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. 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:

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

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

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

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

c) $L_5 = L_1 L_2 = \{ax, ay, cax, cay, cx, cy\}$

d) $L_6 = L_1^2 = \{aa, ac, aca, ca, caa, cac, caca, cc, cca\}$

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 $\{\epsilon, cacaca, yxxyy, caacaxx\}$ è un sottoinsieme di L_7.

f) $L_8 = ({L_1}^2 L_2)^*$
L'insieme L_8 è formato dalla concatenazione di un numero arbitrario (eventualmente nullo) di stringhe composte da due elementi di L_1 e da un elemento di L_2. Pertanto, L_8 è composto da infiniti elementi. L'insieme $\{\epsilon, ccay, cayacycacxacx\}$ 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: $P = \{S ::= K, K ::= c|Hb|Hc, H ::= a|Kd|Ha\}$

Quali fra le seguenti stringhe vengono generate da G?

a) $cabdb$

b) *aaac*

c) *dcdab*

d) *cdab*

e) *adca*

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) *cabdb*

	S
$K ::= Hb$	Hb
$H ::= Kd$	Kdb
$K ::= Hb$	$Hbdb$
$H ::= a$	$abdb$

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

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

b) *aaac*

	S
$S ::= K$	K
$K ::= Hc$	Hc
$H ::= Ha$	Hac
$H ::= Ha$	$Haac$
$H ::= a$	$aaac$

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

c) *dcdab*

	S
$K ::= Hb$	Hb
$H ::= Ha$	Hab
$H ::= Kd$	$Kdab$
$K ::= Hc$	$Hcdab$
$H ::= Kd$	$Kdcdab$

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

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

d) *cdab*

	S
$S ::= K$	K
$K ::= Hb$	Hb
$H ::= Ha$	Hab
$H ::= Kd$	$Kdab$
$K ::= c$	$cdab$

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

e) *adca*

S

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

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

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

- *ecabac*
- *aa*
- *ddec*
- *ceed*

b) quattro stringhe rifiutate da A:

- *badc*
- *aed*
- *ddab*
- *ebdce*

Esercizio 4

Modellare, tramite un automa a stati finiti deterministico, il funzionamento di una doppietta da caccia.

Una doppietta è un fucile da caccia dotato di due canne e due grilletti. Ogni grilletto controlla lo sparo della cartuccia di una canna.

La doppietta dispone di una leva per aprire la doppietta e accedere alle canne. All'apertura, le cartucce presenti vengono espulse automaticamente.

E' inoltre presente una leva di sicura a due posizioni che permette o impedisce lo sparo di entrambe le canne.

Ipotizzare inoltre che non si possano verificare contemporaneamente più azioni. Modellare l'automa in modo che esso accetti solo le stringhe che descrivono il normale funzionamento della doppietta. In particolare, individuare possibili situazioni fisicamente irrealizzabili o pericolose e formalizzarle in modo che l'automa rifiuti le successioni di azioni che porterebbero la doppietta 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 in prima ipotesi possono essere indipendenti: canna destra, canna sinistra, sicura, meccanismo di apertura. Ciascuna canna può trovarsi in due stati (carica o scarica), così come la sicura (inserita e disinserita), e il meccanismo di apertura (doppietta aperta e doppietta chiusa).

Gli stati del doppietta si possono derivare dalle combinazioni degli stati dei sottosistemi sopra descritti. Pertanto, l'insieme degli stati, Q, può essere:

$Q = \{ai00, ai01, ai10, ai11, ad00, ad01, ad10, ad11, ci00, c$

dove la prima lettera indica se la doppietta è aperta (a) o chiusa (c), la seconda lettera indica lo stato della sicura (inserita, i, e disinserita, d), mentre i numeri finali indicano il numero di colpi rispettivamente per la canna destra e quella sinistra. Per esempio, lo stato *ci01* indica che la doppietta è chiusa, con la sicura inserita ed un colpo nella canna sinistra.

