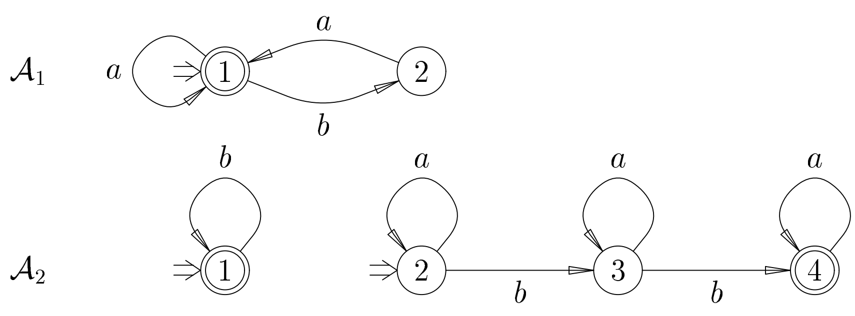
Uitwerkingen geselecteerde opgaven

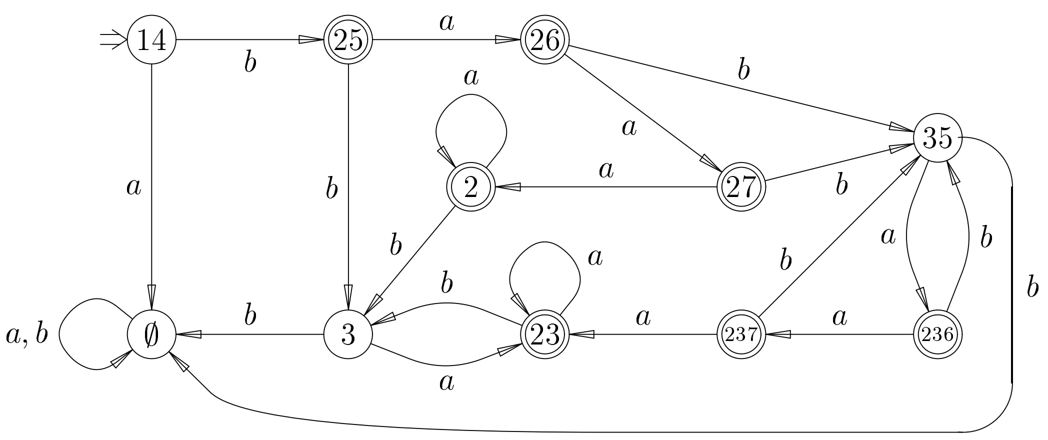
3.1 Onder elkaar: automaten voor K_1 en K_2. Deze laatste automaat bestaat inderdaad uit twee componenten: één voor woorden zonder a's, één voor woorden met precies twee b's. Een automaat voor K_3 ontstaat door die voor K_1 te nemen met daarbij óók 2 als eindtoestand.



3.2:1 We volgen het woord *babaa* vanuit de verzameling begintoestanden $\{1,4\}$.

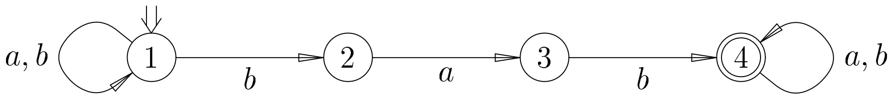
$\delta(\{1,4\},b) = \delta(1,b) \cup \delta(4,b) = \{2\} \cup \{5\}$.
$\delta(\{2,5\},a) = \delta(2,a) \cup \delta(5,a) = \{2\} \cup \{6\}$.
$\delta(\{2,6\},b) = \delta(2,b) \cup \delta(6,b) = \{3\} \cup \{5\}$.
$\delta(\{3,5\},a) = \delta(3,a) \cup \delta(5,a) = \{2,3\} \cup \{6\}$.
$\delta(\{2,3,6\},a) = \delta(2,a) \cup \delta(3,a) \cup \delta(6,a) = \{2\} \cup \{2,3\} \cup \{7\}$; acceptatie vanwege toestand 2, dus $babaa \in L(\mathcal{A})$. Verder is $\delta(\{2,3,7\},a) = \{2,3\}$ en $\delta(\{2,3\},b) = \{3\}$, dus $babaaab \notin L(\mathcal{A})$.

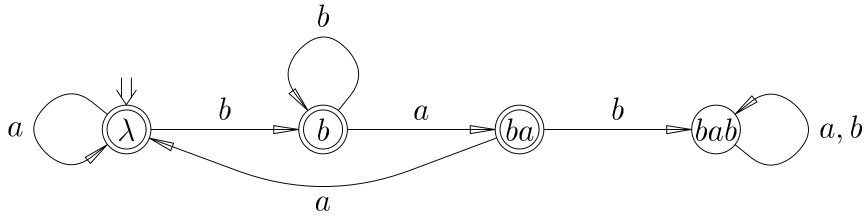
:2 De standaard constructie levert de volgende automaat:



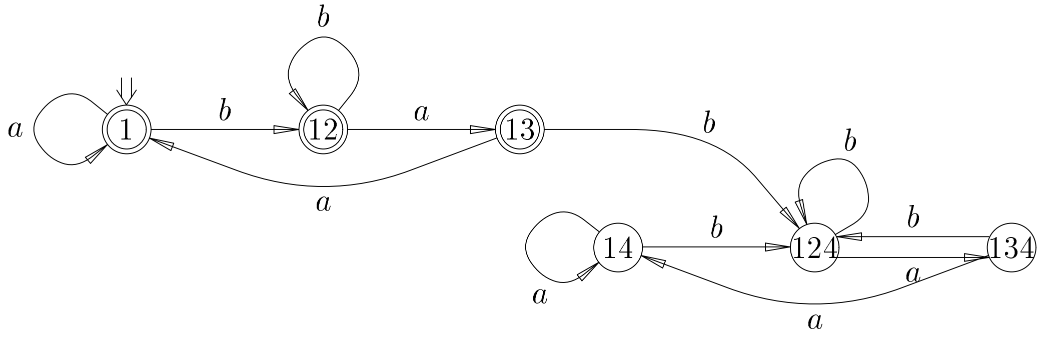
Het kan heel wat kompakter. Merk op dat de taal niet verandert wanneer we in de oorspronkelijke automaat (i) de rechter deelautomaat verwijderen (!), en (ii) de loop van "3" naar zichzelf weghalen. Dit levert een (niet totale) deterministische automaat met drie toestanden voor dezelfde taal.

