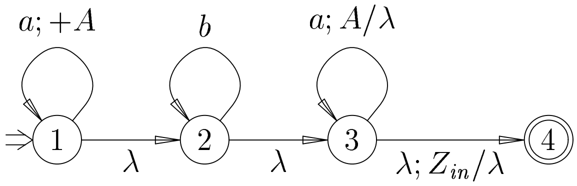
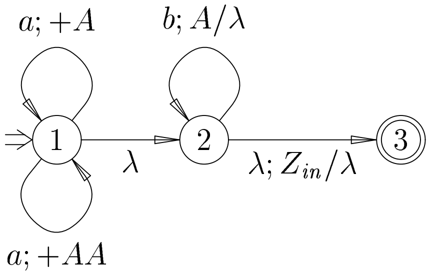
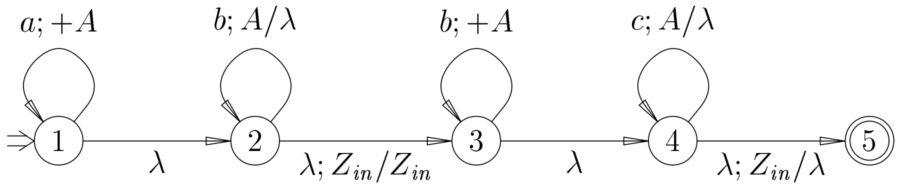
5.1a Acceptatie naar keuze met lege stapel of eindtoestand.

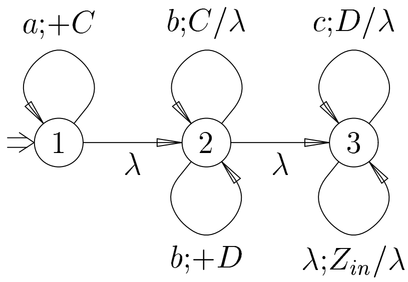


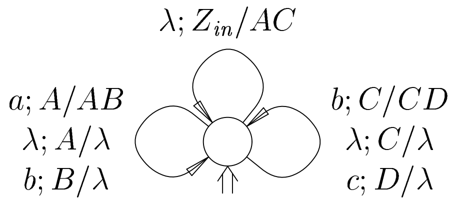
c Stapel voor elke a niet-deterministisch één of twee symbolen op de stapel. Elke b verwijdert één symbool.



d We leven ons uit, en geven drie verschillende automaten voor de taal, in de laatste twee gevallen met lege stapel acceptatie. Alle automaten gebruiken dat L_4 geschreven kan worden als $\{ a^k b^k b^l c^l \mid k, l \in \mathbb{N} \}$.







5.2a Zie de stapelautomaat van Voorbeeld 5.5 in het dictaat, maar verander de overgangen $(p, ¢, X) \mapsto (q, X)$ in $(p, \lambda, X) \mapsto (q, X)$.

b Als w niet van de vorm $x\,\mathrm{mir}(x)$ is, dan is óf de lengte van w oneven, óf er is een k zodat de k-de letter van voren verschilt van de k-de letter van achter. De taal van oneven lengtes is regulier, en daarmee een stapeltaal (bouw de eindige automaat om tot stapelautomaat). Omdat stapeltalen gesloten zijn onder vereniging is het hier voldoende een automaat te maken voor de taal

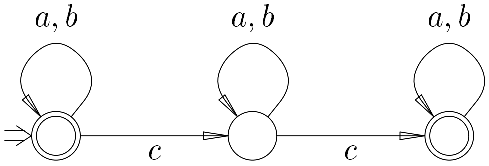
$\{ x\sigma_1 y\sigma_2 z \mid x, y, z \in \{a, b\}^*, |x| = |z|, \sigma_1, \sigma_2 \in \{a, b\}, \sigma_1 \neq \sigma_2 \}$.

