Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

Fondamenti di informatica per la sicurezza

anno accademico 2005–2006

docente: Stefano Ferrari

03.02.2006 — 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:

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

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) $L_6 = {L_2}^2$

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

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

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

c) $L_5 = L_1 L_2 = \{ax, ay, bax, bay, bx, by\}$

d) $L_6 = L_2^2 = \{xx, xy, yx, yy\}$

e) $L_7 = L_2^* L_1^*$
L'insieme L_7 è dato dalle stringhe formate come concatenazione di un numero arbitrario (eventualmente nullo) di elementi di L_2 seguito da una concatenazione di un numero arbitrario (eventualmente nullo) di elementi di L_1. Poiché sia ${L_1}^*$ che ${L_2}^*$ sono composti da infiniti elementi, anche L_7 avrà infiniti elementi. L'insieme $\{\epsilon, bbbaa, xyy, ybaa\}$ è un sottoinsieme di L_7.

f) $L_8 = (L_1 L_2)^*$
L'insieme L_8 è formato dalla concatenazione di un numero arbitrario (eventualmente nullo) di stringhe composte da un elemento di L_1 e da un elemento di L_2. Pertanto, L_8 è composto da infiniti elementi. L'insieme $\{\epsilon, by, axbaxay\}$ 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: $P = \{S ::= H,\ K ::= a|Hb|Hc,\ H ::= c|Ka|Hd\}$

Quali fra le seguenti stringhe vengono generate da G?

a) *aadd*

b) *bbad*

c) *adba*

d) *aadca*

e) *dcca*

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) *aadd*

	S
$S ::= H$	H
$H ::= Hd$	Hd
$H ::= Hd$	Hdd
$H ::= Ka$	$Kadd$
$K ::= a$	$aadd$

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

b) *bbad*

	S
$S ::= H$	H
$H ::= Hd$	Hd
$H ::= Ka$	Kad
$K ::= Hb$	$Hbad$
$H ::= b$	$bbad$

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

c) *adba*

	S
$S ::= H$	H
$H ::= Ka$	Ka
$K ::= Hb$	Hba
$H ::= Hd$	$Hdba$
$H ::= Ka$	$Kadba$

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

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

d) *aadca*

	S
$S ::= H$	H
$H ::= Ka$	Ka
$K ::= Hc$	Hca
$H ::= Hd$	$Hdca$
$H ::= Ka$	$Kadca$
$K ::= a$	$aadca$

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

e) *dcca*

	S
$S ::= H$	H
$H ::= Ka$	Ka
$K ::= Hc$	Hca
$H ::= c$	cca

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

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

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

- 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:

- *ecaba*
- *decc*
- *aebe*
- *abcd*

b) quattro stringhe rifiutate da A:

- *ebdce*
- *dece*
- *ddac*
- *bacaa*

Esercizio 4

Modellare, tramite un automa a stati finiti deterministico, il funzionamento di una macchina per le granite.

Una macchina per le granite è dotata di un serbatoio, di una pala mescolatrice, di un impianto di raffreddamento e di un rubinetto. Nel normale funzionamento, l'utente pone nel serbatoio gli ingredienti (acqua e sciroppo), chiude il coperchio e poi attiva la preparazione della granita mediante l'apposito tasto d'avvio. Dopo l'avvio, si attiva l'impianto di raffreddamento e la pala inizia a mescolare gli ingredienti. Il rubinetto consente di prelevare le granite anche con la macchina in funzione. Il serbatoio contiene fino a tre porzioni.

La macchina è dotata di due sensori: uno rileva il livello degli ingredienti, e l'altro l'apertura del coperchio. Se il coperchio viene aperto durante l'attività della pala, la pala viene arrestata e il rimescolamento riprende alla successiva chiusura del coperchio. Se, non viene rilevata la presenza di ingredienti, la pala e l'impianto di raffreddamento vengono fermati.

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

Il sistema del problema è descrivibile come insieme di sottosistemi che inizialmente possono essere ipotizzati indipendenti: l'alimentazione, il coperchio, il serbatorio, la pala e il dispositivo di raffreddamento. L'alimentazione può trovarsi in due stati (attiva e disattivata), il coperchio può trovarsi in due stati (aperto o chiuso), il serbatoio può trovarsi in quattro stati (vuoto, oppure una, due o tre porzioni), la pala può essere attiva o ferma (2 stati) e il raffreddamento attivo o disattivo (2 stati). Quindi, se tutti i sottosistemi fossero indipendenti, l'automa necessiterebbe di $2 \times 2 \times 4 \times 2 \times 2 = 64$ stati.

