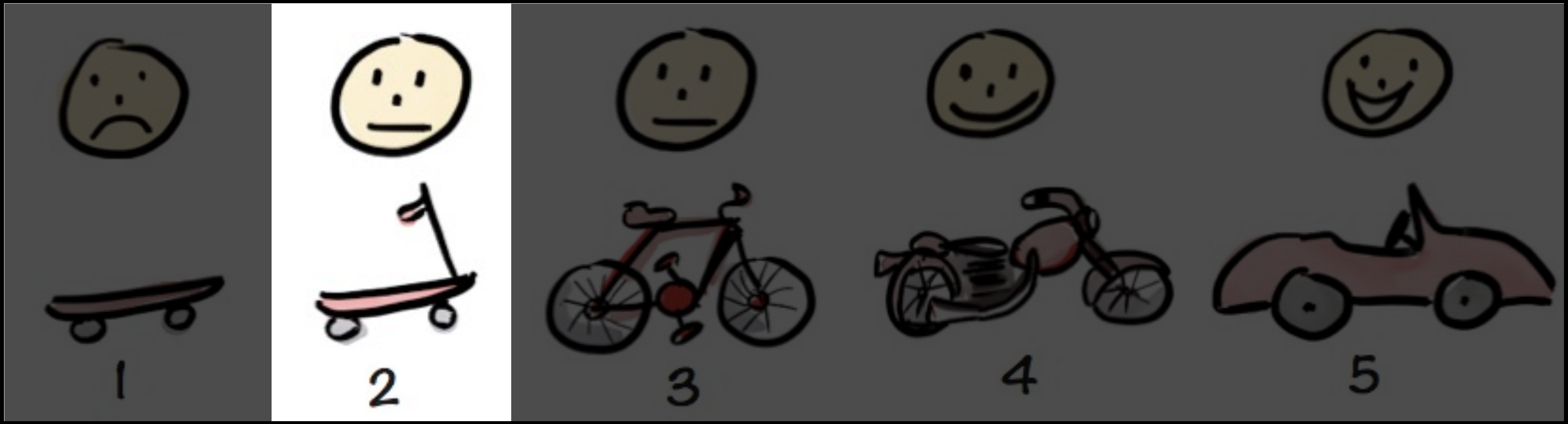
Errungenschaften der letzten Vorlesung



Mithilfe von **formalen Grammatiken** kann die **Syntax aller zulässigen Programme (Wörter) einer Programmiersprache** beschrieben werden. Token, für die ein Automat existiert, werden automatisch klassifiziert. Die Syntax unterliegt noch einer manuellen Prüfung.

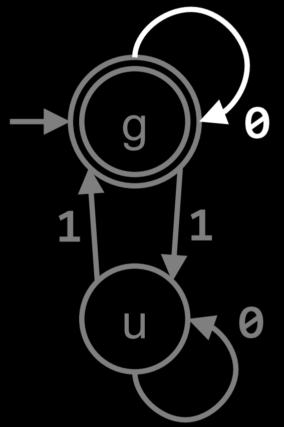
Beschreiben endliche Akzeptoren und Grammatiken die gleichen Sprachen?

$M = (Q, \Sigma, \delta, q_0, F)$ mit

$Q = \{g, u\}$,
$\Sigma = \{0, 1\}$,
$q_0 = g$,
$F = \{g\}$,
und $\delta = \begin{Bmatrix} (g,0) \to g \\ (g,1) \to u \\ (u,0) \to u \\ (u,1) \to g \end{Bmatrix}$

$G = (V_N, V_T, S, P)$ mit

$V_N = Q = \{g, u\}$,
$V_T = \Sigma = \{0, 1\}$,
$S = q_0 = g$,
und $P = \{g \to 0g\}$



Beschreiben endliche Akzeptoren und Grammatiken die gleichen Sprachen?