3.3 Bij L_1 gokken we wanneer het deelwoord *bab* gelezen gaat worden, daarvoor en daarna mogen we alles lezen. Bij L_2 volgen we een andere taktiek. Om *bab* te vermijden houden we precies bij hoeveel we van *bab* gezien hebben; vgl. patroonherkenning volgens Knuth-Morris-Pratt.

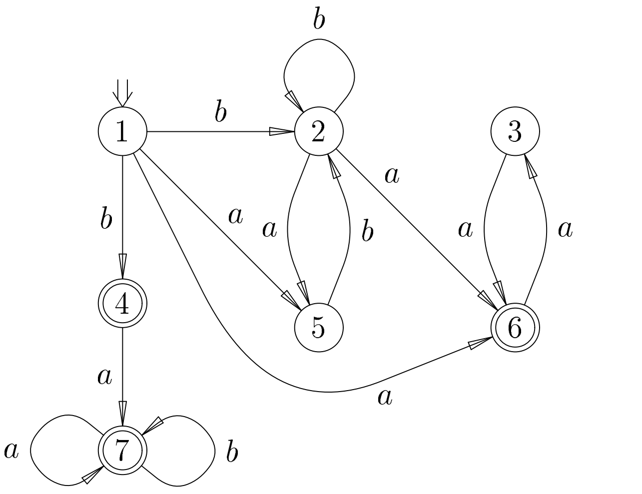


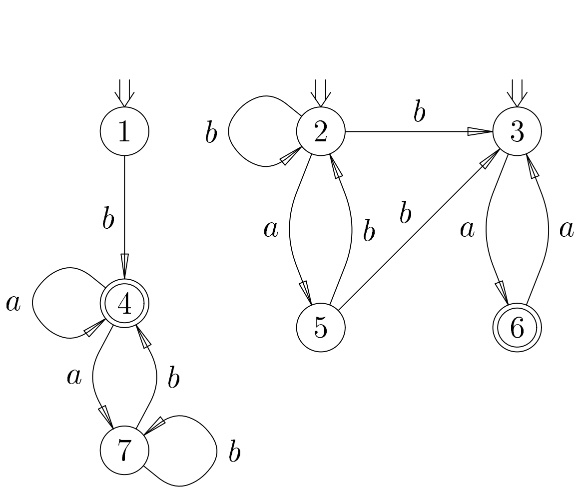


Uiteraard kan de toestand “bab” verwijderd worden. Een andere manier om een automaat voor L_2 te krijgen is als volgt: construeer uit de automaat voor L_1 (met de standaardconstructie) een totaal deterministische automaat voor L_1, en verwissel hierin de eind- en niet-eind-toestanden. Dit geeft een automaat die lijkt op de vorige, maar met een wat gecompliceerde verzameling nutteloze toestanden:

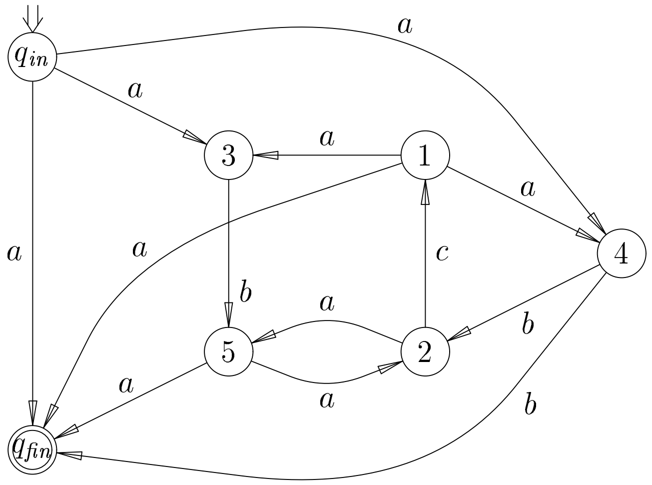


3.4 Met gebruik van de λe-methode, respectievelijk de $e\lambda$-methode:





3.5:1 Voeg nieuwe begin- en eindtoestand toe, en verbind die als volgt met de overige toestanden.



:2 Dan komen er twee takken bij: één van q_{in} naar 2 met label b, en één van q_{in} naar q_{fin} met label b.

3.7:2 We laten zien dat er geen constante n is waarvoor K_2 aan het pomplemma voldoet: voor elke n bestaat er een woord $z \in K_2$, met $|z| > n$, dat niet binnen de taal te pompen is.

Neem $n > 1$ en laat $z = a^{(n^2)} \in K_2$. Schrijf $z = uvw$ als in het pomplemma, dus $1 \leq |v| \leq |uv| \leq n$. Bekijk $z_2 = uv^2w$. Er geldt $|z_2| = n^2 + |v|$, dus $n^2 + 1 \leq |z_2| \leq n^2 + n$. Deze lengte kan geen kwadraat zijn, want $(n+1)^2 = n^2 + 2n + 1$ en dus is $|z_2|$ kleiner dan het eerst mogelijke kwadraat.

:3 Neem weer n en laat $z = a^n b^n \in K_3$. Schrijf $z = uvw$ als in het pomplemma, dus $1 \leq |v| \leq |uv| \leq n$. Bekijk $z_0 = uv^0w = uv$. Het 'pompwoord' v ligt geheel binnen het segment met a's, dus geldt $z_0 = a^k b^n$ met $k < n$. Dit woord zit niet in K_3.