Risulta evidente che, al fine di ridure il numero di stati necessari per modellare il sistema in esame, è necessario individuare delle situazioni di dipendenza tra i sottosistemi o introdurre delle ipotesi semplificative (o entrambe le soluzioni).

Dalle specifiche si evince che lo stato della pala dipende dallo stato del coperchio: non è consentita la situazione in cui la pala è attiva e il coperchio è aperto. Inoltre, sia la pala, sia il raffreddamento, sono dipendenti dall'alimentazione: non è possibile che la pala si muova o che il dispositivo di raffreddamento sia attivo quando l'alimentazione non viene fornita alla macchina. Bisogna notare anche che il raffreddamento viene attivato quando la macchina viene avviata (cioè quando la alimentazione è attiva). Infine, pala e raffreddamento vengono disattivati quando il serbatoio è vuoto.

Queste specifiche consentono di ridurre il numero di stati necessario alla formalizzazione. In un primo momento, è utile notare che la modellazione del serbatoio può essere semplificata, in quanto la funzionalità principale si caratterizza solo per l'assenza degli ingredienti. Si può quindi ipotizzare che gli ingredienti vengano inseriti o prelevati solo in quantità pari a tre porzioni (cioè il serbatoio viene riempito completamente o svuotato completamente).

Gli stati della macchina sono quindi: Pertanto, l'insieme degli stati, Q, può essere:

$$Q = \{dcdd0,\ dcdd3,\ dadd0,\ dadd3,\ acdd0,\ acaa3,\ aadd0,\ aada3\}$$

dove le lettere che costituiscono la stringa identificativa degli stati indicano, rispettivamente, lo stato dell'alimentazione, del coperchio, della pala, del raffreddamento e del serbatoio. In particolare, l'alimentazione, la pala e il raffreddamento possono essere attivi (a) o disattivi (d), il coperchio può essere aperto (a) o chiuso (c), e il serbatoio può essere vuoto (0) o pieno (3). Per indicare il serbatoio è stato usato il numero di porzioni in esso contenute.

Le azioni che possono essere effettuate sul sistema sono l'apertura e la chiusura del coperchio, l'introduzione degli ingredienti nel serbatoio, il prelevamento della granita e l'attivazione/disattivazione della macchina.

Pertanto, insieme dei simboli, Σ, può essere:

$$\Sigma = \{a, c, i, p, o, d\}$$

dove a e c indicano, rispettivamente, l'azione di apertura e di chiusura del coperchio, mentre i indica l'introduzione degli ingredienti, p il prelevamento delle granite, o l'attivazione e d la disattivazione della macchina.

Pare inoltre ragionevole introdurre le seguenti specifiche:

- l'introduzione degli ingredienti può avvenire solo a coperchio aperto;
- l'introduzione di ingredienti quando il serbatoio è già pieno, causa un malfuzionamento;
- il tentativo di apertura (chiusura) del coperchio già aperto (chiuso) è causa di malfunzionamento;
- il prelevamento delle granite quando il serbatoio è vuoto non ha alcun effetto;
- attivare la macchina già attiva e disattivarla quando è già disattiva, non ha alcun effetto.
- attivare la macchina a coperchio aperto attiva solo il raffreddamento.

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 della macchina per granite. 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 serbatoio scarico e macchina disattivata con coperchio chiuso, *dcdd0*.

Con le ipotesi fatte, dovrebbero essere accettate, per esempio, le seguenti sequenze di azioni: *aico*, *acoaicpp*. Al contrario, verrebbero rifiutate, tra le altre, le seguenti sequenze di azioni: *i*, *aici*, *aii*, *aicc*. 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, $\delta : Q \times \Sigma \rightarrow Q$ può essere quella riportata in Tabella 1.