$M = (Q, \Sigma, \delta, q_0, F)$ mit

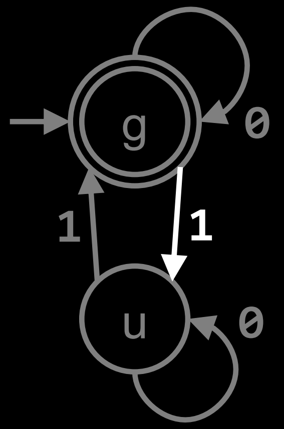
$Q = \{g, u\}$,
$\Sigma = \{0, 1\}$,
$q_0 = g$,
$F = \{g\}$,

und $\delta = \begin{Bmatrix} (g,0) \to g \\ \boldsymbol{(g,1) \to u} \\ (u,0) \to u \\ (u,1) \to g \end{Bmatrix}$

$G = (V_N, V_T, S, P)$ mit

$V_N = Q = \{g, u\}$,
$V_T = \Sigma = \{0, 1\}$,
$S = q_0 = g$,

und $P = \begin{Bmatrix} g \to 0g \\ \boldsymbol{g \to 1u} \end{Bmatrix}$



Beschreiben endliche Akzeptoren und Grammatiken die gleichen Sprachen?

$M = (Q, \Sigma, \delta, q_0, F)$ mit

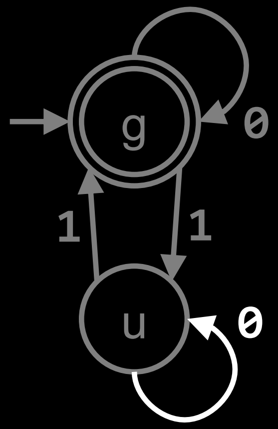
$Q = \{g, u\}$,
$\Sigma = \{0, 1\}$,
$q_0 = g$,
$F = \{g\}$,

und $\delta = \begin{Bmatrix} (g,0) \to g \\ (g,1) \to u \\ (u,0) \to u \\ (u,1) \to g \end{Bmatrix}$

$G = (V_N, V_T, S, P)$ mit

$V_N = Q = \{g, u\}$,
$V_T = \Sigma = \{0, 1\}$,
$S = q_0 = g$,

und $P = \begin{Bmatrix} g \to 0g \\ g \to 1u \\ u \to 0u \end{Bmatrix}$



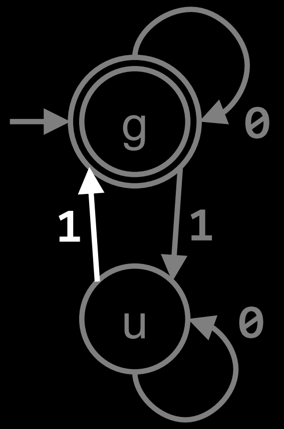
Beschreiben endliche Akzeptoren und Grammatiken die gleichen Sprachen?

$M = (Q, \Sigma, \delta, q_0, F)$ mit

$Q = \{g, u\}$,
$\Sigma = \{0, 1\}$,
$q_0 = g$,
$F = \{g\}$,
und $\delta = \left\{\begin{matrix} (g,0) \to g \\ (g,1) \to u \\ (u,0) \to u \\ (u,1) \to g \end{matrix}\right\}$

$G = (V_N, V_T, S, P)$ mit

$V_N = Q = \{g, u\}$,
$V_T = \Sigma = \{0, 1\}$,
$S = q_0 = g$,
und $P = \left\{\begin{matrix} g \to 0g \\ g \to 1u \\ u \to 0u \\ u \to 1g \end{matrix}\right\}$



Beschreiben endliche Akzeptoren und Grammatiken die gleichen Sprachen?

$M = (Q, \Sigma, \delta, q_0, F)$ mit

$Q = \{g, u\}$,
$\Sigma = \{0, 1\}$,
$q_0 = g$,
$F = \{g\}$,

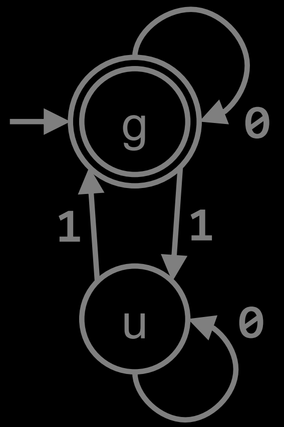
und $\delta = \begin{Bmatrix} (g,0) \to g \\ (g,1) \to u \\ (u,0) \to u \\ (u,1) \to g \end{Bmatrix}$

$G = (V_N, V_T, S, P)$ mit

$V_N = Q = \{g, u\}$,
$V_T = \Sigma = \{0, 1\}$,
$S = q_0 = g$,

und $P = \begin{Bmatrix} g \to 0g \\ g \to 1u \\ u \to 0u \\ u \to 1g \end{Bmatrix} =$

$\{q_A \to x q_B \mid q_B \in \delta(q_A, x)\}$



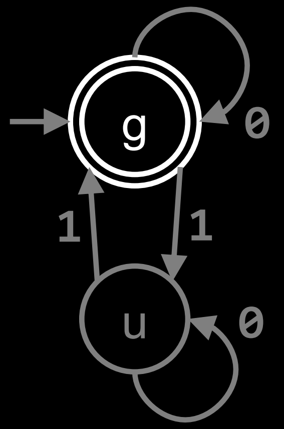
Beschreiben endliche Akzeptoren und Grammatiken die gleichen Sprachen?

