Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

Fondamenti di informatica per la sicurezza

anno accademico 2003–2004

docente: Stefano Ferrari

Soluzione del secondo compitino — 13.01.2004 — Versione D

valutazioni **1** (4) _____ **2** (5) _____ **3** (4) _____ **4** (7) _____ **5** (7) _____ **6** (7) _____

Cognome _____

Nome _____

Matricola _____ **Firma** _____

Esercizio 1

Siano dati i linguaggi L_1 e L_2:

- $L_1 = \{0,\ 1,\ abc\}$
- $L_2 = \{a,\ b,\ ab\}$

Descrivere i linguaggi:

a) $L_3 = L_1 \cup L_2$

b) $L_4 = L_1 L_2$

c) $L_5 = {L_1}^3$

d) $L_6 = L_1 {L_2}^*$

e) $L_7 = (L_1 L_2)^*$

f) $L_8 = {L_1}^* {L_2}^*$

Per quegli insiemi di cui sia troppo lungo (o impossibile) dare una descrizione estensionale, elencare almeno tre elementi, indicando le caratteristiche degli elementi che li compongono.

Soluzione

a) $L_3 = L_1 \cup L_2 = \{0,\ 1,\ abc,\ a,\ b,\ ab\}$

b) $L_4 = L_1 L_2 = \{0a,\ 0b,\ 0ab,\ 1a,\ 1b,\ 1ab,\ abca,\ abcb,\ abcab\}$

c) $L_5 = {L_1}^3$
L'insieme L_5 è formato dalle stringhe che si ottengono concatenando tre stringhe (qualsiasi) appartenenti a L_1. Poiché L_1 ha 3 elementi, possono essere formate $3^3 = 27$ stringhe come concatenazione di tre elementi di L_1. L'insieme $\{110,\ 0abc1,\ abcabc1,\ \}$ è un sottoinsieme di L_5.

d) $L_6 = L_1 {L_2}^*$
L'insieme L_6 è formato da stringhe che hanno un elemento di L_1 come prefisso e una concatenazione di un numero arbitrario (eventualmente nullo) di elementi di L_2 come suffisso. Poiché ${L_2}^*$ è composto da infiniti elementi, anche L_6 avrà infiniti elementi. L'insieme $\{0,\ 1aabbab,\ abcabbaa\}$ è un sottoinsieme di L_6.

e) $L_7 = (L_1 L_2)^*$
L'insieme L_7 è equivalente a ${L_4}^*$. Pertanto, anche L_7 è composto da infiniti elementi. L'insieme $\{\epsilon,\ 0aabcab,\ 1011abaabb\}$ è un sottoinsieme di L_7.

f) $L_8 = {L_1}^* {L_2}^*$
Gli elementi dell'insieme L_8 sono il risultato della concatenazione di una stringa di ${L_1}^*$ con una stringa di ${L_2}^*$. Poiché sia ${L_1}^*$ che ${L_2}^*$ hanno infiniti elementi, anche L_8 avrà cardinalità infinita. L'insieme $\{\epsilon,\ ababbba,\ 01bbaab,\ abcabc\}$ è un sottoinsieme di L_8.

Esercizio 2

Sia data la seguente grammatica, $G = \langle T, V, P, S\rangle$, definita su $A = \{a,\ b,\ c,\ d\}$:

- insieme dei simboli terminali, T: $T = A$
- insieme dei metasimboli, V: $V = \{K,\ H\}$
- insieme delle regole di produzione, P: $P = \{S ::= K,\ K ::= a|Hb,\ H ::= c|Kd|Ha\}$

Quali fra le seguenti stringhe vengono generate da G?

a) cb

b) $caabdb$

c) $adaab$

d) $cbbab$

e) aab

Riportare la successione di regole da applicare per la generazione di tali stringhe e le stringhe parziali ottenute.

Soluzione

a) cb

	S
$S ::= K$	K
$K ::= Hb$	Hb
$H ::= c$	cb

La stringa cb è generata da G: $cb \in L(G)$.

b) $caabdb$

	S
$S ::= K$	K
$K ::= Hb$	Hb
$H ::= Kd$	Kdb
$K ::= Hb$	$Hbdb$
$H ::= Ha$	$Habdb$
$H ::= Ha$	$Haabdb$
$H ::= c$	$caabdb$

La stringa $caabdb$ è generata da G: $caabdb \in L(G)$.

c) $adaab$

	S
$S ::= K$	K
$K ::= Hb$	Hb
$H ::= Ha$	Hab
$H ::= Ha$	$Haab$
$H ::= Kd$	$Kdaab$
$K ::= a$	$adaab$

La stringa $adaab$ è generata da G: $adaab \in L(G)$.

d) $cbbab$

	S
$S ::= K$	K
$K ::= Hb$	Hb
$H ::= Ha$	Hab

Non esiste una regola di produzione in G che permetta di ottenere b come sostituzione di H. La stringa $cbbab$ non è quindi generata da G: $cbbab \notin L(G)$.

e) aab

	S
$S ::= K$	K
$K ::= Hb$	Hb
$H ::= Ha$	Hab
$H ::= Ha$	$Haab$

Non è possibile eliminare il metasimbolo H. La stringa aab non è quindi generata da G: $aab \notin L(G)$.

