Università degli Studi di Milano

Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

Fondamenti di informatica per la sicurezza

anno accademico 2005–2006 docente: Stefano Ferrari

25.03.2006 — Soluzione della seconda parte — versione 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:

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

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) $L_6 = {L_1}^3$

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

f) $L_8 = (L_2 L_1)^*$

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

c) $L_5 = L_1 L_2 = \{ax, az, bax, baz, bx, bz\}$

d) Il linguaggio $L_6 = L_1^3$ ha 26 elementi. L'insieme $\{aaa, aab, bbab, bbaba, bbb, bbba\}$ è un sottoinsieme di L_6.
Gli elementi che possono essere ottenuti in più di un modo devono essere riportati solo una volta.

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) dielementi di L_2. Poiché sia ${L_1}^*$ che ${L_2}^*$ sono composti da infiniti elementi, anche L_7 avrà infiniti elementi. L'insieme $\{\epsilon, bba, zzz, aazxx\}$ è un sottoinsieme di L_7.

f) $L_8 = (L_2 L_1)^*$
L'insieme L_8 è formato dalla concatenazione di un numero arbitrario (eventualmente nullo) di stringhe composte da un elemento di L_2 e da un elemento di L_1. Pertanto, L_8 è composto da infiniti elementi. L'insieme $\{\epsilon, za, xbzbxbzb\}$ 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: $P = \{S ::= H, K ::= a|Hc|Hb, H ::= b|Ka|Hd\}$

Quali fra le seguenti stringhe vengono generate da G?

a) *bbad*

b) *cdca*

c) *baaba*

d) *bcadd*

e) *baca*

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)

bbad	
	S
$S ::= H$	H
$H ::= Hd$	Hd
$H ::= Ka$	Kad
$K ::= Hb$	$Hbad$
$H ::= b$	$bbad$

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

b)

cdca	
	S
$S ::= H$	H
$H ::= Ka$	Ka
$K ::= Hc$	Hca
$H ::= Hd$	$Hdca$

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

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

c)

baaba	
	S
$S ::= H$	H
$H ::= Ka$	Ka
$K ::= Hb$	Hba
$H ::= Ka$	$Kaba$
$K ::= a$	$aaba$

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

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

d)

bcadd	
	S
$S ::= H$	H
$H ::= Hd$	Hd
$H ::= Hd$	Hdd
$H ::= Ka$	$Kadd$
$K ::= Hc$	$Hcadd$
$H ::= b$	$bcadd$

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

e)

baca	
	S
$S ::= H$	H
$H ::= Ka$	Ka
$K ::= Hc$	Hca
$H ::= Ka$	$Kaca$
$K ::= Hb$	$Hbaca$

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

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

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

- 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:

- *ddac*
- *decc*
- *ebdce*

- *bacac*

b) quattro stringhe rifiutate da A:

- *ecaba*
- *dece*
- *abcd*
- *aaeb*

Esercizio 4

Modellare, tramite un automa a stati finiti deterministico, il funzionamento di un distributore di bevande.

Un distributore di bevande è dotato di una fessura per introdurre le monete, di un tastierino per selezionare la bevanda e di un'uscita per la bevanda servita. Nel normale funzionamento, l'utente inserisce una moneta, seleziona la bevanda desiderata e ritira la bevanda.

Selezionare la bevanda senza aver inserito il denaro non ha effetti, mentre selezionare la bevanda senza che la bevanda precedente sia stata ritirata causa un malfunzionamento.

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

La macchina che deve essere descritta dall'automa è composta da alcuni sottosistemi, responsabili di particolari funzioni: la gestione del credito e l'erogazione della bevanda. L'insieme di stati in cui il distributore di bevande può trovarsi sono descritti dalle combinazioni possibili dei suoi sottosistemi.

Le specifiche non pongono condizioni sul tipo e sul numero di monete necessarie per l'acquisto delle bevande. Si suppone quindi che il sottosistema di gestione del credito possa trovarsi solo in due stati: credito insufficiente e credito sufficiente. L'erogazione avviene solo se la selezione viene effettuata in uno stato in cui il credito e' sufficiente. Per semplicità, si suppone che vengano considerate solo monete tali da assicurare credito sufficiente e che la macchina non dia resto. Per raggiungere uno stato di credito sufficiente, quindi, basta una moneta. Inoltre, al termine dell'erogazione il credito viene azzerato e diventa insufficiente.