$M = (Q, \Sigma, \delta, q_0, F)$ mit

$Q = \{g, u\}$,
$\Sigma = \{0, 1\}$,
$q_0 = g$,
$F = \{g\}$,
und $\delta = \begin{Bmatrix} (g,0) \rightarrow g \\ (g,1) \rightarrow u \\ (u,0) \rightarrow u \\ (u,1) \rightarrow g \end{Bmatrix}$

$G = (V_N, V_T, S, P)$ mit

$V_N = Q = \{g, u\}$,
$V_T = \Sigma = \{0, 1\}$,
$S = q_0 = g$,
und $P = \begin{Bmatrix} g \rightarrow 0g \\ g \rightarrow 1u \\ u \rightarrow 0u \\ u \rightarrow 1g \\ g \rightarrow \varepsilon \end{Bmatrix} =$
$\{q_A \rightarrow x q_B \mid q_B \in \delta(q_A, x)\} \cup$
$\{q_C \rightarrow \varepsilon \mid q_C \in F\}$



Beschreiben endliche Akzeptoren und Grammatiken die gleichen Sprachen?

$\boldsymbol{M = (Q, \Sigma, \delta, q_0, F)}$ **mit**

$\boldsymbol{Q = \{g, u\}}$,
$\boldsymbol{\Sigma = \{0, 1\}}$,
$\boldsymbol{q_0 = g}$,
$\boldsymbol{F = \{g\}}$,

und $\boldsymbol{\delta = \begin{Bmatrix} (g,0) \to g \\ (g,1) \to u \\ (u,0) \to u \\ (u,1) \to g \end{Bmatrix}}$

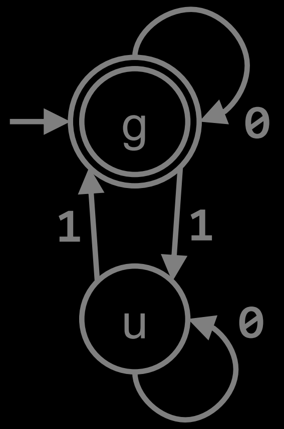
$G = (V_N, V_T, S, P)$ mit

$V_N = Q = \{g, u\}$,
$V_T = \Sigma = \{0, 1\}$,
$S = q_0 = g$,

und $P = \begin{Bmatrix} g \to 0g \\ g \to 1u \\ u \to 0u \\ u \to 1g \\ g \to \varepsilon \end{Bmatrix} =$
$\{q_A \to xq_B \mid q_B \in \delta(q_A, x)\} \cup$
$\{q_C \to \varepsilon \mid q_C \in F\}$

Abbildungsvorschrift

$$V_N = Q$$
$$V_T = \Sigma$$
$$S = q_0$$
$$P = \{q_A \to xq_B \mid q_B \in \delta(q_A, x)\} \cup \{q_C \to \varepsilon \mid q_C \in F\}$$



Eine Grammatik $G = (V_N, V_T, S, P)$ heißt strikt rechtslinear, wenn alle Produktionen P die folgende Form haben:

$A \to \alpha B$ oder $A \to \varepsilon$ wobei $\alpha \in V_T \cup \{\varepsilon\}$ und $A, B \in V_N$.

Eine durch eine strikt rechtslineare Grammatik erzeugte Sprache heißt strikt rechtslinear.

$$V_N = Q$$
$$V_T = \Sigma$$
$$S = q_0$$
$$P = \{q_A \to xq_B \mid q_B \in \delta(q_A, x)\} \cup \{q_C \to \varepsilon \mid q_C \in F\}$$

nicht Ka: Kein Beweis !

Let $D = (Q, \Sigma, \delta, q_0, F)$ be a **deterministic finite acceptor**. Then there is a grammar $G = (V_N, V_T, S, P)$ such that $L(D) = L(G)$.

We take advantage of $A = B \Leftrightarrow A \subseteq B \land B \subseteq A$. And split the proof into two lemmas.