Esercizio 3

Sia dato il seguente automa a stati finiti, A, $A = \langle Q, \Sigma, \delta, q_0, F \rangle$:

- insieme degli stati, Q: $Q = \{q_0, q_1, q_2, q_3\}$
- alfabeto di input, Σ: $\Sigma = \{a, b, c, d, e\}$
- funzione di transizione δ:

	a	b	c	d	e
q_0	q_0	q_1	q_2	q_1	q_1
q_1	q_1	q_0	q_2	q_2	q_3
q_2	q_3	q_2	q_2	q_0	q_3
q_3	q_1	q_3	q_1	q_1	q_1

- stato iniziale, q_0
- insieme di stati finali, F: $F = \{q_1\}$

Indicare:

a) quattro stringhe accettate da A

b) quattro stringhe rifiutate da A

Soluzione

a) quattro stringhe accettate da A:

- eeabd
- dbddac
- cabbd
- cbeba

b) quattro stringhe rifiutate da A:

- ccce
- acda
- caab
- a

Esercizio 4

Modellare tramite un automa a stati finiti deterministico il comportamento di un distributore automatico di bevande. Il distributore accetta monete da 50, 20, 10 e 5 centesimi. La bevanda distribuita costa 35 centesimi, e la macchina la rilascia quando la somma inserita superi tale cifra. Su richiesta, la macchina rilascia il resto.

Progettare un automa che modelli il credito in possesso della macchina. I simboli di input siano 50, 20, 10, 5, e R, dove i primi quattro modellano le monete inserite, e R modella la richiesta di resto.

Soluzione

L'automa deve modellare un sistema fisico. L'insieme dei simboli di input modella quindi gli stimoli che il sistema riceve dall'esterno e gli stati descrivono le situazioni in cui il sistema viene a trovarsi.

Questo permette di vedere l'automa come un simulatore del sistema in esame. È quindi ragionevole pensare ad un automa che accetti le stringhe che rappresentano le sequenze di stimoli fisicamente realizzabili oppure quelle che rappresentano una sequenza di eventi di particolare interesse.

Nel caso in esame, sono date le seguenti informazioni:

- l'automa deve modellare il sistema di credito del distributore;
- l'alfabeto di input, Σ, è dato dalle monete che il distributore può accettare più il segnale di restituzione del resto:
  $\Sigma = \{5c,\ 10c,\ 20c,\ 50c,\ R\}$;

  Il nome ai simboli di input è stato formato posponendo una c al valore delle monete. Questa scelta è stata fatta per facilitare la distinzione tra simboli di input e stati.
- possono essere introdotte solo monete multiple di 5 centesimi e possono essere spese solo quantità di denaro multiple di 35 centesimi: il credito residuo sarà sempre un multiplo di 5 centesimi;
- gli stati dell'automa, Q, sono dati dalla quantità di denaro che il sistema può avere come credito residuo; poiché al raggiungimento o al superamento della cifra pari a 35 centesimi la bevanda viene rilasciata (e i 35 centesimi scalati dal credito), il credito residuo sarà una quantità maggiore o uguale a 0 centesimi e strettamente minore di 35 centesimi:
  $Q = \{0,\ 5,\ 10,\ 15,\ 20,\ 25,\ 30\}$

Da notare che quando vi siano 20 o più centesimi di credito, l'introduzione di una moneta da 50 centesimi causa la distribuzione di due bevande.

Tale comportamento è formalizzato tramite la tabella delle transizioni, $\delta : Q \times \Sigma \to Q$ riportata di seguito:

$\boldsymbol{\delta}$	$5c$	$10c$	$20c$	$50c$	R
0	5	10	20	15	0
5	10	15	25	20	0
10	15	20	30	25	0
15	20	25	0	30	0
20	25	30	5	0	0
25	30	0	10	5	0
30	0	5	15	10	0

Sebbene in assenza di esplicite specifiche si possa attribuire ad ogni elemento di Q il ruolo di stato iniziale, è ragionevole ipotizzare che lo stato iniziale sia 0, che rappresenta l'assenza di credito residuo.

Se lo scopo della modellazione è la simulazione delle sequenze di operazioni che si possono correttamente compiere sul distributore, è sensato porre tutti gli stati utilizzati nell'insieme degli stati finali: $F = Q$. Ciò è dovuto al fatto che non vi sono stati non fisicamente possibili.