De automaat werkt in een aantal fasen: (1) lees letters, en stapel voor elke gelezen letter een X ónder het initiele symbool Z, (2) lees een letter en onthou deze bovenop de stapel, (3) negeer willekeurig veel letters, (4) lees een ándere letter dan de eerder onthouden letter, (5) lees letters en haal per letter een X van de stapel af. Indien na het lezen van het woord de stapel precies leeg is, behoort het gelezen woord tot de taal. Dit lukt allemaal met één toestand:

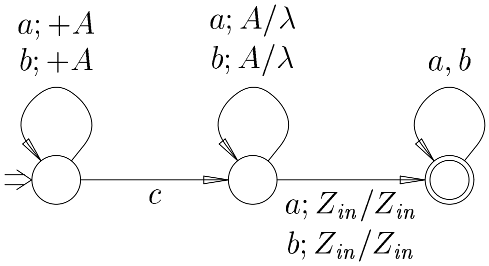
$(p, \sigma, Z) \mapsto (p, ZX) \quad (\sigma \in \{a, b\})$,
$(p, a, Z) \mapsto (p, A)$, $\quad (p, b, Z) \mapsto (p, B)$,
$(p, \sigma, A) \mapsto (p, A)$, $\quad (p, \sigma, B) \mapsto (p, B) \quad (\sigma \in \{a, b\})$,
$(p, a, B) \mapsto (p, \lambda)$, $\quad (p, b, A) \mapsto (p, \lambda)$,
$(p, \sigma, X) \mapsto (p, \lambda) \quad (\sigma \in \{a, b\})$.

c De gegeven taal is te schrijven als $L_1 \cup L_2 \cup L_3 \cup L_4 \cup L_5$ met

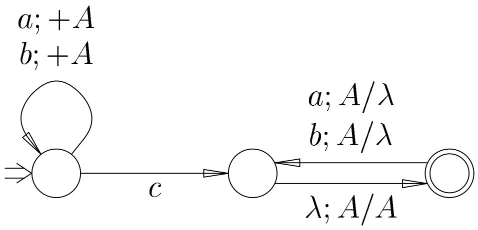
$L_1 = \{ \, w \in \{a, b, c\}^* \mid \#_c(w) \neq 1 \, \}$, een reguliere taal



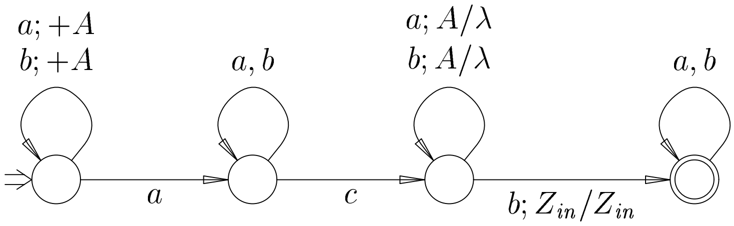
$L_2 = \{ \, x_1 c x_2 \mid x_1, x_2 \in \{a, b\}^*, |x_1| < |x_2| \, \}$,



$L_3 = \{ \, x_1 c x_2 \mid x_1, x_2 \in \{a, b\}^*, |x_1| > |x_2| \, \}$,



$L_4 = \{ \, x_1 a y_1 c x_2 b y_2 \mid x_1, x_2, y_1, y_2 \in \{a, b\}^*, |x_1| = |x_2| \, \}$,



en L_5 is hetzelfde als L_4 maar met woorden van de vorm $x_1 b y_1 c x_2 a y_2$.

De geconstrueerde stapelautomaten kunnen worden samengevoegd dmv. overgangen $(q_{in}, \lambda, Z_{in}) \mapsto (q, Z_{in})$ vanuit een nieuwe begintoestand q_{in}.

d* Als w niet uit twee maal hetzelfde woord bestaat, heeft w oneven lengte, of de eerste helft van w is ongelijk aan de tweede helft, dwz. er is een k zodat de letter op de k-de positie verschilt van de letter op de k-de positie ná het midden. Helaas is hier, anders dan in onderdeel **c**, het midden niet te herkennen, en moet dit punt gegokt worden (en achteraf gecontroleerd).

