Università degli Studi di Milano

Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

Fondamenti di informatica per la sicurezza

anno accademico 2006–2007 docente: Stefano Ferrari

12.01.2007 — Soluzione del secondo compitino — 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:

- $L_1 = \{a, b, ab\}$
- $L_2 = \{x, y\}$

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) $L_4 = L_1 \cup L_2 = \{a, ab, b, x, y\}$

c) $L_5 = L_1 L_2 = \{abx, aby, ax, ay, bx, by\}$

d) $L_6 = L_1^2 = \{aa, aab, ab, aba, abab, abb, ba, bab, bb\}$

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 $\{\epsilon, aabb, yxxy, abbxyy\}$ 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 $\{\epsilon, abbx, babyaaybax\}$ 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: $P = \{S ::= K,\ K ::= c|bH|aH,\ H ::= a|dK|bH\}$

Quali fra le seguenti stringhe vengono generate da G?

a) *bbbdaab*

b) *adbbd*

c) *bdaa*

d) *abdd*

e) *bbbdc*

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)

bbbdaab	
	S
$S ::= K$	K
$K ::= bH$	bH
$H ::= bH$	bbH
$H ::= bH$	$bbbH$
$H ::= dK$	$bbbdK$
$K ::= aH$	$bbbdaH$
$H ::= a$	$bbbdaa$

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

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

b)

adbbd	
	S
$S ::= K$	K
$K ::= aH$	aH
$H ::= dK$	adK
$K ::= bH$	$adbH$
$H ::= bH$	$adbbH$
$H ::= dK$	$adbbdK$

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

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

c)

bdaa	
	S
$S ::= K$	K
$K ::= bH$	bH
$H ::= dK$	bdK
$K ::= aH$	$bdaH$
$H ::= a$	$bdaa$

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

d)

abdd	
	S
$S ::= K$	K
$K ::= aH$	aH
$H ::= bH$	abH
$H ::= dK$	$abdK$

Non esiste regola che permetta di ottenere il simbolo d dal metasimbolo K.

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

e)

bbbdc	
	S
$S ::= K$	K
$K ::= bH$	bH
$H ::= bH$	bbH
$H ::= bH$	$bbbH$
$H ::= dK$	$bbbdK$
$K ::= c$	$bbbdc$

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

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_1	q_0	q_3
q_1	q_2	q_1	q_2	q_3	q_1
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:

- *dbea*
- *bcbe*
- *aaa*
- *accda*

b) quattro stringhe rifiutate da A:

- *abcd*
- *bbab*
- *eeea*
- *eda*

Esercizio 4

Modellare, tramite un automa a stati finiti deterministico, il funzionamento di una taglierina automatica.

La taglierina è composta di una lama, azionata meccanicamente, e di un piano di lavoro. Per controllare la lama sono disponibili due tasti, posti sui lati della taglierina, ben separati. Per azionare la lama i tasti devono essere premuti contemporaneamente.

Sul piano, sotto la lama possono essere posizionati fino a due lamierini. Per il buon funzionamento della taglierina, però, solo un lamierino per volta può essere tagliato.

Ipotizzare che non si possano verificare contemporaneamente più azioni. Modellare l'automa in modo che esso accetti solo le stringhe che descrivono il normale funzionamento della taglierina. In particolare, individuare possibili situazioni fisicamente irrealizzabili o pericolose e formalizzarle in modo che l'automa rifiuti le successioni di azioni che porterebbero la taglierina 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 lama, il tasto destro e quello sinistro il pulsante, e il piano di lavoro. L'insieme di stati in cui la taglierina può trovarsi sono descritti dalle combinazioni possibili dei suoi sottosistemi (o da un suo sottoinsieme).

Ipotizzando che la lama si abbassi e rimanga in tale posizione solo finché entrambi i pulsanti sono premuti e che l'operazione di taglio provveda anche a rimuovere i lamierini dal piano di lavoro, si possono modellare gli stati dei sottosistemi del sistema in esame nel modo che segue.

Lo stato in cui si trova la lama può essere descritto dalla posizione (alta o bassa). I due tasti possono essere premuti o rilasciati, in modo indipendente l'uno dall'altro. Infine, il piano di lavoro può essere caratterizzato dal numero di lamierini presenti (0, 1, o 2). Quindi lo stato della taglierina potrebbe essere descritto da $2 \times 2 \times 2 \times 3 = 24$ stati.

Tuttavia, le specifiche consentono di ridurre il numero di stati necessari. Poiché la posizione della lama dipende esclusivamente dalla posizione dei tasti, lo stato della lama è ridondante: per esempio, se entrambi i tasti sono premuti la lama deve essere abbassata. Quindi, le combinazioni degli stati dei sottosistemi lama e pulsanti di attivazione possono essere ridotti a 4: le combinazioni dei tasti.

Inoltre, le specifiche dicono che la lama non può tagliare due lamierini contemporaneamente. Quindi, lo stato che descrive tale situazione non può mai essere raggiunto. Al suo posto, va invece introdotto uno stato *errore*, utilizzato per catturare tutte le situazioni impossibili o irrealizzabili, quali, per esempio, il tentativo di posizionare più di due lamierini sul piano di lavoro. Inoltre, ciò significa che quando la lama è abbassata, il numero di lamierini sul piano diventa 0.