Lemma 1: $L(D) \subseteq L(G)$

Lemma 2: $L(G) \subseteq L(D)$

$$V_N = Q$$
$$V_T = \Sigma$$
$$S = q_0$$
$$P = \{q_A \to xq_B \mid q_B \in \delta(q_A, x)\} \cup \{q_C \to \varepsilon \mid q_C \in F\}$$

Let $D = (Q, \Sigma, \delta, q_0, F)$ be a **deterministic finite acceptor**. Then there is a grammar $G = (V_N, V_T, S, P)$ such that $L(D) = L(G)$.

We take advantage of $A = B \Leftrightarrow A \subseteq B \land B \subseteq A$. And split the proof into two lemmas.

$L(D) \subseteq L(G) :\Leftrightarrow \forall w: \left(w \in L(D) \Longrightarrow w \in L(G)\right)$ with $L(D) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}$ and $L(G) = \left\{w \in V_T^* \mid S \overset{*}{\Rightarrow} w\right\}$

$V_N = Q$

$\boxed{V_T = \Sigma}$

$S = q_0$

$P = \{q_A \to xq_B \mid q_B \in \delta(q_A, x)\}$
$\cup \{q_C \to \varepsilon \mid q_C \in F\}$

Let $D = (Q, \Sigma, \delta, q_0, F)$ be a deterministic finite acceptor. Then there is a grammar $G = (V_N, V_T, S, P)$ such that $L(D) = L(G)$.

We take advantage of $A = B \Leftrightarrow A \subseteq B \wedge B \subseteq A$. And split the proof into two lemmas.

$L(D) \subseteq L(G) :\Leftrightarrow \forall w: \big(w \in L(D) \Longrightarrow w \in L(G)\big)$ with $L(D) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}$
and $L(G) = \left\{w \in V_T^* \mid S \overset{*}{\Rightarrow} w\right\}$

Let $w \in \Sigma^*$. Due to $\boxed{V_T = \Sigma}$, also $w \in V_T^*$.
Consequently, $\forall w: \left((\delta^*(q_0, w) \in F) \Longrightarrow \left(S \overset{*}{\Rightarrow} w\right) \right)$

$V_N = Q$

$V_T = \Sigma$

$S = q_0$

$P = \{q_A \to x q_B \mid q_B \in \delta(q_A, x)\} \cup \{q_C \to \varepsilon \mid q_C \in F\}$

Let $D = (Q, \Sigma, \delta, q_0, F)$ be a deterministic finite acceptor. Then there is a grammar $G = (V_N, V_T, S, P)$ such that $L(D) = L(G)$.

We take advantage of $A = B \Leftrightarrow A \subseteq B \wedge B \subseteq A$. And split the proof into two lemmas.

$L(D) \subseteq L(G) :\Leftrightarrow \forall w: \big(w \in L(D) \Longrightarrow w \in L(G)\big)$ with $L(D) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}$ and $L(G) = \left\{w \in V_T^* \mid S \overset{*}{\Rightarrow} w\right\}$

Let $w \in \Sigma^*$. Due to $V_T = \Sigma$, also $w \in V_T^*$.
Consequently, $\forall w: \Big(\big(\delta^*(q_0, w) \in F\big) \Longrightarrow \big(S \overset{*}{\Rightarrow} w\big)\Big)$

hypothesis

If $\delta^*(q_A, \tau) = q_B$ then $q_A \overset{*}{\Rightarrow} \tau q_B$.

Furthermore, if $q_B \in F$ then $q_A \overset{*}{\Rightarrow} \tau$.

$q_A, q_B \in Q$
$q_A, q_B \in V_N$
$\tau \in \Sigma^*$

$V_N = Q$

$V_T = \Sigma$

$S = q_0$

$P = \{q_A \to x q_B \mid q_B \in \delta(q_A, x)\} \cup \{q_C \to \varepsilon \mid q_C \in F\}$

hypothesis

If $\delta^*(q_A, \tau) = q_B$ then $q_A \overset{*}{\Rightarrow} \tau q_B$.

Furthermore, if $q_B \in F$ then $q_A \overset{*}{\Rightarrow} \tau$.

$q_A, q_B \in Q$

$q_A, q_B \in V_N$

$\tau \in \Sigma^*$

base

Let $\tau = \varepsilon$. By definition, $\delta^*(q_A, \varepsilon) = q_B$ with $q_A = q_B$. The induction hypothesis implies $q_A \overset{*}{\Rightarrow} \varepsilon q_B$ with $q_A = q_B$. By definition, this is always true.

