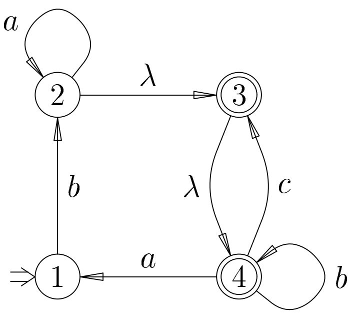
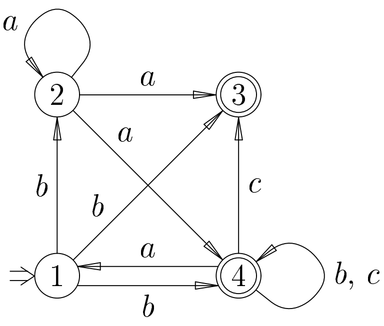
1) **a** De automaat bevat λ-takken. Deze geven keten-regels bij het omzetten naar een grammatica. Omdat dit niet toegestaan is moeten we eerst λ-takken verwijderen (of later keten-regels elimineren). Na verwijdering van λ-takken met de $e\lambda$-methode vinden we een automaat als hieronder (tweede plaatje).



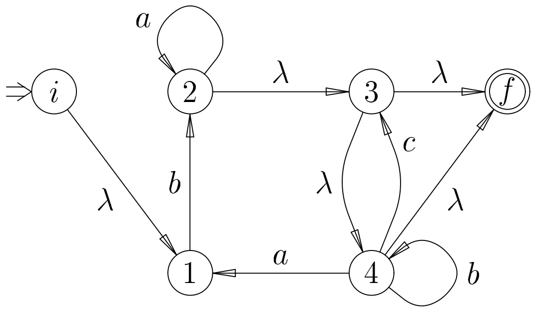


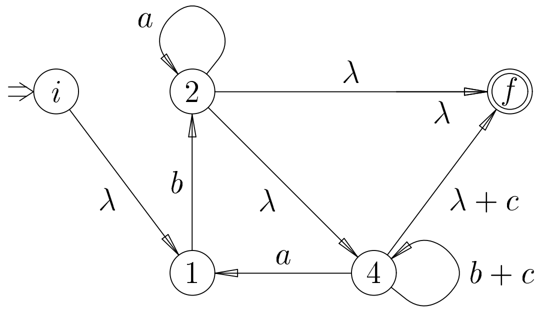
Uit het plaatje is nu de grammatica af te lezen; laat S, A, B, C corresponderen met toestanden $1, 2, 3, 4$.

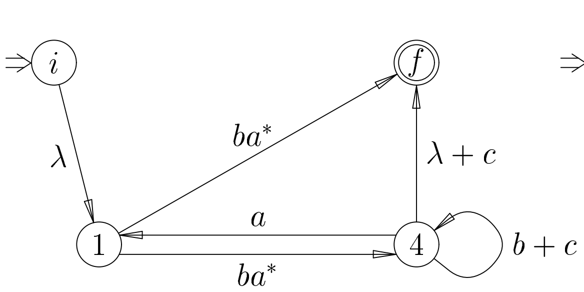
Axioma S, producties

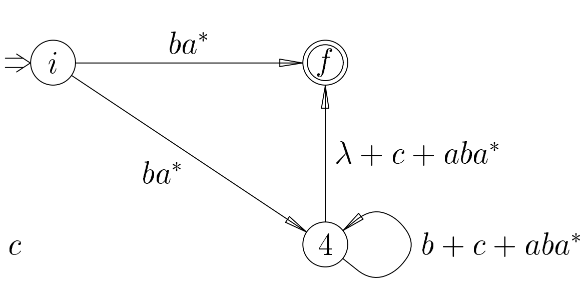
$S \to bA \mid bB \mid bC$
$A \to aA \mid aB \mid aC$
$B \to \lambda$
$C \to aS \mid cB \mid bC \mid cC \mid \lambda$.

b $L(\mathcal{A})$: Voeg toestand i en f toe als nieuwe begin- en eindtoestand, aan de originele automaat. Verwijder achtereenvolgens 3, 2 en 1. Als we daarna ook 4 verwijderen vinden we de expressie $ba^* + ba^*(b + c + aba^*)^*(\lambda + c + aba^*)$. Deze is equivalent met $ba^*(b + c + aba^*)^*$.









