Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

# Fondamenti di informatica per la sicurezza

Università degli Studi di Milano

anno accademico 2007–2008 docente: Stefano Ferrari

## 24.06.2008 — 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 $L_2$:

- $L_1 = \{x, y\}$
- $L_2 = \{a, bx, ab\}$

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) $L_6 = L_1^3$

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 $\epsilon$ 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) $L_4 = L_1 \cup L_2 = \{a, ab, bx, x, y\}$

c) $L_5 = L_1 L_2 = \{xa, xab, xbx, ya, yab, ybx\}$

d) $L_6 = L_1^3 = \{xxx, xxy, xyx, xyy, yxx, yxy, yyx, yyy\}$

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 $\{\epsilon, xyx, ababa, ay\}$ 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 $\{\epsilon, yab, xaya\}$ 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, $P$: $P = \{S ::= H,\ H ::= a|bH|dK,\ K ::= c|aK|bH\}$

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

a) $bdbba$

b) $bdaca$

c) $dabbc$

d) $bbdab$

e) $daaba$

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)

| | $bdbba$ |
|---|---|
| | $S$ |
| $S ::= H$ | $H$ |
| $H ::= bH$ | $bH$ |
| $H ::= dK$ | $bdK$ |
| $K ::= bH$ | $bdbH$ |
| $H ::= bH$ | $bdbbH$ |
| $H ::= a$ | $bdbba$ |

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

b)

| | $bdaca$ |
|---|---|
| | $S$ |
| $S ::= H$ | $H$ |
| $H ::= bH$ | $bH$ |
| $H ::= dK$ | $bdK$ |
| $K ::= aK$ | $bdaK$ |
| $K ::= c$ | $bdac$ |

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

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

c)

| | $dabbc$ |
|---|---|
| | $S$ |
| $S ::= H$ | $H$ |
| $H ::= dK$ | $dK$ |
| $K ::= aK$ | $daK$ |
| $K ::= bH$ | $dabH$ |
| $H ::= bH$ | $dabbH$ |

Non esiste regola che permetta di ottenere il simbolo $c$ dal metasimbolo $H$.

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

d)

| | $bbdab$ |
|---|---|
| | $S$ |
| $S ::= H$ | $H$ |
| $H ::= bH$ | $bH$ |
| $H ::= bH$ | $bbH$ |
| $H ::= dK$ | $bbdK$ |
| $K ::= aK$ | $bbdaK$ |
| $K ::= bH$ | $bbdabH$ |

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

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

e)

| | $daaba$ |
|---|---|
| | $S$ |
| $S ::= H$ | $H$ |
| $H ::= dK$ | $dK$ |
| $K ::= aK$ | $daK$ |
| $K ::= aK$ | $daaK$ |
| $K ::= bH$ | $daabH$ |
| $H ::= a$ | $daaba$ |

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

## 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, $\delta$:

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

- 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 $A$:

- *eaacc*
- *abccd*
- *eaae*
- *aaab*

b) quattro stringhe rifiutate da $A$:

- *ebbc*
- *abcd*
- *ccdecc*
- *dada*

## Esercizio 4

Modellare, tramite un automa a stati finiti deterministico, il voto di un esame.

L'esame è composto da due parti tra loro indipendenti, ciascuna delle quali ha un voto massimo di 32. Una parte è superata con successo quando viene raggiunta una votazione pari o superiore a 18. Per superare l'esame, lo studente deve superare con successo ciascuna parte.

Un voto superiore a 30 viene considerato 30 e lode. Il voto finale dell'esame è 30 e lode solo se in entrambe le parti viene raggiunta la lode.

Ciascuna parte può essere ripetuta e il voto considerato per la parte è il migliore tra quelli raggiunti.

Il voto iniziale per ciascuna parte è zero.

Ipotizzare che non si possano verificare contemporaneamente più azioni e che le azioni abbiano effetto immediato. Modellare l'automa in modo che esso accetti solo le stringhe che descrivono il normale e corretto accredito dei voti che porta al superamento dell'esame con successo. In particolare, individuare possibili situazioni irrealizzabili e formalizzarle in modo che l'automa rifiuti le successioni di azioni che porterebbero il voto 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 la situazione di uno studente nei confronti dell'esame considerato. L'insieme dei simboli di input modella quindi i voti guadagnati nelle singole parti che compongono l'esame e fli stati descrivono le situazioni in cui il voto viene a formarsi.

