Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

Fondamenti di informatica per la sicurezza

anno accademico 2004–2005

docente: Stefano Ferrari

05.02.2005 — 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:

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

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

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

c) $L_5 = L_1 L_2 = \{bx, cx, bxx, cxx\}$

d) $L_6 = {L_1}^2 = \{bb, bc, cb, cc\}$

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 composto da infiniti elementi, anche L_7 avrà infiniti elementi. L'insieme $\{\epsilon, cbc, xxx, bccbcbx\}$ è un sottoinsieme di L_7.

f) $L_8 = ({L_2}^2 L_1)^*$
L'insieme L_8 è formato dalla concatenazione di un numero arbitrario (eventualmente nullo) di stringhe composte da due elementi di L_2 e da un elemento di L_1. Pertanto, L_8 è composto da infiniti elementi. L'insieme $\{\epsilon, xxb, xxxc, xxxxbxxc\}$ 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: $P = \{S ::= K, K ::= d|Hb|Hc, H ::= c|Kd|Ha\}$

Quali fra le seguenti stringhe vengono generate da G?

a) *ddaab*

b) *acbdac*

c) *cacdb*

d) *aaab*

e) *baac*

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) *ddaab*

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

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

b) *acbdac*

	S
$S ::= K$	K
$K ::= Hc$	Hc
$H ::= Ha$	Hac
$K ::= Kd$	$Kdac$
$H ::= Hb$	$Hbdac$
$H ::= c$	$cbdac$

Non esistono altri metasimboli da espandere e mancano alcuni simboli per ottenere la stringa data.

Quindi, la stringa *acbdac* non è generata da G: $acbdac \notin \mathcal{L}(G)$.

c) *cacdb*

	S
$S ::= K$	K
$K ::= Hb$	Hb
$H ::= Kd$	Kdb
$K ::= Hc$	$Hcdb$
$H ::= Ha$	$Hacdb$
$H ::= c$	$cacdb$

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

d) *aaab*

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

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

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

e) *baac*

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

Non è possibile ottenere il simbolo b dal metasimbolo H.

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

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

- *bae*
- *debc*
- *eac*

- *acde*

b) Quattro stringhe rifiutate da A:

- *abcd*
- *ea*
- *ccd*
- *ebea*

Esercizio 4

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

La porta può essere posta in due posizioni: aperta e chiusa. Una porta chiusa può essere e una porta aperta può essere chiusa, ma non si può aprire una porta aperta, nè chiudere una porta chiusa.

La porta è dotata di tasto che agisce su un blocco bistabile: se la porta è libera, la pressione del tasto la blocca nella posizione in cui si trova, mentre se la porta è bloccata, la pressione del tasto la libera.

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

Stati e simboli riportati nel testo sono solo indicativi: possono essere modificati, ridotti ed estesi a secondo delle esigenze del progetto. Al fine di facilitare il progetto e la verifica dell'automa, si suggerisce di fornire qualche esempio di stringa accettata e di stringa rifiutata, modellizzanti, rispettivamente, successioni di azioni consentite e successioni di azioni non consentite.

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.

Le informazioni date dalle specifiche consentono di definire:

- insieme degli stati, Q:
  $Q = \{aperta,\ chiusa,\ ablocc,\ cblocc\}$
  dove *aperta* indica la porta aperta, *chiusa* la porta chiusa, mentre *ablocc* e *cblocc* indicano la porta bloccata rispettivamente nelle posizioni aperta e chiusa;
- insieme dei simboli, Σ: $\Sigma = \{a, b, c\}$
  dove a indica l'azione di apertura, b l'azione di blocco/sblocco e c l'azione di chiusura.

Le specifiche descrivono i seguenti comportamenti:

- la porta aperta non si può aprire;
- la porta chiusa non si può chiudere;
- in seguito ad una azione di blocco, una porta libera si blocca, mentre una porta bloccata torna libera.

Le prime due situazioni descrivono una situazione fisicamente irrealizzabile. Per formalizzare tale condizione, si può aggiungere un altro stato, *errore*, tale per cui una volta raggiunto non lo si possa più lasciare.

Si può ipotizzare che il tentativo di apertura di una porta bloccata in posizione aperta generi errore, mentre non abbia effetto se la porta fosse bloccata in posizione di chiusura. Analoghe considerazioni valgono per la chiusura di una porta bloccata.

Ogni sequenza di azioni che non comporti il raggiungimento dello stato *errore* rappresenta il normale funzionamento della porta. 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 porta chiusa, *chiusa*.

