Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

Fondamenti di informatica per la sicurezza

anno accademico 2004–2005

docente: Stefano Ferrari

25.01.2005 — Soluzione della seconda parte — vers. C

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:

- $L_1 = \{a, b\}$
- $L_2 = \{x, xy, yx\}$

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

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) $L_4 = L_1 \cup L_2 = \{a, b, x, xy, yx\}$

c) $L_5 = L_1 L_2 = \{ax, bx, axy, bxy, ayx, byx\}$

d) $L_6 = {L_2}^2 = \{xx, xyx, yxx, xxy, xyxy, yxxy, xyyx, yxyx\}$

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 composto da infiniti elementi, anche L_7 avrà infiniti elementi. L'insieme $\{\epsilon, baaba, xxyxxy, ababxxxy\}$ è un sottoinsieme di L_7.

f) $L_8 = ({L_2}^2 L_1)^*$
L'insieme L_8 è formato dalla concatenazione di un numero arbitrario (eventualmente nullo) di stringhe composte da due elementi di L_2 e da un elemento di L_1. Pertanto, L_8 è composto da infiniti elementi. L'insieme $\{\epsilon, xxya, yxyxbxyyxa\}$ 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: $P = \{S ::= H,\ K ::= d|aH|cK,\ H ::= c|bK|dH\}$

Quali fra le seguenti stringhe vengono generate da G?

a) $ccdb$

b) *dbac*

c) *ddbac*

d) *badca*

e) *dbcca*

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)

ccdb	
	S
$S ::= H$	H
$H ::= c$	c

Non esistono altri metasimboli da espandere e mancano alcuni simboli per ottenere la stringa data.

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

b)

dbac	
	S
$S ::= H$	H
$H ::= dH$	dH
$H ::= bK$	dbK
$K ::= aH$	$dbaH$
$H ::= c$	$dbac$

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

c)

ddbac	
	S
$S ::= H$	H
$H ::= dH$	dH
$H ::= dH$	ddH
$H ::= bK$	$ddbK$
$K ::= aH$	$ddbaH$
$H ::= c$	$ddbac$

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

d)

badca	
	S
$S ::= H$	H
$H ::= bK$	bK
$K ::= aH$	baH
$H ::= dH$	$badH$
$H ::= c$	$badc$

Non esistono altri metasimboli da espandere e mancano alcuni simboli per ottenere la stringa data.

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

e)

dbcca	
	S
$S ::= H$	H
$H ::= dH$	dH
$H ::= bK$	dbK
$K ::= cK$	$dbcK$
$K ::= cK$	$dbccK$
$K ::= aH$	$dbccaH$

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

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

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

- *becd*
- *dab*
- *ba*
- *e*

b) quattro stringhe rifiutate da A:

- *bbbb*

- *dead*
- *bcb*
- *ddb*

Esercizio 4

Modellare, tramite un automa a stati finiti deterministico, il funzionamento di un personaggio di un videogioco che deve assumere il ruolo di un portiere di calcio.

Il portiere di calcio può trovarsi in tre situazioni: in attesa, in uscita e in tuffo.

Dopo un tuffo, il portiere può solo rialzarsi in piedi e tornare così nello stato di attesa.

Ipotizzare che non si possano verificare contemporaneamente più azioni. Modellare l'automa in modo che esso accetti solo le stringhe che descrivono il comportamento corretto del personaggio del gioco. In particolare, individuare possibili situazioni fisicamente irrealizzabili e formalizzarle in modo che l'automa rifiuti le successioni di azioni che porterebbero il personaggio in tali situazioni (per esempio, tuffarsi se sei trova già nello stato di tuffo).

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 le situazioni di funzionamento di un personaggio di un videogioco. Gli stati rappresenteranno la situazione del personaggio, mentre l'insieme dei simboli di input modelleranno gli stimoli che il personaggio riceve dall'esterno (o le azioni che esso subisce).

Questo permette di vedere l'automa come un simulatore del personaggio in esame: l'automa deve accettare le stringhe che rappresentano le sequenze di stimoli (o azioni) che mantengono il personaggio nei comportamenti consentiti oppure quelle che rappresentano una sequenza di eventi di particolare interesse.

Le informazioni date dalle specifiche consentono di definire:

- insieme degli stati, Q:
  $Q = \{attesa, uscita, tuffo, errore\}$
  dove *attesa*, *uscita* e *tuffo* indicano le azioni che il personaggio sta eseguendo, mentre *errore* viene usato per segnalare situazioni non consentite;
- insieme dei simboli, Σ: $\Sigma = \{a, u, t\}$
  dove a comanda al portiere di portarsi in posizione di attesa, u di uscire dalla porta e andare incontro alla palla e t di tuffarsi.