A derivation step has the form $\alpha v \beta \Rightarrow \alpha w \beta$ where $\alpha, \beta \in V^*$ and $v \to w \in P$.

$\boxed{\delta^*(q, \varepsilon) = q}$

$\forall w \in \Sigma^*\ \forall x \in \Sigma: \delta^*(q, wx) = \delta(\delta^*(q, w), x)$

$$\boxed{w_1 \overset{*}{\Rightarrow} w_2} :\Leftrightarrow \begin{cases} \boxed{w_1 = w_2} \\ \exists w \in V^*: w_1 \overset{*}{\Rightarrow} w \wedge w \Rightarrow w_2 \end{cases}$$

$V_N = Q$

$V_T = \Sigma$

$S = q_0$

$P = \{q_A \to xq_B \mid q_B \in \delta(q_A, x)\} \boxed{\cup \{q_C \to \varepsilon \mid q_C \in F\}}$

hypothesis

If $\delta^*(q_A, \tau) = q_B$ then $q_A \overset{*}{\Rightarrow} \tau q_B$.

Furthermore, if $q_B \in F$ then $q_A \overset{*}{\Rightarrow} \tau$.

$q_A, q_B \in Q$
$q_A, q_B \in V_N$
$\tau \in \Sigma^*$

base

Let $\tau = \varepsilon$. By definition, $\delta^*(q_A, \varepsilon) = q_B$ with $q_A = q_B$. The induction hypothesis implies $q_A \overset{*}{\Rightarrow} \varepsilon q_B$ with $q_A = q_B$. By definition, this is always true.

If and only if $q_B \in F$, then $P \ni q_B \to \varepsilon$. This rule allows a derivation step of the form $\alpha q_B \beta \Rightarrow \alpha \varepsilon \beta$ where $\alpha, \beta \in V^*$. If we set $\alpha = \beta = \varepsilon$, we get $\left(q_A \overset{*}{\Rightarrow} \varepsilon\right) \Leftrightarrow \left(q_A \overset{*}{\Rightarrow} \varepsilon q_B \land \varepsilon q_B \Rightarrow \varepsilon\varepsilon\right)$.

A derivation step has the form $\boxed{\alpha v \beta \Rightarrow \alpha w \beta}$ where $\alpha, \beta \in V^*$ and $v \to w \in P$.

$$\delta^*(q, \varepsilon) = q$$
$$\forall w \in \Sigma^*\ \forall x \in \Sigma: \delta^*(q, wx) = \delta(\delta^*(q, w), x)$$

$$w_1 \overset{*}{\Rightarrow} w_2 :\Leftrightarrow \begin{cases} w_1 = w_2 \\ \exists w \in V^*: \boxed{w_1 \overset{*}{\Rightarrow} w \land w \Rightarrow w_2} \end{cases}$$

$$V_N = Q$$
$$V_T = \Sigma$$
$$S = q_0$$
$$P = \{q_A \to xq_B \mid q_B \in \delta(q_A, x)\} \cup \{q_C \to \varepsilon \mid q_C \in F\}$$

hypothesis

If $\delta^*(q_A, \tau) = q_B$ then $q_A \overset{*}{\Rightarrow} \tau q_B$.

Furthermore, if $q_B \in F$ then $q_A \overset{*}{\Rightarrow} \tau$.

$q_A, q_B \in Q$
$q_A, q_B \in V_N$
$\tau \in \Sigma^*$

step

Let $\tau = \tau' x$. By definition, $\delta^*(q_A, \tau' x) = \delta(\delta^*(q_A, \tau'), x) = q_B$. We‘ve already showed that if $\delta^*(q_A, \tau') = q_C$ then $q_A \overset{*}{\Rightarrow} \tau' q_C$ is also valid.

A derivation step has the form $\alpha v \beta \Rightarrow \alpha w \beta$ where $\alpha, \beta \in V^*$ and $v \to w \in P$.

$$\delta^*(q, \varepsilon) = q$$
$$\forall w \in \Sigma^* \ \forall x \in \Sigma: \delta^*(q, wx) = \boxed{\delta(\delta^*(q, w), x)}$$

$$w_1 \overset{*}{\Rightarrow} w_2 :\Leftrightarrow \begin{cases} w_1 = w_2 \\ \exists w \in V^*: w_1 \overset{*}{\Rightarrow} w \wedge w \Rightarrow w_2 \end{cases}$$

$V_N = Q$

$V_T = \Sigma$

$S = q_0$

$P = \{q_A \to x q_B \mid q_B \in \delta(q_A, x)\} \cup \{q_C \to \varepsilon \mid q_C \in F\}$

hypothesis

If $\delta^*(q_A, \tau) = q_B$ then $q_A \overset{*}{\Rightarrow} \tau q_B$.