Quindi, l'insieme degli stati, Q, può essere:

$$\begin{aligned} Q \;=\; & \{d_0 s_0 l_0,\ d_0 s_0 l_1,\ d_0 s_0 l_2,\ d_0 s_1 l_0,\ d_0 s_1 l_1, \\ & d_0 s_1 l_2,\ d_1 s_0 l_0,\ d_1 s_0 l_1,\ d_1 s_0 l_2,\ d_1 s_1 l_0, \\ & \mathit{errore}\} \end{aligned}$$

dove le prime due lettere indicano i tasti (destro, d, e sinistro, s) e la terza lettera, l, indica lo stato del piano di lavoro. I numeri in pedice indicano lo stato dei tasti (rilasciato, 0, o premuto 1) e il numero di lamierini presenti sul piano (0, 1 o 2).

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 azioni che possono essere effettuate sono:

- il posizionamento di un lamierino;
- la rimozione di un lamierino;
- la pressione di un tasto (destro e sinistro);
- il rilascio di un tasto (destro e sinistro).

Le specifiche non chiariscono cosa succeda se si cerca di premere (o rilasciare) un tasto già premuto (o rilasciato). Si può ipotizzare che questa azione non abbia conseguenze. Allo stesso modo, si può ipotizzare che il tentativo di rimuovere un lamierino non presente non abbia effetto.

Discutibile invece l'effetto di far abbassare le lame a piano di lavoro vuoto. Le specifiche non danno indicazioni, e quindi pare ragionevole che tale situazione non sia causa di malfunzionamento,

ma l'utilizzo della macchina senza materiale appare quantomento uno spreco e come tale potrebbe essere descritto da un automa più sofisticato.

Può invece essere considerato pericoloso (oltreché impossibile per un solo operatore) tentare di posizionare o rimuovere i lamierini a lame abbassate.

L'insieme dei simboli, Σ, può essere:

$$\Sigma = \{p,\ r,\ d_{\text{on}},\ d_{\text{off}},\ s_{\text{on}},\ s_{\text{off}}\}$$

dove p e r rappresentano rispettivamente il posizionamento e la rimozione di un lamierino, mentre i restanti simboli rappresentano le azioni (pressione, *on*) sui tasti (destro, d, e sinistro s).

Ogni sequenza di azioni che non comporti il raggiungimento dello stato *errore* rappresenta il normale funzionamento della taglierina. 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 taglierina a riposo: $d_0 s_0 l_0$.

Con le ipotesi fatte, dovrebbero essere accettate, per esempio, le seguenti sequenze di azioni: $pd_{\text{on}}s_{\text{on}}$, $d_{\text{on}}d_{\text{off}}s_{\text{on}}pd_{\text{on}}r$, $ppd_{\text{on}}rs_{\text{on}}$. Al contrario, verrebbero rifiutate, tra le altre, le seguenti sequenze di azioni: ppp, $pd_{\text{on}}ps_{\text{on}}$. 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.

Una variante possibile può essere individuata osservando che le operazioni sui tasti (pressione e rilascio) sono mutuamente esclusive. Pertanto, potrebbero essere sostituite da una singola azione di pressione/rilascio il cui effetto sarebbe quello di rilasciare il tasto se premuto, e di premerlo se rilasciato.

Un'altra variante possibile riguarda l'insieme degli stati finali, F, il quale potrebbe essere ristretto alle sole situazioni in cui i tasti sono entrambi rilasciati.

Infine, va menzionata una variante più sofisticata, nella quale i lamierini tagliati non vengono automaticamente rimossi. In questo caso, lo stato del piano deve tener conto, oltre che del numero, anche della condizione dei lamierini, aggiungendo un stato in cui il lamierino è stato tagliato.

Esercizio 5

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

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

Individuare, motivando le risposte, quali fra le seguenti stringhe vengono descritte da E:

a) *bbbaacccc*

b) *bbbacbac*

c) *aac*

d) *bacccc*

e) *acaccc*

f) *acbac*

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: $E_1 = (b^*a + ac)^2$ 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.

- *bbbaacccc*
  $bbbaacccc = (b^3a)(ac)(c^3) \subseteq (b^*a)(ac)(c^3) \subseteq (b^*a + ac)^2(c^3 + b^*ac) \subseteq (b^*a + ac)^2(c^3 + b^*ac)^*$

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

  Spiegazione alternativa:

b	b	b	a	a	c	c	c	c
	b^3		a	ac			c^3	
	b^*		a	ac			$(c^3 + b^*ac)$	
	b^*a			ac			$(c^3 + b^*ac)^*$	
	$(b^*a + ac)^2$						$(c^3 + b^*ac)^*$	