Con queste ipotesi aggiuntive, dovrebbero essere accettate, per esempio, le seguenti sequenze di azioni: *acacabcccbc*, *acbbaca*. Al contrario, verrebbero rifiutate, tra le altre, le seguenti sequenze di azioni: *c*, *acaa*, *acbc*. 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.

$\boldsymbol{\delta}$	a	b	c
aperta	*errore*	*ablocc*	*chiusa*
chiusa	*aperta*	*cblocc*	*errore*
ablocc	*errore*	*aperta*	*ablocc*
cblocc	*cblocc*	*chiusa*	*errore*
errore	*errore*	*errore*	*errore*

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

Esercizio 5

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

- $E = (a + ba)^3 + (c^*a + b)^4$

Quali fra le seguenti stringhe vengono descritte da E?

a) *aaba*

b) *bcccabb*

c) *babababa*

d) *bcbcbcbc*

e) *baaba*

f) *abbaabba*

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 che le stringhe che vengono descritte da E sono descritte, in alternativa, o dalla sottoespressione $E_1 = (a + ba)^3$ o dalla sottoespressione $E_2 = (c^*a + b)^4$. Questa premessa semplificherà la spiegazione delle soluzioni di seguito riportate.

a) *aaba*
$aaba \subseteq (a)(a)(ba) \subseteq (a+ba)^3 \subseteq (a+ba)^3 + (c^*a+b)^4$

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

b) *bcccabb*
$bcccabb \subseteq (b)(ccca)(b)(b) \subseteq (b + ccca)^4 \subseteq (b + c^*a)^4 \subseteq (a + ba)^3 + (c^*a + b)^4$

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

c) *babababa*

La stringa *babababa* non può essere descritta da E_1 perché il simbolo b può essere ottenuto solo seguito da un simbolo a e questa sequenza può ripetersi solo per 3 volte. Nella stringa presa in esame, invece, la sequenza ba si ripete per 4 volte. La stringa *babababa* non può inoltre essere descritta da E_2 perché, in tal caso, la sequenza ba potrebbe essere ripetuta solo 2 volte.

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

d) *bcbcbcbc*

La stringa *bcbcbcbc* non può essere descritta da E_1 perché il simbolo c non può essere ottenuto da essa. Inoltre, *bcbcbcbc* non può essere descritta da E_2 perché il simbolo c dovrebbe apparire sempre seguito da un simbolo a.

Quindi, la stringa *bcbcbcbc* non viene descritta da E: $bcbcbcbc \notin \mathcal{L}(E)$.

e) *baaba*
$baaba \subseteq (ba)(a)(ba) \subseteq (a + ba)^3 \subseteq (a + ba)^3 + (c^*a + b)^4$

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

f) *abbaabba*
La stringa *abbaabba* non può essere descritta da E_1 perché quest'ultima non può descrivere stringhe nelle quali il simbolo b si ripeta consecutivamente. La stringa *abbaabba* non può inoltre essere descritta da E_2 perché quest'ultima non può descrivere strin-

ghe nelle quali sia assente il simbolo c ed abbiano lunghezza diversa da 4.

Quindi, la stringa *abbaabba* non viene descritta da E: $abbaabba \notin \mathcal{L}(E)$.

Esercizio 6

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

- *aaabcbc*
- *abcaaabccb*
- *aaabcbcbb*
- *bcaaabcbc*

ma non le seguenti:

- *abcccbabc*
- *aaacca*
- *bbcbabb*
- *bcbbbaac*

Soluzione

Si può notare che nessuna delle stringhe da includere contiene il simbolo a nei suoi ultimi tre simboli. Questa caratteristica può essere descritta dall'espressione regolare $(a + b + c)^*(b + c)^3$:

- suffisso di lunghezza 3 diverso da a: $(a+b+c)^*\underline{(b+c)^3}$;
- qualsiasi sequenza di simboli come prefisso: $\underline{(a + b + c)^*}(b + c)^3$.

Nessuna delle stringhe del secondo gruppo viene descritta da tale espressione regolare in quanto in tutte è presente un simbolo a fra i loro ultimi tre simboli:

- *abcccb*$\underline{a}$*bc*
- *aaacc*$\underline{a}$
- *bbcb*$\underline{a}$*bb*
- *bcbbba*$\underline{a}$*c*

Altre espressioni regolari che rispettano le specifiche del problema sono:

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