:1 Rechtstreeks met het pomplemma, precies als in **:3** hierboven. Alternatief met behulp van afsluitingseigenschappen. In Gevolg 3.19 is bewezen dat $L = \{\ a^m b^m \mid m \geq 0\ \}$ niet representeerbaar is. Als gevolg hiervan kan ook K_1 niet representeerbaar zijn, omdat anders $L = K_1 \cap a^*b^*$ representeerbaar zou moeten zijn, vanwege de geslotenheid van REP onder doorsnede.

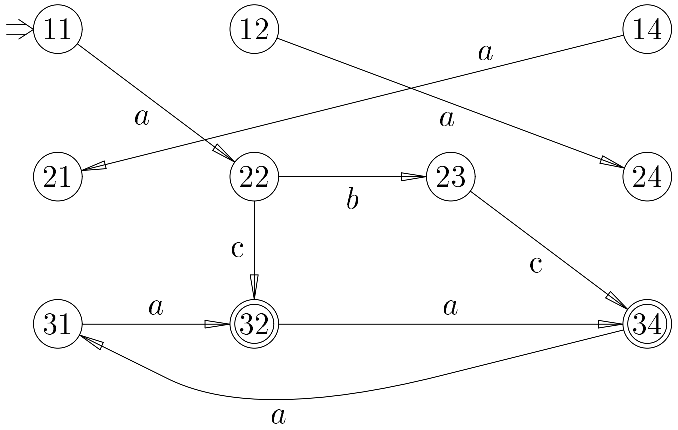
:4 Analoog aan **:3**, maar met $z = a^n b a^n$.

Merk op: $\mathrm{PAL} \cap a^*ba^* = \{a^n b a^n \mid n \geq 0\}$.

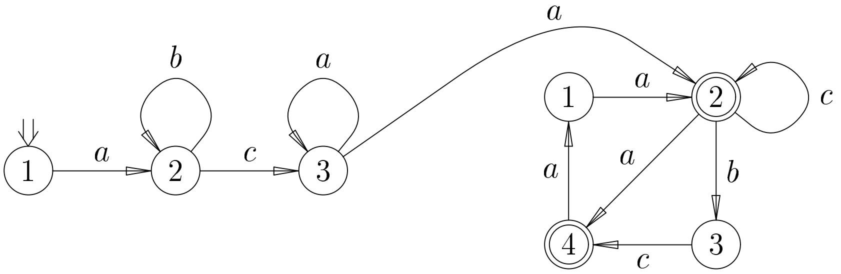
3.8 K is pompbaar, met constante $n = 1$. Woorden uit $\{a, b\}^*$ zijn duidelijk binnen K te pompen. Kijk dus naar een woord $z = c^k a^\ell b^\ell$ in K, met $k \geq 1$. Zo'n woord is pompbaar met $u = \lambda$, $v = c$, en $w = c^{k-1} a^\ell b^\ell$.

K is niet representeerbaar. Bekijk een nieuwe taal $K' = K \cap c\{a, b\}^* = \{\ c a^\ell b^\ell \mid \ell \geq 0\ \}$. Deze taal is niet pompbaar (vgl. het bewijs voor $\{\ a^m b^m \mid m \geq 0\ \}$) en dus niet representeerbaar. Omdat de representeerbare talen gesloten zijn onder doorsnede, en $c\{a,b\}^*$ duidelijk representeerbaar is, kan de oorspronkelijke taal K niet representeerbaar zijn.

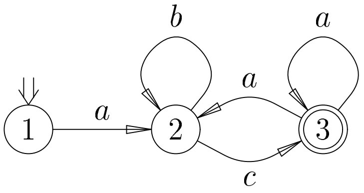
3.10a In de 'product-automaat' correspondeert toestand ij met toestand i in automaat $\mathcal{A}_1$ en j in $\mathcal{A}_2$. De constructie levert zes nutteloze toestanden op (nl. ook "13" en "33", die niet getekend zijn).

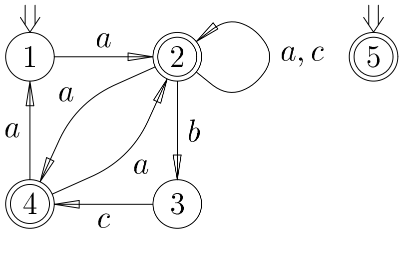


b Een automaat voor $L(\mathcal{A}_1) \cdot L(\mathcal{A}_2)$, verkregen met de λe-methode:

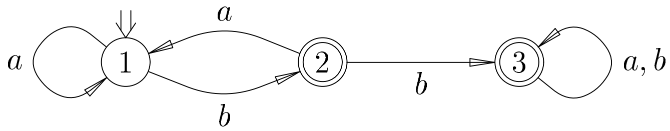


Automaten voor $L(\mathcal{A}_1)^+$ en $L(\mathcal{A}_2)^*$, weer met de λe-methode:

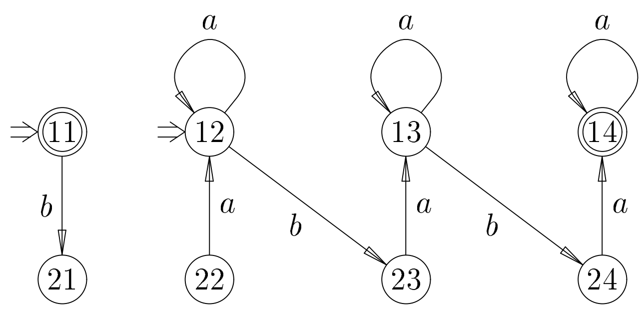




3.11a Automaat voor $\{a,b\}^* - K_1$ (eerst de oorspronkelijke automaat totaal maken):

