Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

Fondamenti di informatica per la sicurezza

Università degli Studi di Milano

anno accademico 2006–2007 docente: Stefano Ferrari

12.01.2007 — Soluzione del secondo compitino — versione B

valutazioni **1** (4) ______ **2** (4) ______ **3** (4) ______ **4** (6) ______ **5** (6) ______ **6** (8) ______

Cognome ____________________

Nome ____________________

Matricola ____________________ **Firma** ____________________

Esercizio 1

Siano dati i linguaggi L_1 e L_2:

- $L_1 = \{a, b, ba\}$
- $L_2 = \{x, y\}$

Descrivere i linguaggi:

a) $L_3 = L_1 \cap L_2$

b) $L_4 = L_1 \cup L_2$

c) $L_5 = L_2 L_1$

d) $L_6 = {L_2}^2$

e) $L_7 = {L_2}^* {L_1}^*$

f) $L_8 = ({L_1}^2 L_2)^*$

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. In particolare, chiarire se la stringa vuota ϵ appartiene al linguaggio.

Soluzione

a) $L_3 = L_1 \cap L_2 = \emptyset$
Gli insiemi L_1 e L_2 non hanno elementi in comune, quindi la loro intersezione è vuota.

Nota: L'insieme vuoto $\emptyset$ è diverso dall'insieme costituito dalla sola stringa vuota, $\{\epsilon\}$.

b) $L_4 = L_1 \cup L_2 = \{a, b, ba, x, y\}$

c) $L_5 = L_2 L_1 = \{xa, xb, xba, ya, yb, yba\}$

d) $L_6 = L_2^2 = \{xx, xy, yx, yy\}$

e) $L_7 = L_2^* L_1^*$
L'insieme L_7 è dato dalle stringhe formate come concatenazione di un numero arbitrario (eventualmente nullo) di elementi di L_2 seguito da una concatenazione di un numero arbitrario (eventualmente nullo) di elementi di L_1. Poiché sia ${L_1}^*$ che ${L_2}^*$ sono composti da infiniti elementi, anche L_7 avrà infiniti elementi. L'insieme $\{\epsilon, ababa, xyy, ybba\}$ è un sottoinsieme di L_7.

f) $L_8 = ({L_1}^2 L_2)^*$
L'insieme L_8 è formato dalla concatenazione di un numero arbitrario (eventualmente nullo) di stringhe composte da due elementi di L_1 e da un elemento di L_2. Pertanto, L_8 è composto da infiniti elementi. L'insieme $\{\epsilon, babx, babyabaybaax\}$ è un sottoinsieme di L_8.

Esercizio 2

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

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

Quali fra le seguenti stringhe vengono generate da G?

a) *dabdd*

b) *caab*

c) *cdaddd*

d) *cdaab*

e) *bbbdd*

Riportare la successione di regole da applicare per la generazione di tali stringhe e le stringhe parziali ottenute, spiegando perché non si possono ottenere le stringhe che eventualmente non risultassero appartenere al linguaggio generato da G.

Soluzione

a) *dabdd*

	S
$S ::= K$	K
$K ::= Hd$	Hd
$H ::= Kd$	Kdd
$K ::= Hb$	$Hbdd$
$H ::= Ha$	$Habdd$
$H ::= Kd$	$Kdabdd$

Non è possibile eliminare il metasimbolo K senza aggiungere un altro simbolo.

La stringa *dabdd* non è generata da G: $dabdd \notin \mathcal{L}(G)$.

b) *caab*

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

Non esiste regola che permetta di ottenere il simbolo c dal metasimbolo H.

La stringa *caab* non è generata da G: $caab \notin \mathcal{L}(G)$.

c) *cdaddd*

	S
$S ::= K$	K
$K ::= Hd$	Hd
$H ::= Kd$	Kdd
$K ::= Hd$	$Hddd$
$H ::= Ha$	$Haddd$
$H ::= Kd$	$Kdaddd$
$K ::= c$	$cdaddd$

La stringa *cdaddd* è generata da G: $cdaddd \in \mathcal{L}(G)$.

d) *cdaab*

	S
$S ::= K$	K
$K ::= Hb$	Hb
$H ::= Ha$	Hab
$H ::= Ha$	$Haab$
$H ::= Kd$	$Kdaab$
$K ::= c$	$cdaab$

La stringa *cdaab* è generata da G: $cdaab \in \mathcal{L}(G)$.

e) *bbbdd*

	S
$S ::= K$	K
$K ::= Hd$	Hd
$H ::= Kd$	Kdd
$K ::= Hb$	$Hbdd$
$H ::= b$	$bbdd$

La stringa generata non coincide con la stringa data, *bbbdd*, e non può essere ulteriormente estesa, per mancanza di metasimboli.

