Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

Fondamenti di informatica per la sicurezza

anno accademico 2003–2004

docente: Stefano Ferrari

Soluzione della seconda parte — 02.04.2004 — Versione A

valutazioni **1** (4) ______ **2** (5) ______ **3** (4) ______ **4** (7) ______ **5** (7) ______ **6** (7) ______

Cognome ____________________

Nome ____________________

Matricola ____________________ **Firma** ____________________

Esercizio 1

Siano dati i linguaggi L_1 e L_2:

- $L_1 = \{0,\ 1,\ x\}$
- $L_2 = \{x,\ a,\ b\}$

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) $L_6 = {L_2}^2$

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

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

g) $L_9 = {L_1}^* {L_2}^*$

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.

Soluzione

a) $L_3 = L_1 \cap L_2 = \{x\}$

b) $L_4 = L_1 \cup L_2 = \{0,\ 1,\ x,\ a,\ b\}$

c) $L_5 = L_1 L_2 = \{0x,\ 1x,\ xx,\ 0a,\ 1a,\ xa,\ 0b,\ 1b,\ xb\}$

d) $L_6 = {L_2}^2 = \{xx,\ ax,\ bx,\ xa,\ aa,\ ba,\ xb,\ ab,\ bb\}$

e) $L_7 = {L_1}^* L_2$
L'insieme L_7 è formato da stringhe che hanno un elemento di L_2 preceduto da una concatenazione di un numero arbitrario (eventualmente nullo) di elementi di L_1. Poiché ${L_1}^*$ è composto da infiniti elementi, anche L_7 avrà infiniti elementi. L'insieme $\{001xb,\ a,\ xxxxx\}$ è un sottoinsieme di L_7.

f) $L_8 = (L_1 L_2)^*$
L'insieme L_8 è equivalente a ${L_5}^*$. Pertanto, L_8 è composto da infiniti elementi. L'insieme $\{\epsilon,\ 0xxx1b,\ xb0a\}$ è un sottoinsieme di L_8.

g) $L_9 = {L_1}^* {L_2}^*$
Gli elementi dell'insieme L_9 sono il risultato della concatenazione di una stringa di ${L_1}^*$ con una stringa di ${L_2}^*$. Poiché sia ${L_1}^*$ che ${L_2}^*$ hanno infiniti elementi, anche L_9 avrà cardinalità infinita. L'insieme $\{\epsilon,\ x1011baxa,\ 0001,\ baxbbx\}$ è un sottoinsieme di L_9.

Esercizio 2

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

- insieme dei simboli terminali, T: $T = A$
- insieme dei metasimboli, V: $V = \{K,\ H\}$
- insieme delle regole di produzione, P: $P = \{S ::= K,\ K ::= d|Ha,\ H ::= c|Ka|Hb\}$

Quali fra le seguenti stringhe vengono generate da G?

a) *dcb*

b) *daba*

c) *caba*

d) *cbaa*

e) *dabbba*

Riportare la successione di regole da applicare per la generazione di tali stringhe e le stringhe parziali ottenute.

Soluzione

a) *dcb*

	S
$S ::= K$	K

Non esiste nessuna regola che permetta di ottenere il simbolo b dal metasimbolo K. La stringa *dcb* non è quindi generata da G: $dcb \notin L(G)$.

b) *daba*

	S
$S ::= K$	K
$K ::= Ha$	Ha
$H ::= Hb$	Hba
$H ::= Ka$	$Kaba$
$K ::= d$	$daba$

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

c) *caba*

	S
$S ::= K$	K
$K ::= Ha$	Ha
$H ::= Hb$	Hba
$H ::= Ka$	$Kaba$

Non esiste nessuna regola che permetta di ottenere il simbolo c dal metasimbolo K. La stringa *caba* non è quindi generata da G: $caba \notin L(G)$.

