Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

Fondamenti di informatica per la sicurezza

anno accademico 2004–2004

docente: Stefano Ferrari

Soluzione esempio — Seconda parte — versione 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:

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

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) $L_7 = L_1 {L_2}^*$

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) $L_3 = L_1 \cap L_2 = \{a\}$

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

c) $L_5 = L_1 L_2 = \{aa, ba, za, ax, bx, zx, ay, by, zy\}$

d) $L_6 = {L_1}^2 = \{aa, ba, za, ab, bb, zb, az, bz, zz\}$

e) $L_7 = L_1 {L_2}^*$
L'insieme L_7 è formato da stringhe che hanno un elemento di L_1 seguito da una concatenazione di un numero arbitrario (eventualmente nullo) di elementi di L_2. Poiché ${L_2}^*$ è composto da infiniti elementi, anche L_7 avrà infiniti elementi. L'insieme $\{z, bxxy, aaaaa\}$ è un sottoinsieme di L_7.

f) $L_8 = (L_1 L_2)^*$
L'insieme L_8 è equivalente a ${L_5}^*$. Pertanto, L_8 è composto da infiniti elementi. L'insieme $\{\epsilon, aaza, zxbyba\}$ 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: $P = \{S ::= K,\ K ::= d|Ha|Hc,\ H ::= c|Kd|Hb\}$

Quali fra le seguenti stringhe vengono generate da G?

a) ddc

b) $cabba$

c) $adcc$

d) $cabcda$

e) $aabba$

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) ddc

	S
$S ::= K$	K
$K ::= Hc$	Hc
$H ::= Kd$	Kdc
$K ::= d$	ddc

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

b) $cabba$

	S
$S ::= K$	K
$K ::= Ha$	Ha
$H ::= Hb$	Hba
$H ::= Hb$	$Hbba$

Non esiste nessuna regola che permetta di ottenere il simbolo a dal metasimbolo H. La stringa $cabba$ non è quindi generata da G: $cabba \notin \mathcal{L}(G)$.

c) $adcc$

	S
$S ::= K$	K
$K ::= Hc$	Hc
$H ::= c$	cc

Non c'è più alcun metasimbolo nella stringa generata, ma essa non coincide con la stringa data, $adcc$. La stringa $adcc$ non è quindi generata da G: $adcc \notin \mathcal{L}(G)$.

d) $cabcda$

	S
$S ::= K$	K
$K ::= Ha$	Ha
$H ::= Kd$	Kda
$K ::= Hc$	$Hcda$
$H ::= Hb$	$Hbcda$

Non esiste alcuna regola che permetta di ottenere il simbolo a dal metasimbolo H. La stringa $cabcda$ non è quindi generata da G: $cabcda \notin \mathcal{L}(G)$. 7

e) $aabba$

	S
$S ::= K$	K
$K ::= Ha$	Ha
$H ::= Hb$	Hba
$H ::= Hb$	$Hbba$

Non esiste alcuna regola che permetta di ottenere il simbolo a dal metasimbolo H. La stringa $aabba$ non è quindi generata da G: $cabcda \notin \mathcal{L}(G)$.

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

- $bacce$
- a
- cc
- dab

b) quattro stringhe rifiutate da A:

- $caae$
- $abdd$
- $bbcb$
- db

Esercizio 4

Il sistema di controllo di una stampante dispone di un sensore per la presenza/assenza della carta, di un pulsante di avvio e di uno di arresto.

La stampante può essere in quattro stati: *stampa*, *attesa carta*, *attesa stampa*, *arresto*.

I segnali che pervengono al controllo della stampante sono: *assenza di carta*, *inizio stampa*, *fine stampa*, *avvio* e *arresto*.

Quando la stampante riceve il segnale *arresto*, smette di compiere l'azione che sta facendo e si pone in stato *arresto*. Da tale stato, potrà uscire solo dopo aver ricevuto il segnale *avvio*, in seguito al quale di porrà nello stato *attesa stampa*.

Modellare il comportamento di tale sistema tramite un automa a stati finiti deterministico, dove gli stati rappresentano gli stati possibili per la stampante, e la stringa di input rappresenta la sequenza di segnali che possono arrivare alla stampante.

Ipotizzare che non si possano verificare contemporaneamente più segnali. Ipotizzare inoltre, per semplicità di progetto, che gli eventi che non si possono fisicamente realizzare quando il sistema si trova in un dato stato abbiano l'effetto di far permanere il sistema in tale stato.

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 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. È quindi ragionevole pensare ad un automa che accetti le stringhe che rappresentano le sequenze di stimoli fisicamente realizzabili oppure quelle che rappresentano una sequenza di eventi di particolare interesse.

Nel caso in esame, sono date le seguenti informazioni:

- insieme degli stati, Q:
  $Q = \{q_s,\, q_{ac},\, q_{as},\, q_a\}$
  per indicare, rispettivamente, *stampa*, *attesa carta*, *attesa stampa*, e, infine, lo stato di *arresto*;
- alfabeto di input, Σ:
  $\Sigma = \{c,\, i,\, f,\, a,\, s\}$
  per indicare, rispettivamente, *assenza di carta*, *inizio stampa*, *fine stampa*, *avvio* e *arresto*.