La stringa *bbbdd* non è generata da G: $bbbdd \notin \mathcal{L}(G)$.

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_2	q_0	q_1	q_0	q_3
q_1	q_3	q_2	q_0	q_0	q_2
q_2	q_2	q_1	q_2	q_3	q_1
q_3	q_1	q_2	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:

- *dbea*
- *bcbe*
- *bdc*
- *acdc*

b) quattro stringhe rifiutate da A:

- *abcd*
- *badb*
- *eeea*
- *eda*

Esercizio 4

Modellare, tramite un automa a stati finiti deterministico, il funzionamento di una mongolfiera.

Una mongolfiera è dotata di un pallone e di un bruciatore.

Quando il pallone viene aperto, la mongolfiera scende, quando il bruciatore viene attivato, la molgolfiera sale.

Ipotizzare che il bruciatore possa essere attivato solo per un periodo di tempo tale per cui la mongolfiera sale di 1000 metri e che il pallone possa essere aperto solo per un periodo di tempo tale per cui la mongolfiera scende di 1000 metri. Ipotizzare inoltre che se il bruciatore non viene attivato e il pallone non viene aperto, la mongolfiera si stabilizza in quota. Se la mongolfiera sale sopra i 3000 metri, la sicurezza per i passeggeri viene considerata compromessa. Anche se la mongolfiera tocca terra troppo velocemente, cioè senza stabilizzarsi alla quota 1000, la sicurezza dei passeggeri va considerata compromessa.

Ipotizzare che non si possano verificare contemporaneamente più azioni. Modellare l'automa in modo che esso accetti solo le stringhe che descrivono il normale funzionamento della mongolfiera. In particolare, individuare possibili situazioni fisicamente irrealizzabili o pericolose e formalizzarle in modo che l'automa rifiuti le successioni di azioni che porterebbero la mongolfiera in tali situazioni.

Stati e simboli riportati o suggeriti nel testo sono solo indicativi: possono essere modificati, ridotti ed estesi a secondo delle esigenze del progetto.

Soluzione

L'automa deve modellare un sistema fisico. L'insieme dei simboli di input modella quindi gli stimoli che il sistema riceve dall'esterno (o le azioni che esso subisce) 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: l'automa deve accettare le stringhe che rappresentano le sequenze di stimoli (o azioni) fisicamente realizzabili oppure quelle che rappresentano una sequenza di eventi di particolare interesse.

Della mongolfiera, le specifiche richiedono che l'automa descriva sia l'altezza alla quale si trova, sia il suo stato dinamico (per esempio, "si trova a quota 2000 m in salita", piuttosto che "è a quota 3000" stabile). Ciò può essere fatto in diversi modi. Tra questi, quello più semplice è quello di considerare (e modellare) solo gli aspetti veramente critici della dinamicità del sistema in questione. Infatti, le specifiche riportano come unico caso critico il passaggio diretto da quota 2000 a 0 m, cioè l'atterraggio senza prima stabilizzarsi a quota 1000 m.

Inoltre, i sottosistemi corrispondenti ai comandi della mongolfiera vanno modellati. La mongolfiera dispone infatti di un bruciatore che può essere attivato per farla salire di quota e di un'apertura nel pallone che consente di farlo scendere di quota. Quindi, il sottosistema bruciatore può trovarsi in due stati e altrettanto vale per il sottosistema pallone. Pertanto, gli stati della mongolfiera potrebbero essere rappresentati dalle combinazioni degli stati dei sottosistemi (quota, dinamica, bruciatore e pallone). Tuttavia, le specifiche dicono che il bruciatore e l'apertura del pallone possono essere attivati solo per intervalli di tempo tali per cui la mongolfiera modifica la quota di 1000 m. Quindi, ipotizzando che bruiciatore e apertura non possano essere attivati contemporaneamente, si può semplificare il modello ipotizzando che dopo l'attivazione dei due dispositivi si ritorni nella situazione di riposo (bruciatore spento e pallone chiuso).

Con queste premesse, l'insieme degli stati, Q, può quindi essere:

$$Q \;=\; \{\mathit{terra},\, 1000s,\, 1000d,\, 2000,\, 3000,\, \mathit{errore}\}$$

dove *terra* indica quota 0 m, i rimanenti numeri (1000, 2000, 3000) indicano le quote permesse e le lettere indicano la situazione dinamica (stabile, s, e discesa, d).

