Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

Fondamenti di informatica per la sicurezza

anno accademico 2005–2006

docente: Stefano Ferrari

20.01.2006 — Soluzione del secondo compitino — vers. D

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:

- $L_1 = \{a, ba, ab\}$
- $L_2 = \{c, bc\}$

Descrivere i linguaggi:

a) $L_3 = L_2 \cap L_1$

b) $L_4 = L_1 \cup L_2$

c) $L_5 = L_1 L_2$

d) $L_6 = {L_2}^3$

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

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

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) $L_4 = L_1 \cup L_2 = \{a, ba, ab, c, bc\}$

c) $L_5 = L_1 L_2 = \{ac, abc, bac, babc, abbc\}$
L'elemento *abc* può essere ottenuto in due modi (*a-bc* e *ab-c*), ma deve essere riportato solo una volta.

d) $L_6 = {L_2}^3 = \{ccc, ccbc, cbcc, cbcbc, bccc, bccbc, bcbcc, bcbcbc\}$

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 $\{\epsilon, babaaab, cccbcc, aaccbc\}$ è un sottoinsieme di L_7.

f) $L_8 = (L_2 {L_1}^2)^*$
L'insieme L_8 è formato dalla concatenazione di un numero arbitrario (eventualmente nullo) di stringhe composte concatenando un elemento di L_1 con due elementi di L_1. Pertanto, L_8 è composto da infiniti elementi. L'insieme $\{\epsilon, caa, bcabacabba\}$ 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: $P = \{S ::= H,\ K ::= b|aH|cH,\ H ::= d|cK|bH\}$

Quali fra le seguenti stringhe vengono generate da G?

a) *cabdc*

b) *cabbda*

c) *bcabd*

d) *bcac*

e) *cabcb*

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)

cabdc	
	S
$S ::= H$	H
$H ::= cK$	cK
$K ::= aH$	caH
$H ::= bH$	$cabH$
$H ::= d$	$cabd$

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

La stringa *cabdc* non è generata da G: $cabdc \notin L(G)$.

b)

cabbda	
	S
$S ::= H$	H
$H ::= cK$	cK
$K ::= aH$	caH
$H ::= bH$	$cabH$
$H ::= bH$	$cabbH$
$H ::= d$	$cabbd$

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

La stringa *cabbda* non è generata da G: $cabbda \notin L(G)$.

c)

bcabd	
	S
$S ::= H$	H
$H ::= bH$	bH
$H ::= cK$	bcK
$K ::= aH$	$bcaH$
$H ::= bH$	$bcabH$
$H ::= d$	$bcabd$

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

d)

bcac	
	S
$S ::= H$	H
$H ::= bH$	bH
$H ::= cK$	bcK
$K ::= aH$	$bcaH$
$H ::= cK$	$bcacK$

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

La stringa *bcac* non è generata da G: $bcac \notin L(G)$.

e)

cabcb	
	S
$S ::= H$	H
$H ::= cK$	cK
$K ::= aH$	caH
$H ::= bH$	$cabH$
$H ::= cK$	$cabcK$
$K ::= b$	$cabcb$

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

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

- 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:

- *ecabac*
- *ddea*
- *cede*
- *badc*

b) quattro stringhe rifiutate da A:

- *aed*
- *ddab*
- *aa*
- *ebdce*

Esercizio 4

Modellare, tramite un automa a stati finiti deterministico, il funzionamento di un trapano a colonna.

Un trapano a colonna è un trapano fissato ad una infrastruttura che gli permette solo movimenti verticali. Una apposita morsa permette di bloccare i pezzi da lavorare in modo che si trovino proprio sulla corsa del trapano.

Il trapano può essere posizionato in alto (posizione di riposo) o in basso (posizione di lavoro). Il trapano è dotato di punte intercambiabili.

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

Stati e simboli riportati 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.

Il sistema del problema è descrivibile come insieme di tre sottosistemi indipendenti: posizione trapano, morsa, punta. Il trapano può trovarsi in posizioni stati (alto e basso), la morsa in due stati (vuota, con pezzo), e la punta in due stati (inserita e assente).

