Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

Fondamenti di informatica per la sicurezza

anno accademico 2005–2006

docente: Stefano Ferrari

20.01.2006 — Soluzione del secondo compitino — vers. A

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:

- $L_1 = \{c, b, cb\}$
- $L_2 = \{x, y\}$

Descrivere i linguaggi:

a) $L_3 = L_1 \cap L_2$

b) $L_4 = L_1 \cup L_2$

c) $L_5 = L_1 L_2$

d) $L_6 = {L_1}^2$

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

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) $L_4 = L_1 \cup L_2 = \{c, b, cb, x, y\}$

c) $L_5 = L_1 L_2 = \{cx, cy, bx, by, cbx, cby\}$

d) $L_6 = {L_1}^2 = \{cc, cb, ccb, bc, bb, bcb, cbc, cbb, cbcb\}$

e) $L_7 = {L_1}^* {L_2}^*$
L'insieme L_7 è dato dalle stringhe formate come concatenazione di un numero arbitrario (eventualmente nullo) di elementi di L_1 seguito da una concatenazione di un numero arbitrario (eventualmente nullo) di elementi di L_2. Poiché sia ${L_1}^*$ che ${L_2}^*$ sono composti da infiniti elementi, anche L_7 avrà infiniti elementi. L'insieme $\{\epsilon, cbbc, yyxy, bbcbyx\}$ 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 $\{\epsilon, bcx, bcbyccx\}$ 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: $P = \{S ::= K, K ::= c|Hb|Hc, H ::= a|Kd|Ha\}$

Quali fra le seguenti stringhe vengono generate da G?

a) *dabdc*

b) *bdaab*

c) *acdb*

d) *baac*

e) *cdaac*

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) *dabdc*

	S
$S ::= K$	K
$K ::= Hc$	Hc
$H ::= Kd$	Kdc
$K ::= Hb$	$Hbdc$
$H ::= Ha$	$Habdc$
$H ::= Kd$	$Kdabdc$

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

La stringa *dabdc* non è generata da G: $dabdc \notin L(G)$.

b) *bdaab*

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

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

La stringa *bdaab* non è generata da G: $bdaab \notin L(G)$.

c) *acdb*

	S
$S ::= K$	K
$K ::= Hb$	Hb
$H ::= Kd$	Kdb
$K ::= Hc$	$Hcdb$
$H ::= a$	$acdb$

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

d) *baac*

	S
$S ::= K$	K
$K ::= Hc$	Hc
$H ::= Ha$	Hac
$H ::= Ha$	$Haac$

Non esiste regola che generi il simbolo b dal metasimbolo H.

La stringa *baac* non è generata da G: $baac \notin L(G)$.

e) *cdaac*

	S
$S ::= K$	K
$K ::= Hc$	Hc
$H ::= Ha$	Hac
$H ::= Ha$	$Haac$
$H ::= Kd$	$Kdaac$
$K ::= c$	$cdaac$

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

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_1	q_2	q_3	q_1
q_1	q_2	q_0	q_1	q_0	q_3
q_2	q_3	q_2	q_0	q_0	q_2
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:

- *ecabaa*
- *ddea*
- *cede*
- *baddd*

b) quattro stringhe rifiutate da A:

- *aed*
- *ddab*
- *aa*
- *ebdce*

Esercizio 4

Modellare, tramite un automa a stati finiti deterministico, il funzionamento di un *carillon*.

Un carillon è posto all'interno di un cofanetto. Quando il coperchio del cofanetto viene aperto, il carillon è libero di suonare. Quando il cofanetto viene chiuso, il carillon viene bloccato.

Il carillon è mosso da una molla che si carica girando una chiavetta. La molla sopporta una carica fino a 3 giri di chiavetta.

Ipotizzare che la carica possa essere effettuata solo per multipli interi di giro e che, una volta aperto il cofanetto, la scarica avvenga solo per multipli interi di giro.