Le azioni che possono essere effettuate sul sistema sono l'apertura e la chiusura della doppietta, l'inserimento e il disinserimento della sicura, tirare i grilletti destro e sinistro, inserire i colpi nella canna destra e sinistra. Non è invece necessario modellare l'azione di estrazione dei colpi dalle canne, in quanto, da specifiche, i colpi vengono espulsi automaticamente all'apertura della doppietta.

Pertanto, insieme dei simboli, Σ, può essere:

$$\Sigma = \{a, c, i, d, g_d, g_s, c_d, c_s\}$$

dove a e c indicano, rispettivamente, l'azione di apertura e di chiusura della doppietta, i e d indicano rispettivamente l'inserimento e il disinserimento della sicura, g_d e g_s indicano rispettivamente le azioni con cui si tirano il grilletto destro e quello sinistro, e c_d e c_s indicano il caricamento della canna destra e sinistra, rispettivamente.

Le specifiche descrivono i seguenti comportamenti:

- la sicura inserita inibisce lo sparo;
- l'apertura della doppietta comporta l'espulsione dei colpi rimanenti.

Le specifiche non impongono ulteriori particolari condizioni. Tuttavia, alcune specifiche aggiuntive è ragionevole supporle:

- il caricamento può avvenire solo a doppietta aperta;
- lo sparo può avvenire solo a doppietta chiusa;
- tirare il grilletto di una canna scarica non ha conseguenze;
- tirare il grilletto di una doppietta aperta non ha conseguenze;
- aprire (chiudere) la doppietta quando essa è aperta (chiusa) non ha conseguenze;

- inserire (disinserire) la sicura quando essa è inserita (disinserita) non ha conseguenze;

- non si può inserire una cartuccia in una canna già occupata.

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 doppietta. 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 doppietta chiusa, scarica, con la sicura inserita, *ci00*.

Con le ipotesi fatte, dovrebbero essere accettate, per esempio, le seguenti sequenze di azioni: $ac_dc_scdg_s$, dac_sca. Al contrario, verrebbero rifiutate, tra le altre, le seguenti sequenze di azioni: c_dad, $iac_sc_dc_s$, acg_dc_d. 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 le azioni di apertura e chiusura hanno effetto solo su sottosistema e tale sottosistema ammette solo due stati, è possibile immaginare solo un'azione collegata alla leva di apertura che abbia come effetto l'apertura della doppietta chiusa oppure la chiusura della doppietta aperta. Analoghe considerazioni valgono per la leva di sicura.

Ulteriori semplificazioni possono essere fatte considerando i due grilletti non come indipendenti, ma come se esistesse un meccanismo che impone un ordine di carico e scarico della doppietta (per esempio, prima la canna destra e poi la sinistra): in tal caso, non potrebbero mai verificarsi quegli stati del tipo *c-10* o *a-01*.

Esercizio 5

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

- $E = (ba^* + ca)^3(c + b^*a)^*$

Quali fra le seguenti stringhe vengono descritte da E?

a) *baaacab*

b) *bccaac*

c) *cababaacbbb*

d) *bbbccc*

e) *aaabaca*

f) *baaaca*

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: $E_1 = (ba^* + ca)^3$ e $E_2 = (c + b^*a)^*$. Quindi, le stringhe descritte da E dovranno obbligatoriamente avere un prefisso descritto da E_1 eventualmente seguito da un suffisso descritto da E_2 (poiché E_2 descrive anche la stringa vuota). Questa premessa semplificherà la spiegazione delle soluzioni di seguito riportate.

a) *baaacab*
$baaacab = (baaa)(ca)(b) \subseteq (ba^3)(ca)(ba^*) \subseteq (ba^*)(ca)(ba^*) \subseteq (ba^* + ca)^3 \subseteq (ba^* + ca)^3(c + b^*a)^*$