Ai sottosistemi esplicitamente citati nelle specifiche si potrebbe aggiungere il sottosistema motore che può essere in moto o fermo, a seconda che il trapano sia attivo o meno e un pulsante per attivare o disattivare il motore. Tuttavia, non è necessario arrivare a questo livello di dettaglio. Basterà ipotizzare che il trapano si attivi quando portato in posizione bassa e si disattivi quando portato in posizione alta.

Gli stati del trapano si possono derivare dalle combinazioni degli stati dei tre sottosistemi. Pertanto, l'insieme degli stati, Q, può essere:

$$Q = \{avi,\ avd,\ api,\ apd,\ bvi,\ bvd,\ bpi,\ bpd\}$$

dove la prima lettera indica se il trapano è alto, a, o basso, b, la seconda lettera indica che il pezzo da lavorare è presente o no (morsa vuota, v, o con pezzo, p) e la terza lettera indica se la punta è presente (inserita, i) oppure no (disinserita, d).

Le azioni che possono essere effettuate sul sistema sono l'alzamento e l'abbassamento del trapano, fissaggio e il rilascio del pezzo da lavorare e l'inserimento e il disinserimento della punta.

Pertanto, insieme dei simboli, Σ, può essere:

$$\Sigma = \{a,\ b,\ f,\ r,\ i,\ d\}$$

dove a e b indicano, rispettivamente, l'azione di posizionamento del trapano (alto, a, e basso, b), f e r indica il fissaggio e il rilascio del pezzo, mentre i e d indicano le azioni sulla punta (inserimento, i, e disinserimento, d).

Le specifiche non descrivono particolari condizioni di funzionamento. Tuttavia, alcune considerazioni appaiono ragionevoli:

- inserire o togliere un pezzo o una punta quando il trapano è in posizione di lavoro va considerato un errore;
- inserire (togliere) un pezzo o una punta in presenza (assenza) di una punta o di un pezzo va considerato errore;
- abbassare (alzare) il trapano quando già si trova in posizione bassa (alta) non ha effetto;

- abbassare il trapano senza punto o senza il pezzo va considerato errore;

Poiché esiste almeno una situazione ritenuta non accettabile, è opportuno aggiungere all'insieme Q un altro stato, *errore*, 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.

Ogni sequenza di azioni che non comporti il raggiungimento dello stato *errore* rappresenta il normale funzionamento del trapano a colonna. 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 trapano in posizione di riposo, senza pezzo, né punta, *avd*.

Con le ipotesi fatte, dovrebbero essere accettate, per esempio, le seguenti sequenze di azioni: $ifba$, $firfbadr$. Al contrario, verrebbero rifiutate, tra le altre, le seguenti sequenze di azioni: bi, $ifbr$, $fibd$. 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.

Dalle assunzioni fatte, deriva che gli stati in cui il trapano è abbassato ma senza pezzo o senza punta non verranno mai raggiunti. In tal senso, è possibile eliminarl dall'automa senza modificare l'insieme delle stringhe accettate. Quindi, eliminando gli stati del tipo *bv-* e *b-d*, si ottiene la tabella delle trasizioni riportata in Tabella 2.

Considerando poi che i sottosistemi sono tutti di tipo binario, le azioni possono essere semplificate. Per esempio, le azioni a e b possono essere sostituite da una sola azione, ab, che ha l'effetto di commutare la posizione del trapano: se è basso lo porta in alto, se è in alto lo porta in basso. Analogamente, introducendo le azioni fr e id al posto dei rimanenti simboli, si può ottenere la tabella delel transizioni riportata in Tabella 3.

Esercizio 5

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

- $E = (b^*ac^2)^* + (ab^*ca)^*$

Quali fra le seguenti stringhe vengono descritte da E?

a) *abbbcac*

b) *bbbaccacc*

c) *accbacc*

d) *acaacaabbca*

e) *abbcaaca*

f) *bbbacca*

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 costituita dall'unione di due sottoespressioni: $E_1 = (b^*ac^2)^*$ e $E_2 = (ab^*ca)^*$. Quindi, le stringhe descritte da E dovranno essere descritte da E_1 o, in alternativa, da E_2. Questa premessa semplificherà la spiegazione delle soluzioni di seguito riportate.

a) *abbbcac*

   L'espressione E_1 descrive stringhe che terminano con due simboli c, mentre E_2 descrive stringhe che iniziano e terminano per a. La stringa data non rientra in nessuna di queste due caratteristiche.

