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. B

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:

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

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

b) $L_4 = L_1 \cup L_2 = \{a,\ ac,\ c,\ ca\}$
Gli elementi comuni non devono essere ripetuti.

c) $L_5 = L_1 L_2 = \{ac,\ aca,\ acc,\ acca,\ cac,\ caca\}$

d) $L_6 = L_2^2 = \{cac,\ caca,\ cc,\ cca\}$

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 $\{\epsilon,\ caaac,\ cccca,\ caccaacca\}$ 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 $\{\epsilon,\ acc,\ accacaaccaacc\}$ 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: $P = \{S ::= H,\ K ::= d|Hb|Hc,\ H ::= c|Kd|Ha\}$

Quali fra le seguenti stringhe vengono generate da G?

a) *dabd*

b) *ddaa*

c) *ddacd*

d) *bbda*

e) *accd*

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) *dbba*

	S
$S ::= H$	H
$H ::= Ka$	Ka
$K ::= Hb$	Hba

Non esiste regola che generi il simbolo b dal metasimbolo H.

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

b) *ddaa*

	S
$S ::= H$	H
$H ::= Ha$	Ha
$H ::= Ha$	Haa
$H ::= Kd$	$Kdaa$
$K ::= d$	$ddaa$

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

c) *ddacd*

	S
$S ::= H$	H
$H ::= Kd$	Kd
$K ::= Hc$	Hcd
$H ::= Ha$	$Hacd$
$H ::= Kd$	$Kdacd$
$K ::= d$	$ddacd$

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

d) *bbda*

	S
$S ::= H$	H
$H ::= Ha$	Ha
$H ::= Kd$	Kda
$K ::= Hb$	$Hbda$

Non esiste regola che generi il simbolo b dal metasimbolo H.

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

e) *accd*

	S
$S ::= H$	H
$H ::= Kd$	Kd
$K ::= Hc$	Hcd
$H ::= c$	ccd

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

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

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_2	q_1	q_2	q_3	q_1
q_2	q_1	q_2	q_1	q_1	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:

- *ebdce*
- *aa*
- *decc*
- *aebe*

b) quattro stringhe rifiutate da A:

- *abcd*
- *ddab*
- *ecabac*
- *badc*

Esercizio 4

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

Una macchina per il pane è dotata di un cestello, di una pala impastatrice e di una resistenza elettrica. Nel normale funzionamento, l'utente pone nel cestello gli ingredienti (farina, lievito, sale e acqua), chiude il coperchio e poi attiva la preparazione del pane mediante un tasto d'avvio. Dopo l'avvio, la pala inizia a impastare per un quarto d'ora. Poi, la pala si ferma e viene attivata la resistenza per mezz'ora.

La macchina è dotata di due sensori: uno rileva l'apertura del coperchio e l'altro rileva la presenza di acqua. Se il coperchio viene aperto durante l'impastatura, la pala viene arrestata e l'impastatura riprende da capo alla successiva chiusura del coperchio. Se, al termine dell'impastatura non viene rilevata la presenza di acqua, la fase di cottura non viene attivata.

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 per il pane. 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 in prima ipotesi possono essere considerati indipendenti: il coperchio della macchina, il sistema di lavorazione (impastamento e cottura), il contenuto del cestello. Il coperchio può trovarsi in due stati (aperto o chiuso), il sistema di lavorazione può trovarsi in 4 stati (spento, fermo, impastamento, cottura), dove lo stato di fermo è la situazione in cui l'impastamento è stato avviato, ma il coperchio è stato aperto. Qualche ipotesi aggiuntiva è necessaria prima di definire gli stati del cestello.

In teoria, ogni combinazione degli ingredienti citati (farina, lievito, sale e acqua) in quantità tali da non superare il volume del cestello è ammessa. Il numero di stati per descrivere tutte queste combinazioni sarebbe però improponibile. Alcune considerazioni permettono di semplificare la modellazione:

- le specifiche descrivono un comportamento più variegato a seconda della presenza o dell'assenza dell'ingrediente *acqua*;
- nelle specifiche non viene fatto alcun accenno alla possibilità di lavorare gli ingredienti in differenti quantità o proporzioni.