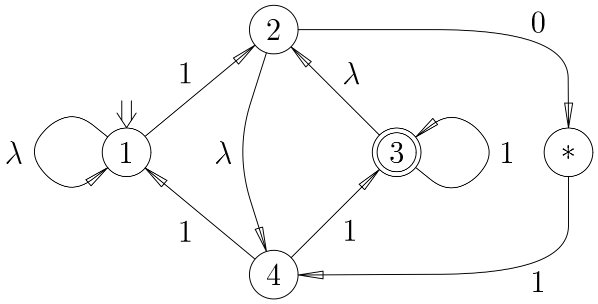


Automaat voor $K_1 \cap K_2$ (toestanden 21 en 22 zijn nutteloos):

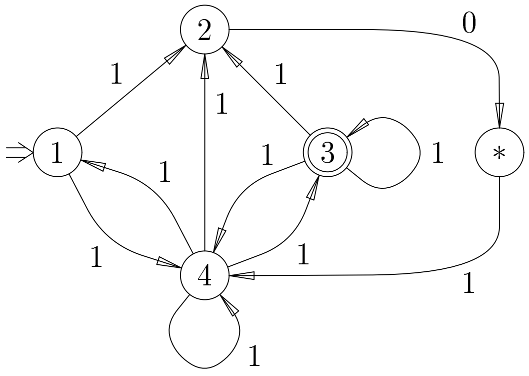


3.12 Bij doorsnede kiezen we als eindtoestanden van de product-automaat $F_1 \times F_2$, bij vereniging $(F_1 \times Q_2) \cup (Q_1 \times F_2)$. Deze constructie werkt alleen als de automaten voor elk woord tenminste één pad hebben vanuit een begintoestand.

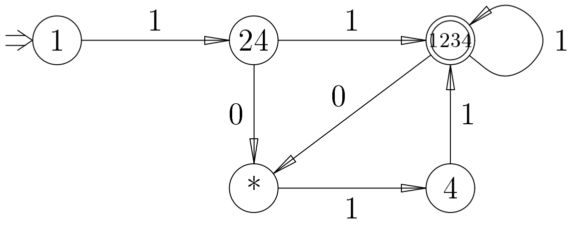
3.14b Vervang elk label door zijn beeld. Wanneer er twee of meer letters langs een tak staan splitsen we deze tak; dit is gebeurd met de tak van '2' naar '4' met de string '01'.



Nu elimineren we de λ-takken, bijvoorbeeld met de $e\lambda$-methode.



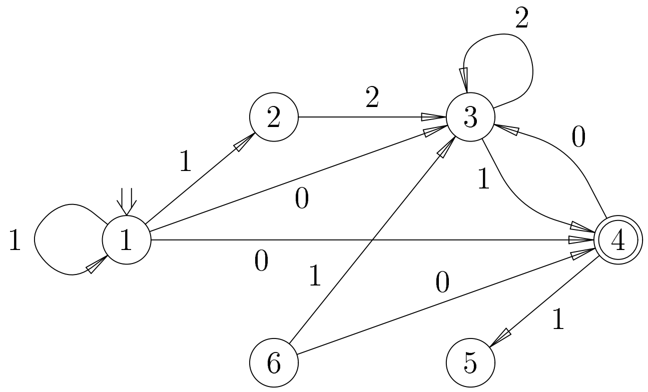
De kluwen van takken met label 1 kan nog sterk vereenvoudigd worden. Gebruik het standaard algoritme om de automaat deterministisch te maken (met weglating van nutteloze toestanden). We vinden:



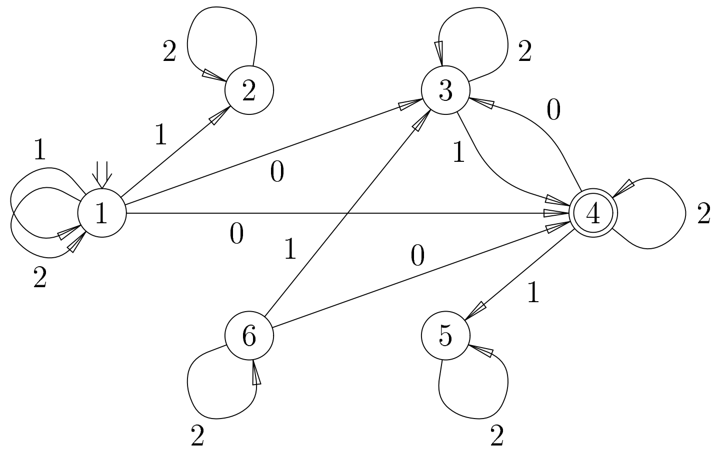
Deze automaat accepteert $\{1\} \cdot \{1, 011\}^+$.

3.15a Als $\mathcal{A}$ een eindige automaat met alfabet Σ is, en $h : \Delta^* \to \Sigma^*$ een homomorfisme, dan construeren we een automaat voor $h^{-1}(L(\mathcal{A}))$ door de toestanden van $\mathcal{A}$ over te nemen, evenals de begin- en eindtoestanden. Telkens wanneer de oorspronkelijke automaat een pad gelabeld $h(a)$ van knoop p naar q heeft krijgt de nieuwe automaat een tak met label a van p naar q, dit voor elke $a \in \Delta$.

Toestanden 5 en 6 zijn nutteloos; er bestaat geen pad dat vanuit 5 een eindtoestand bereikt, toestand 6 wordt niet vanuit de begintoestand bereikt.



b Net als hierboven, met de opmerking dat een pad met label λ alleen een pad ter lengte nul kan zijn, van een willekeurige knoop naar zichzelf.



3.18 Voor de talen bepalen we *reguliere expressies*. Tussen ⟨haakjes⟩ steeds de gebruikte verwijdervolgorde van de toestanden van de getekende automaten.

