Università degli Studi di Milano

Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

Fondamenti di informatica per la sicurezza

anno accademico 2006–2007 docente: Stefano Ferrari

12.01.2007 — Soluzione del secondo compitino — versione 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:

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

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) $L_6 = {L_2}^3$

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

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

c) $L_5 = L_1 L_2 = \{ba, bc, ca, cba, cbc, cc\}$

d) $L_6 = L_2^3 = \{aaa, aac, aca, acc, caa, cac, cca, ccc\}$

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 $\{\epsilon, bccbbb, aaccc, bccc\}$ 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 $\{\epsilon, cbcc, cbcccacbcccbcbc\}$ 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: $P = \{S ::= K,\ K ::= c|Kd|Ha,\ H ::= a|Kb|Kc\}$

Quali fra le seguenti stringhe vengono generate da G?

a) *acdba*

b) *cdcaba*

c) *badba*

d) *cdcad*

e) *bcad*

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) *acdba*

	S
$S ::= K$	K
$K ::= Ha$	Ha
$H ::= Kb$	Kba
$K ::= Kd$	$Kdba$
$K ::= c$	$cdba$

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

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

b) *cdcaba*

	S
$S ::= K$	K
$K ::= Ha$	Ha
$H ::= Kb$	Kba
$K ::= Ha$	$Haba$
$H ::= Kc$	$Kcaba$
$K ::= Kd$	$Kdcaba$
$K ::= c$	$cdcaba$

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

c) *badba*

	S
$S ::= K$	K
$K ::= Ha$	Ha
$H ::= Kb$	Kba
$K ::= Kd$	$Kdba$
$K ::= Ha$	$Hadba$
$H ::= Kb$	$Kbadba$

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

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

d) *cdcad*

	S
$S ::= K$	K
$K ::= Kd$	Kd
$K ::= Ha$	Had
$H ::= Kc$	$Kcad$
$K ::= Kd$	$Kdcad$
$K ::= c$	$cdcad$

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

e) *bcad*

	S
$S ::= K$	K
$K ::= Kd$	Kd
$K ::= Ha$	Had
$H ::= Kc$	$Kcad$

Non esiste regola che permetta di ottenere il simbolo b dal metasimbolo K.

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

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_1	q_2	q_3	q_1
q_1	q_3	q_2	q_0	q_0	q_2
q_2	q_1	q_2	q_1	q_1	q_1
q_3	q_2	q_0	q_1	q_0	q_3

- 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:

- *badb*
- *dabe*
- *cba*
- *debb*

b) quattro stringhe rifiutate da A:

- *acdc*
- *abcd*
- *eeea*
- *eda*

Esercizio 4

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

La macchina è composta da un piano di lavoro, da un utensile che opera la lavorazione, da un coperchio di protezione. Un opportuno pulsante controlla l'attivazione della macchina.

Prima di procedere con la lavorazione, l'operatore deve assicurarsi che l'utensile sia montato, che il pezzo sia posizionato e che il coperchio sia chiuso. Un dispositivo di sicurezza fa sì che il macchinario possa operare solo a coperchio chiuso.

Ipotizzare che non si possano verificare contemporaneamente più azioni. Modellare l'automa in modo che esso accetti solo le stringhe che descrivono il normale funzionamento del macchinario. In particolare, individuare possibili situazioni fisicamente irrealizzabili o pericolose e formalizzarle in modo che l'automa rifiuti le successioni di azioni che porterebbero il macchinario 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.

La macchina che deve essere descritta dall'automa è composta da alcuni sottosistemi, responsabili di particolari funzioni: un piano di lavoro, da un utensile che opera la lavorazione, da un coperchio di protezione e da un pulsante di attivazione.

L'insieme di stati in cui la macchina può trovarsi sono descritti dalle combinazioni possibili dei suoi sottosistemi (o da un suo sottoinsieme).

Il piano di lavoro può essere caratterizzato dalla presenza o dall'assenza del pezzo da lavorare. L'utensile può essere presente o assente. Lo stato in cui si trova il coperchio può essere descritto dalla posizione (alta o bassa). Il tasto può essere premuto o rilasciato. Quindi lo stato della macchina potrebbe essere descritto da $2 \times 2 \times 2 \times 2 = 16$ stati.

Tuttavia, le specifiche consentono di ridurre il numero di stati necessari. Il dispositivo di sicurezza non permette l'attivazione della macchina a coperchio alzato. Si può quindi supporre che non possa mai verificarsi la situazione con pulsante premuto e coperchio alzato, per via della presenza del dispositivo di sicurezza che farebbe scattare il pulsante in posizione di rilascio. Quindi, le combinazioni degli stati dei sottosistemi coperchio e pulsante di attivazione possono essere ridotti a 3: pulsante rilasciato con coperchio alzato o abbassato e coperchio abbassato e pulsante premuto.

