + \mathbf{1})^*$.

Sol. 3.7

Le stringhe di lunghezza 1 sono descritte dall'espressione regolare $\mathbf{0} + \mathbf{1}$.

Le stringhe di lunghezza 2 si ottengono concatenando due stringhe di lunghezza 1. Quindi, sono descritte dall'espressione regolare $(\mathbf{0} + \mathbf{1})(\mathbf{0} + \mathbf{1})$.

Le stringhe di lunghezza pari si ottengono concatenando stringhe di lunghezza 2. Esse sono quindi descritte dalla chiusura dell'insiem delle stringhe di lunghezza 2: $((\mathbf{0} + \mathbf{1})(\mathbf{0} + \mathbf{1}))^*$.

Le stringhe di lunghezza dispari possono essere ottenute concatenando una stringa pari con una stringa di lunghezza 1. Ciò può essere descritto dall'espressione regolare $((\mathbf{0} + \mathbf{1})(\mathbf{0} + \mathbf{1}))^*(\mathbf{0} + \mathbf{1})$.

Sol. 3.8

Definiamo l'insieme delle cifre decimali $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ e l'espressione regolare corrispondente $\boldsymbol{A} = \mathbf{0} + \mathbf{1} + \mathbf{2} + \mathbf{3} + \mathbf{4} + \mathbf{5} + \mathbf{6} + \mathbf{7} + \mathbf{8} + \mathbf{9}$.

L'espressione regolare cercata può essere espressa come: $\boldsymbol{A}^2/\boldsymbol{A}^2/\boldsymbol{A}^4$.

Tale espressione non solo descrive le date, ma anche sequenze di simboli che non rappresentano date (come 00/00/0000, per esempio) o date di giorni inesistenti (31/02/2004, per esempio). Sebbene sia possibile costruire un'espressione regolare che descriva solo le date significative, la sua struttura non è semplice: essa dovrebbe incorporare tutte le regole per il numero di giorni dei mesi e addirittura anche le regole per il calcolo degli anni bisestili.

Sol. 3.9

Assumiamo che un numero telefonico abbia una struttura del tipo:

- 0039 07 3875 5047,
- +39 07 3875 5047,
- 07 3875 5047.

e consideriamo l'espressione regolare $\boldsymbol{A}$ definita nell'esercizio 3.7 e definiamo, per comodità l'espressione rogolare $\boldsymbol{S}$ per definire il carattere spazio (" "). L'espressione regolare cercata è:

- $((\mathbf{0}^2 + \text{“}+\text{”})\boldsymbol{A}^2\boldsymbol{S}) + \boldsymbol{\epsilon}$ per il prefisso internazionale
- $\boldsymbol{A}^2\boldsymbol{S}\boldsymbol{A}^4\boldsymbol{S}\boldsymbol{A}^4$ per il numero di telefono nazionale.