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 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:

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

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

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

c) $L_5 = L_1 L_2 = \{ax, bx, cx, ay, by, cy\}$

d) $L_6 = {L_1}^2 = \{aa, ba, ca, ab, bb, cb, ac, bc, cc\}$

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 composto da infiniti elementi, anche L_7 avrà infiniti elementi. L'insieme $\{\epsilon, xyyxx, cbbcacbab, xxyabac\}$ 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 $\{\epsilon, yyc, xyaxxb\}$ 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: $P = \{S ::= K, K ::= d|bH|cH, H ::= b|dK|aH\}$

Quali fra le seguenti stringhe vengono generate da G?

a) *cdbb*

b) *baad*

c) *cadc*

d) *bdca*

e) *bdbdbc*

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) *cdbb*

	S
$S ::= K$	K
$K ::= cH$	cH
$H ::= dK$	cdK
$K ::= bH$	$cdbH$
$H ::= b$	$cdbb$

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

b) *baad*

	S
$S ::= K$	K
$K ::= bH$	bH
$H ::= aH$	baH
$H ::= aH$	$baaH$
$H ::= dK$	$baadK$

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

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

c) *cadc*

	S
$S ::= K$	K
$K ::= cH$	cH
$H ::= aH$	caH
$H ::= dK$	$cadK$
$K ::= cH$	$cadcH$

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

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

d) *bdca*

	S
$S ::= K$	K
$K ::= bH$	bH
$H ::= dK$	bdK
$K ::= cH$	$bdcH$
$H ::= aH$	$bdcaH$

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

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

e) *bdbdbc*

	S
$S ::= K$	K
$K ::= bH$	bH
$H ::= dK$	bdK
$K ::= bH$	$bdbH$
$H ::= dK$	$bdbdK$
$K ::= bH$	$bdbdbH$

Non è possibile ottenere il simbolo c dal metasimbolo H.

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

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

- *abdde*
- *bdeea*

- *ceabdb*
- *abcec*

b) quattro stringhe rifiutate da A:

- *abdce*
- *bdeed*
- *ceabd*
- *debad*

Esercizio 4

Modellare, tramite un automa a stati finiti deterministico, il funzionamento del prestito bibliotecario di un libro.

Il libro può trovarsi nelle seguenti condizioni: *libero*, *prenotato*, *imprestato*, *scaduto*, *bloccato*.

Un libro *libero* può venire prenotato oppure ritirato. Dopo una settimana, se un libro *prenotato* non viene ritirato, torna *libero*. Un libro ritirato diventa *imprestato* e rimane in quello stato finché non viene restituito (ritornando *libero*) oppure scade il termine di una settimana, dopo il quale il libro diventa *scaduto*. Un libro *scaduto*, quando viene restituito viene dichiarato *bloccato* e rimane tale per una settimana (durante la quale non può essere né prenotato, né ritirato).

Ipotizzare che non si possano verificare contemporaneamente più azioni. Modellare l'automa in modo che esso accetti solo le stringhe che descrivono il funzionamento del prestito. In particolare, individuare possibili situazioni fisicamente irrealizzabili e formalizzarle in modo che l'automa rifiuti le successioni di azioni che porterebbero il libro in tali situazioni (per esempio, un libro *libero* o *bloccato* non può essere restituito).

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 in cui si può trovare un libro all'interno di un servizioo di prestito bibliotecario. Gli stati rappresenteranno la situazione del libro, mentre l'insieme dei simboli di input modelleranno le operazioni che l'oggetto subisce.

Questo permette di vedere l'automa come un simulatore del sistema di prestito: l'automa deve accettare le stringhe che rappresentano le sequenze di azioni che mantengono il libro nelle situazioni consentiti oppure quelle che rappresentano una sequenza di eventi di particolare interesse.

Le informazioni date dalle specifiche consentono di definire:

- insieme degli stati, Q:
  $Q = \{$*libero*, *prenotato*, *imprestato*, *scaduto*, *bloccato*, *errore*$\}$
  dove *libero* indica che il libro è disponibile al pubblico, *prenotato* indica che il libro è fisicamente in biblioteca, ma è stato impegnato e quindi non è disponibile, *imprestato* indica che il libro è stato ritirato e che quindi non è fisicamente presente in biblioteca, *scaduto* indica che il libro è in mano all'utente da un tempo più lungo del periodo di prestito, e *bloccato* indica che il libro era *scaduto* ed è stato restituito alla biblioteca, mentre, infine, *errore* viene usato per segnalare situazioni non consentite;

- insieme dei simboli, Σ: $\Sigma = \{p, r, c, t\}$
  dove p indica la prenotazione, r indica il ritiro del libro dalla biblioteca, c indica la riconsegna del libro alla biblioteca e t la scadenza temporale di una settimana.