Gli unici comportamenti che riguardano gli ingredienti sembrano ragionevolmente essere:

- la presenza o l'assenza di acqua;
- l'impossibilità di sovraccaricare il estello con una quantità eccessiva di ingredienti.

Pertanto, pare ragionevole supporre che:

- gli ingredienti vengao inseriti in quantità tali da permettere la lavorazione di una porzione;
- non si considerino differenti fra loro gli ingredienti farina, lievito, e sale: d'ora in poi ogni azione eseguita sulla farina varrà anche per gli altri due ingredienti.

Perciò il cestello potrà avere quattro stati (vuoto, solo acqua, solo farina, pieno).

Gli stati della macchina si possono derivare dalle combinazioni degli stati dei sottosistemi precedentemente descritti. Essi dovrebbero quindi essere $2 \times 4 \times 4 = 32$. Tuttavia, poiché i sottosistemi non sono indipendenti, alcuni di essi sono inutili perché non descrivono situazioni realizzabili:

- la stato di cottura non può essere raggiunto se manca l'acqua;
- lo stato di impastamento non può essere raggiunto se il coperchio è aperto;

- lo stato di fermo non può essere raggiunto se il coperchio è chiuso.

Pertanto, l'insieme degli stati, Q, può essere:

$$
\begin{aligned}
Q \;=\; \{&asv,\ asf,\ asa,\ asp,\ afv,\ aff,\ afa,\ afp,\\
&aca,\ acp,\ csv,\ csf,\ csa,\ csp,\ civ,\ cif,\ cia,\\
&cip,\ cca,\ ccp\}
\end{aligned}
$$

dove la prima lettera indica se il coperchio è aperto (a), o chiuso (c), la seconda lettera indica se la macchina è spenta (s), se sta impastando (i), se è ferma (f) o se sta cuocendo (c), ed infine la terza lettera indica il contenuto del cestello: vuoto, (v), solo con farina (f), solo con acqua (a) o con tutti gli ingredienti (pieno, p).

Le azioni che possono essere effettuate sul sistema sono l'apertura e la chiusura del coperchio, il caricamento degli ingredienti, lo svuotamento del cestello, e l'avviamento della lavorazione. Inoltre, è necessario modellare il passare del tempo. In generale, ciò sarebbe un problema perchè il tempo è una grandezza continua, mentre con un automa a stati finiti possono solo essere modellate grandezze discrete (o che possono essere discretizzate). Le specifiche del problema indicano però che eventi particolari avvengono solo ogni quarto d'ora. Ciò significa che non è necessario fare assunzioni per discretizzare il tempo: è sufficiente introdurre un evento che indichi che è trascorso un quarto d'ora. Tuttavia, così facendo, sarebbe necessario duplicare gli stati nei quali avviene la cottura (caratterizzandoli co il primo quarto d'ora e il secondo quarto d'ora). Poiché il numero di stati è già sufficientemente elevato e tale livello di dettaglio non aggiunge molto al comportamento modellato, è ragionevole introdurre due azioni legate al passare del tempo: uno che modella il trascorre di un quarto d'ora e uno che modella il trascorrere di mezz'ora. Questa scelta può creare problemi solo quando l'automa si trova nello stato di impastamento o di cottura: nella realtà, non potrebbe passare mezzora senza che sia trascorso il primo quarto d'ora e il trascorrere di due quarti d'ora dovrebbe equivalere al trascorrere di mezz'ora. Per semplificare, si possono ignorare queste eventualità, come nella modellazione di seguito riportata, oppure si possono catturare in uno stato di errore.

Pertanto, insieme dei simboli, Σ, può essere:

$$\Sigma = \{a,\ c,\ h,\ f,\ s,\ o,\ t_1,\ t_2\}$$

dove a e c indicano, rispettivamente, l'azione di apertura e di chiusura del coperchio, h e f l'inserimento di acqua e farina, s lo svuotamento del cestello, o l'avvio della lavorazione, e t_1 e t_2 il trascorrere di un quarto d'ora e di mezz'ora, rispettivamente.

Le specifiche descrivono i seguenti comportamenti:

- l'inserimento degli ingredienti può avvenire solo a coperchio aperto;
- l'apertura del coperchio in fase di impastamento blocca l'impastamento e lo fa ripartire alla chiusura del coperchio;
- dopo un quarto d'ora di impastamento, la macchina passa allo stato di cottura;
- dopo mezz'ora di cottura, la macchina si spegne automaticamente;
- l'apertura del coperchio in fase di cottura non ha conseguenze.

