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. 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:

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

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) $L_6 = {L_1}^2$

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

f) $L_8 = ({L_1}^* L_2)^3$

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

c) $L_5 = L_1 L_2 = \{aa,\ ac,\ ba,\ bc,\ caa,\ cac\}$

d) $L_6 = L_1^2 = \{aa,\ ab,\ aca,\ ba,\ bb,\ bca,\ caa,\ cab,\ caca\}$

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 $\{\epsilon,\ babca,\ ccc,\ cbab\}$ è un sottoinsieme di L_7.

f) $L_8 = (({L_1}^* L_2)^3$
L'insieme L_8 è formato dalla concatenazione di 3 stringhe composte dalla numero arbitrario (eventualmente nullo) di elementi di L_1 e da un elemento di L_2. Pertanto, L_8 è composto da infiniti elementi ma, poiché deve sempre esserci un elemento di L_2 ed L_2 non contiene la stringa vuota, ϵ non appartiene a L_8. L'insieme $\{cac,\ cabcabc\}$ 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: $P = \{S ::= K,\ K ::= b|aH|cH,\ H ::= d|cK|bH\}$

Quali fra le seguenti stringhe vengono generate da G?

a) $accbd$

b) $accb$

c) *acaba*

d) *cbcb*

e) *ccabac*

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) $accbd$

	S
$S ::= K$	K
$K ::= aH$	aH
$H ::= cK$	acK
$K ::= cH$	$accH$
$H ::= bH$	$accbH$
$H ::= d$	$accbd$

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

b) $accb$

	S
$S ::= K$	K
$K ::= aH$	aH
$H ::= cK$	acK
$K ::= cH$	$accH$
$H ::= bH$	$accbH$

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

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

c) $acaba$

	S
$S ::= K$	K
$K ::= aH$	aH
$H ::= cK$	acK
$K ::= aH$	$acaH$
$H ::= bH$	$acabH$

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

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

d) $cbcb$

	S
$S ::= K$	K
$K ::= cH$	cH
$H ::= bH$	cbH
$H ::= cK$	$cbcK$
$K ::= b$	$cbcb$

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

e) $ccabac$

	S
$S ::= K$	K
$K ::= cH$	cH
$H ::= cK$	ccK
$K ::= aH$	$ccaH$
$H ::= bH$	$ccabH$

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

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

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_0	q_1	q_0	q_3
q_2	q_1	q_2	q_1	q_1	q_1
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:

- *ddab*
- *ecabac*
- *decc*
- *bacaa*

b) quattro stringhe rifiutate da A:

- *aebe*
- *abcd*
- *ebdce*
- *dece*

Esercizio 4

Modellare, tramite un automa a stati finiti deterministico, il funzionamento di una *jukebox*.

Un jukebox permette di riprodurre due dischi a scelta dell'utente. Per ogni canzone singola, l'utente deve introdurre un euro. Il jukebox accetta monete da 1 e 2 euro. Se l'utente introduce 2 euro, ha diritto a tre canzoni. Dopo aver introdotto il denaro, l'utente può selezionare la sequenza di canzoni. Al termine della selezione, parte la riproduzione.

Solo a credito esaurito, l'utente può introdurre altro denaro. Il tentativo di forzare l'introduzione di monete causa un malfunzionamento del jukebox.

Ipotizzare che non si possano verificare contemporaneamente più azioni. Modellare l'automa in modo che esso accetti solo le stringhe che descrivono il normale utilizzo del jukebox. In particolare, individuare possibili situazioni fisicamente irrealizzabili o pericolose e formalizzarle in modo che l'automa rifiuti le successioni di azioni che porterebbero il jukebox 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 può trovarsi in tre modalità: attesa, selezione e riproduzione dei brani. Le modalità sono alternative: quando si trova in modalità selezione, le canzoni non vengono suonate, mentre non ammette nuove selezioni quando riproduce le canzoni. In tutte le situazioni, il sistema deve gestire una coda di al più tre canzoni da suonare.

Il jukebox del problema ha solo due possibili canzoni da selezionare e suonare. Pertanto, si possono utilizzare stringhe di tre simboli per codificare la coda, dove il simbolo *0* viene usato per indicare l'assenza di selezione, mentre i simboli *1* e *2* indicano la selezione della prima o della seconda canzone. Inoltre la posizione dei simboli indica la posizione nella coda. In questo modo, per esempio, la stringa *122* indica che nella lista delle canzoni selezionate la prima canzone è la canzone del disco 1, mentre la seconda e la terza canzone selezionata è la canzone del disco 2. Analogamente, la stringa *021* indica che nella lista delle canzoni selezionate la prima canzone è già stata riprodotta e che saranno riprodotte in sequenza la canzone del disco 2 e poi quella del disco 1, mentre la stringa *210* indica che la canzone del disco 2 e quella del disco 1 sono state selezionate per le prime due posizioni, mentre la terza canzone non è stata ancora selezionata.