Il sottosistema di erogazione fornisce un bicchiere al momento della selezione, se il credito è sufficiente. Altrimenti rimane nulla si verifica (e quindi l'automa rimane nello stato in cui è). L'utente può ritirare il bicchiere con la bevanda al termine dell'erogazione e prima dell'erogazione seguente. Se non ritira il bicchiere prima della successiva erogazione, la macchina distributrice manifesta un malfunzionamento. L'automa raggiunge quindi uno stato *errore*. Lo stato *errore* 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.

Le specifiche non chiariscono cosa succeda se si cerca di prelevare un bicchiere quando quest'ultimo sia assente. Si può ipotizzare che questa azione non abbia conseguenze.

L'insieme degli stati, Q, può quindi essere:

$$Q = \{c_i v,\ c_i b,\ c_s v,\ c_s b,\ \mathit{errore}\}$$

dove la prima lettera indica lo stato relativo al sottosistema di credito (c_s sta per credito sufficiente, mentre c_i sta per credito insufficiente) e la seconda lettera indica lo stato del sistema di erogazione (b sta per presenza del bicchiere, v per assenza).

Le azioni che possono essere effettuate sul sistema sono l'inserimento di una moneta (m) la selezione di una bevanda (s), e il prelevamento della bevanda (p).

Pertanto, insieme dei simboli, Σ, può essere:

$$\Sigma = \{m,\ s,\ p\}$$

Ogni sequenza di azioni che non comporti il raggiungimento dello stato *errore* rappresenta il normale funzionamento del distributore di bevande. 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 credito insufficiente e assenza di bicchiere, $c_i v$.

Con le ipotesi fatte, dovrebbero essere accettate, per esempio, le seguenti sequenze di azioni: *msp*, *mmmmssssp*, *ssssmsp*. Al contrario, verrebbero rifiutate, tra le altre, le seguenti sequenze di azioni: *msms*, *sssmmmsssms*. 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.

È possibile rendere l'automa più complesso, modellando, per esempio anche la situazione di erogazione. In tal caso, bisogna anche modellare la situazione in cui il prelevamento del bicchiere avviene mentre l'erogazione della bevanda è in atto. È ragionevole pensare che ciò rappresenti un malfunzionamento (del sistema utente-distributore, per la verità) e che quindi comporti il passaggio nello stato *errore*.

Per modellare quensta nuova situazione, sono necessari un nuovo simbolo, f, che serve per modellare la fine dell'erogazione e gli stati $c_i e$ e $c_s e$, che modellano l'erogazione in situazione di credito insufficiente e credito sufficiente, rispettivamente. Va notato che il passaggio ad uno stato di erogazione può avvenire solo con credito sufficiente e in assenza di bicchiere (stato $c_s v$). L'azione di selezione causa il passaggio nello stato $c_i e$. Se durante l'erogazione vengono inserite delle monete, si ha il passaggio allo stato $c_s e$. Al termine dell'erogazione, se il bicchiere non viene prelevato, si ha il passaggio allo stato con bicchiere presente (e con lo stato del credito coerente con quello dell'erogazione).

La tabella delle transizioni, $\delta : Q \times \Sigma \to Q$ può essere quella riportata in Tabella 2.

Esercizio 5

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

- $E = (a + bc)^*(ba^* + c)^2$

Quali fra le seguenti stringhe vengono descritte da E?

a) *aaaaccc*

b) *aaabcc*

c) *bcbcbac*

d) *cbaacac*

e) *aaabbc*

f) *aaababa*

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: $E_1 = (a+bc)^*$ e $E_2 = (ba^*+c)^2$. Quindi, le stringhe descritte da E dovranno obbligatoriamente avere un suffisso descritto da E_2 eventualmente preceduto da un prefisso descritto da E_1 (poiché E_1 descrive anche la stringa vuota). Questa premessa semplificherà la spiegazione delle soluzioni di seguito riportate.

a) *aaaaccc*

Il suffisso *cc* può essere descritto da E_2, ma la rimanente sottostringa, *aaaac* non può essere descritta da E_1 perché presenta un simbolo c preceduto da un simbolo a. Nelle stringhe descritte da E_1, invece, i simboli c devono sempre essere preceduti da un simbolo b.