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

b) *bccaac*
Nella stringa data è presente una sequenza

$\boldsymbol{\delta}$	a	c	i	d	g_d	g_s	c_d	c_s
ai00	*ai00*	*ci00*	*ai00*	*ad00*	*ai00*	*ai00*	*ai10*	*ai01*
ai01	*ai01*	*ci01*	*ai01*	*ad01*	*ai01*	*ai01*	*ai11*	*errore*
ai10	*ai10*	*ci10*	*ai10*	*ad10*	*ai10*	*ai10*	*errore*	*ai11*
ai11	*ai11*	*ci11*	*ai11*	*ad11*	*ai11*	*ai11*	*errore*	*errore*
ad00	*ad00*	*cd00*	*ai00*	*ad00*	*ad00*	*ad00*	*ad10*	*ad01*
ad01	*ad01*	*cd01*	*ai01*	*ad01*	*ad01*	*ad01*	*ad11*	*errore*
ad10	*ad10*	*cd10*	*ai10*	*ad10*	*ad10*	*ad10*	*errore*	*ad11*
ad11	*ad11*	*cd11*	*ai11*	*ad11*	*ad11*	*ad11*	*errore*	*errore*
ci00	*ai00*	*ci00*	*ci00*	*cd00*	*ci00*	*ci00*	*errore*	*errore*
ci01	*ai01*	*ci01*	*ci01*	*cd01*	*ci01*	*ci01*	*errore*	*errore*
ci10	*ai10*	*ci10*	*ci10*	*cd10*	*ci10*	*ci10*	*errore*	*errore*
ci11	*ai11*	*ci11*	*ci11*	*cd11*	*ci11*	*ci11*	*errore*	*errore*
cd00	*ad00*	*cd00*	*ci00*	*cd00*	*cd00*	*cd00*	*errore*	*errore*
cd01	*ad01*	*cd01*	*ci01*	*cd01*	*cd01*	*cd00*	*errore*	*errore*
cd10	*ad10*	*cd10*	*ci10*	*cd10*	*cd00*	*cd10*	*errore*	*errore*
cd11	*ad11*	*cd11*	*ci11*	*cd11*	*cd01*	*cd10*	*errore*	*errore*
errore	*errore*	*errore*	*errore*	*errore*	*errore*	*errore*	*errore*	*errore*

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

di due simboli c (*bccaac*) che non può essere descritta dall'espressione E_1.

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

c) *cababaacbbb*
Il prefisso *cababaa* della stringa data può essere descritto dall'espressione E_1, ma la rimanente sottostringa, *cbbb*, non può essere descritta da E_2 in quanto termina per b.

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

d) *bbbccc*
$bbbccc = (b)(b)(b)(c)(c)(c) \subseteq (ba^*)(ba^*)(ba^*)(c)(c)(c) \subseteq (ba^* + ca)(ba^* + ca)(ba^* + ca)(c)(c)(c) \subseteq (ba^* + ca)^3(c + b^*a)(c + b^*a)(c + b^*a) \subseteq (ba^* + ca)^3(c + b^*a)^3 \subseteq (ba^* + ca)^3(c + b^*a)^*$

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

e) *aaabaca*
La sottoespressione regolare E_1 non può descrivere stringhe che iniziano per a.

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

f) *baaaca*
L'espressione regolare E_1 descrive stringhe che contengono almeno tre simboli diversi da a (b o c).

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

Esercizio 6

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

- *aaaaa*
- *bbababca*
- *cacabc*
- *cacaa*

ma non le seguenti:

- *bbbacbc*
- *abbaab*
- *aabcbacbc*
- *caacacac*

Soluzione

Questi due insiemi di stringhe possono essere differenziati in molti modi. Fra tutti, forse, il più semplice lo si può individuare notando che, in 3 su 4 casi, le stringhe da includere terminano per a, e la rimanente stringa, *cacabc*, termina per *abc*.

Queste caratteristiche possono essere descritte dall'espressione regolare $(a + b + c)^*(a + abc)$.

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

- *bbbacbc*: termina per *cbc*;
- *abbaab*: termina per *b*;
- *aabcbacbc*: termina per *cbc*;
- *caacacac*: termina per *cac*.

Altre espressioni regolari che rispettano le specifiche del problema sono:

- $(b^*a + cac)^*(a + bc)^2$;
- $(a + b + c)^*(a + bc)^2$;
- $(ca + bc)^* + (a + b + c)^*a$;
- $a^* + (b^*a + ca)^*(bc)^*a^*$;
- $(a + b + c)^2(ab + bc + c^*a)^*$.