3.1 Zie de automaten in de uitwerkingen.

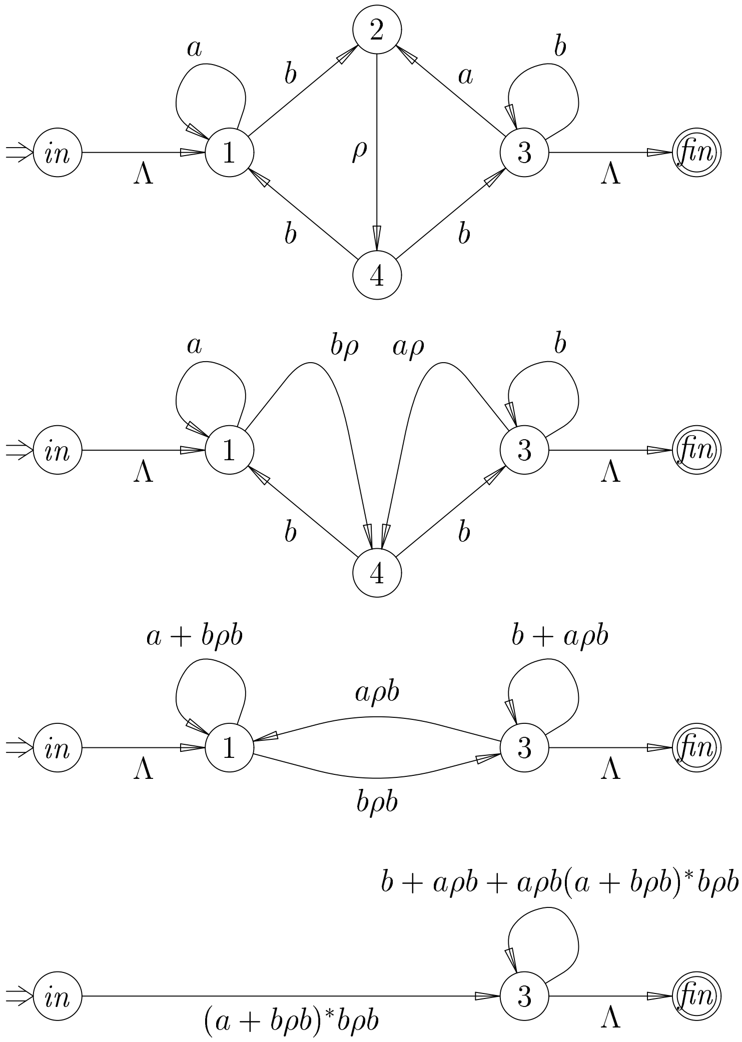
1:	$(a + ba)^*$	$\langle 2, 1 \rangle$
2:	$b^* + a^*ba^*ba^*$	$\langle 1, 2, 3, 4 \rangle$
3:	$(a + ba)^*(\Lambda + b)$	$\langle 2, 1 \rangle$

3.2 $b(a + ba^*a)^* + ba((b + ab)a)^*$ $\langle 3, 2, 1, 7, 5, 6, 4 \rangle$

3.10 1: ab^*ca^* $\langle 1,2,3\rangle$

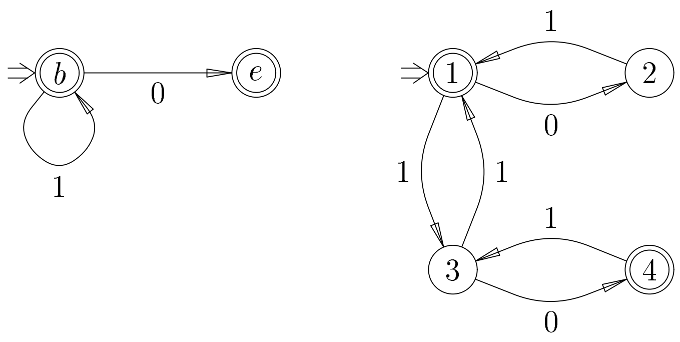
2: $a(c+(a+bc)aa)^*(\Lambda+a+bc)$ $\langle 3,4,1,2\rangle$

3.14 $(a+b(a{+}c)b)^*b(a{+}c)b\ [\ b+a(a{+}c)b+a(a{+}c)b(a+b(a{+}c)b)^*b(a{+}c)b\ ]^*$
$\langle 2,4,1,3\rangle$, als volgt (ρ staat voor $a{+}c$):

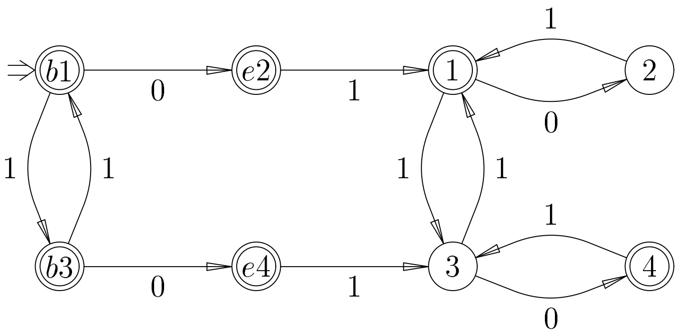


3.15 $a^*(ab+ba)b^*a(abab^*a)^*$ $\langle 2,5,6,1,3,4\rangle$

3.19 $L = L_1 \cup L_2$, waarbij L_1 uit alle woorden bestaat die geen deelwoord 00 én geen deelwoord 01 bevatten, en L_2 uit alle woorden bestaat die geen deelwoord 00 bevatten én een even lengte hebben. Een niet-deterministische automaat voor L is hieronder getekend, waarbij het rechtergedeelte verkregen is door de product-constructie toe te passen op automaten voor de taal van alle woorden zonder deelwoord 00 én de taal van alle woorden van even lengte.



a Deterministisch maken van bovenstaande automaat geeft de automaat



b Een reguliere expressie voor L_1 is $1^*(\Lambda + 0)$. Een reguliere expressie voor L_2 kan met de methode van Brzozowski en McCluskey verkregen worden door bijvoorbeeld achtereenvolgens de toestanden 2,3,4,1 te verwijderen. Dat geeft $(01 + 11 + 10(10)^*11)^*(\Lambda + 10(10)^*)$.