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

$\boldsymbol{\delta}$	m	s	p
$c_i v$	$c_s v$	$c_i v$	$c_i v$
$c_i b$	$c_s b$	$c_i b$	$c_i v$
$c_s v$	$c_s v$	$c_i b$	$c_s v$
$c_s b$	$c_s b$	*errore*	$c_s v$
errore	*errore*	*errore*	*errore*

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

$\boldsymbol{\delta}$	m	s	p	f
$c_i v$	$c_s v$	$c_i v$	$c_i v$	$c_i v$
$c_i b$	$c_s b$	$c_i b$	$c_i v$	$c_i b$
$c_s v$	$c_s v$	$c_i b$	$c_s v$	$c_s v$
$c_s b$	$c_s b$	*errore*	$c_s v$	$c_s b$
$c_i e$	$c_s e$	$c_s e$	*errore*	$c_i b$
$c_s e$	$c_s e$	$c_s e$	*errore*	$c_s b$
errore	*errore*	*errore*	*errore*	

Tabella 2: Tabella delle transizioni dell'automa esteso dell'esercizio 4.

b) *aaabcc*

Il suffisso *cc* può essere descritto da E_2, ma la rimanente sottostringa, *aaab* non può essere descritta da E_1 perché termina con un simbolo *b*. Nelle stringhe descritte da E_1, invece, i simboli *b* devono sempre essere seguiti da un simbolo *c*.

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

c) *bcbcbac*
$bcbcbac = (bc)(bc)(ba)(c) \subseteq (a + bc)(a + bc)(ba^*)(c) \subseteq (a + bc)^2(ba^* + c)^2 \subseteq (a + bc)^*(ba^* + c)^2$

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

d) *cbaacac*

La stringa data termina per *cac*. Tale suffisso non può essere descritto da E_2 in quanto il simbolo *a* non viene preceduto dal simbolo *b*.

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

e) *aaabbc*

Il suffisso *bc* può essere descritto da E_2, ma la rimanente sottostringa, *aaab* non può essere descritta da E_1 perché termina con un simbolo *b*. Nelle stringhe descritte da E_1, invece, i simboli *b* devono sempre essere seguiti da un simbolo *c*.

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

f) *aaababa*
$aaababa = (a)(a)(a)(ba)(ba) \subseteq (a + bc)(a + bc)(a + bc)(ba^*)(ba^*) \subseteq (a + bc)^3(ba^* + c)(ba^* + c) \subseteq (a + bc)^*(ba^* + c)^2 \subseteq (a + bc)^*(ba^* + c)^2$

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

Esercizio 6

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

- *abbcaaccc*
- *abcaa*
- *acaccc*
- *abcacca*

ma non le seguenti:

- *acbbba*
- *aacbaa*
- *ccaacb*
- *accbcc*

Soluzione

Si può notare che tutte le stringhe da includere iniziano con un simbolo a e che, se ve ne sono, tutti i simboli b sono tutti in una sequenza che segue il primo simbolo a.

Queste caratteristica può essere descritta dall'espressione regolare $ab^*(a+c)^*$. Nessuna delle stringhe del secondo gruppo viene descritta da tale espressione regolare:

- *acbbba*: la sequenza di b segue un simbolo c;
- *aacbaa*: il simbolo b segue un simbolo c;
- *ccaacb*: non inizia per a;
- *accbcc*: il simbolo b segue un simbolo c.

Altre espressioni regolari che rispettano le specifiche:

- $(a+b)^*c(a+c^2)^*$;
- $(a+b+c)^*(a+c)^3$;
- $(ab^*+ca^*)^*$;
- $(a+b+c)^*c(a+c)^2$.