Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

# Fondamenti di informatica per la sicurezza

anno accademico 2004–2005

docente: Stefano Ferrari

## 14.01.2005 — Soluzione del econdo compitino — 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 $L_2$:

- $L_1 = \{ab,\ b,\ c\}$
- $L_2 = \{a,\ bc,\ c\}$

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) $L_7 = {L_1}^* L_2$

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 $\epsilon$ appartiene al linguaggio.

### Soluzione

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

b) $L_4 = L_1 \cup L_2 = \{ab,\ b,\ c,\ a,\ bc\}$

c) $L_5 = L_1 L_2 = \{aba,\ ba,\ ca,\ abbc,\ bbc,\ cbc,\ abc,\ bc,\ cc\}$

d) $L_6 = {L_2}^2 = \{aa,\ abc,\ ac,\ bca,\ bcbc,\ bcc,\ ca,\ cbc,\ cc\}$

e) $L_7 = {L_1}^* L_2$
L'insieme $L_7$ è formato da stringhe che hanno un elemento di $L_2$ preceduto da una concatenazione di un numero arbitrario (eventualmente nullo) di elementi di $L_1$. Poiché ${L_1}^*$ è composto da infiniti elementi, anche $L_7$ avrà infiniti elementi. L'insieme $\{a,\ bbaba,\ abbbc\}$ 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 $\{\epsilon,\ aba,\ abbccbc\}$ 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, $P$: $P = \{S ::= K,\ K ::= a|bH|cK,\ H ::= c|dK|bH\}$

Quali fra le seguenti stringhe vengono generate da $G$?

a) *bbda*

b) *cbdbc*

c) *cbdc*

d) *bdca*

e) *bcd*

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)

| *bbda* | |
|---|---|
| | $S$ |
| $S ::= K$ | $K$ |
| $K ::= bH$ | $bH$ |
| $H ::= bH$ | $bbH$ |
| $H ::= dK$ | $bbdK$ |
| $K ::= a$ | $bbda$ |

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

b)

| *cbdbc* | |
|---|---|
| | $S$ |
| $S ::= K$ | $K$ |
| $K ::= cK$ | $cK$ |
| $K ::= bH$ | $cbH$ |
| $H ::= dK$ | $cbdK$ |
| $K ::= bH$ | $cbdbH$ |
| $H ::= c$ | $cbdbc$ |

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

c)

| *cbdc* | |
|---|---|
| | $S$ |
| $S ::= K$ | $K$ |
| $K ::= cK$ | $cK$ |
| $K ::= bH$ | $cbH$ |
| $H ::= dK$ | $cbdK$ |
| $K ::= cK$ | $cbdcK$ |

Non è possibile eliminare il metasimbolo $K$. Pertanto, la stringa *cbdc* non è generata da $G$: $cbdc \notin \mathcal{L}(G)$.

d)

| *bdca* | |
|---|---|
| | $S$ |
| $S ::= K$ | $K$ |
| $K ::= bH$ | $bH$ |
| $H ::= dK$ | $bdK$ |
| $K ::= cK$ | $bdcK$ |
| $K ::= a$ | $bdca$ |

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

e)

| *bcd* | |
|---|---|
| | $S$ |
| $S ::= K$ | $K$ |
| $K ::= bH$ | $bH$ |
| $H ::= c$ | $bc$ |

Non è possibile aggiungere altri simboli, in quanto non ci sono metasimboli nella stringa generata fino a questo punto. La stringa *bcd* non è generata da $G$: $bcd \notin \mathcal{L}(G)$.