$\boldsymbol{\delta}$	p	r	d_{on}	d_{off}	s_{on}	s_{off}
$d_0s_0l_0$	$d_0s_0l_1$	$d_0s_0l_0$	$d_1s_0l_0$	$d_0s_0l_0$	$d_0s_1l_0$	$d_0s_0l_0$
$d_0s_0l_1$	$d_0s_0l_2$	$d_0s_0l_0$	$d_1s_0l_1$	$d_0s_0l_1$	$d_0s_1l_1$	$d_0s_0l_1$
$d_0s_0l_2$	*errore*	$d_0s_0l_1$	$d_1s_0l_2$	$d_0s_0l_2$	$d_0s_1l_2$	$d_0s_0l_2$
$d_0s_1l_0$	$d_0s_1l_1$	$d_0s_1l_0$	$d_1s_1l_0$	$d_0s_1l_0$	$d_0s_1l_0$	$d_0s_0l_0$
$d_0s_1l_1$	$d_0s_1l_2$	$d_0s_1l_0$	$d_1s_1l_0$	$d_0s_1l_1$	$d_0s_1l_1$	$d_0s_0l_1$
$d_0s_1l_2$	*errore*	$d_0s_1l_1$	*errore*	$d_0s_1l_2$	$d_0s_1l_2$	$d_0s_0l_2$
$d_1s_0l_0$	$d_1s_0l_1$	$d_1s_0l_0$	$d_1s_0l_0$	$d_0s_0l_0$	$d_1s_1l_0$	$d_1s_0l_0$
$d_1s_0l_1$	$d_1s_0l_2$	$d_1s_0l_0$	$d_1s_0l_1$	$d_0s_0l_1$	$d_0s_1l_1$	$d_0s_0l_1$
$d_1s_0l_2$	*errore*	$d_1s_0l_1$	$d_1s_0l_2$	$d_0s_0l_2$	*errore*	$d_1s_0l_2$
$d_1s_1l_0$	*errore*	*errore*	$d_1s_1l_0$	$d_0s_1l_0$	$d_1s_1l_0$	$d_1s_0l_0$
errore	*errore*	*errore*	*errore*	*errore*	*errore*	*errore*

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

- *bbbacbac*
  Non esiste un prefisso di *bbbacbac* che possa essere descritto da E_1. Infatti, *bbba* è descritto da b^*a, ma il simbolo seguente, c, non può in alcun modo essere descritto né da b^*a, né da ac.

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

- *aac*
  $aac = (a)(ac) \subseteq (b^*a)(ac) \subseteq (b^*a + ac)^2 \subseteq (b^*a + ac)^2(c^3 + b^*ac)^*$

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

  Spiegazione alternativa:

a	a	c	
a	a	c	ϵ
b^*a		ac	$(c^3 + b^*ac)^*$
$(b^*a + ac)^2$			$(c^3 + b^*ac)^*$

- *bacccc*
  Non esiste un prefisso di *bacccc* che possa essere descritto da E_1. Infatti, *ba* è descritto da b^*a, ma il simbolo seguente, c, non può in alcun modo essere descritto né da b^*a, né da ac.

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

- *acaccc*
  $acaccc = (ac)(a)(ccc) \subseteq (ac)(b^*a)(c^3) \subseteq (b^*a + ac)^2(c^3) \subseteq (b^*a + ac)^2(c^3 + b^*ac) \subseteq (b^*a + ac)^2(c^3 + b^*ac)^*$

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

  Spiegazione alternativa:

a	c	a	c	c	c
	ac		a		c^3
	ac		b^*a		$(c^3 + b^*ac)$
	$(b^*a + ac)^2$				$(c^3 + b^*ac)$

- *acbac*

  Il prefisso *acba* può essere descritto da E_1, ma la stringa rimanente, c non può essere descritto da E_2.

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

Esercizio 6

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

- *bbabbca*
- *aacacaca*
- *bbbbabacbcb*
- *abbbbbbbbba*

ma non le seguenti:

- *caaac*
- *bcccabbc*
- *baaacacaca*
- *bbbaab*

Soluzione

Si può notare che tutte le stringhe da includere hanno nel suffisso sequenze di almeno 2 simboli a e b. Questa caratteristica può essere descritta dall'espressione regolare $(a + b)^2(a + b + c)^*$. Questa espressione regolare descrive tutte le stringhe del primo insieme, ma anche due stringhe del secondo insieme: *baaacacaca* e *bbbaab*. Queste ultime hanno un numero dispari di b, caratteristica che invece non possiedono le stringhe del primo insieme. Modificando l'espressione regolare precedente

in $(a+b^2)^2(a+b+c)^*$ si ottiene una stringa regolare che soddisfa le specifiche. Infatti, pur descrivendo tutte le stringhe del primo gruppo, nessuna delle stringhe del secondo gruppo viene descritta da tale espressione regolare:

- *caaac*: inizia per *c*;
- *bcccabbc*: inizia per *bc*;
- *baaacacaca*: inizia per *ba*;
- *bbbaab*: dopo una prima coppia di *b*, la stringa prosegue con *ba*.

Altre espressioni regolari che rispettano le specifiche:

- $(a^*b^2 + a)^2(a + b + c)^*$
- $(a + b)(b^*a)(bb + ca + ba + cb)^*$
- $(a + bb)^2(a + b + c)^*(bba + bcb + ca)$
- $(b^2 + a)(a + b + c)^*(a + cb)$