We komen zo op de taal

$$K_e = \{ \, x_1\sigma_1 y_1 x_2\sigma_2 y_2 \mid x_1, y_1, x_2, y_2 \in \{a,b\}^*, \\ |x_1| = |x_2|, |y_1| = |y_2|, \sigma_1, \sigma_2 \in \{a,b\}, \sigma_1 \neq \sigma_2 \, \}.$$

Op deze manier gegeven is de taal niet te accepteren. Wél als we ons realiseren dat het midden niet precies hoeft te worden vastgesteld, maar dat het volstaat om na te gaan of tussen de σ's evenveel letters staan als erna en ervoor: $|y_1x_2| = |x_1| + |y_2|$ vervangt de eis $|x_1| = |x_2|, |y_1| = |y_2|$.

5.3 $B = B_1 \cup B_2 \cup B_3$ met

$B_1 = \{ \, 1x01^n\$0^n1 \,\mathrm{mir}(x)1 \mid x \in \{0,1\}^*, n \geq 0 \, \}$,
$B_2 = \{ \, 1^n\$0^n1 \mid n \geq 1 \, \}$, en
$B_3 = \{ \, 0\$1 \, \}$.

Hieruit is een stapelautomaat voor B te maken op voor de hand liggende wijze.

5.4 Zie Opmerking 5.11 in het dictaat. De stapelautomaat die correspondeert met de cf grammatica van Opgave 4.11 heeft één toestand q, initieel stapelsymbool S, en overgangen
$(q, \lambda, S) \mapsto (q, SS)$, $\quad (q, \lambda, S) \mapsto (q, \mathbf{(}S\mathbf{)})$, $\quad (q, \lambda, S) \mapsto (q, \lambda)$,
$(q, \mathbf{(}, \mathbf{(}) \mapsto (q, \lambda)$, $\quad (q, \mathbf{)}, \mathbf{)}) \mapsto (q, \lambda)$

5.5 Met de constructie in het bewijs van Lemma 5.12 vinden we 15 producties: (In alle gevallen geldt: $p, r \in Q$, waarbij $Q = \{q_{in}, q_1\}$. We schrijven Z ipv. Z_{in}.)

$S \to [q_{in}Zp]$,
$[q_{in}Zq_{in}] \to a$, $\quad [q_{in}Zr] \to a[q_{in}Xp][pZr]$,
$[q_{in}Xr] \to a[q_{in}Xp][pXr]$, $\quad [q_{in}Xq_1] \to b$,
$[q_1Xq_1] \to b$, $\quad [q_1Zr] \to a[q_{in}Zr]$.

De achtereenvolgens gevonden productieve niet-terminalen:
(1) $[q_{in}Zq_{in}]$, $[q_{in}Xq_1]$, $[q_1Xq_1]$, $\quad$ (2) S, $[q_1Zq_{in}]$.

De producties die mee blijven doen:
$S \to [q_{in}Zq_{in}]$,
$[q_{in}Zq_{in}] \to a$, $\quad [q_{in}Zq_{in}] \to a[q_{in}Xq_1][q_1Zq_{in}]$,
$[q_{in}Xq_1] \to a[q_{in}Xq_1][q_1Xq_1]$, $\quad [q_{in}Xq_1] \to b$,
$[q_1Xq_1] \to b$, $\quad [q_1Zq_{in}] \to a[q_{in}Zq_{in}]$.

De bereikbare niet-terminalen:
(1) S $\quad$ (2) $[q_{in}Zq_{in}]$ $\quad$ (3) $[q_{in}Xq_1]$, $[q_1Zq_{in}]$ $\quad$ (4) $[q_1Xq_1]$.
Dit zijn alle overgebleven variabelen; de eerder gegeven producties vormen dus een gereduceerde grammatica G met axioma S. We kunnen de productie voor S verwijderen, en $[q_{in}Zq_{in}]$ als axioma nemen.

De grammatica is nog wat overzichtelijker te maken. Schrijf Z voor $[q_{in}Zq_{in}]$, X voor $[q_{in}Xq_1]$, en substitueer $[q_1Zq_{in}] \to a[q_{in}Zq_{in}]$ in $[q_{in}Zq_{in}] \to a[q_{in}Xq_1][q_1Zq_{in}]$, en $[q_1Xq_1] \to b$ in $[q_{in}Xq_1] \to a[q_{in}Xq_1][q_1Xq_1]$. We vinden dan de equivalente grammatica:

$Z \to a, \quad Z \to aXaZ,$
$X \to aXb, \quad X \to b$

Hieruit volgt dat $L(G) = L_e(\mathcal{A}) = (\{ a^nb^n \mid n \geq 1\}a)^* \cdot a$.