Ogni parte può ricevere una votazione che va tra 0 e 32 (senza considerare eventuali decimali). Poichè i voti delle due parti sono indipendenti, ne consegue che gli stati possibili sono pari alle disposizioni con ripetizione di 33 oggetti (i voti possibili) su due posti (le parti dell'esame), risultando quindi $33^2 = 1089$ stati: evidentemente troppi!

Una prima semplificazione è quindi necessaria. Poiché un qualsiasi voto insufficiente impedisce di superare con successo l'esame, i voti insufficienti (quelli inferiori a 18) potrebbero essere rappresentati da un unico voto (per esempio, 0). In tal modo, ogni parte riceverebbe una votazione esprimibile da 16 valori (lo 0 più i voti tra 18 e 32), ma anche così l'automa dovrebbe avere $16^2 = 256$ stati: comunque troppi.

Dovendo semplificare ulteriormente, è necessario individuare altri intervalli in cui raggruppare i possibili voti. Una semplificazione estrema potrebbe prevedere 3 possibili risultati per ciascuna parte: l'insufficienza (0–17), la sufficienza (18–30) e la lode (31–32). Questo approccio permette di ridurre il numero di stati a 9 ($3^2$).

Gli stati verranno indicati con una coppia di simboli, $XY$, dove ciascun simbolo può assumere i valori:

- $I$, per insufficiente
- $S$, per suffficiente
- e $L$ per la lode.

e dove la posizione del simbolo indica la parte dell'esame. Per esempio, lo stato $IL$ indicherà che nella prima parte il voto raggiunto è insufficiente, mentre nella seconda parte si è guadagnata la lode. Pertanto, l'insieme degli stati, $Q$, è:

$$Q = \{II, SI, LI, IS, SS, LS, IL, SL, LL\}$$

Con questa formalizzazione, anche il numero dei simboli dell'automa può essere ridotto a tre per ciascuna parte. Con queste considerazioni l'alfabeto dei simboli, $\Sigma$, è:

$$\Sigma = \{i_1, s_1, l_1, i_2, s_2, l_2\}$$

| $\delta$ | $i_1$ | $s_1$ | $l_1$ | $i_2$ | $s_2$ | $l_2$ |
|---|---|---|---|---|---|---|
| $II$ | $II$ | $SI$ | $LI$ | $II$ | $IS$ | $IL$ |
| $SI$ | $SI$ | $SI$ | $LI$ | $SI$ | $SS$ | $SL$ |
| $LI$ | $LI$ | $LI$ | $LI$ | $LI$ | $LS$ | $LL$ |
| $IS$ | $IS$ | $SS$ | $LS$ | $IS$ | $IS$ | $IL$ |
| $SS$ | $SS$ | $SS$ | $LS$ | $SS$ | $SS$ | $SL$ |
| $LS$ | $LS$ | $LS$ | $LS$ | $LS$ | $LS$ | $LL$ |
| $IL$ | $IL$ | $SL$ | $LL$ | $IL$ | $IL$ | $IL$ |
| $SL$ | $SL$ | $SL$ | $LL$ | $SL$ | $SL$ | $SL$ |
| $LL$ | $LL$ | $LL$ | $LL$ | $LL$ | $LL$ | $LL$ |

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

dove la lettera indica l'esito (insufficiente, sufficiente o lode) e il numero in pedice indica la parte alla quale il voto si riferisce.

Per ogni parte, viene considerato il voto migliore. Quindi, un voto ricevuto ha effetto solo se indica un intervallo di voti superiore a quello attuale. Per esempio, se l'automa si trovasse nello stato $IL$ e ricevesse il simbolo $s_1$, il suo stato diventarebbe $SL$, mentre lo stato rimarrebbe $IL$ se ricevesse il simbolo $s_2$.

Il voto iniziale per ciascuna parte è 0, quindi lo stato iniziale dell'automa è $II$.

L'automa deve accettare tutte le stringhe che formalizzano una serie di prove d'esame che porta almeno alla sufficienza in entrambe le parti. Questo significa che l'insieme degli stati finali dell'automa, $F$, è: $\{SS, LS, SL, LL\}$.

Con le ipotesi fatte, dovrebbero essere accettate, per esempio, le seguenti sequenze di azioni: $i_1 i_2 s_2 l_1 s_2$, $l_2 i_1 i_1 s_1$, $l_1 s_2 i_1 i_2$. Al contrario, verrebbero rifiutate, tra le altre, le seguenti sequenze di azioni: $i_1 l_2 l_2$, $s_1 l_1 l_1 l_1$.