Furthermore, if $q_B \in F$ then $q_A \overset{*}{\Rightarrow} \tau$.

$q_A, q_B \in Q$
$q_A, q_B \in V_N$
$\tau \in \Sigma^*$

step

Let $\tau = \tau' x$. By definition, $\delta^*(q_A, \tau' x) = \delta(\delta^*(q_A, \tau'), x) = q_B$. We‘ve already showed that if $\delta^*(q_A, \tau') = q_C$ then $q_A \overset{*}{\Rightarrow} \tau' q_C$ is also valid. If $\delta(q_C, x) = q_B$, then $P \ni q_C \to x q_B$. This rule allows a derivation step of the form $\alpha q_C \beta \Rightarrow \alpha x q_B \beta$.

If we set $\alpha = \tau'$ and $\beta = \varepsilon$, we get $\tau' q_C \Rightarrow \tau' x q_B$. Since we have already proven $q_A \overset{*}{\Rightarrow} \tau' q_C$, we can now combine them to $\left(q_A \overset{*}{\Rightarrow} \tau' q_C \land \tau' q_C \Rightarrow \tau' x q_B\right) \Leftrightarrow \left(q_A \overset{*}{\Rightarrow} \tau' x q_B\right) \Leftrightarrow \left(q_A \overset{*}{\Rightarrow} \tau q_B\right)$.

A derivation step has the form $\alpha v \beta \Rightarrow \alpha w \beta$ where $\alpha, \beta \in V^*$ and $v \to w \in P$.

$$\delta^*(q, \varepsilon) = q$$
$$\forall w \in \Sigma^* \ \forall x \in \Sigma: \delta^*(q, wx) = \delta(\delta^*(q, w), x)$$

$$w_1 \overset{*}{\Rightarrow} w_2 :\Leftrightarrow \begin{cases} w_1 = w_2 \\ \exists w \in V^*: w_1 \overset{*}{\Rightarrow} w \land w \Rightarrow w_2 \end{cases}$$

$$V_N = Q$$
$$V_T = \Sigma$$
$$S = q_0$$
$$P = \{q_A \to x q_B \mid q_B \in \delta(q_A, x)\} \cup \boxed{\{q_C \to \varepsilon \mid q_C \in F\}}$$

hypothesis

If $\delta^*(q_A, \tau) = q_B$ then $q_A \overset{*}{\Rightarrow} \tau q_B$.

Furthermore, if $q_B \in F$ then $q_A \overset{*}{\Rightarrow} \tau$.

$q_A, q_B \in Q$
$q_A, q_B \in V_N$
$\tau \in \Sigma^*$

step

Let $\tau = \tau' x$. By definition, $\delta^*(q_A, \tau' x) = \delta(\delta^*(q_A, \tau'), x) = q_B$. We've already showed that if $\delta^*(q_A, \tau') = q_C$ then $q_A \overset{*}{\Rightarrow} \tau' q_C$ is also valid. If $\delta(q_C, x) = q_B$, then $P \ni q_C \to x q_B$. This rule allows a derivation step of the form $\alpha q_C \beta \Rightarrow \alpha x q_B \beta$.

If we set $\alpha = \tau'$ and $\beta = \varepsilon$, we get $\tau' q_C \Rightarrow \tau' x q_B$. Since we have already proven $q_A \overset{*}{\Rightarrow} \tau' q_C$, we can now combine them to $\left(q_A \overset{*}{\Rightarrow} \tau' q_C \wedge \tau' q_C \Rightarrow \tau' x q_B\right) \Leftrightarrow \left(q_A \overset{*}{\Rightarrow} \tau' x q_B\right) \Leftrightarrow \left(q_A \overset{*}{\Rightarrow} \tau q_B\right)$.