Alcune ipotesi aggiuntive paiono ragionevoli:

- lo svuotamento può avvenire solo a coperchio aperto;
- lo svuotamento del cestello vuoto non ha effetto;
- l'introduzione di un ingrediente in un cestello che già ne contiene è un errore;
- l'avvio in condizioni di lavorazione non ha effetto;
- l'apertura (chiusura) del coperchio se quest'ultimo è già aperto (chiuso), non ha effetto.

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 il pane. 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 macchina spenta, vuota e con coperchio chiuso, *csv*.

Con le ipotesi fatte, dovrebbero essere accettate, per esempio, le seguenti sequenze di azioni: $afcoahct_1t_2s$, $ahfct_1aaac$. Al contrario, verrebbero rifiutate, tra le altre, le seguenti sequenze di azioni: acf, ahh, h. 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.

Questo problema si presta a molteplici semplificazioni. Per esempio, si può considerare l'ipotesi semplificativa che tutti gli ingredienti (acqua inclusa, quindi) siano inseriti contemporaneamente. In tal caso, il contenuto del cestello può essere o pieno, o vuoto, riducendo così gli stati relativi a tale sottosistema a due (in pratica, si dimezza il numero di stati dell'automa) e riducendo anche le azioni di inserimento degli ingredienti (anziché le azioni f e h, un'unica azione di inserimento degli ingredienti, i). Inoltre, alcune azioni possono essere accorpate, dato il loro carattere binario. Per esempio, l'azione di apertura e chiusura può essere fatta eseguire tramite una più generica azione a/c che ha come effetto quella di aprire un coperchio chiuso e di chiudere un coperchio aperto (quindi, scambia lo stato del sottosistema corpechio).

La tabella delle transizioni, $\delta : Q \times \Sigma \to Q$ della versione semplificata può essere quella riportata in Tabella 2.

Esercizio 5

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

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

Quali fra le seguenti stringhe vengono descritte da E?

a) *bbbbaaa*

b) *bbacaa*

c) *acaccbca*

d) *acbbaba*

e) *bbbcca*

f) *abbbcbbca*

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: $E_1 = (ac + b^2)^*$ e $E_2 = (a + cb^*c)^2$. 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) *bbbbaaa*
Il suffisso aa può essere descritto da E_2, ma la rimanente stringa, $bbbba$ non può essere descritto da E_1 perché questa espressione regolare non descrive stringhe che terminano per a.

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

b) *bbacaa*
$bbacaa = (bb)(ac)(a)(a) \subseteq (b^2)(ac)(a + cb^*c)(a + cb^*c)$
$subseteq (b^2 + ac)^2(a + cb^*c)^2 subseteq (ac + b^2)^*(a + cb^*c)^2$

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

c) *acaccbca*
$acaccbca = (ac)(ac)(cbc)(a) \subseteq (ac+b^2)(ac+b^2)(cb^*c)(a) \subseteq (ac + b^2)^2(cb^*c + a)^2 \subseteq (ac + b^2)^*(a + cb^*c)^2$

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

d) *acbbaba*
L'espressione E_2 non può descrivere stringhe che contengano simboli b a meno che questi non siano delimitati da simboli c.