5.7 Kies een cf-taal L en een reguliere taal R. Laat $\mathcal{A}_R = (Q_R, \Delta, \delta_R, q_R, F_R)$ een eindige automaat zijn voor R en $\mathcal{A}_L = (Q_L, \Delta, \Gamma, \delta_L, q_L, Z_{in}, F_L)$ een stapelautomaat voor L, met eindtoestandsacceptatie. We nemen aan dat $\mathcal{A}_L$ zijn stapel nooit leegt.

We construeren een stapelautomaat $\mathcal{A} = (Q, \Delta, \Gamma, \delta, q, Z_{in}, F)$ met $L_f(\mathcal{A}) = L/R$ als volgt. $\mathcal{A}$ simuleert $\mathcal{A}_L$ op een prefix x van een woord uit L, en kiest (niet-deterministisch) het moment om een suffix y simultaan op $\mathcal{A}_L$ en $\mathcal{A}_R$ te accepteren, zónder de letters van y te lezen. Op deze manier accepteren we x als er een woord $y \in R$ bestaat zo dat $xy \in L$ – het gevraagde rechterquotient.

Formeel gespecificeerd: $Q = Q_L \cup (Q_L \times Q_R)$, $q = q_L$, $F = F_L \times F_R$ en δ bevat de volgende overgangen:
simulatie prefix
$(p, a, A) \mapsto (q, \alpha)$ als $(p, a, A) \mapsto (q, \alpha) \in \delta_L$,
overstap naar suffix
$(p, \lambda, A) \mapsto (\langle p, q_R\rangle, A)$ voor alle $p \in Q_L$ en alle $A \in \Gamma$,
simultane simulatie
$(\langle p, p'\rangle, \lambda, A) \mapsto (\langle q, q'\rangle, \alpha)$
als $a \in \Delta$ bestaat met $(p, a, A) \mapsto (q, \alpha) \in \delta_L$ en $(p', a, q') \in \delta_R$,
$(\langle p, p'\rangle, \lambda, A) \mapsto (\langle q, p'\rangle, \alpha)$ als $(p, \lambda, A) \mapsto (q, \alpha) \in \delta_L$.

5.8a Elke context-vrije grammatica (met regels van de vorm $A \to a\alpha$, met $a \in \Delta \cup \{\lambda\}$ en $\alpha \in (\Sigma - \Delta)^*$) correspondeert op eenvoudige wijze met een stapelautomaat met één toestand en lege-stapel acceptatie: linkspreferente afleidingen van de grammatica komen overeen met de berekeningen van de automaat (zie het bewijs van Lemma 5.10).

Wanneer we aannemen dat de grammatica in Chomsky normaalvorm is (en dat mag) vinden we een stapelautomaat die nul of twee symbolen op de stapel terugzet:

$A \to BC$ geeft $(q, \lambda, A) \mapsto (q, BC)$
$A \to a$ geeft $(q, a, A) \mapsto (q, \lambda)$

b Voeg een (niet-gebruikte) stapel toe aan de eindige automaat (met $\Gamma = \{Z\}$), en zorg ervoor dat die stapel in een eindtoestand leeggemaakt kan worden:

(p, a, q) geeft $(p, a, Z) \mapsto (q, Z)$
$(p, \lambda, Z) \mapsto (p, \lambda)$ voor elke $p \in F$.

c Stapelautomaten met $k = 1$ halen een symbool van de stapel en zetten er ten hoogste een voor terug. Tijdens een berekening staat er precies één symbool op de stapel;

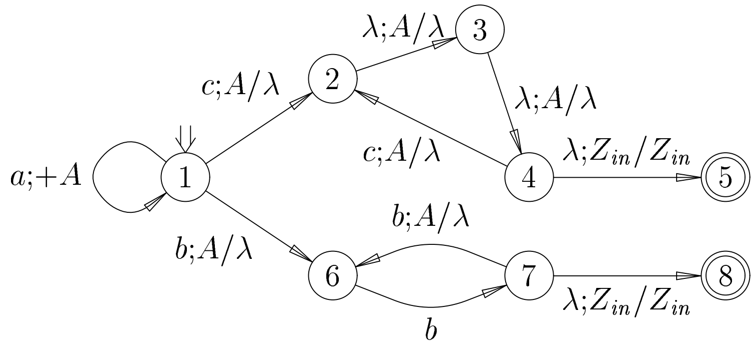
de automaat stopt (noodgedwongen) wanneer er niets meer op de stapel staat– de invoer (mits gelezen) wordt dan geaccepteerd.