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

b) *bbbaccacc*
   $bbbaccacc = (bbbacc)(acc) = (b^3ac^2)(ac^2) \subseteq (b^*ac^2)(b^*ac^2) \subseteq (b^*ac^2)^2 \subseteq (b^*ac^2)^* \subseteq (b^*ac^2)^* + (ab^*ca)^*$

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

$\boldsymbol{\delta}$	a	b	f	r	i	d
avi	avi	$errore$	api	$errore$	$errore$	avd
avd	avd	$errore$	apd	$errore$	avi	$errore$
api	api	bpi	$errore$	avi	$errore$	apd
apd	apd	$errore$	$errore$	avd	api	$errore$
bvi	avi	$errore$	$errore$	$errore$	$errore$	$errore$
bvd	avd	$errore$	$errore$	$errore$	$errore$	$errore$
bpi	api	bpi	$errore$	$errore$	$errore$	$errore$
bpd	apd	$errore$	$errore$	$errore$	$errore$	$errore$
$errore$	$errore$	$errore$	$errore$	$errore$	$errore$	$errore$

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

$\boldsymbol{\delta}$	a	b	f	r	i	d
avi	avi	$errore$	api	$errore$	$errore$	avd
avd	avd	$errore$	apd	$errore$	avi	$errore$
api	api	bpi	$errore$	avi	$errore$	apd
apd	apd	$errore$	$errore$	avd	api	$errore$
bpi	api	bpi	$errore$	$errore$	$errore$	$errore$
$errore$	$errore$	$errore$	$errore$	$errore$	$errore$	$errore$

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

$\boldsymbol{\delta}$	ab	fr	id
avi	$errore$	api	avd
avd	$errore$	apd	avi
api	bpi	avi	apd
apd	$errore$	avd	api
bpi	api	$errore$	$errore$
$errore$	$errore$	$errore$	$errore$

Tabella 3: Tabella delle transizioni dell'automa dell'esercizio 4 dopo la semplificazione delle azioni.

c) *accbacc*
$accbacc = (acc)(bacc) = (ac^2)(bac^2) \subseteq (b^*ac^2)(b^*ac^2) \subseteq (b^*ac^2)^2 \subseteq (b^*ac^2)^* \subseteq (b^*ac^2)^* + (ab^*ca)^*$

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

d) *acaacaabbca*
$acaacaabbca = (aca)(aca)(abbca) = (aca)(aca)(ab^2ca) \subseteq (ab^*ca)(ab^*ca)(ab^*ca) \subseteq (ab^*ca)^3 \subseteq (ab^*ca)^* \subseteq (b^*ac^2)^* + (ab^*ca)^*$

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

e) *abbcaaca*
$abbcaaca = (abbca)(aca) = (ab^2ca)(aca) \subseteq (ab^*ca)(ab^*ca) \subseteq (ab^*ca)^2 \subseteq (ab^*ca)^* \subseteq (b^*ac^2)^* + (ab^*ca)^*$

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

f) *bbbacca*

La stringa data inizia per b, e quindi non può essere descritta da E_2. Poiché termina per a, non può neanche essere descritta da E_1.

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

Esercizio 6

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

- *ccacacc*
- *acaacacab*
- *cbbcacaabc*
- *abab*

ma non le seguenti:

- *caacbac*
- *acaabb*
- *cccacca*
- *baccaab*

Soluzione

Le stringhe da accettare hanno nella parte terminale le stringhe c e ab. In particolare, almeno due occorrenze di queste stringhe costituiscono il suffisso delle stringhe da accettare. Infatti:

- *ccacacc*: termina con due occorrenze di c;
- *acaacacab*: termina con c seguita da ab;
- *cbbcacaabc*: termina con ab seguita da c;
- *abab*; termina con (anzi è composta da) due occorrenze di ab.

Questa caratteristica può essere descritta dall'espressione regolare $(a + b + c)^*(c + ab)^2$.

Tutte le stringhe del secondo gruppo non vengono descritte da tale espressione regolare:

- *caacbac*: termina con c, ma è preceduta da ba;
- *acaabb*: termina con bb;
- *cccacca*: termina con a;
- *baccaab*: termina con a.