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

Il sistema del problema è descrivibile come insieme di due sottosistemi indipendenti: il coperchio del cofanetto e la molla del carillon. Il coperchio può trovarsi in due stati (aperto o chiuso), mentre la molla può trovarsi in 4 stati (scarica, 1, 2 e 3 giri di carica).

Gli stati del carillon si possono derivare dalle combinazioni degli stati dei due sottosistemi. Pertanto, l'insieme degli stati, Q, può essere:

$$Q = \{a0,\ a1,\ a2,\ a3,\ c0,\ c1,\ c2,\ c3\}$$

dove la lettera a indica se il coperchio è aperto, in contrapposizione a c che indica che il coperchio è chiuso, mentre i numeri 0–3 indicano la carica della molla.

Le azioni che possono essere effettuate sul sistema sono l'apertura e la chiusura del coperchio e il caricamento della molla. Inoltre, è necessario modellare il passare del tempo per tener conto dello scaricamento della molla. In generale, ciò sarebbe un problema perchè il tempo è una grandezza continua, mentre con un automa a stati finiti possono solo essere modellate grandezze discrete (o che possono essere discretizzate). Le specifiche del problema indicano che la scarica avviene solo per multipli di giro. Ciò significa che non è necessario fare assunzioni per discretizzare il tempo: è sufficiente introdurre un evento che indichi che è trascorso il tempo sufficiente per far scaricare la molla della carica equivalente ad un giro.

Pertanto, insieme dei simboli, Σ, può essere:

$$\Sigma = \{a,\ c,\ m,\ t\}$$

dove a e c indicano, rispettivamente, l'azione di apertura e di chiusura del coperchio, mentre m indica il caricamento della molla per un giro di chiavetta, e t indica che è trascorso il tempo corrispondente alla scarica di un giro.

Le specifiche descrivono i seguenti comportamenti:

- la scarica può avvenire solo a coperchio aperto;
- il tentativo di caricare la molla per più di 3 giri causa la rottura della molla.

Le specifiche non impongono particolari condizioni per il caricamento del carillon. Pertanto, si assumerà che il carillon possa essere caricato in ogni momento, indipendentemente dallo stato del coperchio.

Poiché esiste almeno una situazione ritenuta non accettabile, è opportuno aggiungere all'insieme Q un altro stato, *errore*, 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.

Si può inoltre ipotizzare che il tentativo di apertura (chiusura) del coperchio già aperto (chiuso) generi errore. In tal caso, l'automa viene portato nello stato *errore*.

Ogni sequenza di azioni che non comporti il raggiungimento dello stato *errore* rappresenta il normale funzionamento del carillon. Pertanto, qualsiasi sequenza di simboli che non porti 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 al carillon scarico e con coperchio chiuso, *c0*.

Quindi, l'insieme degli stati, Q, qui utilizzato sarà:

$$Q = \{a0,\ a1,\ a2,\ a3,\ c0,\ c1,\ c2,\ c3,\ errore\}$$

Con le ipotesi fatte, dovrebbero essere accettate, per esempio, le seguenti sequenze di azioni: *mmmatttc*, *attmmtttmc*. Al contrario, verrebbero rifiutate, tra le altre, le seguenti sequenze di azioni: *mmmm*, *mc*, *ammtmm*. 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 in Tabella 1.

Esercizio 5

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

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

Quali fra le seguenti stringhe vengono descritte da E?

a) *bcaccc*

b) *caabbac*

c) *caabbcccac*

d) *abbabbcc*

e) *abbcabacbbac*

f) *acabbbac*

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.

a) *bcaccc*
La sottoespressione E_1 non descrive stringhe che iniziano per *b*.

Pertanto, la stringa *bcaccc* non può essere descritta da E: $bcaccc \notin \mathcal{L}(E)$.

