Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

Fondamenti di informatica per la sicurezza

anno accademico 2005–2006

docente: Stefano Ferrari

08.02.2006 — Soluzione della seconda parte — 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:

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

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) $L_6 = {L_2}^3$

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

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. 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) $L_6 = L_2^3 = \{xxx, xzz, zxx, zzz\}$
Gli elementi che possono essere ottenuti in più di un modo devono essere riportati solo una volta.

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 $\{\epsilon, babb, xxxz, zxb\}$ è un sottoinsieme di L_7.

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

b) *aadd*

c) *aadba*

d) *adca*

e) *dbba*

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)

ccad	
	S
$S ::= H$	H
$H ::= Hd$	Hd
$H ::= Ka$	Kad
$K ::= Hc$	$Hcad$

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

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

b)

aadd	
	S
$S ::= H$	H
$H ::= Hd$	Hd
$H ::= Hd$	Hdd
$H ::= Ka$	$Kadd$
$K ::= a$	$aadd$

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

c)

aadba	
	S
$S ::= H$	H
$H ::= Ka$	Ka
$K ::= Hb$	Hba
$H ::= Hd$	$Hdba$
$H ::= Ka$	$Kadba$
$K ::= a$	$aadba$

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

d)

adca	
	S
$S ::= H$	H
$H ::= Ka$	Ka
$K ::= Hc$	Hca
$H ::= Hd$	$Hdca$
$H ::= Ka$	$Kadca$

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

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

e)

dbba	
	S
$S ::= H$	H
$H ::= Ka$	Ka
$K ::= Hb$	Hba
$H ::= b$	bba

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

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

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_3	q_0	q_1	q_2	q_3
q_1	q_2	q_2	q_2	q_0	q_0
q_2	q_1	q_1	q_1	q_3	q_1
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:

- *ecaba*
- *dece*
- *decc*
- *aaeb*

b) quattro stringhe rifiutate da A:

- *abcd*
- *ebdce*
- *bacaa*
- *ddac*

Esercizio 4

Modellare, tramite un automa a stati finiti deterministico, il funzionamento di una macchina da scrivere.

Una macchina da scrivere è dotata di una tastiera, un rullo per lo scorrimento del foglio e di un nastro inchiostrato. Nel normale funzionamento, l'utente inserisce un foglio, pigia i tasti della tastiera e, al termine della scrittura, toglie il foglio dal rullo.

Agire sui tasti senza il foglio rovina il rullo, così come inserire più di due fogli per volta.

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 della macchina. In particolare, individuare possibili situazioni fisicamente irrealizzabili o pericolose e formalizzarle in modo che l'automa rifiuti le successioni di azioni che porterebbero la macchina da scrivere 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.

L'insieme di stati in cui la macchina da scrivere può trovarsi sono descritti dal numero di fogli inseriti. Le specifiche non pongono un limite massimo ai fogli che si possono inserire, ma solo ai fogli inseriti *se* si batte un tasto. Pertanto, arbitrariamente, si può fissare tale limite a 4 e ipotizzare che il tentativo di inserire un altro foglio oltre tale valore causi il malfunzionamento della macchina da scrivere. Tale malfunzionamento verrà equiparato a quello causato da battere un tasto senza foglio inserito o con più di due fogli inseriti e verrà rappresentato mediante 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.

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

$$Q = \{q_0,\, q_1,\, q_2,\, q_3,\, q_4,\, errore\}$$

dove la lettera q indica uno stato che contraddistingue una situazione di normale funzionamento e il numero in pedice rappresenta il numero di fogli inseriti.

Le azioni che possono essere effettuate sul sistema sono l'inserimento (i) e il disinserimento (d) di un foglio, e la pigiatura di un tasto (t).

Pertanto, insieme dei simboli, Σ, può essere:

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

Sebbene non sia esplicitamente specificato, è ragionevole supporre che la rimozione di un foglio quando la macchina è vuota non abbia alcun effetto. Anche in questo caso, si tratta di una specifica aggiuntiva, totalmente arbitraria: differenti scelte progettuali conducono ad automi differenti, ma comunque accettabili.

Ogni sequenza di azioni che non comporti il raggiungimento dello stato *errore* rappresenta il normale funzionamento del timbro. 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 alla macchina da scrivere vuota, q_0.

Con le ipotesi fatte, dovrebbero essere accettate, per esempio, le seguenti sequenze di azioni: *iittttdttit*, *idittttditt*. Al contrario, verrebbero rifiutate, tra le altre, le seguenti sequenze di azioni: *t*, *iittiittti*, *itdt*. 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 l'azione di inserimento e rimozione del nastro inchiostrato (ed, eventualmente modellando anche un evento di esaurimento del nastro inchiostrato). Ciò richiede l'introduzione degli stati necessari a tener conto di tutte le possibili combinazioni tra numero di fogli inseriti e stato del nastro. Si può

$\boldsymbol{\delta}$	i	d	t
q_0	q_1	q_0	*errore*
q_1	q_2	q_0	q_1
q_2	q_3	q_1	q_2
q_3	q_4	q_2	*errore*
q_4	*errore*	q_3	*errore*
errore	*errore*	*errore*	*errore*

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

ipotizzare che il tentativo di scrivere senza nastro o con nastro esaurito comporti un malfunzionamento e quindi porti l'automa nello stato *errore*, così come il tentativo di inserire un nastro quando il nastro fosse già presente. Inoltre, poiché un nastro non inserito non può esaurirsi, la lettura del simbolo di esaurimento del nastro porta l'automa, nel caso in cui si trovi in uno stato che descrive la situazione di assenza del nastro, nello stato *errore*.

