Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

Fondamenti di informatica per la sicurezza

anno accademico 2005–2006

docente: Stefano Ferrari

20.01.2006 — 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:

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

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) $L_3 = L_1 \cap L_2 = \{ba\}$
Gli insiemi L_1 e L_2 hanno un unico elemento in comune, ba.

b) $L_4 = L_1 \cup L_2 = \{a, ba, ab, b\}$
L'elemento comune, ba, non deve essere ripetuto due volte.

c) $L_5 = L_1 L_2 = \{ab, aba, bab, baba, abb, abba\}$

d) $L_6 = {L_2}^2 = \{bb, bba, bab, baba\}$

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 $\{\epsilon, bbabb, aabaa, bbababbbaaaab\}$ 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 $\{\epsilon, ab, abaabba\}$ 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) *ddcda*

c) *ccdbd*

d) *bbda*

e) *cddaa*

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)

dabd	
	S
$S ::= H$	H
$H ::= Kd$	Kd
$K ::= Hb$	Hbd
$H ::= Ha$	$Habd$
$H ::= Kd$	$Kdabd$

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

La stringa *dabd* non è generata da G: $dabd \notin L(G)$.

b)

ddcda	
	S
$S ::= H$	H
$H ::= Ha$	Ha
$H ::= Kd$	Kda
$K ::= Hc$	$Hcda$
$H ::= Kd$	$Kdcda$
$K ::= d$	$ddcda$

La stringa *ddcda* è generata da G: $ddcda \in L(G)$.

c)

ccdbd	
	S
$S ::= H$	H
$H ::= Kd$	Kd
$K ::= Hb$	Hbd
$H ::= Kd$	$Kdbd$
$K ::= Hc$	$Hcdbd$
$H ::= c$	$ccdbd$

La stringa *ccdbd* è generata da G: $ccdbd \in L(G)$.

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: $bbda \notin L(G)$.

e)

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

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

La stringa *cddaa* non è generata da G: $cddaa \notin L(G)$.

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

- *ecabac*
- *ddea*

- *cede*
- *baddd*

b) quattro stringhe rifiutate da A:

- *aed*
- *ddab*
- *aa*
- *ebdce*

Esercizio 4

Modellare, tramite un automa a stati finiti deterministico, il funzionamento di un robot da cucina.

Il robot da cucina ha in dotazione due lame che possono essere montate in alternativa. Il robot è dotato di un coperchio e di un pulsante bistabile per attivare (e disattivare) il movimento delle lame. Il coperchio è montato su un sensore che disattiva il movimento in caso di apertura durante l'attività.

Ipotizzare che non si possano verificare contemporaneamente più azioni. Modellare l'automa in modo che esso accetti solo le stringhe che descrivono il funzionamento sicuro del robot da cucina. 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 indipendenti: il coperchio, le lame, il pulsante. Il coperchio può trovarsi in due stati (aperto o chiuso), il pulsante può trovarsi in due stati (posizione di azione e di riposo), mentre le lame possono trovarsi in tre stati (prima lama, seconda lama, senza lama).

Il pulsante bistabile controlla l'attività del motore dell'elettrodomestico. Quando il pulsante è nella posizione di azione, il motore è attivo e muove le lame. Tuttavia, le specifiche forniscono un comportamento più complesso: quando il coperchio viene aperto, il motore si ferma. Quindi, sarebbe necessario modellare anche il sottosistema motore (con due stati: in moto e fermo).

Una semplificazione è possibile: considerando che (modellando correttamente il funzionamento del robot da cucina) non è possibile che si verifichi la situazione in cui il motore è attivo e il pulsante è in posizione di riposo (o il coperchio è aperto), si potrebbero sostituire i due sottosistemi motore e pulsante con un unico sottosistema pulsante-motore a tre stati (azione-attivo, azione-fermo e riposo-fermo). Una analoga semplificazione potrebbe essere fatta considerando il sottosistema coperchio-motore. Tuttavia, in un primo momento non sarà considerata questa semplificazione.