Lo stato *errore* indica le situazioni pericolose, quelle che individuano malfunzionamenti o quelle che non si possono realizzare. Esso è tale per cui una volta raggiunto non lo si possa più lasciare. Questa caratteristica formalizza il fatto che la situazione di errore è irreversibile, cioè non esiste una sequenza di azioni che permetta di riassorbire una situazione non accettabile.

Con le semplificazioni sopra descritte, le azioni che possono essere effettuate sono:

- salita di 1000 m della mongolfiera;
- discesa di 1000 m della mongolfiera;

Per modellare la stabilizzazione della quota, si può aggiungere una azione che rappresenti un intervallo di tempo adeguato alla stabilizzazione.

Le specifiche non chiariscono cosa succeda se si apre il pallone a quota 0, ma si può ipotizzare che questa azione non abbia conseguenze. È invece chiaramente specificato che superare quota 3000 m costituisce un pericolo: il verificarsi di tale situazione porterà l'automa nello stato *errore*.

L'insieme dei simboli, Σ, può essere:

$$\Sigma = \{a,\, d,\, t\}$$

dove a e d rappresentano rispettivamente l'ascesa e la discesa, mentre t rappresenta il trascorre di un adeguato intervallo di tempo.

Ogni sequenza di azioni che non comporti il raggiungimento dello stato *errore* rappresenta il normale funzionamento della mongolfiera. Pertanto, qualsiasi sequenza di simboli che non porti l'automa nello stato *errore* deve venire accettata, e, quindi, tutti gli stati tranne *errore* compongono l'insieme degli stati finali, F.

Si può ipotizzare che lo stato iniziale sia quello relativo alla mongolfiera a terra: *terra*.

Con le ipotesi fatte, dovrebbero essere accettate, per esempio, le seguenti sequenze di azioni: *aaadttd*, *dddaad*. Al contrario, verrebbero rifiutate, tra le altre, le seguenti sequenze di azioni: *aadd*, *atadd*, *aaaa*. Va notato che aggiungendo un qualsiasi suffisso ad una stringa rifiutata, si ottiene sempre una stringa rifiutata: se una certa sequenza di azioni porta in uno stato non accettabile, qualsiasi sequenza di azioni ad essa successiva non può renderla accettabile.

La tabella delle transizioni, $\delta : Q \times \Sigma \to Q$ può essere quella riportata di seguito:

$\boldsymbol{\delta}$	a	d	t
terra	$1000s$	*terra*	*terra*
$1000s$	2000	*terra*	$1000s$
$1000d$	2000	*errore*	$1000s$
2000	3000	$1000d$	2000
3000	*errore*	2000	3000
errore	*errore*	*errore*	*errore*

Diverse varianti possono essere immaginate. Per esempio, la dinamicità de sistema può essere modellata sia in salita, sia in discesa, per tutte le quote. Inoltre, si può pensare di modellare più in dettaglio anche i sottosistemi bruciatore e pallone. In tal caso, si può ipotizzare che bruciatore attivo e pallone aperto, così come bruciatore spento e pallone chiuso non facciano variare la quota della mongolfiera, e quindi potrebbero essere interpretate come azioni di stabilizzazione.

Esercizio 5

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

- $E = (ab^* + ca)^2(c^3 + b^*ac)^*$

Individuare, motivando le risposte, quali fra le seguenti stringhe vengono descritte da E:

a) *abbcaccc*

b) *aabbb*

c) *cacabac*

d) *caaab*

e) *ccabba*

f) *bacaac*

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 $s \subseteq E_1 \subseteq E_2 \subseteq \cdots \subseteq E_k \equiv E$.

Osserviamo innanzitutto l'espressione regolare E è la concatenazione di due sottoespressioni: $E_1 = (ab^* + ca)^2$ e $E_2 = (c^3 + b^*ac)^*$. Quindi, le stringhe descritte da E dovranno obbligatoriamente avere un prefisso descritto da E_1 eventualmente seguito da un suffisso descritto da E_2 (poiché E_2 descrive anche la stringa vuota). Questa premessa semplificherà la spiegazione delle soluzioni di seguito riportate.