If $q_B \in \boldsymbol{F}$ then $\boldsymbol{P} \ni \boldsymbol{q_B} \to \boldsymbol{\varepsilon}$. This rule allows a derivation step of the form $\alpha q_B \beta \Rightarrow \alpha \varepsilon \beta$. If we set $\alpha = \tau$ and $\beta = \varepsilon$, we get $\tau q_B \Rightarrow \tau \varepsilon$. Together with $q_A \overset{*}{\Rightarrow} \tau q_B$ from above, we get $\left(q_A \overset{*}{\Rightarrow} \tau q_B \wedge \tau q_B \Rightarrow \tau \varepsilon\right) \Leftrightarrow \left(q_A \overset{*}{\Rightarrow} \tau \varepsilon\right) \Leftrightarrow \left(q_A \overset{*}{\Rightarrow} \tau\right)$, proving the lemma.

A derivation step has the form $\boxed{\alpha v \beta \Rightarrow \alpha w \beta}$ where $\alpha, \beta \in V^*$ and $v \to w \in P$.

$$\delta^*(q, \varepsilon) = q$$
$$\forall w \in \Sigma^* \ \forall x \in \Sigma: \delta^*(q, wx) = \delta(\delta^*(q, w), x)$$

$$w_1 \overset{*}{\Rightarrow} w_2 :\Leftrightarrow \begin{cases} w_1 = w_2 \\ \exists w \in V^*: \boxed{w_1 \overset{*}{\Rightarrow} w \wedge w \Rightarrow w_2} \end{cases}$$

$$\boldsymbol{V_N = Q}$$
$$\boldsymbol{V_T = \Sigma}$$
$$\boldsymbol{S = q_0}$$
$$\boldsymbol{P = \{q_A \to xq_B \mid q_B \in \delta(q_A, x)\} \cup \{q_C \to \varepsilon \mid q_C \in F\}}$$

Let $D = (Q, \Sigma, \delta, q_0, F)$ be a deterministic finite acceptor. Then there is a grammar $G = (V_N, V_T, S, P)$ such that $L(D) = L(G)$.

We take advantage of $A = B \Leftrightarrow A \subseteq B \land B \subseteq A$. And split the proof into two lemmas.

Lemma 1: $\boldsymbol{L(D) \subseteq L(G)}$

correct

Lemma 2: $\boldsymbol{L(G) \subseteq L(D)}$

Abkürzung

Eigentlich müsste noch das zweite Lemma $L(G) \subseteq L(D)$ bewiesen werden. Weil der Beweis aber sehr ähnlich erfolgt und wahrscheinlich wenig verständnisfördernd ist, verzichten wir an dieser Stelle darauf.

Außerdem haben wir bisher nur bewiesen, dass zu jedem deterministischen endlichen Automaten (DEA) eine äquivalente, strikt rechtslineare Grammatik existiert. Ob die Umkehrung dessen gilt, ist noch unklar…

Beschreiben endliche Akzeptoren und Grammatiken die gleichen Sprachen?

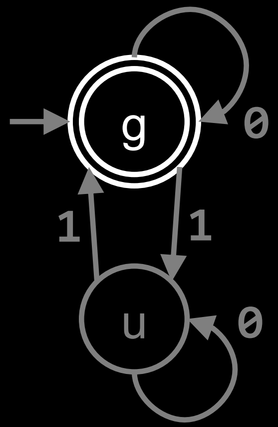
$G = (V_N, V_T, S, P)$ mit

$V_N = \{g, u\}$,
$V_T = \{0, 1\}$,
$S = g$,

und $P = \begin{Bmatrix} g \to 0g \\ g \to 1u \\ u \to 0u \\ u \to 1g \\ g \to \varepsilon \end{Bmatrix}$

$D = (Q, \Sigma, \delta, q_0, F)$ mit

$Q = V_N = \{g, u\}$,
$\Sigma = V_T = \{0, 1\}$,
$q_0 = S = g$,



Beschreiben endliche Akzeptoren und Grammatiken die gleichen Sprachen?

$G = (V_N, V_T, S, P)$ mit

$V_N = \{g, u\}$,
$V_T = \{0, 1\}$,
$S = g$,

und $P = \begin{Bmatrix} g \to 0g \\ g \to 1u \\ u \to 0u \\ u \to 1g \\ g \to \varepsilon \end{Bmatrix}$