Sebbene non citato nelle specifiche, potrebbe essere considerato anche un simbolo per tener conto del fatto che lo studente possa ritirarsi da una prova. Tuttavia, poiché l'effetto di un ritiro è lo stesso di una insufficienza, l'introduzione di un simbolo per considerare anche questo esito comporterebbe una tabella delle transizioni con due colonne identiche. Per evitare una inutile ridondanza, non si considererà il ritiro come un caso a se stante.

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

È possibile estendere l'automa in modo da considerare differenti intervalli di voti. Per esempio si potrebbe sdoppiare il voto sufficiente in due intervalli per indicare un voto appena sufficiente e un voto medio alto (per esempio, usando gli intervalli 18–24 e 25–30). In tal caso, andrebbero sdoppiati anche i simboli che rappresentano il voto sufficiente in ciascuna delle due parti.

Una ulteriore estensione potrebbe essere introdotta considerando anche la parte di verbalizzazione dell'esame. In tal caso, considerand che dopo la verbalizzazione il voto non può più essere modificato, andrebbe introdotto anche lo stato di verbalizzato e il simbolo che formalizza l'operazione di

| $\boldsymbol{\delta}$ | $i_1$ | $s_1$ | $l_1$ | $i_2$ | $s_2$ | $l_2$ | $v$ |
|---|---|---|---|---|---|---|---|
| $II$ | $II$ | $SI$ | $LI$ | $II$ | $IS$ | $IL$ | $err$ |
| $SI$ | $SI$ | $SI$ | $LI$ | $SI$ | $SS$ | $SL$ | $err$ |
| $LI$ | $LI$ | $LI$ | $LI$ | $LI$ | $LS$ | $LL$ | $err$ |
| $IS$ | $IS$ | $SS$ | $LS$ | $IS$ | $IS$ | $IL$ | $err$ |
| $SS$ | $SS$ | $SS$ | $LS$ | $SS$ | $SS$ | $SL$ | $V$ |
| $LS$ | $LS$ | $LS$ | $LS$ | $LS$ | $LS$ | $LL$ | $V$ |
| $IL$ | $IL$ | $SL$ | $LL$ | $IL$ | $IL$ | $IL$ | $err$ |
| $SL$ | $SL$ | $SL$ | $LL$ | $SL$ | $SL$ | $SL$ | $V$ |
| $LL$ | $LL$ | $LL$ | $LL$ | $LL$ | $LL$ | $LL$ | $V$ |
| $V$ | $err$ | $err$ | $err$ | $err$ | $err$ | $err$ | $err$ |
| $err$ | $err$ | $err$ | $err$ | $err$ | $err$ | $err$ | $err$ |

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

verbalizzazione. Inoltre, considerando che l'esame non può essere verbalizzato se anche solo una parte risulta insufficiente, si potrebbe introdurre uno stato di errore che catturi il tentativo di verbalizzazione indebita. In tal caso, l'automa risulterebbe descritto dal seguente insieme degli stati, $Q$, è:

$$Q = \{II, SI, LI, IS, SS, LS, IL, SL, LL, V, err\}$$

dove lo stato $V$ indica l'avvenuta verbalizzazione e lo stato $err$ indica l'errore, dal seguente alfabeto dei simboli, $\Sigma$:

$$\Sigma = \{i_1, s_1, l_1, i_2, s_2, l_2, v\}$$

dove il simbolo $v$ indica la verbalizzazione, e dalla tabella delle transizioni, $\delta : Q \times \Sigma \to Q$ riportata in Tabella 2. Infine, appare sensato limitare l'insieme degli stati finali dell'automa, $F$, al solo stato della verbalizzazione: $F = \{V\}$. Ciò significa che l'automa riconosce tutte le sequenze di simboli che portano al superamento ed alla verbalizzazione dell'esame.

