Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

Fondamenti di informatica per la sicurezza

anno accademico 2005–2006

docente: Stefano Ferrari

24.01.2006 — 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:

- $L_1 = \{z, xz, zx\}$
- $L_2 = \{c, xc\}$

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) $L_3 = L_2 \cap L_1 = \emptyset$
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) $L_4 = L_1 \cup L_2 = \{c, xc, xz, z, zx\}$

c) $L_5 = L_1 L_2 = \{xzc, xzxc, zc, zxc, zxxc\}$
Gli elementi che possono essere ottenuti in più di un modo devono essere riportati solo una volta.

d) $L_6 = L_2^3 = \{ccc, cxcxc, xccc, xcxcxc\}$
Gli elementi che possono essere ottenuti in più di un modo devono essere riportati solo una volta.

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 $\{\epsilon, zxzxz, cxcxc, zxzxc\}$ 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 da un elemento di L_2 e da due elementi di L_1. Pertanto, L_8 è composto da infiniti elementi. L'insieme $\{\epsilon, cxzxz, czzxczxzcxzz\}$ 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) *bcaa*

c) *cabbd*

d) *bccc*

e) *bcad*

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: $cabdc \notin \mathcal{L}(G)$.

b) bcaa NO

bcaa	
	S
$S ::= H$	H
$H ::= bH$	bH
$H ::= cK$	bcK
$K ::= aH$	$bcaH$

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

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

c)

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

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

d)

bccc	
	S
$S ::= H$	H
$H ::= bH$	bH
$H ::= cK$	bcK
$K ::= cH$	$bccH$
$H ::= cK$	$bcccK$

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

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

e)

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

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

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

- *ebdce*
- *dece*
- *ddac*
- *ecaba*

b) quattro stringhe rifiutate da A:

- *decc*
- *bacaa*
- *aebe*
- *abcd*

Esercizio 4

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

Per annullare un francobollo, l'impiegato di un ufficio postale pone la busta sul tavolo e, batte il timbro (opportunamente intinto nell'inchiostro) sul francobollo stesso. Dopo un'inchiostratura, il timbro può essere usato efficacemente al massimo tre volte. Ipotizzare che, ai fini della timbratura, una busta non possa essere posta sopra un'altra e che battere il timbro senza una busta sottostante rovini il timbro stesso.

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 timbro. In particolare, individuare possibili situazioni fisicamente irrealizzabili o pericolose e formalizzarle in modo che l'automa rifiuti le successioni di azioni che porterebbero il timbro 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 sottosistemi che in prima ipotesi possono essere indipendenti: il timbro e la scrivania. Il timbro può trovarsi in quattro stati diversi, a seconda del numero di timbrature che il suo livello di inchiostratura gli permette (da 0 a 3). La scrivania, invece si può trovare in diversi stati a seconda del numero di buste presenti su di essa. Ai fini della modellizzazione richiesta dalle specifiche, ha senso considerare solo la scrivania vuota, con una busta o con due o più buste.

Volendo dare una descrizione dettagliata del sistema da modellare, si potrebbe anche tener conto dello stato di timbratura delle buste, ma ciò non verrà considerato in un primo momento.

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

$$Q = \{v0,\ v1,\ v2,\ v3,\ b0,\ b1,\ b2,\ b3,\ m0,\ m1,\ m2,\ m3\}$$

dove la lettera indica se lo stato della scrivania (vuota, v, con una busta, b, o con un mucchio di buste, m), mentre il numero indica il numero di timbrature ancora possibili. Per esempio, lo stato *v2* indica che la scrivania è vuota e che il timbro porta abbastanza inchiostro per effettuare ancora due timbrature.

Le azioni che possono essere effettuate sul sistema sono l'inchiostratura del timbro (i), la timbratura della busta (t), e il posizionamento di una busta (b). Non è invece necessario modellare l'azione di rimozione di una busta, in quanto si può supporre che la busta timbrata sia rimossa immediatamente dopo l'azione di timbratura.

Pertanto, insieme dei simboli, Σ, può essere:

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

Le specifiche descrivono i seguenti comportamenti:

- l'inchiostratura del timbro gli consente tre timbrature;
- la timbratura non può essere effettuata se sulla scrivania ci sono più buste;
- eseguire la timbratura sulla scrivania vuota rovina il timbro.

Le specifiche non impongono ulteriori particolari condizioni. Tuttavia, è ragionevole supporre che inchiostrare un timbro che ha ancora abbastanza inchiostro per almeno un'altra timbratura, porta il livello di inchiostro a tre timbrature. Quindi, inchiostrare due volte di seguito un timbro

non è un errore e permette comunque solo tre timbrature.

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 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 scrivania vuota, con il timbro senza inchiostro, *v0*.

Con le ipotesi fatte, dovrebbero essere accettate, per esempio, le seguenti sequenze di azioni: *ibtbti*, *ibtbtbtbit*. Al contrario, verrebbero rifiutate, tra le altre, le seguenti sequenze di azioni: *bt*, *bbit*, *it*. 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.

Alcune ipotesi semplificative possono essere fatte. Per esempio, poiché una volta raggiunto uno stato del tipo *m-* non si può fare altro che rimanere in uno stato di quel tipo o raggiungere lo stato *errore*, gli stati *m-* potrebbero essere eliminiati e sostituiti dallo stato *errore* all'interno della tabella delle transizioni.