Poiché la riproduzione inizia automaticamente dopo la selezione dei brani (nel numero dettato dalla quantità di denaro introdotto), la stringa in cui ci sia un simbolo *0* nella seconda o nella terza posizione indica una situazione di selezione, mentre negli altri casi si può supporre che la stringa indichi che il jukebox sta suonando la canzone del disco indicato dal primo simbolo diverso da *0*. Per esempio, la stringa *021* indicherà lo stato in cui il jukebox suonerà la canzone del disco 2 (e al termine, passando nello stato *001*, quella del disco 1).

Agli stati espressi tramite questa codifica vanno aggiunti almeno altri tre stati: lo stato di attesa (*attesa*), e quelli in cui l'automa si dispone alla selezione di un brano (*sel1*) e alla selezione di tre brani (*sel3*).

L'insieme degli stati del juke box, Q, può quindi essere:

$$\begin{aligned} Q \;=\; \{ & \mathit{attesa}, \mathit{sel1}, \mathit{sel3}, \mathit{100}, \mathit{200}, \mathit{110}, \mathit{120}, \\ & \mathit{210}, \mathit{220}, \mathit{111}, \mathit{112}, \mathit{121}, \mathit{122}, \mathit{211}, \mathit{212}, \\ & \mathit{221}, \mathit{222}, \mathit{011}, \mathit{012}, \mathit{021}, \mathit{022}, \mathit{001}, \mathit{002}\} \end{aligned}$$

con il significato dei nomi degli stati sopra descritto.

Le azioni che possono essere effettuate sul sistema sono l'inserimento delle monete (da uno o due euro) e la selezione dei dischi da cui prendere i brani (disco 1 e disco 2).

Inoltre, è necessario modellare il passare del tempo per la riproduzione dei brani, o meglio, un evento che segnali la fine della riproduzione di un brano musicale.

Pertanto, insieme dei simboli, Σ, può essere:

$$\Sigma = \{m_1, m_2, s_1, s_2, f\}$$

dove m_1 e m_2 indicano, rispettivamente, l'inserimento di una moneta da 1€ e 2€, s_1 e s_2 indicano, rispettivamente, la selezione del disco 1 e 2, e f indica la fine della riproduzione di un brano.

Le specifiche descrivono i seguenti comportamenti:

- l'introduzione di 1 euro dà diritto a un brano;
- l'introduzione di 2 euro dà diritto a tre brano;
- introdurre più di 2 euro o introdurre denaro mentre il jukebox sta ancora suonando causa un malfunzionamento.

Le specifiche non chiariscono se il totale di 2€ necessario per aver diritto a tre brani debba essere pagato mediante una sola moneta o se sia possibile accumularlo con due monete da 1€. Pertanto, si assumerà valida questa seconda ipotesi.

Poiché esiste almeno una situazione di malfunzionamento, 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 sia irreversibile, che cioè non esista una sequenza di azioni che permetta di riassorbire una situazione non accettabile.

Si può inoltre ipotizzare che il tentativo di selezionare un brano durante la riproduzione (o durante l'attesa) non abbia alcun effetto. Analogamente, l'evento di fine brano non ha senso che si verifichi in situazioni differenti da quella della modalità di riproduzione dei brani e al di fuori di tale contesto non avrà effetto.

Ogni sequenza di azioni che non comporti il raggiungimento dello stato *errore* rappresenta il normale funzionamento del jukebox. 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 al jukebox in attesa, *attesa*.

Con le ipotesi fatte, dovrebbero essere accettate, per esempio, le seguenti sequenze di azioni: m_1s_1f, $m_2s_1s_1s_2fff$, $m_1m_1s_2s_1s_2fffs_1$. Al contrario, verrebbero rifiutate, tra le altre, le seguenti sequenze di azioni: m_1m_2, $m_1s_2m_1$, $m_2s_1s_1s_2fm_1$. 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, $\delta : Q \times \Sigma \rightarrow Q$ può essere quella riportata in Tabella 1.

Alcune ipotesi semplificative possono essere fatte per rendere più semplice il progetto dell'automa. Per esempio, anziché gestire esplicitamente la coda di selezione è possibile immaginare una azione di selezione per ciascuna sequenza possibile e formalizzare la riproduzione dei brani o delle sequenze tramite un solo evento. In tal caso, aumentando l'insieme di simboli di input, è possibile ridurre notevolmente il numero di stati.