$\boldsymbol{\delta}$	a	c	h	f	s	o	t_1	t_2
asv	asv	csv	asa	asf	asv	asv	asv	asv
asf	asf	csf	asp	asf	asv	asf	asf	asf
asa	asa	csa	asa	asp	asv	asa	asa	asa
asp	asp	csp	asp	asp	asv	asp	asp	asp
afv	afv	civ	afa	aff	afv	afv	afv	afv
aff	aff	cif	afp	aff	afv	aff	aff	aff
afa	afa	cia	afa	afp	afv	afa	afa	afa
afp	afp	cip	afp	afp	afv	afp	afp	afp
aca	aca	cca	aca	acp	acv	aca	aca	aca
acp	acp	ccp	acp	acp	acv	acp	acp	acp
csv	asv	csv	$errore$	$errore$	$errore$	civ	csv	csv
csf	asf	csf	$errore$	$errore$	$errore$	cif	csf	csf
csa	asa	csa	$errore$	$errore$	$errore$	cia	csa	csa
csp	asp	csp	$errore$	$errore$	$errore$	cip	csp	csp
civ	afv	civ	$errore$	$errore$	$errore$	civ	csv	civ
cif	aff	cif	$errore$	$errore$	$errore$	cif	csf	cif
cia	afa	cia	$errore$	$errore$	$errore$	cia	cca	cia
cip	afp	cip	$errore$	$errore$	$errore$	cip	ccp	cip
cca	aca	cca	$errore$	$errore$	$errore$	cca	cca	csa
ccp	acp	ccp	$errore$	$errore$	$errore$	ccp	ccp	csp
$errore$	$errore$	$errore$	$errore$	$errore$	$errore$	$errore$	$errore$	$errore$

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

$\boldsymbol{\delta}$	a/c	i	s	o	t_1	t_2
asv	csv	asp	asv	asv	asv	asv
asp	csp	$errore$	asv	asp	asp	asp
afv	civ	afp	afv	afv	afv	afv
afp	cip	$errore$	afv	afp	afp	afp
acp	ccp	$errore$	acv	acp	acp	acp
csv	asv	$errore$	$errore$	civ	csv	csv
csp	asp	$errore$	$errore$	cip	csp	csp
civ	afv	$errore$	$errore$	civ	csv	civ
cip	afp	$errore$	$errore$	cip	ccp	cip
ccp	acp	$errore$	$errore$	ccp	ccp	csp
$errore$	$errore$	$errore$	$errore$	$errore$	$errore$	$errore$

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

Poiché la stringa data ha come penultimo simbolo una b ($acbbab\underline{b}a$), il suo suffisso non può essere descritto da E_2.

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

e) $bbbcca$
L'espressione E_2 può descrivere il suffisso cca ($bbb\underline{cca}$), ma la rimanente sottostringa, bbb, non può essere descritta da E_1. Quest'ultima espressione regolare, infatti, può descrivere sequenze di b solo se di lunghezza pari.

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

f) $abbbcbbca$
Il suffisso $cbbca$ può essere descritto da E_2, ma la rimanente sottostringa, $abbb$, non può essere descritta da E_1 perché il simbolo a non è seguito da c.

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

Esercizio 6

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

- $abbbabbb$
- $aaca$
- $abbabbbcb$
- $ababbacb$

ma non le seguenti:

- $bbaab$
- $acacbac$
- $ababbabb$
- $ccbabbc$

Soluzione

Si può notare che tutte le stringhe da accettare hanno iniziano per a e che 3 su 4 proseguono con una sequenza di b. Tale schema si ripete per due volte.

L'espressione regolare che descrive questa caratteristica $(ab^*)^2$ descrive buona aprte delle stringhe da includere: $\underline{abbbabbb}$, $\underline{aa}ca$, $\underline{abbabbb}cb$ e $\underline{ababb}acb$. I suffissi rimanenti possono essere descritti dalla espressione regolare $(a^*c(a+b))^*$, dove la chiusura viene posta perché la prima stringa viene completamente descritta dalla prima espressione regolare (e quindi ha come suffisso la stringa vuota).

L'espressione regolare che descrive le stringhe da includere è quindi: $(ab^*)^2(a^*c(a+b))^*$.

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

- $bbaab$: non inizia per a;
- $acacbac$: non inizia per $(ab^*)^2$;
- $ababbabb$: non termina per $a^*c(a+b)$;
- $ccbabbc$: non inizia per a.

Altre espressioni regolari che rispettano le specifiche del problema sono:

- $(ab^*)^2(ca + a^*cb)^*$;
- $a(a+b)^*(c(a+b) + b^3)$;
- $a^*(b^*ab^2 + b^*a^*b^*a^*c)(a+b)$;
- $(a+b)^*c(a+b) + (a+b^3)^*$;
- $ab(a+b+c)^*cb + (a+b^3+c)^*$;
- $(ab^*a)(b^*a^*c + c^*)(b^* + a)$;
- $a(a+b+c)^*(b^2+c)(a+b+c)$;
- $(ab^3)^2 + a(a+b)^*(ca+cb)$.