Esercizio 5

Sia data l'espressione regolare E, definita su $\Sigma = \{a,\ b,\ c\}$:

- $E = a(bb + ac)^2(cb^*)^*$

Quali fra le seguenti stringhe vengono descritte da E?

a) *abbac*

b) *aabccab*

c) *aacaccb*

d) *aaacbcbcbc*

e) *aacbbcbbcb*

f) *acacbcccbb*

g) *aacaccbcbbb*

Soluzione

Le espressioni regolari denotano degli insiemi di stringhe. In tal senso, possiamo applicare l'operatore di relazione insiemistica $\subseteq$ alle espressioni regolari per indicare che l'insieme denotato da un'espressione contiene l'insieme denotato da una seconda espressione regolare. Per esempio, $E_1 \subseteq E_2$ significa che tutte le stringhe descritte da E_1 sono descritte anche da E_2.

Ricordando che l'espressione regolare s descrive l'insieme di stringhe composto dalla sola s, $\{s\}$, si può dimostrare che tale stringa viene descritta da un'espressione regolare E derivando una catena di inclusioni del tipo $s \subseteq E_1 \subseteq E_2 \subseteq \cdots \subseteq E_k \equiv E$.

a) *abbac*
$abbac = a(bbac) \subseteq a(bb + ac)^2 \subseteq a(bb + ac)^2(cb^*)^*$.

La stringa *abbac* viene descritta da E.

b) *aabccab*
Le stringhe descritte da E iniziano per *abb* o per *aac*. *aabccab* inizia per *aab*, quindi non può essere descritta da E.

c) *aacaccb*
$aacaccb = a(acac)(cb) \subseteq a(ac)^2(cb^*) \subseteq a(bb + ac)^2(cb^*)^*$.

La stringa *aacaccb* viene descritta da E.

d) *aaacbcbcbc*
Le stringhe descritte da E iniziano per *abb* o per *aac*. *aaacbcbcbc* inizia per *aaa*, quindi non può essere descritta da E.

e) *aacbbcbbcb*
$aacbbcbbcb = a(acbb)(cbb)(cb) \subseteq a(bb + ac)^2(cb^*)(cb^*) = a(bb + ac)^2(cb^*)^2 \subseteq a(bb + ac)^2(cb^*)^*$

f) *acacbcccbb*
Le stringhe descritte da E iniziano per *abb* o per *aac*. *acacbcccbb* inizia per *aca*, quindi non può essere descritta da E.

g) *aacaccbcbbb*
$aacaccbcbbb = a(acac)(cb)(cbbb) = a(ac)^2(cb)(cb^3) \subseteq a(bb + ac)^2(cb^*)(cb^*) = a(bb + ac)^2(cb^*)^2 \subseteq a(bb + ac)^2(cb^*)^*$

Esercizio 6

Indicare una espressione regolare definita su $\Sigma = \{a, b, c\}$ che descriva le seguenti stringhe:

- *cabbb*
- *cbbbabbc*
- *cbcbc*
- *cbabab*

ma non le seguenti:

- *abbabbc*
- *ccababab*
- *cbbacac*
- *bccabbc*

Soluzione

Per individuare un'espressione regolare accettabile (che non sia banalmente l'unione delle quattro stringhe del primo gruppo) bisogna individuare delle regolarità nel primo insieme di stringhe che non siano presenti nel secondo insieme.

Si possono notare alcune caratteristiche comuni alle stringhe del primo gruppo:

- tutte le stringhe iniziano per *c*;
- tutte tranne la terza, dopo la *c* iniziale sono composte da una successione di *a* e *b*;
- la seconda e la terza terminano con la stringa *bc*, eventualmente (è il caso della terza) ripetuta.

Queste proprietà sono sufficienti a discriminare il primo gruppo di stringhe dal secondo. Infatti:

- *abbabbc*: non inizia per *c*;
- *ccababab*: alla *c* iniziale non segue né una sequenza di *a* e *b*, né una sequenza di *bc*;
- *cbbacac* alla *c* iniziale segue sì una sequenza di *a* e *b*, ma con qualche *c* inframezzata, e comunque senza formare la sottostringa *bc*;
- *bccabbc* non inizia per *c*.

Le proprietà summenzionate sono formalizzabili con le seguenti espressioni regolari:

- *c* iniziale: c;
- successione di *a* e *b*: $(a + b)^*$;
- successione di *bc*: $(bc)^*$;

Unendo le due espressioni, si ottiene $c^*(a + b)^*(bc)^*$.

Da notare che il fatto che la seconda e la terza sottoespressione abbiano l'operatore di chiusura fa sì che, indipendentemente le une dalle altre, possano essere presenti o assenti le sottostringhe da esse generate. L'unico vincolo è che, se entrambe generano una stringa non vuota, la sottostringa generata dalla seconda espressione regolare deve precedere la sottostringa generata dalla terza espressione regolare.

La soluzione proposta, non è tuttavia l'unica accettabile. Per esempio, anche $cb^*(ab+bb+bc)^2$ risponde ai requisiti richiesti.