Infatti, l'insieme Q diverrebbe:

$$\begin{aligned} Q \;=\; &\{\mathit{att},\ \mathit{sel1},\ \mathit{sel3},\ \mathit{001},\ \mathit{002},\ \mathit{111},\ \mathit{112},\\ &\mathit{121},\ \mathit{122},\ \mathit{211},\ \mathit{212},\ \mathit{221},\ \mathit{222},\ \mathit{err}\} \end{aligned}$$

mentre l'insieme Σ, sarebbe:

$$\begin{aligned} \Sigma \;=\; &\{m_1,\ m_2,\ s_{001},\ s_{002},\ s_{111},\ s_{112},\ s_{121},\ s_{122},\\ &s_{211},\ s_{212},\ s_{221},\ s_{222},\ f_1,\ f_3\} \end{aligned}$$

dove, oltre alla selezione esplicita delle sequenze è stato differenziato l'evento di terminazione della riproduzione del singolo brano (f_1) oppure della sequenza (f_3).

La tabella delle transizioni, $\delta : Q \times \Sigma \rightarrow Q$ diventerebbe quindi quella riportata in Tabella 2.

Ulteriori ipotesi semplificative portano ad un automa veramente minimale il quale non tiene conto della coda di selezione, ma solo del fatto di dover riprodurre un solo brano oppure tre brani. In tal caso, l'insieme degli stati, Q sarebbe:

$$Q = \{\mathit{attesa},\ \mathit{sel1},\ \mathit{sel3},\ \mathit{ripr1},\ \mathit{ripr3},\ \mathit{errore}\}$$

mentre l'insieme dei simboli di input, Σ, diventerebbe:

$$\Sigma = \{m_1,\ m_2,\ s_1,\ s_3,\ f_1,\ f_3\}$$

e la tabella delle transizioni, $\delta : Q \times \Sigma \rightarrow Q$ diventerebbe quella riportata in Tabella 3.

$\boldsymbol{\delta}$	m_1	m_2	s_1	s_2	f
attesa	*sel1*	*sel3*	*attesa*	*attesa*	*attesa*
sel1	*sel3*	*errore*	*001*	*002*	*sel1*
sel3	*errore*	*errore*	*100*	*200*	*sel3*
100	*errore*	*errore*	*110*	*120*	*100*
200	*errore*	*errore*	*210*	*220*	*200*
110	*errore*	*errore*	*111*	*112*	*110*
120	*errore*	*errore*	*121*	*122*	*120*
210	*errore*	*errore*	*211*	*212*	*210*
220	*errore*	*errore*	*221*	*222*	*220*
111	*errore*	*errore*	*111*	*111*	*011*
112	*errore*	*errore*	*112*	*112*	*012*
121	*errore*	*errore*	*121*	*121*	*021*
122	*errore*	*errore*	*122*	*122*	*022*
211	*errore*	*errore*	*211*	*211*	*011*
212	*errore*	*errore*	*212*	*212*	*012*
221	*errore*	*errore*	*221*	*221*	*021*
222	*errore*	*errore*	*222*	*222*	*022*
011	*errore*	*errore*	*011*	*011*	*001*
012	*errore*	*errore*	*012*	*012*	*002*
021	*errore*	*errore*	*021*	*021*	*001*
022	*errore*	*errore*	*022*	*022*	*002*
001	*errore*	*errore*	*001*	*001*	*attesa*
002	*errore*	*errore*	*002*	*002*	*attesa*
errore	*errore*	*errore*	*errore*	*errore*	*errore*