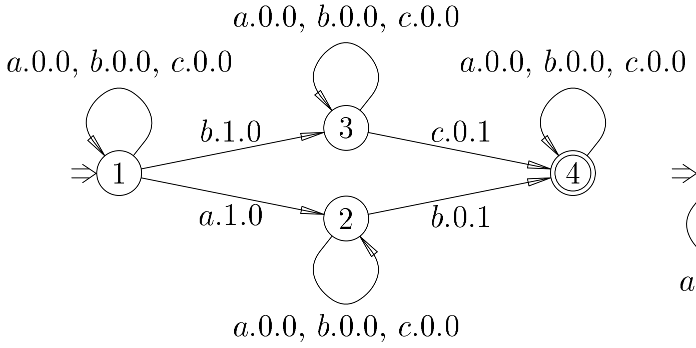
$\{aubvc \mid u \in L(\mathcal{A}), v \in a^*\}$: Ingewikkelde notatie voor de taal $a \cdot L(\mathcal{A}) \cdot b \cdot a^* \cdot c$. Expressie dus $a\rho ba^*c$, waarbij ρ de gevonden expressie voor $L(\mathcal{A})$ is.

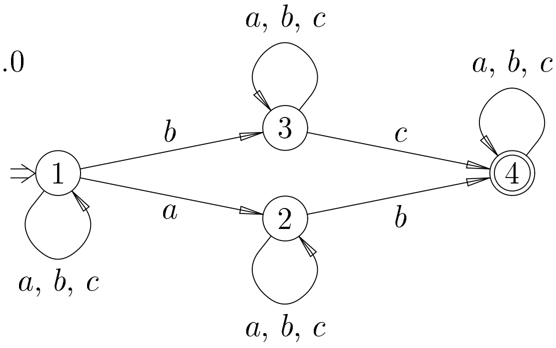
2) **a** Er zijn geen twee posities met c's waartussen alleen a's te vinden zijn:

$$\neg(\exists i)(\exists j)[\ i < j \wedge c(i) \wedge c(j) \wedge (\forall k)((i < k \wedge k < j) \to a(k))\ ]$$

b Formule ϕ_2 heeft twee vrije variabelen, i en j. Letters in de automaat coderen dus letters uit de taal én de posities van i en j in het woord. Omdat gegeven is dat $i < j$ wordt het aantal mogelijkheden beperkt. (De notatie $\sigma.i.j$ geeft hier een vector van de drie componenten weer omdat dat sneller typt.)

Formule ϕ_3 kwantificeert ϕ_2. Daarmee verdwijnen de i, j-componenten.





3) **a** Verwijder linksrecursie voor S, voer de 'alias' B voor b in:

$S \to b \mid bX, \quad X \to aT \mid aTX, \quad T \to aTB \mid b, \quad B \to b.$

b G_1 is niet sterk LL(1) vanwege de productie $S \to SaT$, G_2 is niet sterk LL(1) vanwege $S \to b \mid bX$ (of $X \to aT \mid aTX$). Zie Lemma 5.37.

Factorizatie is het afsplitsen van gelijke prefixen van rechterkanten van producties voor dezelfde letter, hier voor S en X van toepassing.

$S \to bU, \quad X \to aTU, \quad T \to aTB \mid b, \quad B \to b, \quad U \to \lambda \mid X.$

c Bereken $\text{first}_1(\sigma)$ voor alle symbolen σ. Omdat de meeste rechterkanten van producties beginnen met een terminaal is dat eenvoudig: $\text{first}_1(S) = \{b\}$, $\text{first}_1(X) = \{a\}$, $\text{first}_1(T) = \{a, b\}$, $\text{first}_1(B) = \{b\}$, $\text{first}_1(U) = \{\lambda\} \cup \text{first}_1(X) = \{\lambda, a\}$.

Bereken $\text{follow}_1(\sigma)$ voor niet-terminalen σ, met de 'benaderingen' $FL_i(\sigma)$.

Er geldt $FL_0(S) = \{\lambda\}$ (S is axioma, komt rechts niet voor), $FL_0(X) = \emptyset$, $FL_0(T) = \{a, b\}$ (vanwege $X \to aTU$, $T \to aTB$), $FL_0(B) = \emptyset$, $FL_0(U) = \emptyset$.

$FL_{i+1}(S) = FL_i(S)$, want S komt rechts niet voor

$FL_{i+1}(X) = FL_i(X) \cup FL_i(U)$, vanwege $U \to X$,

$FL_{i+1}(T) = FL_i(T) \cup FL_i(X)$, vanwege $X \to aTU$, $\lambda \in \text{first}_1(U)$,

$FL_{i+1}(B) = FL_i(B) \cup FL_i(T)$, vanwege $T \to aTB$,

$FL_{i+1}(U) = FL_i(U) \cup FL_i(S) \cup FL_i(X)$, vanwege $S \to bU$, $X \to aTU$.