## Esercizio 5

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

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

Individuare, motivando le risposte, quali fra le seguenti stringhe vengono descritte da $E$:

a) *ccaaaa*

b) *aaaa*

c) *cabaa*

d) *ababccaa*

e) *aabcac*

f) *ccbaac*

## 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: $E_1 = (c^*a + ab)^2$ e $E_2 = (a^2 + bc^*)^*$. 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) *ccaaaa*
$ccaaaa = (cca)(a)(aa) \subseteq (c^2a)(a)(a^2) \subseteq (c^*a)(c^*a)(a^2) \subseteq\subseteq (c^*a + ab)(c^*a + ab)(a^2 + bc^*) \subseteq (c^*a + ab)^2(a^2 + bc^*)^*$

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

b) *aaaa*
$aaaa = (a)(a)(aa) \subseteq (c^*a)(c^*a)(a^2) \subseteq (c^*a + ab)(c^*a + ab)(a^2 + bc^*) \subseteq (c^*a + ab)^2(a^2 + bc^*)^*$

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

c) *cabaa*
Poiché la stringa considerata inizia per $c$, la sottoespressione che genera il prefisso $ca$ ($\underline{ca}baa$) non può essere che $c^*a$. Questo significa che il seguente simbolo $b$ non può essere generato da alcuna sottoespressione di $E_1$, in quanto l'unica sottoespressione che contiene $b$ richiede che esso sia preceduta da un simbolo $a$.

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

d) *ababccaa*
$ababccaa = (ab)(a)(bcc)(aa) \subseteq (ab)(c^*a)(bc^2)(a^2) \subseteq (ab)(c^*a)(bc^*)(a^2) \subseteq (c^*a + ab)^2(a^2 + bc^2)^2 \subseteq (c^*a + ab)^2(a^2 + bc^*)^*$

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

e) *aabcac*
Fra le sottoespressioni che compongono $E$, l'unica che consente di avere un suffisso che termina per $c$ è la sottoespressione $bc^*$. Essa richiede

però che $c$ sia preceduto da un simbolo $b$, mentre nel suffisso della stringa data $b$ è preceduto da $a$ ($aabc\underline{ac}$).

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

f) $ccbaac$
Come la stringa precedente, anche questa stringa termina per $ac$, sottostringa che non può essere generata da $E$.

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

## Esercizio 6

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

- $bacccb$
- $bbaba$
- $cccacccbcb$
- $bcabacb$

ma non le seguenti:

- $cccccc$
- $bbcaba$
- $bacc$
- $aaacbabba$

## Soluzione

Si può notare che tre stringhe su quattro tra quelle che devono essere descritte hanno $cb$ per suffisso ($bacc\underline{cb}$, $cccacccb\underline{cb}$, $bcaba\underline{cb}$). Analizzando i simboli che precedono il suffisso $cb$, ci si accorge che:

- in $ba\underline{cc}cb$ sono una sequenza di $c$ (preceduta da $ba$),
- in $ccca\underline{cccb}cb$ sono una sequenza di $c$ seguito da $b$,
- e in $bca\underline{ba}cb$ è la sequenza $ba$.

Considerando inoltre che anche la stringa rimanente, $bbaba$ ha come suffisso la ripetizione di $ba$, si può pensare di usare come elemento discriminante l'espressione regolare $(c*b+ba)^2$ (due istanze di $c*b$ o $ba$) (come suffisso). In tal caso, le sottostrighe descritte sono quelle sottolineate:

- $\underline{bacccb}$
- $b\underline{baba}$
- $ccca\underline{cccbcb}$
- $bca\underline{bacb}$

Poiché la prima stringa è già descritta dall'espressione regolare considerata, il prefisso dovrà essere una espressione che consideri anche la stringa vuota. Diverse espressioni regolari possono essere usate per descrivere le sottostringhe $b$, $ccca$ e $bca$. Allo scopo, possono essere usate: $b^*c^*a^*$, $b^*(c^*a)^*$ e $(b+c^*a)^*$, sono ugualmente valide. L'espressione risultante, $b^*c^*a^*(c*b+ba)^2$, non descrive le quattro stringhe da escludere:

- $cccccc$: non termina per $b$ o per $a$ ($b^*c^*a^*(c*\underline{b}+b\underline{a})^2$);
- $bbcaba$: nel suffisso, presenta un simbolo $a$ non preceduto da $b$ ($bbc\underline{a}ba$), caratteristica imposta nell'espressione regolare considerata ($b^*c^*a^*(c*b+\underline{ba})^2$);
- $bacc$: non termina per $b$ o per $a$ ($b^*c^*a^*(c*\underline{b}+b\underline{a})^2$);
- $aaacbabba$: nel suffisso, presenta un simbolo $b$ non seguito da $a$ ($aaacba\underline{b}ba$), caratteristica imposta nell'espressione regolare considerata ($b^*c^*a^*(c*b+\underline{ba})^2$);

Altre espressioni regolari che rispettano le specifiche:

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