d) *cbaa*

	S
$S ::= K$	K
$K ::= Ha$	Ha
$H ::= Ka$	Kaa

Non esiste nessuna regola che permetta di ottenere il simbolo b dal metasimbolo K. La stringa *cbaa* non è quindi generata da G: $cbaa \notin L(G)$.

e) *dabbba*

	S
$S ::= K$	K
$K ::= Ha$	Ha
$H ::= Hb$	Hba
$H ::= Hb$	$Hbba$
$H ::= Hb$	$Hbbba$
$H ::= Ka$	$Kabbba$
$K ::= d$	$dabbba$

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

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

- aacb
- baaae
- dbcc
- a

b) quattro stringhe rifiutate da A:

- abcd
- eea
- caacd
- bbcd

Esercizio 4

Un robot pulisci-pavimenti percorre traiettorie casuali, evitando gli ostacoli.

Il robot è dotato di un sensore di prossimità che gli consente di rilevare gli ostacoli. In assenza di ostacoli, il robot procede in linea retta. Quando viene incontrato un ostacolo, il robot ruota finché l'ostacolo rimane nel campo del sensore e poi riprende ad andare diritto. Il senso di rotazione è alternativamente in senso orario ed in senso antiorario: se l'ultima volta ha ruotato in senso orario, la volta seguente ruoterà in senso antiorario.

Modellare il comportamento del robot tramite un automa a stati finiti deterministico, dove gli stati rappresentano le manovre che il robot deve eseguire (*vai diritto*, *ruota a destra*, *ruota a sinistra*) e i valori letti dal sensore (*libero*, *ostacolo*) rappresentano la stringa di input.

Soluzione

L'automa deve modellare un sistema fisico. L'insieme dei simboli di input modella quindi gli stimoli che il sistema riceve dall'esterno 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. È quindi ragionevole pensare ad un automa che accetti le stringhe che rappresentano le sequenze di stimoli fisicamente realizzabili oppure quelle che rappresentano una sequenza di eventi di particolare interesse.

Nel caso in esame, sono date le seguenti informazioni:

- insieme degli stati, Q:
  $Q = \{$*vai diritto*, *ruota a destra*, *ruota a sinistra*$\}$
- alfabeto di input, Σ:
  $\Sigma = \{$*libero*, *ostacolo*$\}$

Dalle specifiche è possibile dedurre i seguenti comportamenti:

- se il robot trova libera la sua traiettoria, va diritto;
- se trova un ostacolo e l'ultima volta ha ruotato a destra, ruota a sinistra;
- se trova un ostacolo e l'ultima volta ha ruotato a sinistra, ruota a destra;
- se mentre sta ruotando rileva un ostacolo, continua a ruotare nella stessa direzione.

Dalle specifiche risulta evidente la necessità di memorizzare il senso di rotazione. Ruotare a destra o a sinistra, infatti, non dipende dalla situazione (presenza di un ostacolo), ma dalla storia passata (ciò che ha fatto l'ultima volta). Ciò può essere agevolmente realizzato sdoppiando lo stato *vai diritto* in due stati, *vai diritto-dx* e *vai diritto-sx*, in modo che lo stato memorizzi anche la direzione di rotazione in caso di ostacolo.

Mancano invece specifiche riguardo allo stato finale ed allo stato iniziale.

La tabella delle transizioni, $\delta : Q \times \Sigma \to Q$ può essere quella riportata in Tabella 1.

Per completezza, deve essere definito uno stato iniziale ed almeno uno finale. Lo stato iniziale può essere scelto arbitrariamente fra i quattro stati elencati precedentemente. Lo stato finale sarà uno stato q_1 non raggiungibile da nessuno degli stati precedentemente descritti.