L'insieme dei simboli di input potrebbe essere ridotto notando che le operazioni di apertura e chiusura del coperchio e di attivazione e disattivazione della macchina sono tra loro complementari e potrebbero essere sostituite da singole azioni più generiche di mobilitazione del coperchio (se il coperchio è chiuso, viene aperto, mentre se è aperto, viene chiuso) e di attivazione/disattivazione della macchina. In tal modo, però, verrebbe persa la modellazione dell'errore che si commette cercando di aprire (chiudere) il coperchio già aperto (chiuso).

La rimozione dell'ipotesi semplificativa, estende l'insieme degli stati aggiungendo gli stati relativi alle situazioni in cui nel serbatoio sono present gli ingredienti per 1 e 2 porzioni:

$$Q = \{dcdd0, dcdd1, dcdd2, dcdd3, dadd0, dadd1, dadd2, dadd3, acdd0, acaa1, acaa2, acaa3, aadd0, aada1, aada2, aada3\}$$

In tal caso, l'operazione di introduzione ingredienti e prelevamento granite si riferisce a quantità relative ad una sola porzione per volta e la tabella delle transizioni, $\delta : Q \times \Sigma \rightarrow Q$ può essere quella riportata in Tabella 2.

Esercizio 5

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

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

Quali fra le seguenti stringhe vengono descritte da E?

a) *ccccaaa*

b) *ccabaa*

c) *ababbcba*

d) *abccaca*

e) *cccbba*

f) *acccbccba*

$\boldsymbol{\delta}$	a	c	i	p	o	d
dcdd0	*dadd0*	*errore*	*errore*	*dcdd0*	*acdd0*	*dcdd0*
dcdd3	*dadd3*	*errore*	*errore*	*dcdd0*	*acaa3*	*dcdd3*
dadd0	*errore*	*dcdd0*	*dadd3*	*dadd0*	*aadd0*	*dadd0*
dadd3	*errore*	*dcdd3*	*errore*	*dadd0*	*aada3*	*dadd3*
acdd0	*aadd0*	*errore*	*errore*	*acdd0*	*acdd0*	*dcdd0*
acaa3	*aada3*	*errore*	*errore*	*acdd0*	*acaa3*	*dcdd3*
aadd0	*errore*	*acdd0*	*aada3*	*aadd0*	*aadd0*	*dadd0*
aada3	*errore*	*acaa3*	*errore*	*aadd0*	*aada3*	*dadd3*
errore	*errore*	*errore*	*errore*	*errore*	*errore*	*errore*

Tabella 1: Tabella delle transizioni dell'automa semplificato dell'esercizio 4. Il nome degli stati indica, rispettivamente, lo stato dell'alimentazione, del coperchio, della pala, del raffreddamento e del serbatoio.

$\boldsymbol{\delta}$	a	c	i	p	o	d
dcdd0	*dadd0*	*errore*	*errore*	*dcdd0*	*acdd0*	*dcdd0*
dcdd1	*dadd1*	*errore*	*errore*	*dcdd0*	*acaa1*	*dcdd1*
dcdd2	*dadd2*	*errore*	*errore*	*dcdd1*	*acaa2*	*dcdd2*
dcdd3	*dadd3*	*errore*	*errore*	*dcdd2*	*acaa3*	*dcdd3*
dadd0	*errore*	*dcdd0*	*dadd1*	*dadd0*	*aadd0*	*dadd0*
dadd1	*errore*	*dcdd1*	*dadd2*	*dadd0*	*aada1*	*dadd1*
dadd2	*errore*	*dcdd2*	*dadd3*	*dadd1*	*aada2*	*dadd2*
dadd3	*errore*	*dcdd3*	*errore*	*dadd2*	*aada3*	*dadd3*
acdd0	*aadd0*	*errore*	*errore*	*acdd0*	*acdd0*	*dcdd0*
acaa1	*aada1*	*errore*	*errore*	*acdd0*	*acaa1*	*dcdd1*
acaa2	*aada2*	*errore*	*errore*	*acaa1*	*acaa2*	*dcdd2*
acaa3	*aada3*	*errore*	*errore*	*acaa2*	*acaa3*	*dcdd3*
aadd0	*errore*	*acdd0*	*aada1*	*aadd0*	*aadd0*	*dadd0*
aada1	*errore*	*acaa1*	*aada2*	*aadd0*	*aada1*	*dadd1*
aada2	*errore*	*acaa2*	*aada3*	*aada1*	*aada2*	*dadd2*
aada3	*errore*	*acaa3*	*errore*	*aada2*	*aada3*	*dadd3*
errore	*errore*	*errore*	*errore*	*errore*	*errore*	*errore*

Tabella 2: Tabella delle transizioni dell'automa dell'esercizio 4. Il nome degli stati indica, rispettivamente, lo stato dell'alimentazione, del coperchio, della pala, del raffreddamento e del serbatoio.