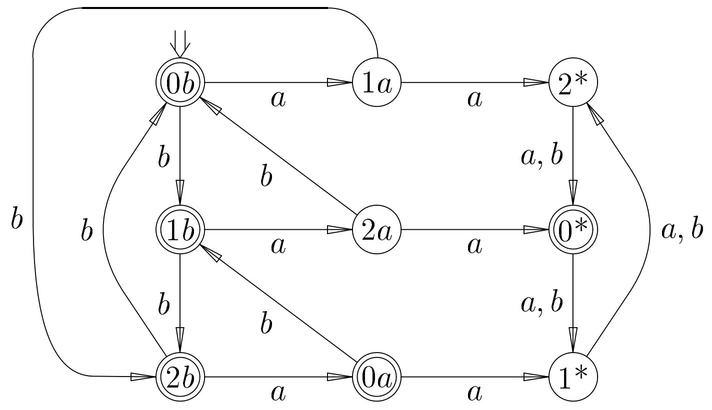
Een reguliere expressie voor L is dus $1^*(\Lambda+0)+(01+11+10(10)^*11)^*(\Lambda+10(10)^*)$.

3.20b Ook als $|w|$ wél een drievoud is mag eventueel elke a door een b gevolgd worden – een simpele expressie is derhalve $((a+b)(a+b)(a+b))^* \ + \ (b+ab)^*$.

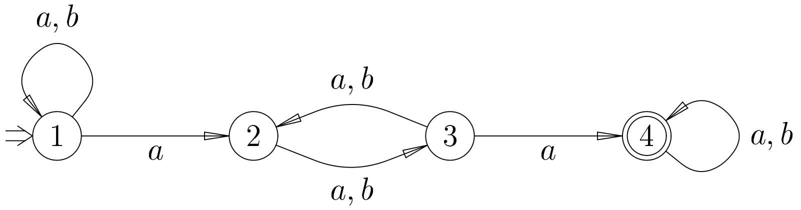
a Maak een automaat voor drievouden en een automaat voor $(b+ab)^*$, en maak de vereniging deterministisch. U krijgt zoiets:

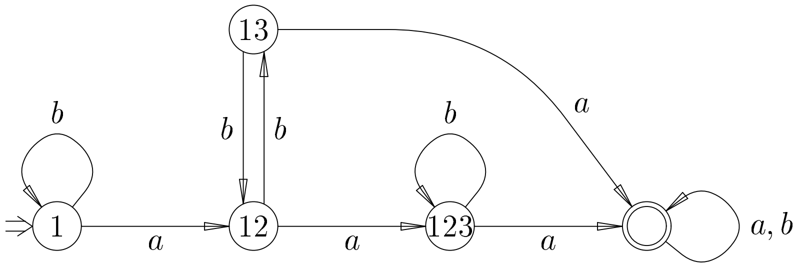
Laat $\mathcal{A} = (Q, \Delta, \delta, I, F)$ met $Q = \{0,1,2\} \times \{b, a, *\}$, $\Delta = \{a, b\}$, $I = \{\langle 0, b\rangle\}$, $F = \{\langle 0,b\rangle, \langle 0,a\rangle, \langle 0,*\rangle, \langle 1,b\rangle, \langle 2,b\rangle\}$;
δ bevat de volgende overgangen – met $i\flat$ bedoelen we hier $i+1$ (modulo 3), dus $0\flat = 1$ $1\flat = 2$ en $2\flat = 0$:
$(\langle i,b\rangle, a, \langle i\flat, a\rangle)$, $\quad(\langle i,b\rangle, b, \langle i\flat, b\rangle)$,
$(\langle i,a\rangle, a, \langle i\flat, *\rangle)$, $\quad(\langle i,a\rangle, b, \langle i\flat, b\rangle)$,
$(\langle i,*\rangle, a, \langle i\flat, *\rangle)$, $\quad(\langle i,*\rangle, b, \langle i\flat, *\rangle)$, $\quad$ voor $i = 0,1,2$.

Dat is de volgende automaat:



3.22a Eerst niet-deterministisch (boven), dan deterministisch gemaakt, en tenslotte de toestanden 124, 134 en 1234 tot één eindtoestand samengetrokken (onder).





b Expressie voor L, afgeleid uit de niet-deterministische automaat: $(a+b)^*a[(a+b)(a+b)]^*(a+b)a(a+b)^*$ (goed kijken, of $\langle 1,4,3,2\rangle$).

Expressie voor L, afgeleid uit de deterministische automaat: $b^*a(bb)^*(ab^*a+ba)(a+b)^*$ (verwijdervolgorde $\langle 1, 13, 123, 12, -\rangle$) en voor $\{a,b\}^*-L$: $b^*+b^*a(bb)^*+b^*a(bb)^*b+b^*a(bb)^*ab^*$.

Woorden in $\{a,b\}^* - L$ hebben ten hoogste twee a's; elk woord met tenminste drie a's zit dus in L.

3.23a Een automaat voor L/K wordt verkregen uit de automaat $\mathcal{A}$ door toestand 2 tot eindtoestand te bevorderen.

b Laat $\mathcal{A}$ een automaat voor L zijn. Een automaat $\mathcal{A}'$ voor L/K is hetzelfde als $\mathcal{A}$, maar met de volgende eindtoestanden: p is een eindtoestand van $\mathcal{A}'$ desdals er een pad met label w, voor een of andere w in K, is van p naar een eindtoestand van $\mathcal{A}$.

Opmerking. De constructie is in het algemeen niet effectief, dwz. het is onbeslisbaar welke toestanden tot eindtoestand moeten worden bevorderd. De constructie is wél effectief voor context-vrije (en dus reguliere) talen K. We kunnen dan nl. beslissen of $R \cap K = \emptyset$, waarbij R een reguliere taal is (van alle woorden vanuit een toestand p naar een eindtoestand).