i	S	X	T	B	U
0	$\{\lambda\}$	$\emptyset$	$\{a, b\}$	$\emptyset$	$\emptyset$
1	$\{\lambda\}$	$\emptyset$	$\{a, b\}$	$\{a, b\}$	$\{\lambda\}$
2	$\{\lambda\}$	$\{\lambda\}$	$\{a, b\}$	$\{a, b\}$	$\{\lambda\}$
3	$\{\lambda\}$	$\{\lambda\}$	$\{\lambda, a, b\}$	$\{a, b\}$	$\{\lambda\}$
4	$\{\lambda\}$	$\{\lambda\}$	$\{\lambda, a, b\}$	$\{\lambda, a, b\}$	$\{\lambda\}$
5	$\{\lambda\}$	$\{\lambda\}$	$\{\lambda, a, b\}$	$\{\lambda, a, b\}$	$\{\lambda\}$

Hiermee berekenen we de look-ahead van de producties:

$A \to w$	$\text{first}_1(w)$	$\text{follow}_1(A)$	$\text{LA}_1(A \to w)$
$S \to bU$	$\{b\}$	$\{\lambda\}$	$\{b\}$
$X \to aTU$	$\{a\}$	$\{\lambda\}$	$\{a\}$
$T \to aTB$	$\{a\}$	$\{\lambda, a, b\}$	$\{a\}$
$T \to b$	$\{b\}$	$\{\lambda, a, b\}$	$\{b\}$
$B \to b$	$\{b\}$	$\{\lambda, a, b\}$	$\{b\}$
$U \to \lambda$	$\{\lambda\}$	$\{\lambda\}$	$\{\lambda\}$
$U \to X$	$\{a\}$	$\{\lambda\}$	$\{a\}$

Producties voor dezelfde niet-terminaal hebben steeds verschillende look-ahead; de grammatica is dus sterk LL(1). Merk op dat we alleen de follow_1-waarde van U gebruikt hebben (voor het berekenen van $\text{LA}_1(U \to \lambda)$).

4) **a** $S \to c$
$S \to aaaSaaa \mid aabSbaa \mid abaSaba \mid abbSbba$
$S \to baaSaab \mid babSbab \mid bbaSabb \mid bbbSbbb$

Of telkens tot drie tellen (met axioma S_0):

$S_0 \to aS_1a \mid bS_1b \mid c$
$S_1 \to aS_2a \mid bS_2b$
$S_2 \to aS_0a \mid bS_0b$

b Stel K is regulier, dan voldoet K aan het pomplemma voor de representeerbare/representeerbare talen voor zekere constante n. Neem nu $z = a^{3n}ca^{3n}$. Dan $z \in K$, en $|z| = 6n + 1 > n$. Er bestaat daarom een verdeling $z = uvw$ met $|uv| \leq n$, $v \neq \lambda$, zodat $uv^iw \in K$ voor elke i. Omdat $|uv| \leq n < 3n$, zal w het suffix ca^{3n} geheel bevatten, v bestaat geheel uit a's die in het oorspronkelijke woord z voor de c staan. Bekijk $uw = uv^0w$. Dit woord is van de vorm a^kca^{3n} met $k = 3n - |v| < 3n$, en zit dus niet in K (omdat $\text{mir}(a^{3n}) = a^{3n} \neq a^k$).

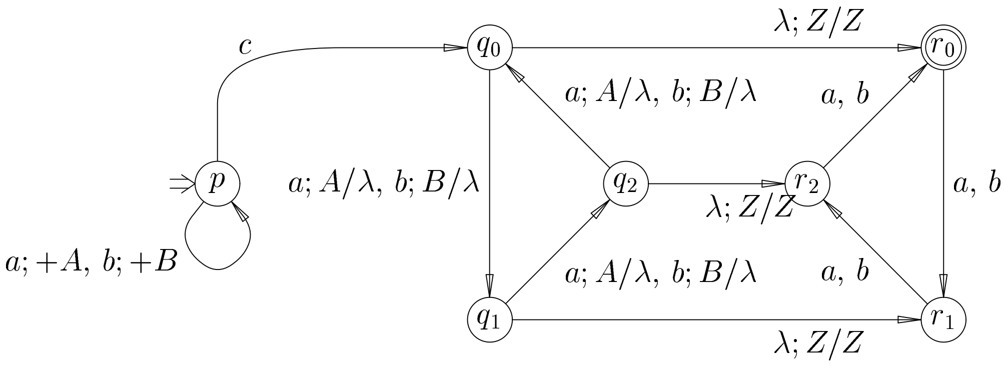
Conclusie: z kan niet binnen K gepompt worden; K is niet regulier.

5) **a** Doe de tweede grammatica uit de vorige opgave na; acceptatie dmv. lege stapel. Initieel stapelsymbool S_0, begintoestand p.