Con le ipotesi aggiuntive sopra descritte, l'insieme degli stati, Q, sarebbe:

$$Q = \{n_0, n_1, n_2, n_3, n_4, a_0, a_1, a_2, a_3, a_4, e_0, e_1, e_2, e_3, e_4, \textit{errore}\}$$

dove la lettera n indica che il nastro è presente ed efficiente, la lettera a indica che il nastro è assente e la lettera e indica che è esaurito.

L'insieme dei simboli, Σ, diverrebbe:

$$\Sigma = \{i_f, d_f, t, e, i_n, d_n\}$$

dove i simboli i_f e d_f indicano l'azione di inserimento e rimozione del foglio, t indica la battitura di un tasto, e indica l'esaurimento del nastro e i simboli i_n e d_n indicano rispettivamente l'inserimento e la rimozione del nastro.

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 $\Sigma = \{a, b, c\}$:

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

Quali fra le seguenti stringhe vengono descritte da E?

a) *aaaaccc*

b) *aaabcc*

c) *cbcbbabc*

d) *cbaacac*

e) *aaabbc*

f) *caaabaabc*

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: $E_1 = (a^2 + cb)^*$ e $E_2 = (ba^*b + c)^2$. 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 stringa rimanente, *aaaac* non può essere descritta da E_1. Infatti, E_1 non è in grado di descrivere stringhe nelle quali il simbolo c non sia seguito da un simbolo b.

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

b) *aaabcc*
Il suffisso *cc* può essere descritto da E_2, ma la stringa rimanente, *aaab* non può essere descritta da E_1. Infatti, E_1 non è in grado

$\boldsymbol{\delta}$	i_f	d_f	t	e	i_n	d_n
n_0	n_1	n_0	*errore*	e_0	*errore*	a_0
n_1	n_2	n_0	n_1	e_1	*errore*	a_1
n_2	n_3	n_1	n_2	e_2	*errore*	a_2
n_3	n_4	n_2	*errore*	e_3	*errore*	a_3
n_4	*errore*	n_3	*errore*	e_4	*errore*	a_4
a_0	a_1	a_0	*errore*	*errore*	n_0	a_0
a_1	a_2	a_0	*errore*	*errore*	n_1	a_1
a_2	a_3	a_1	*errore*	*errore*	n_2	a_2
a_3	a_4	a_2	*errore*	*errore*	n_3	a_3
a_4	*errore*	a_3	*errore*	*errore*	n_4	a_4
e_0	e_1	e_0	*errore*	e_0	n_0	a_0
e_1	e_2	e_0	*errore*	e_1	n_1	a_1
e_2	e_3	e_1	*errore*	e_2	n_2	a_2
e_3	e_4	e_2	*errore*	e_3	n_3	a_3
e_4	*errore*	e_3	*errore*	e_4	n_4	a_4
errore	*errore*	*errore*	*errore*	*errore*	*errore*	*errore*

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

di descrivere stringhe nelle quali il simbolo b non sia preceduto da un simbolo b.

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

c) *cbcbbabc*
$cbcbbabc = (cb)(cb)(bab)(c) \subseteq (a^2 + cb)(a^2 + cb)(ba^*b)(c) \subseteq (a^2 + cb)^2(ba^*b + c)^2 \subseteq (a^2 + cb)^*(ba^*b + c)^2$

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

d) *cbaacac*
L'espressione regolare E_2 non può descrivere stringhe in cui il simbolo a non sia delimitato da simboli b. Quindi, la stringa data non ha suffissi che siano descrivibili da E_2 ($cbaac\underline{ac}$).

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

e) *aaabbc*
Il suffisso *bbc* può essere descritto da E_2, ma la stringa rimanente, *aaa* non può essere descritta da E_1. Infatti, E_1 non è in grado di descrivere sequenze di a di lunghezza dispari.

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

f) *caaabaabc*
Il suffisso *baabc* può essere descritto da E_2, ma la stringa rimanente, *caaa* non può essere descritta da E_1. Infatti, E_1 non è in grado di descrivere stringhe nelle quali il simbolo c non sia seguito da un simbolo b.

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

Esercizio 6

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

- *acccaccc*
- *aaba*
- *bcaccaccc*
- *bcacacca*

ma non le seguenti:

- *ccaac*
- *ababcab*
- *acaccacc*
- *bbcaccb*

Soluzione

Si può notare che tutte le stringhe da includere terminano con un simbolo a o, in alternativa, con una sequenza di c lunga 3.

Queste caratteristica può essere descritta dall'espressione regolare $(a+b+c)^*(a+c^3)$. Nessuna delle stringhe del secondo gruppo viene descritta da tale espressione regolare:

- *ccaac*: termina per *ac*;
- *ababcab*: termina per *b*;
- *acaccacc*: termina per *acc*;
- *bbcaccb*: termina per *b*.

In aggiunta, si può notare che le stringhe da includere non iniziano mai per c. Quindi, una soluzione alternativa può essere $(a + b)(a + b + c)^*(a + c^3)$.

L'insieme delle stringhe descritte può essere ulteriormente ristretto imponendo che il simbolo a terminale sia eventualmente preceduto da una sequenza di c: $(a + b)(a + b + c)^*(c^*a + c^3)$.