Een te construeren eindige automaat onthoudt de toestand van de stapelautomaat, maar ook het ene stapelsymbool op de stapel.

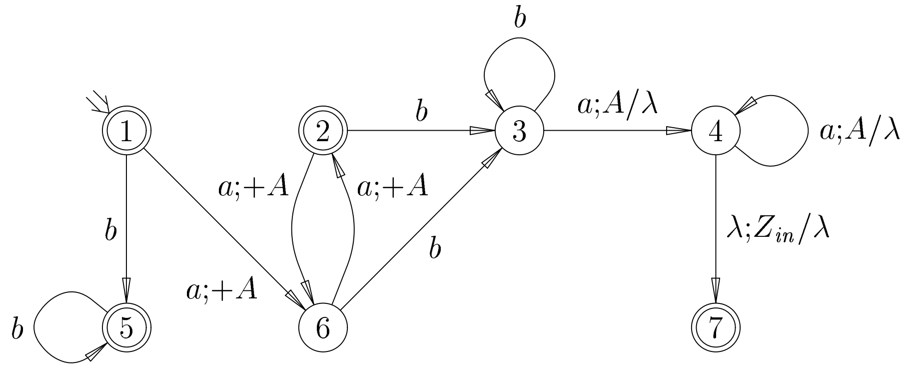
Toestandsverzameling $Q \times (\Gamma \cup \{\lambda\})$, begintoestand $\langle q_{in}, Z_{in}\rangle$, eindtoestanden $Q \times \{\lambda\}$,
$(p, a, Z) \mapsto (q, X)$ geeft $(\langle p, Z\rangle, a, \langle q, X\rangle)$, $\quad (X \in \Gamma \cup \{\lambda\})$.

Er zullen zo λ-overgangen ontstaan in de eindige automaat, maar die zijn weg te werken.

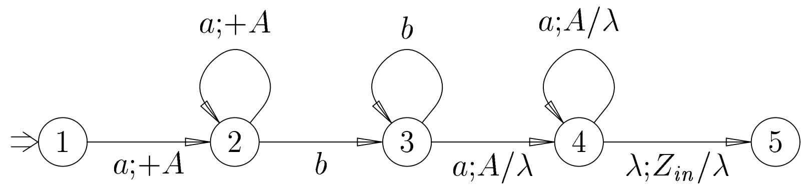
5.9 .



5.10a De taal $K_1 = \{\ a^n b^m a^n \mid m, n \in \mathbb{N}\ \}$ is duidelijk niet prefix-vrij. Dus $K_1 \notin \mathsf{DSA_e}$. Maar $K_1 \in \mathsf{DSA_f}$, want $K_1 = L_f(\mathcal{A})$ voor de deterministische stapelautomaat $\mathcal{A}$:



De taal $\{\ a^n b^m a^n \mid m, n \geq 1\ \}$ zit wél in $\mathsf{DSA_e}$, en wordt door middel van de lege stapel geaccepteerd door de volgende deterministische stapelautomaat:



b Deze taal zit niet in $\mathsf{DSA_e}$ (niet prefix-vrij) maar wel in $\mathsf{DSA_f}$. De taal $\{\ a^n b^m c^m \mid m, n \geq 1\ \}$ zit in $\mathsf{DSA_e}$.

c De taal $L = \{\ a^n b^m \mid n \leq m \leq 2n\ \}$ uit Opgave 5.1c is niet deterministisch te accepteren, maar dat is vooralsnog alleen intuitief te beredeneren.