Gli stati del robot da cucina si possono derivare dalle combinazioni degli stati dei quattro sottosistemi (coperchio, pulsante, motore e lame). Pertanto, l'insieme degli stati, Q, può essere:

$$\begin{aligned} Q \;=\; \{ & \mathit{aaf0},\ \mathit{aaf1},\ \mathit{aaf2},\ \mathit{arf0},\ \mathit{arf1},\ \mathit{arf2},\ \mathit{aam0}, \\ & \mathit{aam1},\ \mathit{aam2},\ \mathit{arm0},\ \mathit{arm1},\ \mathit{arm2} \\ & \mathit{caf0},\ \mathit{caf1},\ \mathit{caf2},\ \mathit{crf0},\ \mathit{crf1},\ \mathit{crf2},\ \mathit{cam0}, \\ & \mathit{cam1},\ \mathit{cam2},\ \mathit{crm0},\ \mathit{crm1},\ \mathit{crm2}\} \end{aligned}$$

dove il primo simbolo rappresenta lo stato del coperchio (a sta per aperto e c per chiuso), il secondo simbolo rappresenta lo stato del pulsante (a sta per azione e r per riposo), il terzo simbolo rappresenta lo stato del motore (m sta per in movimento e f per fermo), e, infine, il numero indica lo stato della lama (0 sta per senza lama, 1 con la prima lama, 2 con la seconda lama).

Le azioni che possono essere effettuate sul sistema sono l'apertura e la chiusura del coperchio, l'inserimento e il disinserimento della lama, e la pressione del pulsante.

Pertanto, insieme dei simboli, Σ, può essere:

$$\Sigma = \{a,\, c,\, m_1,\, m_2,\, t,\, p\}$$

dove a e c indicano, rispettivamente, l'azione di apertura e di chiusura del coperchio, $m1$ e

$m1$ indicano l'inserimento di una lama (la prima e la seconda, rispettivamente), t il disinserimento della lama (non è necessario distinguerle, in questo caso) e p la pressione del pulsante di avviamento.

Oltre alla già citata specifica che riguarda il comportamento del motore all'apertura del coperchio, si possono esplicitare i seguenti comportamenti:

- la lama può essere messa e tolta solo a motore fermo e coperchio aperto;

- solo una lama per volta può essere montata.

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.

Si può inoltre ipotizzare che il tentativo di apertura (chiusura) del coperchio già aperto (chiuso) generi errore. Analogamente per il tentativo di mettere (o togliere) una lama già inserita (o assente). In tal caso, l'automa viene portato nello stato *errore*.

Ogni sequenza di azioni che non comporti il raggiungimento dello stato *errore* rappresenta il normale funzionamento del robot da cucina. 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 robot spento, senza lame e con coperchio chiuso, *crf0*.

Con le ipotesi fatte, dovrebbero essere accettate, per esempio, le seguenti sequenze di azioni: $am_1cpatm2c$, am_2cppp. Al contrario, verrebbero rifiutate, tra le altre, le seguenti sequenze di azioni: $m_1tm_2tm_1$, am_2m_1, t. 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.

L'automa fin qui descritto è però eccessivamente scrupoloso: alcuni stati non possono essere mai raggiunti (né ha senso che esistano). Per esempio, non è possibile (non deve esserlo) che il robot abbia il coperchio aperto o il pulsante in posizione di riposo e il motore funzionante. Infatti, gli stati *a-m-* *-rm-* (dove - indica un qualsiasi simbolo) non possono essere raggiunti.

Tali stati possono essere eliminati dall'automa senza che ciò modifichi l'insieme delle stringhe accettate (cioè il comportamento dell'automa). La tabella delle transizioni così semplificata è riportata in Tabella 2.

Una ulteriore semplificazione è possibile. Poiché le due diverse lame non hanno, stando alle specifiche, funzionalità differenti, si può considerare anche solo una lama. Il dato riguardante il numero di lame a disposizione è inutile ai fini del comportamento dell'automa, pertanto può non essere considerato. Ciò porta a eliminare tutti gli stati del tipo *—2* e l'azione $m2$. La tabella delle transionzioni risultante è quella riportata in Tabella 3.

Altre modellazioni possono essere operando differenti scelte progettuali:

- se si considera solo lo stato $crf0$ come stato finale, le stringhe accettate sono solo quelle che riportano il robot nell ostato iniziale (che idealmente è lo stato che consente al robot di essere riposto dopo l'utilizzo);

- far funzionare il robot senza lama potrebbe essere considerato errore.