## 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, $\Sigma$: $\Sigma = \{a, b, c, d, e\}$
- funzione di transizione $\delta$:

| | $a$ | $b$ | $c$ | $d$ | $e$ |
|---|---|---|---|---|---|
| $q_0$ | $q_3$ | $q_0$ | $q_3$ | $q_0$ | $q_3$ |
| $q_1$ | $q_3$ | $q_2$ | $q_1$ | $q_0$ | $q_2$ |
| $q_2$ | $q_2$ | $q_1$ | $q_2$ | $q_3$ | $q_1$ |
| $q_3$ | $q_1$ | $q_2$ | $q_1$ | $q_2$ | $q_1$ |

- 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 $A$:

- *dacc*
- *abbc*
- *ac*
- *adec*

b) quattro stringhe rifiutate da $A$:

- *edccc*
- *cdedc*
- *bab*
- *dead*

## Esercizio 4

Modellare, tramite un automa a stati finiti deterministico, l'utilizzo di una caffettiera.

Una caffettiera può essere sottoposta alle seguenti azioni: riscaldamento, caricamento, pulizia, svuotamento.

L'uso tipico della caffettiera consiste nel riscaldarla solo dopo averla caricatata (d'acqua e di polvere di caffè), nello svuotarla solo dopo averla riscaldata, nel pulirla solo dopo averla svuotata e nel caricarla solo dopo averla pulita.

Modellare un automa a stati finiti deterministico dove gli stati rappresentano le condizioni in cui una caffettiera si può trovare, e la stringa di input rappresenta la sequenza di azioni che possono essere operate sulla caffettiera stessa. Ipotizzare che non si possano verificare contemporaneamente più azioni. Modellare l'automa in modo che esso accetti solo le stringhe che descrivono un utilizzo accettabile della caffettiera.

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 (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.

Nel caso in esame, sono date le seguenti informazioni:

- insieme degli stati, $Q$:
  $Q = \{c,\, p,\, s,\, v\}$
  dove $c$ sta per "carica" (acqua e polvere di caffè), $p$ sta per "pronta" (il caffè è pronto), $s$ sta per "sporca" (il caffè è stato versato) e $v$ sta per "vuota" (senza acqua, né caffè);

- alfabeto di input, $\Sigma$:
  $\Sigma = \{r,\, v,\, l,\, c\}$
  per indicare, rispettivamente, le azioni di riscaldamento, di versamento, di lavaggio e di caricamento della caffettiera.

Dalle specifiche è possibile dedurre i seguenti comportamenti:

- è corretto riscaldare una una caffettiera che sia nello stato $c$;

- è corretto versare il caffé da una caffettiera che sia nello stato $p$;

- è corretto pulire una caffettiera che sia nello stato $s$;

- è corretto caricare una caffettiera che sia nello stato $v$.

Riscaldare una caffettiera che non sia nello stato $c$ produce evidenti malfunzionamenti nella caffettiera stessa. È ragionevole ipotizzare, quindi di aggiungere uno stato, $e$ ("errore"), per poter catturare le sequenze di azioni ritenute non corrette. Una volta raggiunto lo stato $e$, l'automa non possa più uscirne. Poiché continuare a versare il caffé da una caffettiera ormai vuota non produce effetti particolarmente negativi, si ipotizza che applicare l'azione di versamento su una caffettiera che si trovi nello stato $s$, lascia la caffettiera nello stato $s$ stesso. Analoghe considerazioni consentono di stabilire che l'operazione di pulizia di una caffettiera già pulita (stato $v$) non ha effetto.

Si può ipotizzare che lo stato iniziale sia quello relativo alla caffettiera pulita ($q_0 = v$), e che gli stati finali siano tutti, escluso $e$ ($F = \{c,\, p,\, s,\, v\}$).

La tabella delle transizioni, $\delta : Q \times \Sigma \to Q$ può essere quella riportata in Tabella 1.

Molte varianti sono possibili. Per esempio, è discutibile che l'operazione di versamento di una caffettiera vuota vada ritenuto accettabile: l'effetto dell'operazione non è quello che l'utente della caffettiera si attende (il travaso di caffè dal serbatorio superiore della caffettiera) e pertanto potrebbe benissimo essere considerata un'azione da rifiutare. Analoghe condizioni riguardano altre operazioni, come lavare una caffettiera già pulita o versare il caffè da una caffettiera pulita. Non essendo esplicitamente trattate nelle specifiche, queste situazioni vanno giudicate e classificate dal progettista.