Inoltre, le specifiche dicono che per lavorare correttamente deve esserci un pezzo sul piano di lavoro. La lavorazione senza pezzo, per quanto inutile, non è di per sé pericolosa. Analogamente, anche lavorare un pezzo gia' lavorato non dovrebbe avere conseguenze negative.

Le specifiche non dicono se il piano di lavoro può ospitare più pezzi contemporaneamente, ma pare ragionevole supporre che solo un pezzo per volta possa essere lavorato. Quindi, il tentativo di posizionare più di un pezzo sul piano deve essere considerato scorretto. Inoltre, vi sono alcuni vincoli fisici da tenere in considerazione: a coperchio chiuso, non è possibile agire sul piano di lavoro (posizionare o rimuovere un pezzo) e sull'utensile (montarlo o smontarlo). Queste operazioni impossibili devono essere modellate dall'automa mediante uno stato particolare: lo stato *errore*. Lo stato *errore* è 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.

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

$$\begin{aligned} Q \;=\; & \{c_a p_{\text{off}} u_a 0,\ c_a p_{\text{off}} u_a 1,\ c_a p_{\text{off}} u_p 0,\ c_a p_{\text{off}} u_p 1, \\ & c_b p_{\text{off}} u_a 0,\ c_b p_{\text{off}} u_a 1,\ c_b p_{\text{off}} u_p 0,\ c_b p_{\text{off}} u_p 1, \\ & c_b p_{\text{on}} u_a 0,\ c_b p_{\text{on}} u_a 1,\ c_b p_{\text{on}} u_p 0,\ c_b p_{\text{on}} u_p 1,\ \} \\ & \mathit{errore} \end{aligned}$$

dove la prima lettera, c, indica il coperchio (alzato, a, e abbassato, b), la seconda lettera, p, indica lo stato del pulsante (rilasciato, off, e premuto, on), la terza lettera, u, indica lo stato dell'utensile (assente, a, e presente, p), e il numero finale indica il numero di pezzi posizionati sul piano di lavoro.

Le azioni che possono essere effettuate sono:

- il posizionamento di un pezzo;
- la rimozione di un pezzo;
- la pressione del tasto di attivazione;
- il rilascio del tasto di attivazione;

Le specifiche non chiariscono cosa succeda se si cerca di premere (o rilasciare) un tasto già premuto (o rilasciato). Si può ipotizzare che questa azione non abbia conseguenze. Allo stesso modo, si può ipotizzare che il tentativo di rimuovere un pezzo non presente, alzare (abbassare) il coperchio già alzato (abbassato), o montare (rimuovere) l'utensile già montato (assente) non abbia effetto.

L'insieme dei simboli, Σ, può essere:

$$\Sigma = \{c_a, c_b, p_{\text{on}}, p_{\text{off}}, m, t, p, r\}$$

dove c_a e c_b rappresentano rispettivamente l'atto di alzare e abbassare il coperchio, i due seguenti rappresentano le azioni sul pulsante (pressione, p_{on} e rilascio p_{off}), m e t rappresentano le azioni sull'utensile (inserimento, m, e rimozione, r), e, infine, i simboli p e r rappresentano le azioni di posizionamento e rimozione del pezzo dal piano di lavoro.

Ogni sequenza di azioni che non comporti il raggiungimento dello stato *errore* rappresenta il normale funzionamento della macchina. 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 a riposo: $c_b p_{\text{off}} u_a 0$.

Con le ipotesi fatte, dovrebbero essere accettate, per esempio, le seguenti sequenze di azioni: $apmbp_{\text{on}}$, $amprpbp_{\text{on}}arb$. Al contrario, verrebbero rifiutate, tra le altre, le seguenti sequenze di azioni: app, abm, $ampbt$. 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.

Una variante possibile può essere individuata osservando che le operazioni sui tasti (pressione e rilascio) sono mutuamente esclusive. Pertanto, potrebbero essere sostituite da una singola azione di pressione/rilascio il cui effetto sarebbe quello di rilasciare il tasto se premuto, e di premerlo se rilasciato. Analoghe considerazioni valgono per il coperchio e per l'utensile.

Un'altra variante possibile riguarda l'insieme degli stati finali, F, il quale potrebbe essere ristretto alle sole situazioni in cui la macchina si trova in condizioni di riposo.