Le specifiche descrivono i seguenti comportamenti:

- dallo stato *libero*, il libro può passare negli stati *prenotato* e *imprestato*;

- la scadenza settimanale fa transitare un libro dallo stato *prenotato* allo stato *libero*, da *imprestato* a *scaduto*, e da *bloccato* a *libero*;

- un libro *scaduto* diventa *bloccato*, quando riconsegnato;

- un libro può subire le zioni di chi lo ha in carico: per esempio, un libro *imprestato* non si può prenotare e un libro *bloccato* non si può restituire.

Quest'ultima situazione descrive un insieme di situazioni 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 del prestito. 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, *libero*.

Con queste ipotesi aggiuntive, dovrebbero essere accettate, per esempio, le seguenti sequenze di azioni: *ptrc*, *rtctp*, *prcpr*. Al contrario, verrebbero rifiutate, tra le altre, le seguenti sequenze di azioni: *pp*, *rtp*, *prtcr*. 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.

Esercizio 5

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

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

Quali fra le seguenti stringhe vengono descritte da E?

a) *cabcaabbca*

b) *bcbcbbca*

c) *cacabbcbbc*

d) *cabc*

e) *cbbaaaabbc*

f) *abbcbbc*

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) *cabcaabbca*
$cabcaabbca \subseteq (ca)(bc)(a)(a)(bbc)(a) \subseteq (ca+bc)^2(a+bbc)^4 \subseteq (bc+ca)^2(a+b^2c)^4 \subseteq (bc+ca)^2(a+b^2c)^*$

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

b) *bcbcbbca*
$bcbcbbca \subseteq (bc)(bc)(bbc)(a) \subseteq (bc)^2(bbc+a)^2 \subseteq (bc+ca)^2(a+b^2c)^2 \subseteq (bc+ca)^2(a+b^2c)^*$

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

c) *cacabbcbbc*
$cacabbcbbc \subseteq (ca)(ca)(bbc)(bbc) \subseteq (ca)^2(bbc)^2 \subseteq (bc+ca)^2(b^2c)^* \subseteq (bc+ca)^2(a+b^2c)^*$

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

d) *cabc*
$cabc \subseteq (ca)(bc) \subseteq (ca+bc)^2 \subseteq (bc+ca)^2 \subseteq (bc+ca)^2(a+b^2c)^*$

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

e) *cbbaaaabbc*

Le stringhe descritte da E devono avere per prefisso la stringa *bc* oppure la stringa *ca*.

La stringa *cbbaaaabbc* non può quindi essere descritta da E: $cbbaaaabbc \notin \mathcal{L}(E)$.

f) *abbcbbc*

Le stringhe descritte da E devono avere per prefisso la stringa *bc* oppure la stringa *ca*.

La stringa *abbcbbc* non gode di questa proprietà e, quindi, non può essere descritta da E: $abbcbbc \notin \mathcal{L}(E)$.

Esercizio 6

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

- *bbbcbcaaabc*
- *bcbcbcbc*
- *ccbbbabc*
- *bbcbbaaaabcaaabc*

δ	p	r	c	t
libero	*prenotato*	*imprestato*	*errore*	*errore*
prenotato	*errore*	*imprestato*	*errore*	*libero*
imprestato	*errore*	*errore*	*libero*	*scaduto*
scaduto	*errore*	*errore*	*bloccato*	*scaduto*
bloccato	*errore*	*errore*	*errore*	*libero*
errore	*errore*	*errore*	*errore*	*errore*

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

ma non le seguenti:

- *abccccbabc*
- *aaacca*
- *bbcbabb*
- *bcbbbaac*

Soluzione

Si può notare che tutte le stringhe da includere terminano per bc e che iniziano per b o per c. Questa caratteristica può essere descritta dall'espressione regolare $(b+c)(a+b+c)^*bc$:

- b o c come prefisso: $\underline{(b+c)}(a+b+c)^*bc$;
- bc come suffisso: $(b+c)(a+b+c)^*\underline{bc}$;
- una qualsiasi stringa all'interno: $(b+c)\underline{(a+b+c)^*}bc$.

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

- *abccccbabc*: inizia per a;
- *aaacca*: inizia per a;
- *bbcbabb*: non termina per bc;
- *bcbbbaac*: non termina per bc.

Altre espressioni regolari che rispettano le specifiche del problema sono:

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