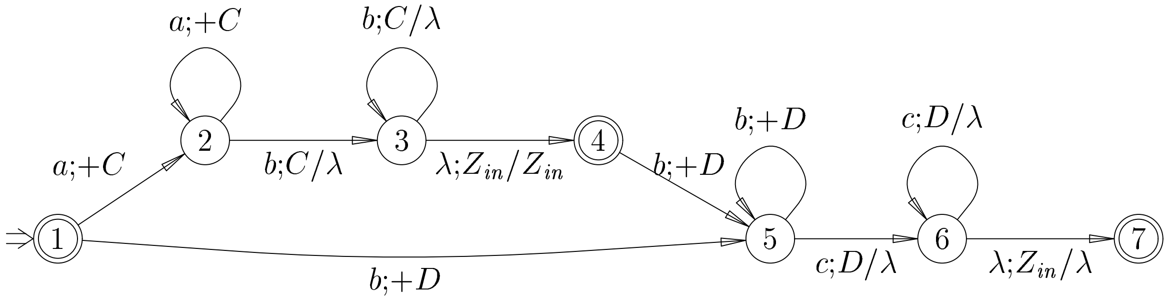
Een echt bewijs is nog wat werk: Voer een nieuwe operatie op talen over $\{a, b\}$ in, gemodelleerd op de operatie pre. Deze operatie verandert, als een prefix van een woord tot de taal behoort, alle letters ná dat prefix in de andere letter. Formeel :

swap$(K) = \{ \; wh(y) \mid w \in K$, geen echte prefix van w in $K, wy \in K \; \}$,
waarbij $h(a) = b, h(b) = a$.

Er geldt swap$(L) = \{ \; a^n b^n a^k \mid k \leq n \; \}$; deze taal is niet context-vrij.

De deterministische stapeltalen (acceptatie met eindtoestand) zijn gesloten onder swap: Een gegeven deterministische stapelautomaat passen we aan door twee kopieen van de automaat te nemen; in de tweede kopie zijn alle a's en b's in elkaar veranderd. Zolang nog geen eindtoestand is aangedaan blijven we in de eerste kopie; bij een opdracht die naar een eindtoestand leidt stappen we over naar de tweede kopie.

d De taal $K_2 = \{ \; a^k b^l c^m \mid k, l, m \in \mathbb{N}, k + m = l \; \}$ is duidelijk niet prefix-vrij. Dus $K_2 \notin \mathsf{DSA_e}$. Maar $K_2 \in \mathsf{DSA_f}$, want $K_2 = L_f(\mathcal{A})$ voor de deterministische stapelautomaat $\mathcal{A}$:

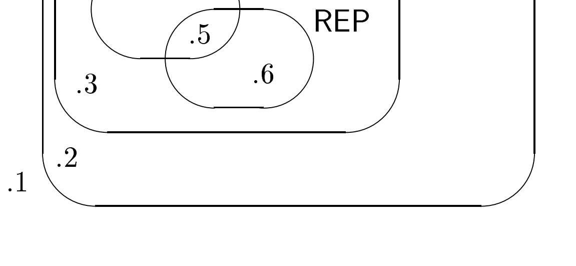


5.11a Gegeven een deterministische stapelautomaat $\mathcal{A}$ met $L_f(\mathcal{A}) = L$. Laat $\mathcal{A}'$ de deterministische stapelautomaat zijn die uit $\mathcal{A}$ verkregen wordt door alle overgangen (p, a, Z, q, w) met $p \in F$ te verwijderen. Dan is $L_f(\mathcal{A}') = \min(L)$.

b $\min(L)$ is prefix-vrij.

c $\min(L) = \{ \; a^i b^j c^k \mid i, j \geq 1, k = \min(i, j) \; \}$ is niet context-vrij volgens Opgave 4.14 (K_4). Stel $L \in \mathsf{DSA_f}$. Dan $\min(L) \in \mathsf{DSA_f}$ volgens **a**, en dus $\min(L) \in \mathsf{CF}$, een tegenspraak. Dus $L \notin \mathsf{DSA_f}$.

5.12 CF=SA, REP=REG=RLIN=DREP, REP $\cup$ $\mathsf{DSA_e}$ $\subset$ $\mathsf{DSA_f}$ $\subset$ CF (echte inclusie), en $\mathsf{DSA_e}$ $-$ REP, REP $\cap$ $\mathsf{DSA_e}$, REP $-$ $\mathsf{DSA_e}$ zijn niet leeg.