Esercizio 5

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

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

Quali fra le seguenti stringhe vengono descritte da E?

a) *acbbbacbb*

b) *cbbbacacb*

c) *cbbbcb*

d) *accacccb*

e) *accbbbacbb*

f) *cbbbacbcbbbbc*

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

$\boldsymbol{\delta}$	a	c	m_1	m_2	t	p
aaf0	*errore*	*cam0*	*aaf1*	*aaf2*	*errore*	*arf0*
aaf1	*errore*	*cam1*	*errore*	*errore*	*arf0*	*arf1*
aaf2	*errore*	*cam2*	*errore*	*errore*	*aaf0*	*arf2*
arf0	*errore*	*crf0*	*arf1*	*arf2*	*arf0*	*aaf0*
arf1	*errore*	*crf1*	*errore*	*errore*	*arf0*	*aaf1*
arf2	*errore*	*crf2*	*errore*	*errore*	*arf0*	*aaf2*
aam0	*errore*	*cam0*	*errore*	*errore*	*errore*	*arf0*
aam1	*errore*	*cam1*	*errore*	*errore*	*errore*	*arf1*
aam2	*errore*	*cam2*	*errore*	*errore*	*errore*	*arf2*
arm0	*errore*	*crf0*	*errore*	*errore*	*errore*	*aaf0*
arm1	*errore*	*crf1*	*errore*	*errore*	*errore*	*aaf1*
arm2	*errore*	*crf2*	*errore*	*errore*	*errore*	*aaf2*
caf0	*aaf0*	*errore*	*errore*	*errore*	*errore*	*crf0*
caf1	*aaf1*	*errore*	*errore*	*errore*	*errore*	*crf1*
caf2	*aaf2*	*errore*	*errore*	*errore*	*errore*	*crf2*
crf0	*arf0*	*errore*	*errore*	*errore*	*errore*	*cam0*
crf1	*arf1*	*errore*	*errore*	*errore*	*errore*	*cam1*
crf2	*arf2*	*errore*	*errore*	*errore*	*errore*	*cam2*
cam0	*aaf0*	*errore*	*errore*	*errore*	*errore*	*crf0*
cam1	*aaf1*	*errore*	*errore*	*errore*	*errore*	*crf1*
cam2	*aaf2*	*errore*	*errore*	*errore*	*errore*	*crf2*
crm0	*arf0*	*errore*	*errore*	*errore*	*errore*	*cam0*
crm1	*arf1*	*errore*	*errore*	*errore*	*errore*	*cam1*
crm2	*arf2*	*errore*	*errore*	*errore*	*errore*	*cam2*
errore	*errore*	*errore*	*errore*	*errore*	*errore*	*errore*

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

$\boldsymbol{\delta}$	a	c	m_1	m_2	t	p
aaf0	*errore*	*cam0*	*aaf1*	*aaf2*	*errore*	*arf0*
aaf1	*errore*	*cam1*	*errore*	*errore*	*arf0*	*arf1*
aaf2	*errore*	*cam2*	*errore*	*errore*	*aaf0*	*arf2*
arf0	*errore*	*crf0*	*arf1*	*arf2*	*arf0*	*aaf0*
arf1	*errore*	*crf1*	*errore*	*errore*	*arf0*	*aaf1*
arf2	*errore*	*crf2*	*errore*	*errore*	*arf0*	*aaf2*
caf0	*aaf0*	*errore*	*errore*	*errore*	*errore*	*crf0*
caf1	*aaf1*	*errore*	*errore*	*errore*	*errore*	*crf1*
caf2	*aaf2*	*errore*	*errore*	*errore*	*errore*	*crf2*
crf0	*arf0*	*errore*	*errore*	*errore*	*errore*	*cam0*
crf1	*arf1*	*errore*	*errore*	*errore*	*errore*	*cam1*
crf2	*arf2*	*errore*	*errore*	*errore*	*errore*	*cam2*
cam0	*aaf0*	*errore*	*errore*	*errore*	*errore*	*crf0*
cam1	*aaf1*	*errore*	*errore*	*errore*	*errore*	*crf1*
cam2	*aaf2*	*errore*	*errore*	*errore*	*errore*	*crf2*
errore	*errore*	*errore*	*errore*	*errore*	*errore*	*errore*

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

$\boldsymbol{\delta}$	*a*	*c*	*m*	*t*	*p*
aaf0	*errore*	*cam0*	*aaf1*	*errore*	*arf0*
aaf1	*errore*	*cam1*	*errore*	*arf0*	*arf1*
arf0	*errore*	*crf0*	*arf1*	*arf0*	*aaf0*
arf1	*errore*	*crf1*	*errore*	*arf0*	*aaf1*
caf0	*aaf0*	*errore*	*errore*	*errore*	*crf0*
caf1	*aaf1*	*errore*	*errore*	*errore*	*crf1*
crf0	*arf0*	*errore*	*errore*	*errore*	*cam0*
crf1	*arf1*	*errore*	*errore*	*errore*	*cam1*
cam0	*aaf0*	*errore*	*errore*	*errore*	*crf0*
cam1	*aaf1*	*errore*	*errore*	*errore*	*crf1*
errore	*errore*	*errore*	*errore*	*errore*	*errore*

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

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

La stringa data termina con *bb*, sottostringa che può essere descritta solo da E_2. Quindi, *acbbbacbb* può essere descritta da E solo se E_1 descrive *acbbbac*.