Voor recursieve talen K (Hoofdstuk 6) kunnen we leegheid niet beslissen. Laat Δ^* geaccepteerd worden door een automaat met één toestand. Er geldt $\Delta^*/K = \Delta^*$ als $K \neq \emptyset$ en $\Delta^*/K = \emptyset$ als $K = \emptyset$. Omdat $K = \emptyset$ onbeslisbaar is, kunen we niet bepalen of de enige toestand van de automaat eindtoestand moet worden.

3.24b Laat $\mathcal{A} = (Q, \Delta, \delta, I, F)$ een eindige automaat zijn. De te construeren automaat $\mathcal{A}' = (Q, \Delta, \delta', I, F)$ verschilt alleen van $\mathcal{A}$ in de takken. Voor elk paar opvolgende takken $(p, a, q)(q, b, r)$ in δ bevat δ' de tak (p, b, r).

c Een woord $x = a_1 a_2 \ldots a_{2n}$ behoort tot $L(\mathcal{A})$ als er een pad $(p_0, a_1, p_1)(p_1, a_2, p_2) \cdots (p_{2n-1}, a_{2n}, p_{2n})$ in $\mathcal{A}$ bestaat met $p_0 \in I$ en $p_{2n} \in F$. Precies dan is er een pad $(p_0, a_2, p_2) \cdots (p_{2n-2}, a_{2n}, p_{2n})$ in $\mathcal{A}'$, dwz. $a_2 \cdots a_{2n} \in L(\mathcal{A}')$.

d Kies taal K over Σ. Dan is $g^{-1}(K) = \{ \langle a_1, a_2 \rangle \langle a_3, a_4 \rangle \cdots \langle a_{2n-1}, a_{2n} \rangle \mid a_1 a_2 a_3 \cdots a_{2n} \in K \}$; dus in deze taal over $\Sigma \times \Sigma$ is elk tweetal opeenvolgende letters uit K tot een paar samengevoegd. Nu projecteren we op de tweede component: $\pi_2(g^{-1}(K)) = \text{even}(K)$.

3.26 De equivalentieklassen zijn:

$\{ x \in \{a, b\}^* \mid x$ bevat een deelwoord bab $\}$

$\{ x \in \{a, b\}^* \mid x$ bevat geen deelwoord bab, maar eindigt op ba $\}$

$\{ x \in \{a, b\}^* \mid x$ bevat geen deelwoord bab, maar eindigt op b $\}$

$\{ x \in \{a, b\}^* \mid x$ bevat geen deelwoord bab, en eindigt niet op b of ba $\}$

Er zijn dus minimaal vier toestanden, nl. voor iedere equivalentieklasse één toestand.

3.27 Neem aan dat $x\, E_L\, y$, en kies een woord z.

Stel nu $xz \in M = L/K$, met andere woorden er is een woord $w \in K$ zo dat $xzw \in L$. Vanwege $x E_L y$ impliceert $x(zw) \in L$ dat ook $y(zw) \in L$, en dus $yz \in M = L/K$. Deze argumentatie is symmetrisch in x en y en geldt twee kanten op; dus $xz \in M$ desdals $yz \in M$, ofwel $x\, E_M\, y$.

Inderdaad: $x\, E_L\, y$ impliceert dat $x\, E_M\, y$ – woorden equivalent voor E_L zijn dat ook voor E_M. Dit betekent dat de equivalentieklassen van E_L steeds geheel binnen een van die van E_M vallen. Als E_L eindige index heeft (L is regulier) dan heeft ook E_M eindige index, want E_M heeft maximaal evenveel klassen als E_L.

3.28i $\{a, b\}^* \cdot a$.

ii $\{a, b\}^* \cdot a \cup \{\lambda\}$.

iii $\text{last}(i) \to a(i)$ is waar (voor positie i in een woord) als i *niet* de laatste positie is, of als a de letter op die positie is. Alle woorden van lengte twee of meer hebben een positie die voldoet aan het eerste alternatief. Het lege woord heeft geen posities die

voldoen.
$L(\ (\exists i)(\mathrm{last}(i) \to a(i))\ ) = \{a,b\}^* - \{\lambda, b\}$.

iv $\{\lambda, a\}$.

3.29i
- $(\forall i)[\ b(i) \to (\exists j)(\mathrm{next}(i,j) \wedge a(j))\ ]$.
- $(\exists i)(a(i)) \to (\exists j)(\exists k)(j \neq k \wedge b(j) \wedge b(k) \wedge \neg(\exists \ell)(b(\ell) \wedge \ell \neq j \wedge \ell \neq k))$.
- $\neg(\exists i)(\exists j)(\mathrm{next}(i,j) \wedge b(i) \wedge b(j))$.

ii Zij $\phi = (\exists i)(\exists j)(\exists k)[\mathrm{next}(i,j) \wedge \mathrm{next}(j,k) \wedge b(i) \wedge a(j) \wedge b(k)]$.
Dan $L_1 = L(\phi)$, $L_2 = L(\neg\phi)$.

3.31i

$(\forall i)(\ \mathrm{first}(i) \to (a(i) \wedge \neg(i \in X)) \vee (b(i) \wedge i \in X)\ )$	eerste stap
$\wedge\ (\forall i)(\forall j)[\ \mathrm{next}(i,j) \to$	transities:
$\quad (i \in X \wedge a(j) \wedge \neg(j \in X))$	$(1,a,2)$
$\quad \vee\ (i \in X \wedge b(j) \wedge j \in X)$	$(1,b,1)$
$\quad \vee\ (\neg(i \in X) \wedge j \in X)\ ]$	$(2,a,1)$,$(2,b,1)$

ii Bij een gegeven woord codeert de verzameling X waarvoor $\phi(X)$ waar is de berekening van de automaat op dat woord. Voor acceptatie eisen we dat zo'n berekening bestaat en eindigt in de eindtoestand: $L(\mathcal{A}) = L(\psi)$ met

$\psi = (\exists X)(\ \phi(X) \wedge (\forall i)(\mathrm{last}(i) \to i \in X)\ )$

3.32 De constructie uit het dictaat werkt ook voor niet-deterministische automaten. De automaat is overigens deterministisch te maken (met behoud van taal) door de tak $(3,a,3)$ weg te laten: elk deelpad $(2,b,3)(3,a,3)^k(3,a,2)$ kan vervangen worden door $(2,b,3)(3,a,2)(2,a,2)^k$.