Talen uit de diverse gebieden:

1. $\{\ a^n b^n c^n \mid n \in \mathbb{N}\ \} \notin \mathsf{CF}$.
2. $\{\ a^i b^j c^k \mid i \neq j \vee j \neq k\ \} \in \mathsf{CF} - \mathsf{DSA_f}$.
3. $\{\ a^n b^m \mid 1 \leq m \leq n\ \} \in \mathsf{DSA_f} - (\mathsf{REP} \cup \mathsf{DSA_e})$.
4. $\{\ a^n b^n \mid n \geq 1\ \} \in \mathsf{DSA_e} - \mathsf{REP}$.
5. $a^*b \in \mathsf{REP} \cap \mathsf{DSA_e}$.
6. $a^* \in \mathsf{REP} - \mathsf{DSA_e}$.

5.13 Het complement van de taal doorsnijden we met $a^*b^*c^*$ en we krijgen $\{\ a^m b^n c^p \mid m \geq n \geq p\ \}$. Deze taal is niet context-vrij (pomplemma).

Zou de gegeven taal in $\mathsf{DSA_f}$ zitten, dan ook het complement, want $\mathsf{DSA_f}$ is gesloten onder complement. Meer algemeen is het complement dan context-vrij en de doorsnijding met $a^*b^*c^*$ ook. Omdat dit laatste niet het geval is, is de gegeven taal geen element van $\mathsf{DSA_f}$.

5.14 $\mathrm{mir}(L(G))$ wordt gegenereerd door de cf grammatica met producties $S \to c_i S\, \mathrm{mir}(w_i)$ voor $1 \leq i \leq n$, en $S \to \lambda$. Een deterministische stapelautomaat die $\mathrm{mir}(L(G))$ accepteert (met eindtoestanden) heeft $Q = \{1,2,3,4\}$, $q_{in} = 1$, $F = \{1,4\}$, $\Gamma = \Delta \cup \{Z_{in}\}$, en de volgende overgangen:

$(1, c_i, Z_{in}) \mapsto (1, Z_{in})$, voor alle i met $w_i = \lambda$,
$(1, c_i, Z_{in}) \mapsto (2, \mathrm{mir}(w_i) Z_{in})$, voor alle i met $w_i \neq \lambda$,
$(2, c_i, X) \mapsto (2, \mathrm{mir}(w_i) X)$ voor $1 \leq i \leq n$, $X \in \Gamma$,
$(2, a, a) \mapsto (3, \lambda)$, voor $a \in \Delta$,
$(3, a, a) \mapsto (3, \lambda)$, voor $a \in \Delta$, en
$(3, \lambda, Z_{in}) \mapsto (4, \lambda)$.

Dus $\mathrm{mir}(L(G)) \in \mathsf{DSA_f}$, en dus ook $(\Delta \cup \Delta')^* - \mathrm{mir}(L(G)) \in \mathsf{DSA_f}$, want $\mathsf{DSA_f}$ is gesloten onder het nemen van het complement.

Dus is $(\Delta \cup \Delta')^* - L(G) = \mathrm{mir}(\ (\Delta \cup \Delta')^* - \mathrm{mir}(L(G))\ ) \in \mathsf{CF}$, want CF is gesloten onder het nemen van het spiegelbeeld.

5.15a $L(G) = \{a^n a^m c^m b^n a^k c^k \mid n, m, k \geq 0\}$.

b. Het bepalen van de look-ahead verzamelingen voor elk van de producties gaat als volgt. Eerst bepalen we first$_1$ 'F' en follow$_1$ 'FL' :

F	a	b	c	S	A	B
0	a	b	c	$\emptyset$	a	a, λ
1	a	b	c	a	a, λ	a, λ
2	a	b	c	a, λ	a, λ	a, λ

FL	S	A	B
0	λ	a, b	c
1	λ	a, b, λ	c, λ, a, b

En dan

$A \to w$	$\text{first}_1(w)$	$\text{follow}_1(A)$	$LA_1(A \to w)$
$S \to AB$	a, λ	λ	a, λ
$A \to aAb$	a	a, b, λ	a
$A \to B$	a, λ	a, b, λ	a, b, λ
$B \to aBc$	a	a	a
$B \to \lambda$	λ	a, b, c, λ	a, b, c, λ

Dus G is niet sterk $LL(1)$, want $LA_1(A \to aAb) \cap LA_1(A \to B) = \{a\} \neq \emptyset$ (en $LA_1(B \to aBc) \cap LA_1(B \to \lambda) \neq \emptyset$).