Dalle specifiche è possibile dedurre i seguenti comportamenti:

- quando viene premuto il pulsante di arresto, la stampante si porta nella stato di *arresto*, qualsiasi cosa stesse facendo;
- il tasto di avvio riporta la stampante nella normale modalità di funzionamento quando essa si trova in stato di *arresto*;
- i segnali di *inizio stampa* e di *fine stampa* servono rispettivamente a porre la stampante nello stato di *stampa* e di *attesa stampa*;
- quando giunge il segnale di assenza carta la stampante si pone nello stato di *attesa carta*, e vi rimane evidentemente finché non viene rifornita di carta.

Non viene specificato come la stampante può rendersi conto del rifornimento di carta. È realistico supporre che ciò possa essere realizzato in due modi:

- tramite un segnale aggiuntivo, *presenza carta*, (che, quando usato, indicheremo con p e che va ad arricchire l'insieme Σ già descritto sopra);
- la situazione di ripristino carta viene segnalata dall'utente, premendo il tasto di avvio.

Mancano inoltre specifiche riguardo allo stato finale ed allo stato iniziale. Si può ipotizzare che entrambi gli stati siano quello di *attesa stampa*, che rappresenta la condizione di normale inattività della stampante: quando la stampante è in *attesa stampa* significa che tutto funziona a dovere e che eventuali incarichi sono stati espletati correttamente.

Nell'ipotesi di utilizzare il tasto di avvio per indicare alla stampante l'effettuazione del rifornimneto di carta, la tabella delle transizioni, $\delta : Q \times \Sigma \to Q$ può essere quella riportata in Tabella 1.

Ipotizzando, invece, l'utilizzo di un segnale aggiuntivo per indicare alla stampante l'effettuazione del rifornimneto di carta, la tabella delle transizioni, $\delta : Q \times \Sigma \to Q$ può essere quella riportata in Tabella 2.

δ	c	i	f	a	s
q_s	q_{ac}	q_s	q_{as}	q_s	q_a
q_{ac}	q_{ac}	q_{ac}	q_{ac}	q_{as}	q_a
q_{as}	q_{ac}	q_s	q_{as}	q_{as}	q_a
q_a	q_a	q_a	q_a	q_{as}	q_a

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

δ	c	i	f	a	s	p
q_s	q_{ac}	q_s	q_{as}	q_s	q_a	q_s
q_{ac}	q_{ac}	q_{ac}	q_{ac}	q_{ac}	q_a	q_{as}
q_{as}	q_{ac}	q_s	q_{as}	q_{as}	q_a	q_{as}
q_a	q_a	q_a	q_a	q_{as}	q_a	q_a

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

Esercizio 5

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

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

Quali fra le seguenti stringhe vengono descritte da E?

a) *aabbbabcc*

b) *aabaababb*

c) *aabaabaccb*

d) *babb*

e) *accab*

f) *baaabb*

g) *cccccc*

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) *aabbbabcc*
$aabbbabcc = (aa)(bbb)(a)(b)(cc) = (a^2)(b^3)(a)(b)(c^2) \subset (a^*)(b^*)(a)(b)(c^2) \subset (a + b)^*(a)(b + c^2)^2 \subset (ac^* + b)^* a(b + c^2)^2$

La stringa *aabbbabcc* viene descritta da E.

b) *aabaababb*
$aabaababb = (aabaab)a(bb) \subset (a + b)^6 a(b^2) \subset (ac^* + b)^6 a(b + c^2)^2 \subset (ac^* + b)^* a(b + c^2)^2$

La stringa *aabaababb* viene descritta da E.

c) *aabaabaccb*
$aabaabaccb = (aabaab)(a)(cc)(b) \subset (a + b)^6 a(c^2 + b)^2 \subset (ac^* + b)^6 a(b + c^2)^2 \subset (ac^* + b)^* a(b + c^2)^2$

La stringa *aabaabaccb* viene descritta da E.

d) *babb*
$babb = (b)(a)(bb) \subset (ac^* + b)a(b)^2 \subset (ac^* + b)^* a(b + c^2)^2$

La stringa *babb* viene descritta da E.

e) *accab*
La stringa *accab* non può essere descritta da E in quanto una stringa di E dopo l'ultima a deve avere o due b, o la stringa bcc, o la stringa ccb, o quattro c.

f) *cccccc*
La stringa *cccccc* non può essere descritta da E perché tutte le stringhe di E contengono almeno una a.

Esercizio 6

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

- $acaabc$
- $acaaacc$
- $caacb$
- $aaabb$

ma non le seguenti:

- bcb
- $cabcca$
- $cbcba$
- $ccbcbb$

Soluzione

Si può notare che le stringhe da includere hanno il prefisso di lunghezza due composto solo da a o c. Inoltre, il suffisso di lunghezza due è composto solo da b o c. Fra tale prefisso e tale suffisso si trovano stringhe di lunghezza arbitraria composte solo da a. Queste caratteristiche possono essere descritte dall'espressione regolare $(a+c)^2a^*(b+c)^2$:

- solo a e c nel prefisso di lunghezza due: $\underline{(a+c)^2}a^*(b+c)^2$;
- solo b e c nel suffisso di lunghezza due: $(a+c)^2a^*\underline{(b+c)^2}$;
- sequenza di a nel mezzo: $(a+c)^2\underline{a^*}(b+c)^2$.

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

- bcb: ha b nel prefisso di lunghezza due;
- $cabcca$: ha a nel suffisso di lunghezza due;
- $cbcba$: ha a nel suffisso di lunghezza due;
- $ccbcbb$: tra il prefisso e il suffisso di lunghezza due ha bc.

Un'altra espressione regolare che rispetta le specifiche del problema è la seguente:

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