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: $E_1 = (ab + c^2)^*$ e $E_2 = (a + bc^*b)^2$. Quindi, le stringhe descritte da E dovranno obbligatoriamente avere un suffisso descritto da E_2 eventualmente preceduto da un prefisso descritto da E_1 (poiché E_1 descrive anche la stringa vuota). Questa premessa semplificherà la spiegazione delle soluzioni di seguito riportate.

a) *ccccaaa*
Il suffisso *aa* può essere descritto da E_2, ma la sottostringa rimanente, *cccca*, termina per a e non può quindi essere descritta da E_1.

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

b) *ccabaa*
$ccabaa = (cc)(ab)(a)(a) = (c^2)(ab)(a)(a) \subseteq (c^2 + ab)^2(a)(a) \subseteq (c^2 + ab)^*(a)(a) \subseteq (c^2 + ab)^*(a+bc^*b)(a+bc^*b) = (ab+c^2)^*(a+bc^*b)^2$

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

c) *ababbcba*
$ababbcba = (ab)(ab)(bcb)(a) \subseteq (ab+c^2)(ab+c^2)(bc^*b)(a) \subseteq (ab + c^2)^2(bc^*b + a)^2 \subseteq (ab + c^2)^*(a + bc^*b)^2$

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

d) *abccaca*
L'espressione regolare E_2 non può descrivere stringhe in cui il simbolo c non sia delimitato da simboli b. Quindi, la stringa data non ha suffissi che siano descrivibili da E_2 ($abcca\underline{c}a$).

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

e) *cccbba*
Il suffisso *bba* può essere descritto da E_2, ma la stringa rimanente, *ccc* non può essere descritta da E_1. Infatti, E_1 non è in grado di descrivere sequenze di c di lunghezza dispari.

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

f) *acccbccba*
Il suffisso *bccba* può essere descritto da E_2, ma la stringa rimanente, *accc* non può essere descritta da E_1. Infatti, E_1 non è in grado di descrivere stringhe nelle quali il simbolo a non sia seguito da un simbolo b.

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

Esercizio 6

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

- *acccaccc*
- *aaba*
- *accacccbc*
- *acaccabc*

ma non le seguenti:

- *ccaac*
- *ababcab*
- *acaccacc*
- *bbcaccb*

Soluzione

Questi due insiemi di stringhe possono essere differenziati in molti modi. Fra tutti, forse, il più semplice lo si può individuare notando che tutte le stringhe da includere iniziano per a, e che 3 su 4 hanno come penultimo simbolo b.

Questa descrizione $(a(a + b + c)^*b(a + b + c))$ esclude tutte le stringhe del secondo insieme, ma deve essere ampliata per poter descrivere la stringa *acccaccc*. Ammettere anche le stringhe terminanti per c o cc, includerebbe ancora anche una stringa da escludere, *acaccacc*. Per differenziare ulteriormente, si può considerare anche il terzultimo simbolo, descrivendo le stringhe da includere come le stringhe che iniziano per a e che terminano per ba, bc o ccc. Va notato che con la descrizione così specifica del suffisso non è più necessario imporre la descrizione del prefisso.

Queste caratteristiche possono essere descritte dall'espressione regolare $a(a+b)^*(b(a+c)+ccc)$ (o anche da $(a+b)^*(b(a+c)+ccc)$).

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

- *ccaac*: termina per ac;
- *ababcab*: termina per ab;
- *acaccacc*: termina per acc;
- *bbcaccb*: termina per cb.

Altre espressioni regolari che rispettano le specifiche del problema sono:

- $(ac^*)^2(ba + a^*bc)^*$;
- $(a + c^3)^* + (a + b + c)^*b(a + c)$;
- $(ac^*)^2(bc + ba + abc)^*$;
- $(ac^*a)c^*(ba)^*(cb)^*(ab)^*c^*$.