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

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

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

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

d) *abbabbcc*
Il suffisso *cc* non può essere descritto né da E_1 (*c* deve esempre essere seguito da *a*), né da E_2 (*c* deve essere presente sequenze di lunghezza multipla di 3 oppure singolo, preceduto da *a*).

Pertanto, la stringa *abbabbcc* non può essere descritta da E: $abbabbcc \notin \mathcal{L}(E)$.

δ	a	c	m	t
a0	*errore*	*c0*	*a1*	*a0*
a1	*errore*	*c1*	*a2*	*a0*
a2	*errore*	*c2*	*a3*	*a1*
a3	*errore*	*c3*	*errore*	*a2*
c0	*a0*	*errore*	*c1*	*c0*
c1	*a1*	*errore*	*c2*	*c1*
c2	*a2*	*errore*	*c3*	*c2*
c3	*a3*	*errore*	*errore*	*c3*
errore	*errore*	*errore*	*errore*	*errore*

Tabella 1: Tabella delle transizioni dell'automa dell'esercizio 4.

e) *abbcabacbbac*
$abbcabacbbac = (abb)(ca)(bac)(bbac) \subseteq (ab^2)(ca)(bac)(b^2ac) \subseteq (ab^*)(ca)(b^*ac)(b^*ac) \subseteq (ab^*+ca)^2(b^*ac)^2 \subseteq (ab^* + ca)^2(c^3 + b^*ac)^2 \subseteq (ab^* + ca)^2(c^3 + b^*ac)^*$

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

f) *acabbbac*
$acabbbac = (a)(ca)(bbbac) \subseteq (ab^*)(ca)(b^3ac) \subseteq (ab^*+ca)^2(b^*ac) \subseteq (ab^*+ca)^2(c^3 + b^*ac) \subseteq (ab^* + ca)^2(c^3 + b^*ac)^*$

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

Esercizio 6

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

- *abbbabbaaacbca*
- *abacaca*
- *abbbab*
- *aa*

ma non le seguenti:

- *cabbbbca*
- *bcaba*
- *abbabbac*
- *abbbabccb*

Soluzione

Questi due insiemi di stringhe possono essere differenziati in molti modi. Fra tutti, forse, il più semplice lo si può individuare notando che tutte le stringhe da includere iniziano per a, e che 3 su 4 terminano per a.

Questa descrizione esclude tutte le stringhe del secondo insieme, ma deve essere ampliata per poter descrivere la stringa *abbbab*. Ammettere anche le stringhe terminanti per b, includerebbe anche una stringa da escludere, *abbbabccb*. Per differenziare ulteriormente, si può considerare il penultimo simbolo, descrivendo le stringhe da includere come le stringhe che iniziano per a e che terminano per a o per ab. Quest'ultima caratteristica può essere ampliata considerando le stringhe aventi come suffisso un simbolo a eventualmente seguito da una sequenza di b.

Queste caratteristiche possono essere descritte dall'espressione regolare $a(a + b + c)^*ab^*$: sequenze di caratteri qualsiasi ($a(a + b + c)^*ab^*$), con un simbolo a alle estremità ($\underline{a}(a+b+c)^*\underline{a}b^*$) ed eventualmente terminate con una sequenza di b ($a(a + b + c)^*a\underline{b^*}$).

Nessuna delle stringhe del secondo gruppo viene descritta da tale espressione regolare:

- *cabbbbca*: non inizia per a;
- *bcaba*: non inizia per a;
- *abbabbac*: non termina per ab^*;
- *abbbabccb*: non termina per ab^*.

Altre espressioni regolari che rispettano le specifiche del problema sono:

- $(ab^*)^2a^*(cb + ca)^*$;
- $(ab^*)^*(cb)^*(ca)^*$;
- $(ab^* + cb + ca)^*$;

- $ab^*a(a+b+c(a+b))^*$;
- $(ab^*)^2a^*(ca+cb)^*$;
- $ab(a+b+c)^*ca+(a+b)^*$.