Le specifiche descrivono i seguenti comportamenti:

- nello stato *tuffo*, le uniche azioni consentite sono quelle che lo portano nello stato *attesa*.

Ogni altra transizione descrive una situazione fisicamente irrealizzabile. Per formalizzare queste condizioni, si può usare lo stato *errore*, tale per cui una volta raggiunto non lo si possa più lasciare.

Ogni sequenza di azioni che non comporti il raggiungimento dello stato *errore* rappresenta il normale funzionamento della porta. 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, *attesa*.

Con queste ipotesi aggiuntive, dovrebbero essere accettate, per esempio, le seguenti sequenze di azioni: *aauauuata*, *taua*, *t*. Al contrario, verrebbero rifiutate, tra le altre, le seguenti sequenze di azioni: *autt*, *uuatt*, *uauuaatt*. 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.

Le specifiche sono abbastanza vaghe da rendere immaginabili altre formalizzazioni. Per esempio, l'azione di uscita dopo un tuffo potrebbe essere permessa, o rendendo più complesso l'automa, potrebbe essere aggiunta un'azione di ripristino della situazione precedente al tuffo che permetterebbe di tornare allo stato *attesa* o *uscita* in funzione dello stato in cui si trovava l'automa prima di leggere il simbolo *t*.

Esercizio 5

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

- $E = a(b + bc + bca)^3 + c^*(a + b)^3$

Quali fra le seguenti stringhe vengono descritte da E?

a) *abcbcbca*

$\boldsymbol{\delta}$	a	u	t
attesa	*attesa*	*uscita*	*tuffo*
uscita	*attesa*	*uscita*	*tuffo*
tuffo	*attesa*	*errore*	*errore*
errore	*errore*	*errore*	*errore*

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

b) *abbbc*

c) *cccabb*

d) *aaa*

e) *cabca*

f) *cccabac*

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) *abcbcbca*
$abcbcbca \subseteq (a)(bc)(bc)(bca) \subseteq a(bc+bca)^3 \subseteq a(b+bc+bca)^3 \subseteq a(b+bc+bca)^3+c^*(a+b)^3$

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

b) *abbbc*
$abbbc \subseteq (a)(b)(b)(bc) \subseteq a(b+bc)^3 \subseteq a(b+bc+bca)^3 \subseteq a(b+bc+bca)^3+c^*(a+b)^3$

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

c) *cccabb*
$cccabb \subseteq (ccc)(a)(b)(b) \subseteq c^3(a+b)^3 \subseteq c^*(a+b)^3 \subseteq a(b+bc+bca)^3+c^*(a+b)^3$

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

d) *aaa*
$aaa \subseteq a^3 \subseteq (a+b)^3 \subseteq c^*(a+b)^3 \subseteq a(b+bc+bca)^3+c^*(a+b)^3$

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

e) *cabca*

Le stringhe descritte da E che hanno c come prefisso possono essere descritte solo dalla sottoespressione $c^*(a+b)^3$. Tali stringhe, però, non possono contenere simboli c negli ultimi 3 simboli.

La stringa *cabca* ha come penultimo simbolo un simbolo c Pertanto, essa non può essere descritta da E: $cabca \notin \mathcal{L}(E)$.

f) *cccabac*

Le stringhe descritte da E non possono terminare per ac.

Quindi, la stringa *cccabac* non può essere descritta da E: $cccabac \notin \mathcal{L}(E)$.

Esercizio 6

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

- *abccabcc*
- *cccccc*
- *bbbaac*
- *babbbbac*

ma non le seguenti:

- *cacbbb*
- *acbbbac*
- *cccabba*
- *cbbab*

Soluzione

Si può notare che nelle prime due stringhe da includere il simbolo c è presente a coppie ($ab(cc)ab(cc)$ e $(cc)(cc)(cc)$), mentre le altre due iniziano entrambe per b e terminano per c. La prima caratteristica può essere descritta dall'espressione regolare $(a+b+c^2)^*$, mentre la seconda viene descritta da $b(a+b+c)*c$. L'alternativa tra le due espressioni regolari viene formalizzata unendo le due sottoespressioni mediante l'operatore "+": $(a+b+c^2)^* + b(a+b+c)*c$.

Nessuna delle stringhe del secondo gruppo viene descritta da tale espressione regolare in quanto tutte non iniziano per b e non hanno c a coppie.

Altre espressioni regolari che rispettano le specifiche del problema sono:

- $(ab+c)^* + (b^*a)^2c$
- $(a+b+c)^5(bc)^*(ba)^*c$
- $c^* + (a+b)^2(a+b+c)^*c$
- $(a+b+c^2)^*c^*$
- $(ab+c)*+b(a+b+c)*c$