## Esercizio 5

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

- $E = (ac + bc)^2 b^* (c + a)^3$

Quali fra le seguenti stringhe vengono descritte da $E$?

| $\boldsymbol{\delta}$ | $r$ | $v$ | $l$ | $c$ |
|---|---|---|---|---|
| $c$ | $p$ | $e$ | $e$ | $e$ |
| $p$ | $e$ | $s$ | $e$ | $e$ |
| $s$ | $e$ | $s$ | $v$ | $e$ |
| $v$ | $e$ | $e$ | $v$ | $c$ |
| $e$ | $e$ | $e$ | $e$ | $e$ |

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

a) *ababbbbccc*

b) *bcacccc*

c) *acbcbcac*

d) *acacbaca*

e) *cababbc*

f) *aaacccaa*

## 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) *ababbbbccc*
Poiché $E$ ha per prefisso l'espressione $(ac + bc)^2$, tutte le stringhe descritte da $E$ dovranno iniziare con un stringa dell'insieme $\{acac, acbc, bcac, bcbc\}$.

La stringa *ababbbbccc* non rispetta questa regola, quindi non può essere descritta da $E$: $ababbbbccc \notin \mathcal{L}(E)$.

b) *bcacccc*
$bcacccc \subseteq (bc)(ac)(ccc) \subseteq (bc + ac)^2(c)^3 \subseteq (bc + ac)^2(c + a)^3 \subseteq (ac + bc)^2b^*(c + a)^3$

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

c) *acbcbcac*
$acbcbcac \subseteq (ac)(bc)(b)(cac) \subseteq (ac + bc)^2(b)(c + a)^3 \subseteq (ac + bc)^2b^*(c + a)^3$

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

d) *acacbaca*
$acacbaca \subseteq (ac)(ac)(b)(aca) \subseteq (ac)^2(b)(c + a)^3 \subseteq (ac + bc)^2b^*(c + a)^3$

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

e) *cababbc*
Il prefisso di lunghezza 4 della stringa data ($caba$) non può essere descritto dal prefisso di $E$ ($(ac + bc)^2$).

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

f) *aaacccaa*
Il prefisso di lunghezza 4 della stringa data ($aaac$) non può essere descritto dal prefisso di $E$ ($(ac + bc)^2$).

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

## Esercizio 6

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

- *aaabcbbbc*
- *bbbcbbcbc*
- *abcaaaabc*
- *baabcb*

ma non le seguenti:

- *abbc*
- *acbbaab*
- *bcaabca*
- *aabcaabcc*

Si può notare che in tutte le stringhe da includere i simboli $c$ sono preceduti da almeno un simbolo $b$. Inoltre, se fra due simboli $c$ vi sono più di un simbolo $b$, non c'è alcuna $a$, o, in altri termini, fra $a$ e $c$ vi è solo una occorrenza del simbolo $b$. Infine, tutte le sequenze di $a$ sono seguiti da $bc$. Quindi, utilizzando i simboli $c$ per individuare le sottostringhe su cui costruire l'espressione regolare cercata, queste caratteristiche possono essere descritte dall'espressione regolare $(a^*bc + b)^*$:

- una sola $b$ tra $a$ e $c$ e sequenze di $a$ seguite da $bc$: $(\underline{a^*bc} + b)^*$;
- sequenze di $b$: $(a^*bc + \underline{b})^*$.

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

- *abbc*: ha più di un simbolo $b$ fra $a$ e $c$;
- *acbbaab*: fra $a$ e $c$ iniziali non ha il simbolo $b$;
- *bcaabca*: l'ultimo simbolo è $a$, ma non è seguito da $bc$;
- *aabcaabcc*: gli ultimi due simboli sono due $c$.

Un'altra espressione regolare che rispetta le specifiche del problema è $(a + b + c)^3 a^*(cb + bc + bb)^*b^*$.