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

$\boldsymbol{\delta}$	m_1	m_2	s_{001}	s_{002}	s_{111}	s_{112}	s_{121}	s_{122}	s_{211}	s_{212}	s_{221}	s_{222}	f_1	f_3
att	*sel1*	*sel3*	*att*	*att*	*att*	*att*	*att*	*att*	*att*	*att*	*att*	*att*	*att*	*att*
sel1	*sel3*	*err*	*001*	*002*	*sel1*	*sel1*	*sel1*	*sel1*	*sel1*	*sel1*	*sel1*	*sel1*	*sel1*	*sel1*
sel3	*err*	*err*	*sel3*	*sel3*	111	112	121	122	211	212	221	222	*sel3*	*sel3*
001	*err*	*err*	*001*	*001*	*001*	*001*	*001*	*001*	*001*	*001*	*001*	*001*	*att*	*001*
002	*err*	*err*	*002*	*002*	*002*	*002*	*002*	*002*	*002*	*002*	*002*	*002*	*att*	*002*
111	*err*	*err*	*111*	*111*	*111*	*111*	*111*	*111*	*111*	*111*	*111*	*111*	*111*	*att*
112	*err*	*err*	*112*	*112*	*112*	*112*	*112*	*112*	*112*	*112*	*112*	*112*	*112*	*att*
121	*err*	*err*	*121*	*121*	*121*	*121*	*121*	*121*	*121*	*121*	*121*	*121*	*121*	*att*
122	*err*	*err*	*122*	*122*	*122*	*122*	*122*	*122*	*122*	*122*	*122*	*122*	*122*	*att*
211	*err*	*err*	*211*	*211*	*211*	*211*	*211*	*211*	*211*	*211*	*211*	*211*	*211*	*att*
212	*err*	*err*	*212*	*212*	*212*	*212*	*212*	*212*	*212*	*212*	*212*	*212*	*212*	*att*
221	*err*	*err*	*221*	*221*	*221*	*221*	*221*	*221*	*221*	*221*	*221*	*221*	*221*	*att*
222	*err*	*err*	*222*	*222*	*222*	*222*	*222*	*222*	*222*	*222*	*222*	*222*	*222*	*att*
err	*err*	*err*	*err*	*err*	*err*	*err*	*err*	*err*	*err*	*err*	*err*	*err*	*err*	*err*

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

$\boldsymbol{\delta}$	m_1	m_2	s_1	s_3	f_1	f_3
attesa	*sel1*	*sel3*	*attesa*	*attesa*	*attesa*	*attesa*
sel1	*sel3*	*errore*	*ripr1*	*sel1*	*sel1*	*sel1*
sel3	*errore*	*errore*	*sel3*	*ripr3*	*sel3*	*sel3*
ripr1	*errore*	*errore*	*ripr1*	*ripr1*	*attesa*	*ripr1*
ripr3	*errore*	*errore*	*ripr3*	*ripr3*	*ripr3*	*attesa*
errore	*errore*	*errore*	*errore*	*errore*	*errore*	*errore*

Tabella 3: Tabella delle transizioni dell'automa minimale dell'esercizio 4.

Esercizio 5

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

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

Quali fra le seguenti stringhe vengono descritte da E?

a) *aacbaacb*

b) *cabbca*

c) *aaabaaaccb*

d) *bbcaabc*

e) *ccbbacca*

f) *cbbbaca*

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 l'alternativa tra due sottoespressioni: $E_1 = a^3c^*b)^*$ e $E_2 = (c^2b^*a)^*$. Quindi, le stringhe descritte da E dovranno obbligatoriamente essere descritte o da E_1 o da E_2. L'espressione E_1 descrive stringhe che iniziano con 3 simboli a, mentre l'espressione E_2 descrive stringhe che iniziano con 2 simboli c. Questa premessa semplificherà la spiegazione delle soluzioni di seguito riportate.

a) *aacbaacb*
La stringa data inizia con 2 simboli a seguiti da c.

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

b) *cabbca*
La stringa data inizia con ca.

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

c) *aaabaaaccb*
$aaabaaaccb = (aaa)(b)(aaa)(cc)(b) \subseteq (a^3)c^*(b)(a^3)(c^2)(b) \subseteq (a^3c^*b)(a^3c^*b) = (a^3c^*b)^2 \subseteq (a^3c^*b)^* \subseteq (a^3c^*b)^* + (c^2b^*a)^*$

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

d) *bbcaabc*
La stringa data inizia con b.

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

e) *ccbbacca*
$ccbbacca = (cc)(bb)(a)(cc)(a) \subseteq (c^2)(b^2)(a)(c^2)b^*(a) \subseteq (c^2)(b^*)(a)(c^2)b^*(a) = (c^2b^*a)^2 \subseteq (c^2b^*a)^* \subseteq (a^3c^*b)^* + (c^2b^*a)^*$

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

f) *cbbbaca*
La stringa data inizia con cb.

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

Esercizio 6

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

- *cbbccab*
- *cbcabc*
- *acaabab*
- *acaaaabc*

ma non le seguenti:

- *cbbccba*
- *acaacabb*
- *cbcaba*
- *acabcc*

Soluzione

Le stringhe da accettare hanno come suffisso le stringhe *ab* e *bc*. L'espressione regolare che descrive le stringhe da includere è quindi: $(a+b+c)^*(ab+bc)$.

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

- $cbbccba$: termina per ba;
- $acaacabb$: termina per bb;
- $cbcaba$: termina per ba;
- $acabcc$: termina per cc.

Altre espressioni regolari che rispettano le specifiche del problema sono:

- $(ac + cb)a^*b^*c^*(ab + abc)$;
- $(cb^* + a)c^*a^*((ab) * +bc)$;
- $(ac + cb)(a + b + c)^*(bc + ab)$.