Poiché ciò non è vero, la stringa *acbbbacbb* non viene descritta da E: $acbbbacbb \notin \mathcal{L}(E)$.

b) *cbbbacacb*

Nessun suffisso della stringa data può essere descritto da E_2. Il simbolo b ($cbbbacac\underline{b}$) può essere descritto, ma non cb ($cbbbaca\underline{cb}$). Considerando un simbolo in più, si ha la stringa *acb* ($cbbbac\underline{acb}$) che proprio non può essere descritta da E_2 in quanto il simbolo a non appare in E_2.

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

c) *cbbbcb*
$cbbbcb = (cbbbc)(b) = (cb^3c)(b) \subseteq (cb^*c)(b) \subseteq (cb^*c + b)^2 \subseteq (ac + cb^3)^*(b + cb^*c)^2$

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

d) *accacccb*

La stringa data termina con *ccb*, sottostringa che può essere descritta da E_2, ma la stringa rimanente, *accac*, non può essere descritta da E_1 (non possono esserci due c prima di una a).

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

e) *accbbbacbb*
$accbbbacbb = (ac)(cbbb)(ac)(bb) = (ac)(cb^3)(ac)(b^2) \subseteq (ac + cb^3)^3 b^2 \subseteq (ac+cb^3)^3(b+cb^*c)^2 \subseteq (ac+cb^3)^*(b+cb^*c)^2$

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

f) *cbbbacbcbbbbc*
$cbbbacbcbbbbc = (cbbb)(ac)(b)(cbbbbc) = (cb^3)(ac)(b)(cb^4c) \subseteq (cb^3 + ac)^2(b)(cb^*c) \subseteq (cb^3+ac)^*(b+cb^*c)^2 \subseteq (ac+cb^3)^*(b+cb^*c)^2$

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

Esercizio 6

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

- $bbaacacbcbc$
- bca
- $cacbbaaa$
- $abaabbaabc$

ma non le seguenti:

- $aacaccba$
- $bbabbbababa$
- $cacbb$
- $ccabca$

Soluzione

Le stringhe da accettare hanno nella parte terminale le stringhe a e bc. In particolare, almeno due occorrenze di queste stringhe costituiscono il suffisso delle stringhe da accettare. Infatti:

- $bbaacac\underline{bcbc}$: termina con due occorrenze di bc;
- $\underline{bca}$: termina con (anzi è composta da) bc concatenato con a;
- $cacbba\underline{aa}$: termina con due occorrenze di a;
- $abaabba\underline{abc}$: termina con a concatenato con bc.

Questa caratteristica può essere descritta dall'espressione regolare $(a+b+c)^*(a+bc)^2$.

Delle stringhe del secondo gruppo solo una, $ccabca$, viene descritta da tale espressione regolare; riguardo le altre, infatti, si può notare che:

- $aacac\underline{cb}a$: termina per cb e a;
- $bbabbbab\underline{ab}a$: termina per ab e a;
- $cacbb$: termina per b.

Per escludere la stringa rimanente, bisogna rendere un po' più specifica l'espressione regolare fin qui individuata. Ciò può essere fatto osservando che nella parte iniziale delle stringhe da accettare, il simbolo c, se compare, è sempre a coppie, intervallato da un simbolo a. In altri termini, il simbolo c compare solo all'interno di sottostringhe cac. Pertanto, l'espressione $(a+b+cac)^*(a+bc)^2$ risolve il problema dato.

Altre espressioni regolari che rispettano le specifiche del problema sono:

- $(b^*a+cac)^*(a+bc)^2$;
- $(b^*a^*+cac)*(bc+a)^2$;
- $(b^2a+a^*c+ab+a^2)^*(bc+a)^2$;
- $(cac+ab)^*(a^*b^2a^2+b)(c^*a^*+cb+bc)^*$;
- $(bc+cac)(a+b)^*a+(a+b+c)^*bc$;
- $(b^2a^2+cac+bc+ab^*)^*$.