Si può inoltre considerare una variante con uno stato iniziale fittizio: si aggiunge all'automa di cui sopra uno stato che non rappresenta nessuna manovra. È uno stato (non finale) q_0 con le seguenti transizioni: $\delta(q_0, \textit{libero}) = \textit{vai diritto-dx}$, $\delta(q_0, \textit{ostacolo}) = \textit{curva a destra}$; da notare che la scelta degli stati *vai diritto-dx* e *curva a destra* è totalmente arbitraria, e potrebbe essere sostituita, rispettivamente con *vai diritto-sx* e *curva a sinistra*.

Esercizio 5

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

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

Quali fra le seguenti stringhe vengono descritte da E?

a) *ababccacc*

b) *abb*

c) *acacb*

d) *abcabc*

e) *abbcabb*

f) *ababaccc*

g) *aaababb*

$\boldsymbol{\delta}$	*libero*	*ostacolo*
vai diritto-dx	*vai diritto-dx*	*curva a destra*
vai diritto-sx	*vai diritto-sx*	*curva a sinistra*
curva a destra	*vai diritto-sx*	*curva a destra*
curva a sinistra	*vai diritto-dx*	*curva a sinistra*

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

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

a) *ababccacc*
$ababccacc = (ababcc)a(cc) = ((ab)^2c^2)a(c)^2 \subset (ab+c)^2a(b+c)^2 \subset (ab+c)^*a(b+c)^2$.

La stringa *ababccacc* viene descritta da E.

b) *abb*
$abb = a(bb) = a(b^2) \subset a(b+c)^2 \subset (ab+c)^*a(b+c)^2$.

La stringa *abb* viene descritta da E.

c) *acacb*

La stringa *acacb* non può essere descritta da E. Infatti, se una stringa di E inizia per *ac*, non c'è altro modo per descriverla che utilizzare la sottoespressione $a(b+c)^2$. In tal caso, però, la stringa non può essere più lunga di tre caratteri.

d) *abcabc*
$abcabc = (abc)a(bc) = ((ab)(c))a(bc) \subset (ab+c)^2a(b+c)^2 \subset (ab+c)^*a(b+c)^2$

La stringa *abcabc* viene descritta da E.

e) *abbcabb*
La stringa *abbcabb* inizia per *abb* e poi seguono altri caratteri. Essa non può essere quindi descritta da E in quanto E richiede che la doppia b possa essere solo finale.

f) *ababaccc*

La stringa *ababaccc* non può essere descritta da E in quanto termina per *ccc*. Le stringhe descritte da E devono avere per terzultimo carattere solo il simbolo a.

g) *aaababb*
La stringa *aaababb* non può essere descritta da E in quanto una stringa di E non può iniziare con *aa*.

Esercizio 6

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

- *ccaaaabb*
- *cc*
- *ccaacbbc*
- *ccbbbbccc*

ma non le seguenti:

- *aaabbc*
- *ccababa*
- *ccbbbcccca*
- *bcabbca*

Soluzione

Si può notare che le stringhe da includere hanno tutte *cc* come prefisso. Inoltre, esse terminano eventualmente con una sequenza di simboli b e c. Ciò non accade per la seconda stringa, dove eliminando il prefisso *cc*, rimane la stringa vuota ϵ. Se ci sono dei simboli a, essi sono presenti in sequenze e comunque solo dopo il *cc* iniziale e prima della sequenza do b e c finale. Queste caratteristiche possono essere descritte dall'espressione regolare $c^2a^*(b+c)^*$:

- *cc* come prefisso: $\underline{c^2}a^*(b+c)^*$

- eventuale suffisso composto da una sequenza di b e c: $c^2a^*\underline{(b+c)^*}$
- ed eventuale sequenza di a nella parte centrale: $c^2\underline{a^*}(b+c)^*$

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

- *aaabbc*: non ha *cc* come prefisso;
- *ccababa*: termina per a e tale a non fa parte di una sequenza che segue immediatamente il prefisso *cc*;
- *ccbbbccca*: termina per a e tale a non fa parte di una sequenza che segue immediatamente il prefisso *cc*;
- *bcabbca*: non ha *cc* come prefisso.