Inoltre, si può interpretare il funzionamento del sistema descritto nel problema in modo che il pulsante dia l'avvio alla lavorazione e che tale attività termini automaticamente. In tal caso, lo stato del pulsante può non essere esplicitato, venendo sostituito dallo stato di attività della macchina e le azioni riguardanti il pulsante possono essere sostituite da un'azione che rappresenti l'evento di avvio della lavorazione e da una che rappresenti l'evento "terminazione della lavorazione".

Infine, va menzionata una variante più sofisticata, nella quale lo stato di lavorazione del pezzo presente sul piano di lavoro viene considerato, e l'automa reagisce con un errore se viene forzato a lavorare un pezzo già lavorato.

Esercizio 5

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

- $E = (ab^* + ca)^2(c^3 + b^*ac)^*$

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

a) *abbbcaccc*

b) *acacc*

c) *ccabac*

d) *bbbaac*

e) *cabbb*

f) *abbabbbbac*

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: $E_1 = (ab^* + ca)^2$ e $E_2 = (c^3 + b^*ac)^*$. 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.

- *abbbcaccc*
  $abbbcaccc = (abbb)(ca)(ccc) \subseteq (ab^*)(ca)(c^3) \subseteq (ab^* + ca)^2(c^3 + b^*ac) \subseteq (ab^* + ca)^2(c^3 + b^*ac)^*$

$\boldsymbol{\delta}$	c_a	c_b	p_{on}	p_{off}	m	t	p	r
$c_a p_{\text{off}} u_a 0$	$c_a p_{\text{off}} u_a 0$	$c_b p_{\text{off}} u_a 0$	$c_a p_{\text{off}} u_a 0$	$c_a p_{\text{off}} u_a 0$	$c_a p_{\text{off}} u_p 0$	$c_a p_{\text{off}} u_a 0$	$c_a p_{\text{off}} u_a 1$	$c_a p_{\text{off}} u_a 0$
$c_a p_{\text{off}} u_a 1$	$c_a p_{\text{off}} u_a 1$	$c_b p_{\text{off}} u_a 1$	$c_a p_{\text{off}} u_a 1$	$c_a p_{\text{off}} u_a 1$	$c_a p_{\text{off}} u_p 1$	$c_a p_{\text{off}} u_a 1$	*errore*	$c_a p_{\text{off}} u_a 0$
$c_a p_{\text{off}} u_p 0$	$c_a p_{\text{off}} u_p 0$	$c_b p_{\text{off}} u_p 0$	$c_a p_{\text{off}} u_p 0$	$c_a p_{\text{off}} u_p 0$	$c_a p_{\text{off}} u_p 0$	$c_a p_{\text{off}} u_a 0$	$c_a p_{\text{off}} u_p 1$	$c_a p_{\text{off}} u_p 0$
$c_a p_{\text{off}} u_p 1$	$c_a p_{\text{off}} u_p 1$	$c_a p_{\text{off}} u_p 1$	$c_a p_{\text{off}} u_p 1$	$c_a p_{\text{off}} u_p 1$	$c_b p_{\text{off}} u_p 1$	$c_a p_{\text{off}} u_a 1$	*errore*	$c_a p_{\text{off}} u_p 0$
$c_b p_{\text{off}} u_a 0$	$c_a p_{\text{off}} u_a 0$	$c_b p_{\text{off}} u_a 0$	$c_b p_{\text{on}} u_a 0$	$c_b p_{\text{off}} u_a 0$	*errore*	*errore*	*errore*	*errore*
$c_b p_{\text{off}} u_a 1$	$c_a p_{\text{off}} u_a 1$	$c_b p_{\text{off}} u_a 1$	$c_b p_{\text{on}} u_a 1$	$c_b p_{\text{off}} u_a 1$	*errore*	*errore*	*errore*	*errore*
$c_b p_{\text{off}} u_p 0$	$c_a p_{\text{off}} u_p 0$	$c_b p_{\text{off}} u_p 0$	$c_b p_{\text{on}} u_p 0$	$c_b p_{\text{off}} u_p 0$	*errore*	*errore*	*errore*	*errore*
$c_b p_{\text{off}} u_p 1$	$c_a p_{\text{off}} u_p 1$	$c_b p_{\text{off}} u_p 1$	$c_b p_{\text{on}} u_p 1$	$c_b p_{\text{off}} u_p 1$	*errore*	*errore*	*errore*	*errore*
$c_b p_{\text{on}} u_a 0$	$c_a p_{\text{off}} u_a 0$	$c_b p_{\text{on}} u_a 0$	$c_b p_{\text{on}} u_a 0$	$c_b p_{\text{off}} u_a 0$	*errore*	*errore*	*errore*	*errore*
$c_b p_{\text{on}} u_a 1$	$c_a p_{\text{off}} u_a 1$	$c_b p_{\text{on}} u_a 1$	$c_b p_{\text{off}} u_a 1$	$c_b p_{\text{on}} u_a 1$	*errore*	*errore*	*errore*	*errore*
$c_b p_{\text{on}} u_p 0$	$c_a p_{\text{off}} u_p 0$	$c_b p_{\text{on}} u_p 0$	$c_b p_{\text{on}} u_p 0$	$c_b p_{\text{off}} u_p 0$	*errore*	*errore*	*errore*	*errore*
$c_b p_{\text{on}} u_p 1$	$c_a p_{\text{off}} u_p 1$	$c_b p_{\text{on}} u_p 1$	$c_b p_{\text{on}} u_p 1$	$c_b p_{\text{off}} u_p 1$	*errore*	*errore*	*errore*	*errore*
errore	*errore*	*errore*	*errore*	*errore*	*errore*	*errore*	*errore*	*errore*