È inoltre possibile rendere l'automa più complesso, modellando, per esempio anche l'azione di rimozione della busta timbrata (per esempio attraverso un opportuno simbolo di input, r). Ciò richiede l'introduzione degli stati necessari a tener conto dello stato della busta. Per esempio, gli stati *t0–t3* possono essere introdotti a questo scopo. Quando l'automa raggiunge uno di questi stati e legge il simbolo r, si porta nel corrispondente stato *v-*, mentre se legge il simbolo b, si porta nel corrispondente stato *m-* e se legge il simbolo i, si porta nello stato *t3*.

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) *bbaccacc*

b) *acabbab*

c) *ccacbacc*

d) *bcaabcca*

e) *aacbcabb*

f) *acaabbbca*

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 l'alternativa tra due sottoespressioni: $E_1 = (b^*ac^2)^*$ e $E_2 = (ab^*ca)^*$. Quindi, le stringhe descritte da E dovranno obbligatoriamente essere descritte o da E_1 o da E_2. L'espressione E_1 descrive stringhe che terminano con *acc*, mentre l'espressione E_2 descrive stringhe che iniziano per a e terminano per *ca*. Questa premessa semplificherà la spiegazione delle soluzioni di seguito riportate.

a) *bbaccacc*

   $bbaccacc = (bb)(a)(cc)(a)(cc) \subseteq (b^2)(a)(c^2)b^*(a)(c^2) \subseteq (b^*)(a)(c^2)b^*(a)(c^2) = (b^*ac^2)^2 \subseteq (b^*ac^2)^* \subseteq (b^*ac^2)^* + (ab^*ca)^*$

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

$\boldsymbol{\delta}$	i	t	b
v0	*v3*	*errore*	*b0*
v1	*v3*	*errore*	*b1*
v2	*v3*	*errore*	*b2*
v3	*v3*	*errore*	*b3*
b0	*b3*	*errore*	*m0*
b1	*b3*	*v0*	*m1*
b2	*b3*	*v1*	*m2*
b3	*b3*	*v2*	*m3*
m0	*m3*	*errore*	*m0*
m1	*m3*	*errore*	*m1*
m2	*m3*	*errore*	*m2*
m3	*m3*	*errore*	*m3*
errore	*errore*	*errore*	*errore*

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

b) *acabbab*
La stringa data termina per b.

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

c) *ccacbacc*
L'espressione E_1 non può descrivere stringhe che hanno simboli c isolati (*cca*$\underline{c}$*bacc*), e l'espressione E_2 non può descrivere stringhe che terminano per b.

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

d) *bcaabcca*
La stringa data non può venire descritta dall'espressione E_1 perché termina per a, e non può essere descritta da E_2 perché non inizia per a.

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

e) *aacbcabb*

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

f) *acaabbbca*
$acaabbbca = (a)(ca)(a)(bbb)(ca) \subseteq (a)b^*(ca)(a)(b^3)(ca) \subseteq (a)b^*(ca)(a)(b^*)(ca) = (ab^*ca)^2 \subseteq (ab^*ca)^* \subseteq (b^*ac^2)^* + (ab^*ca)^*$

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

Esercizio 6

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

- *bbbcca*
- *acbacbaba*
- *bacac*
- *acacbcca*

ma non le seguenti:

- *cccbaba*
- *cabcaba*
- *acac*
- *bbabcac*

Soluzione

Si può notare che tutte le stringhe da accettare hanno iniziano per b o per ac e che tale schema si ripete più volte.

L'espressione regolare $(b + ac)$ descrive il prefisso di tutte le stringhe da includere (*b*$\underline{b}$*bcca*, *ac*$\underline{a}$*cbacbaba*, *b*$\underline{a}$*cac* e *ac*$\underline{a}$*cacbcca*), ma anche quello di due stringhe da escludere (*ac*$\underline{a}$*c* e *b*$\underline{b}$*abcac*). Ripetendo due volte questa espressione regolare ($(b + ac)^2$) si descrive ancora il prefisso delle stringhe da includere (*bb*$\underline{b}$*bcca*, *acb*$\underline{a}$*cbacbaba*, *bac*$\underline{a}$*c* e *acac*$\underline{b}$*cca*), ma anche quello delle due stringhe da escludere (*ac*$\underline{a}$*c* e *bb*$\underline{b}$*abcac*). Invece, ripetendo tre volte l'espressione regolare considerata ($(b+ac)^3$) si descrive ancora il prefisso delle stringhe da includere (*bbb*$\underline{b}$*cca*, *acbac*$\underline{b}$*aba*, *bacac* e *acacb*$\underline{c}$*ca*),

ma non più quello delle due stringhe da escludere (*acac* è ora troppo breve e *bb*$\underline{ab}$*cac* non ha *b* o *ac* come terza sottostringa).

Quindi, $(b + ac)^3(a + b + c)^*$ descrive tutte le stringhe da includere e nessuna di quelle da escludere.

Altre espressioni regolari che rispettano le specifiche del problema sono:

- $(b + ac)^3(c^2a + ba)^*$;
- $a(a + b + c)^*a + bb^*(a + c)^*$;
- $b(b + c + ac + ca)^* + a(a + b + c)^*a$;
- $b(b + c + ac + ca)^* + a(a + b + c)^*a$;
- $b(b^2c^2 + ac^*)^2 + aca^*(ba + c^*a + cb)^*$;
- $(b2 + ac)(a + b + c)^*a + ba(a + b + c)^*c$;
- $(a + b)((a + b + c)^*ca + (ac + cb + ba)^*)$;
- $(a + b)(a + b + c)^*a + b(a + c)^*$;
- $a(a + b + c)^*a + bb^*(a + c)^*$;
- $(ac)^*(b^*c^*a) + (a + b)(c^*ba + ac)^*$.