$\phi_{\mathcal{A}} = (\exists X_1)(\exists X_2)(\exists X_3)$	
$[\ \ (\forall i)(\ i \in X_1 \vee i \in X_2 \vee i \in X_3\ )$	partitie
$\wedge\ (\forall i)\neg(\ (i \in X_1 \wedge i \in X_2) \vee (i \in X_2 \wedge i \in X_3)$	
$\quad \vee\ (i \in X_3 \wedge i \in X_1)\ )$	
$\wedge\ (\forall i)(\ \mathrm{first}(i) \to (b(i) \wedge i \in X_2)\ )$	eerste stap
$\wedge\ (\forall i)(\forall j)(\ \mathrm{next}(i,j) \to$	transities:
$\quad (i \in X_2 \wedge a(j) \wedge j \in X_2)$	$(2,a,2)$
$\quad \vee\ (i \in X_2 \wedge b(j) \wedge j \in X_3)$	$(2,b,3)$
$\quad \vee\ (i \in X_3 \wedge a(j) \wedge j \in X_2)$	$(3,a,2)$
$\quad \vee\ (i \in X_3 \wedge a(j) \wedge j \in X_3)$	$(3,a,3)$
$\quad)$	
$\wedge\ (\forall i)(\ \mathrm{last}(i) \to (i \in X_2)\ )$	eindtoestand
$]$	
$\wedge(\exists i)(\mathrm{first}(i))$	geen λ

Transitie $(1,b,2)$ is alleen opgenomen als eerste stap, en niet bij de lijst van transities omdat deze transitie inderdaad alleen als eerste kan voorkomen.

3.36 Ga uit van een eindige automaat $\mathcal{A} = (Q, \Delta, \delta, I, F)$ voor L. Een (mogelijke) automaat voor Half(L) heeft een toestandsverzameling $Q \times Q$. De eerste component werkt als in $\mathcal{A}$, de tweede component werkt 'achterstevoren' en gokt een pad naar een eindtoestand toe dat even lang is als de gelezen invoer.

Constructie: $\mathcal{A}_h = (Q \times Q, \Delta, \delta_h, I_h, F_h)$ met
$I_h = I \times F = \{\langle p, q\rangle \mid p \in I, q \in F\}$
$F_h = \{\langle p, p\rangle \mid p \in Q\}$ (het pad heeft een toestand bereikt vanwaaruit een eindtoestand te bereiken is in evenveel stappen)
$(\langle p, q\rangle, a, \langle p', q'\rangle) \in \delta_h$ als $(p, a, p') \in \delta$ en er is een $b \in \Delta$ met $(q', b, q) \in \delta$.

3.37 Nee, Middle$(a^+ba^+ba^+) \cap a^*bba^* = \{a^nbba^n \mid n \geq 0\}$.

3.39:1 Er geldt: $L(\mathcal{A})$ bevat een woord van de gegeven vorm desdals $L(\mathcal{A}) \cap (ab)^+ \neq \emptyset$. Construeer, om dit te testen, eerst een automaat $\mathcal{A}'$ voor de taal $(ab)^+$, en dan een automaat $\mathcal{A}_d$ voor de doorsnede $L(\mathcal{A}) \cap L(\mathcal{A}')$ met de productconstructie. Dus $L(\mathcal{A}_d) = L(\mathcal{A}) \cap (ab)^+$. Test nu of $L(\mathcal{A}_d)$ niet leeg is, met Stelling 3.35.

:2 Construeer als boven een automaat $\mathcal{A}_d$ voor de taal $L(\mathcal{A}) \cap (ab)^+$. Test of $L(\mathcal{A}_d)$ oneindig is, met Stelling 3.35.

:3 Construeer ditmaal een automaat voor de taal $L(\mathcal{A}) - (ab)^+$, en bepaal of deze automaat de lege taal accepteert.

3.41 Alternatieve beschrijving: x heeft oneven lengte, of x bevat een deelwoord baa. Eerst twee losse grammatica's, axioma R en X respectievelijk

$R \to aT, \quad R \to bT$
$T \to aR, \quad T \to bR, \quad T \to \lambda$

$X \to aX, \quad X \to bX, \quad X \to bX_{aa}$
$X_{aa} \to aX_a, \quad X_a \to aY$
$Y \to aY, \quad Y \to bY, \quad Y \to \lambda.$

Samenvoegen, met axioma S en de extra producties:
$S \to aT, \quad S \to bT$
$S \to aX, \quad S \to bX, \quad S \to bX_{aa}$

3.42:1 Opgave 3.2. Axioma S.

$S \to bA_2, \quad S \to bA_5,$
$A_2 \to aA_2, \quad A_2 \to bA_3, \quad A_2 \to \lambda,$
$A_3 \to aA_3, \quad A_3 \to aA_2,$
$A_5 \to aA_6,$
$A_6 \to aA_7, \quad A_6 \to bA_5, \quad A_6 \to \lambda,$
$A_7 \to bA_5.$

:2 Opgave 3.5. Axioma A_1.

$A_1 \to aA_3, \quad A_1 \to aA_4,$
$A_2 \to aA_5, \quad A_2 \to cA_1, \quad A_2 \to \lambda,$
$A_3 \to bA_5, \quad A_3 \to \lambda,$
$A_4 \to bA_2, \quad A_5 \to aA_2.$

:3 Dan moet de productie $A_1 \to \lambda$ worden toegevoegd.