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

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

Spiegazione alternativa:

$$\begin{array}{ccccccccc}
a & b & b & b & c & a & c & c & c \\
\multicolumn{4}{c}{ab^3} & \multicolumn{2}{c}{ca} & \multicolumn{3}{c}{c^3} \\
\multicolumn{4}{c}{ab^*} & \multicolumn{2}{c}{ca} & \multicolumn{3}{c}{(c^3+b^*ac)} \\
\multicolumn{6}{c}{(ab^*+ca)^2} & \multicolumn{3}{c}{(c^3+b^*ac)}
\end{array}$$

- *acacc*
  Il prefisso *aca* è descritto da E_1, ma la rimanente sottostringa, *cc*, non può essere descritto da E_2.

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

- *ccabac*
  Le stringhe descritte da E non possono iniziare per *cc*.

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

- *bbbaac*
  Le stringhe descritte da E non possono iniziare per b.

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

- *cabbb*
  La sottoespressione E_1 non è in grado di descrivere il prefisso della stringa data, *cab*, in quanto dopo *ca* non può esserci b, se non preceduta da un ulteriore simbolo a.

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

- *abbabbbbac*
  $abbabbbbac = (abb)(abbb)(bac) \subseteq (ab^*)(ab^*)(b^*ac) \subseteq (ab^*+ca)(ab^*+ca)(c^3+b^*ac) \subseteq (ab^*+ca)^2(c^3+b^*ac)^*$

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

  Spiegazione alternativa:

$$\begin{array}{cccccccccc}
a & b & b & a & b & b & b & b & a & c \\
\multicolumn{3}{c}{ab^2} & \multicolumn{4}{c}{ab^3} & \multicolumn{3}{c}{bac} \\
\multicolumn{3}{c}{ab^*} & \multicolumn{4}{c}{ab^*} & \multicolumn{3}{c}{b^*ac} \\
\multicolumn{3}{c}{(ab^*+ca)} & \multicolumn{4}{c}{(ab^*+ca)} & \multicolumn{3}{c}{(c^3+b^*ac)} \\
\multicolumn{7}{c}{(ab^*+ca)^*} & \multicolumn{3}{c}{(c^3+b^*ac)^*}
\end{array}$$

Esercizio 6

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

- *aacccba*
- *acacaabc*
- *accaccbc*
- *acaba*

ma non le seguenti:

- *accbaba*
- *bcbaca*
- *caacba*
- *aaaca*

Soluzione

Si può notare che tutte le stringhe da includere hanno a come suffisso. Inoltre, tutte contengono solo un simbolo b. Queste caratteristiche possono essere descritte dall'espressione regolare $a(a+c)^*b(a+c)^*$. Nessuna delle stringhe del secondo gruppo viene descritta da tale espressione regolare:

- *accbaba*: contiene due b;

- *bcbaca*: non inizia per a;
- *caacba*: non inizia per a;
- *aaaca*: non contiene neanche un simbolo b.

Altre espressioni regolari che rispettano le specifiche:

- $(ac^*)^2(ba + a^*bc)^*$
- $(ac^*)^2(a + c)^*b(a + c)$
- $a(a + c)^*b(a + c)$
- $a(a + c)^*(ba + bc)$
- $(a^2 + ac)(a + c)^*b(a + c)^*$
- $a(c^*a^*)^*(ba + bc)$
- $a(ac^* + ca + cc)^*ba^*c^*$
- $(a^2 + ac)(a + c)^*(ba + bc)$