$(p, a, S_0) \mapsto (p, S_1A)$, $(p, b, S_0) \mapsto (p, S_1B)$, $(p, c, S_0) \mapsto (p, \lambda)$,
$(p, a, S_1) \mapsto (p, S_2A)$, $(p, b, S_1) \mapsto (p, S_2B)$,
$(p, a, S_2) \mapsto (p, S_0A)$, $(p, b, S_2) \mapsto (p, S_0B)$,
$(p, a, A) \mapsto (p, \lambda)$, $(p, b, B) \mapsto (p, \lambda)$.

b We zullen nu eerst stapelen, en bij het afstapelen van het spiegelbeeld gaan tellen. Na het spiegelbeeld mogen we alles lezen, zolang de lengte maar totaal een drievoud is. Acceptatie dmv. eindtoestand. Initieel stapelsymbool Z, begintoestand p, eindtoestand r_0. De instructies ($\gamma = A, B, Z$):

$(p, a, \gamma) \mapsto (p, A\gamma)$, $(p, b, \gamma) \mapsto (p, B\gamma)$,
$(p, c, \gamma) \mapsto (q_0, \gamma)$,
$(q_0, a, A) \mapsto (q_1, \lambda)$, $(q_0, b, B) \mapsto (q_1, \lambda)$, $(q_0, \lambda, Z) \mapsto (r_0, Z)$,
$(q_1, a, A) \mapsto (q_2, \lambda)$, $(q_1, b, B) \mapsto (q_2, \lambda)$, $(q_1, \lambda, Z) \mapsto (r_1, Z)$,
$(q_2, a, A) \mapsto (q_0, \lambda)$, $(q_2, b, B) \mapsto (q_0, \lambda)$, $(q_2, \lambda, Z) \mapsto (r_2, Z)$,
$(r_0, a, Z) \mapsto (r_1, Z)$, $(r_0, b, Z) \mapsto (r_1, Z)$,
$(r_1, a, Z) \mapsto (r_2, Z)$, $(r_1, b, Z) \mapsto (r_2, Z)$,
$(r_2, a, Z) \mapsto (r_0, Z)$, $(r_2, b, Z) \mapsto (r_0, Z)$.



c K zit in $\mathsf{DSA_e}$; we hebben er immers een deterministische stapelautomaat met lege stapel acceptatie voor gevonden.

K' zit niet in $\mathsf{DSA_e}$; de taal is niet prefix-vrij, want bijvoorbeeld zowel c als $caaa$ behoren tot de taal.

6) **a** Gegeven $\mathcal{A} = (Q, \{a, b\}, \delta, I, F)$. We construeren $\mathcal{B} = (Q', \{a, b\}, \delta', I', F')$ voor $l(L(\mathcal{A}))$ als volgt.

$Q' = Q \times \{1, 2\}$: elke toestand q krijgt twee kopieën, die we noteren als $q1$ en $q2$. In de eerste kopie is nog geen letter verkeerd gelezen, in de tweede kopie is dat wel gebeurd.

$I' = I \times \{1\}$: we starten in de eerste kopie van een begintoestand,

$F' = F \times \{2\}$: we eindigen in de tweede kopie van een eindtoestand.

Voor elke instructie (p, a, q) in δ, bevat δ' de instructies $(p1, a, q1)$, $(p2, a, q2)$ en $(p1, b, q2)$, en evenzo
voor elke instructie (p, b, q) in δ, bevat δ' de instructies $(p1, b, q1)$, $(p2, b, q2)$ en $(p1, a, q2)$.

Voor elke reguliere K construeren we een automaat voor $l(K)$ als boven; $l(K)$ is dus regulier. Merk op dat de reguliere talen gesloten zijn onder verschil, aangezien dat volgt uit de afsluiting onder doorsnede en complement (Stelling 3.20). Hiermee is ook $f(K) = l(K) - K$ regulier.

b Ga uit van een grammatica in Chomsky normaalvorm voor de taal K, met axioma S. Neem een extra kopie van elke niet-terminaal A, die we noteren als A'. Deze nieuwe kopie houdt bij dat nog een letter veranderd moet worden. Deze eigenschap wordt telkens aan één van de kinderen doorgegeven.

De nieuwe grammatica heeft axioma S', en bevat de producties

$A \to BC$, $A' \to B'C$, $A' \to BC'$, voor elke oorspronkelijke productie $A \to BC$,

$A \to a$, $A' \to b$ voor elke oorspronkelijke productie $A \to a$, en

$A \to b$, $A' \to a$ voor elke oorspronkelijke productie $A \to b$.

bb Alternatief, constructie als bij **a**, nu uitgevoerd voor stapelautomaten.

Opmerking, voor de liefhebber. Neem $K = \{\ a^k b^\ell c^m \mid k \neq \ell \text{ of } \ell \neq m\ \}$. Merk op dat voor alle $n \geq 1$: $a^n b^n c^n \in l(K)$, omdat $a^{n+1} b^{n-1} c^n \in K$. Dan is $f(K) \cap a^+ b^+ c^+ = \{\ a^n b^n c^n \mid n \geq 1\ \}$; deze taal is niet context-vrij, terwijl K dat wel is.