5.16 *(jan 2002) De opgave is gewijzigd. Verander de productie $A \to Ab$ in $A \to Ad$.*

a G is gereduceerd en bevat een links-recursieve productie, dus G is niet sterk $LL(1)$, volgens Lemma 5.37.

b Links-recursie verwijderen levert G' :

$A \to aB \mid aBX$
$X \to cX \mid dX \mid c \mid d$
$B \to bBc \mid \lambda$

Factorizatie levert nu G'' :

$A \to aBZ$
$X \to cZ \mid dZ$
$Z \to X \mid \lambda$
$B \to bBc \mid \lambda$

Nu de look-ahead verzamelingen bepalen :

F	A	X	Z	B
0	a	c, d	λ	b, λ
1	a	c, d	λ, c, d	b, λ

FL	A	X	Z	B
0	λ	$\emptyset$	$\emptyset$	c, d
1	λ	$\emptyset$	λ	c, d, λ
2	λ	λ	λ	c, d, λ

En dan

$A \to w$	$\text{first}_1(w)$	$\text{follow}_1(A)$	$LA_1(A \to w)$
$A \to aBZ$	a	λ	a
$X \to cZ$	c	λ	c
$X \to dZ$	d	λ	d
$Z \to X$	c, d	λ	c, d
$Z \to \lambda$	λ	λ	λ
$B \to bBc$	b	c, d, λ	b
$B \to \lambda$	λ	c, d, λ	c, d, λ

G'' is dus $LL(1)$ omdat voor ieder tweetal producties $A \to w_1$ en $A \to w_2$ met $w_1 \neq w_2$ geldt dat $LA_1(A \to w_1) \cap LA_1(A \to w_2) = \emptyset$.

5.17a De grammatica is niet sterk $LL(1)$: aan de producties voor S is te zien dat minstens look-ahead 3 nodig is om te beslissen welke productie voor S toegepast moet worden.

b. We passen eerst factorizatie toe :

$S \to aX$
$X \to A \mid bY$
$Y \to B \mid cC$
$A \to aA \mid \lambda$
$B \to bB \mid \lambda$
$C \to cC \mid \lambda$

Het bepalen van de first$_1$-, follow$_1$- en look-ahead verzamelingen levert dan het volgende.

F	S	X	Y	A	B	C
0	a	b	c	a, λ	b, λ	c, λ
1	a	b, a, λ	c, b, λ	a, λ	b, λ	c, λ
2	a	b, a, λ	c, b, λ	a, λ	b, λ	c, λ

FL	S	X	Y	A	B	C
0	λ	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
1	λ	λ	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
2	λ	λ	λ	λ	$\emptyset$	$\emptyset$
3	λ	λ	λ	λ	λ	λ
4	λ	λ	λ	λ	λ	λ

En dan

$A \to w$	first$_1(w)$	follow$_1(A)$	$LA_1(A \to w)$
$S \to aX$	a	λ	a
$X \to A$	a, λ	λ	a, λ
$X \to bY$	b	λ	b
$Y \to B$	b, λ	λ	b, λ
$Y \to cC$	c	λ	c
$A \to aA$	a	λ	a
$A \to \lambda$	λ	λ	λ
$B \to bB$	b	λ	b
$B \to \lambda$	λ	λ	λ
$C \to cC$	c	λ	c
$C \to \lambda$	λ	λ	λ

Voor ieder tweetal producties $A \to w_1$ en $A \to w_2$ met $w_1 \neq w_2$ geldt dat $LA_1(A \to w_1) \cap LA_1(A \to w_2) = \emptyset$, dus de grammatica is sterk $LL(1)$.