- *abbcaccc*
  $abbcaccc = (abb)(ca)(ccc) \subseteq (ab^*)(ca)(c^3) \subseteq (ab^* + ca)^2(c^3 + b^*ac) \subseteq (ab^* + ca)^2(c^3 + b^*ac)^*$

  La stringa *abbcaccc* viene descritta da E: $abbcaccc \in \mathcal{L}(E)$.

  Spiegazione alternativa:

  $$\begin{array}{cccccccc} a & b & b & c & a & c & c & c \\ & abb & & & ca & & c^3 & \\ & ab^* & & & ca & & (c^3 + b^*ac) & \\ & (ab^* + ca) & & & & & (c^3 + b^*ac) & \end{array}$$

- *aabbb*
  $aabbb = (a)(abbb) \subseteq (a)(ab^*) \subseteq (ab^*)(ab^*) \subseteq (ab^* + ca)(ab^* + ca) \subseteq (ab^* + ca)^2(c^3 + b^*ac)^*$

  La stringa *aabbb* viene descritta da E: $aabbb \in \mathcal{L}(E)$.

  Spiegazione alternativa:

  $$\begin{array}{ccc} a & a\ b\ b\ b & \\ a & abbb & \epsilon \\ ab^* & ab^* & (c^3 + b^*ac)^* \\ (ab^* + ca) & (ab^* + ca) & (c^3 + b^*ac)^* \\ \multicolumn{2}{c}{(ab^* + ca)^2} & (c^3 + b^*ac)^* \end{array}$$

- *cacabac*
  $cacabac = (ca)(ca)(bac) \subseteq (ab^* + ca)(ab^* +$

$ca)(b^*ac) \subseteq (ab^* + ca)^2(c^3 + b^*ac) \subseteq (ab^* + ca)^2(c^3 + b^*ac)^*$

La stringa *cacabac* viene descritta da E: $cacabac \in \mathcal{L}(E)$.

Spiegazione alternativa:

$$\begin{array}{ccc}
c \quad a & c \quad a & b \quad a \quad c \\
ca & ca & ccc \\
(ab^* + ca) & (ab^* + ca) & c^3 \\
\multicolumn{2}{c}{(ab^* + ca)^2} & (c^3 + b^*ac) \\
\multicolumn{2}{c}{(ab^* + ca)^2} & (c^3 + b^*ac)^*
\end{array}$$

- *caaab*
  Il prefisso *caaa* può essere descritto da E_1, ma la rimanente sottostringa, *ab* non può essere descritto da E_2.

  La stringa *caaab* non viene descritta da E: $caaab \notin \mathcal{L}(E)$.

- *ccabba*
  La stringa *cc* non può essere prefisso di una delle stringhe descritte da E.

  La stringa *ccabba* non viene descritta da E: $ccabba \notin \mathcal{L}(E)$.

- *bacaac*
  Nessuna delle stringhe descritte da E può iniziare per b.

  La stringa *bacaac* non viene descritta da E: $bacaac \notin \mathcal{L}(E)$.

Esercizio 6

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

- *cbbbcbbbc*
- *cccacaca*
- *cbbcbbb*
- *cbbbcacb*

ma non le seguenti:

- *cccbb*
- *acbbac*
- *bccbccc*
- *cacbaacb*

Soluzione

Si può notare che tutte le stringhe da includere hanno nel prefisso sequenze di almeno 2 simboli c e b. Più precisamente, tutte le stringhe da descrivere iniziano per c, e poi sono seguiti da c o b. Questa caratteristica può essere descritta dall'espressione regolare $c(b + c)(a + b + c)^*$. Questa espressione regolare descrive tutte le stringhe del primo insieme, ma anche una stringa del secondo insieme: *cccbb*. Per escludere quest'ultima, bisogna modificare l'espressione regolare precedente in $c(b+cca)(a+b+c)^*$ si ottiene una stringa regolare che soddisfa le specifiche. Infatti, pur descrivendo tutte le stringhe del primo gruppo, nessuna delle stringhe del secondo gruppo viene descritta da tale espressione regolare:

- *cccbb*: b è il simbolo che segue i primi tre simboli c;
- *acbbac*: non inizia per c;
- *bccbccc*: non inizia per c;
- *cacbaacb*: il secondo simbolo non è né b, né c.

Altre espressioni regolari che rispettano le specifiche:

- $c(b+c)^2(b+a+c)(c+b)(b+a)(b+c)(b+a)^*c^*$
- $cb^*(b3 + c)^*(ca + cb)^*$
- $cc^*b^*c(b^3 + c + ac^*)^*(acb)^*$