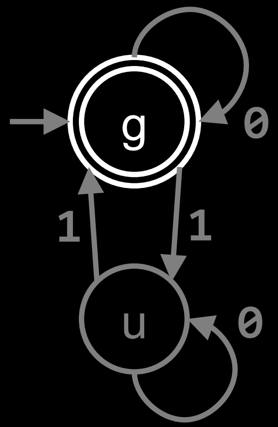
$D = (Q, \Sigma, \delta, q_0, F)$ mit

$Q = V_N = \{g, u\}$,
$\Sigma = V_T = \{0, 1\}$,
$q_0 = S = g$,
$F = \{n_C \mid n_C \to \varepsilon \in P\} = \{g\}$ und
$\delta(n_A, x) = \{n_B \mid n_A \to x n_B \in P\}$,

also $\delta = \begin{Bmatrix} (g, 0) \to g \\ (g, 1) \to u \\ (u, 0) \to u \\ (u, 1) \to g \end{Bmatrix}$

Abbildungsvorschrift

$$Q = V_N$$
$$\Sigma = V_T$$
$$\delta(n_A, x) = \{n_B \mid n_A \to x n_B \in P\}$$
$$q_0 = S$$
$$F = \{n_C \mid n_C \to \varepsilon \in P\}$$



Beschreiben endliche Akzeptoren und Grammatiken die gleichen Sprachen?

$G = (V_N, V_T, S, P)$ mit

$V_N = \{g, u\}$,
$V_T = \{0, 1\}$,
$S = g$,

und $P = \left\{ \begin{matrix} g \to 0g \\ g \to 1u \\ g \to 1g \\ u \to 0u \\ u \to 1g \\ g \to \varepsilon \end{matrix} \right\}$

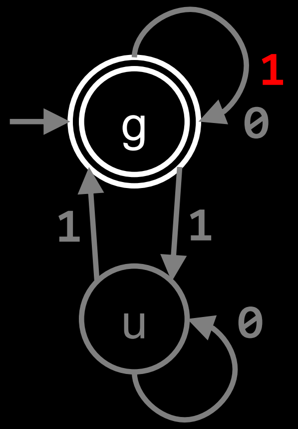
$D = (Q, \Sigma, \delta, q_0, F)$ mit

$Q = V_N = \{g, u\}$,
$\Sigma = V_T = \{0, 1\}$,
$q_0 = S = g$,
$F = \{n_C \mid n_C \to \varepsilon \in P\} = \{g\}$ und
$\delta(n_A, x) = \{n_B \mid n_A \to x n_B \in P\}$,

also $\delta = \left\{ \begin{matrix} (g, 0) \to g \\ (g, 1) \to u \\ (g, 1) \to g \\ (u, 0) \to u \\ (u, 1) \to g \end{matrix} \right\}$

Abbildungsvorschrift

$$\begin{gathered} Q = V_N \\ \Sigma = V_T \\ \delta(n_A, x) = \{n_B \mid n_A \to x n_B \in P\} \\ q_0 = S \\ F = \{n_C \mid n_C \to \varepsilon \in P\} \end{gathered}$$



Wer jetzt dachte wir können den Beweis exakt so führen wie zuvor, läuft in eine Falle. Wir sind nämlich von einem deterministischen Automaten ausgegangen. Eine strikt rechtslineare Grammatik unterbindet die hervorgerufene Regel jedoch nicht. Der deterministische endliche Automat dagegen schon. Wie sollten wir jetzt weiter vorgehen?

non-deterministic finite acceptor

A non-deterministic finite acceptor (NFA) is specified by a tuple $(Q, \Sigma, \delta, q_0, F)$ where

Q	is a finite set of states,
Σ	is a finite alphabet,
$\delta: Q \times \Sigma_\varepsilon \to P(Q)$	is the transition function,
$q_0 \in Q$	is the initial state,
$F \subseteq Q$	is the set of final states.

For we write $q_1 \overset{w}{\rightarrow}_* q_2$ if

- if $q_1 = q_2$ and $w = \varepsilon$
- or $\exists q' \in Q$ and $\exists w', w'' \in \Sigma^*$ so that $w = w'w''$ and $q_1 \overset{w'}{\rightarrow}_* q' \overset{w''}{\longrightarrow} q_2$

$g^*(Q \times \Sigma^*) \to P(Q)$ specifies the states that are reached last when a word $w \in \Sigma^*$ is entered.

$$g^*(q_1, w) = \left\{q_2 \mid q_1 \overset{w}{\rightarrow}_* q_2\right\}$$

A non-deterministic finite acceptor $N = (Q, \Sigma, \delta, q_0, F)$ recognizes a language $L \subseteq \Sigma^*$ if

$$L(N) = \{w \in \Sigma^* \mid g^*(z_0, w) \cap F \neq \emptyset\}.$$

Can every deterministic acceptor be classified as non-deterministic?

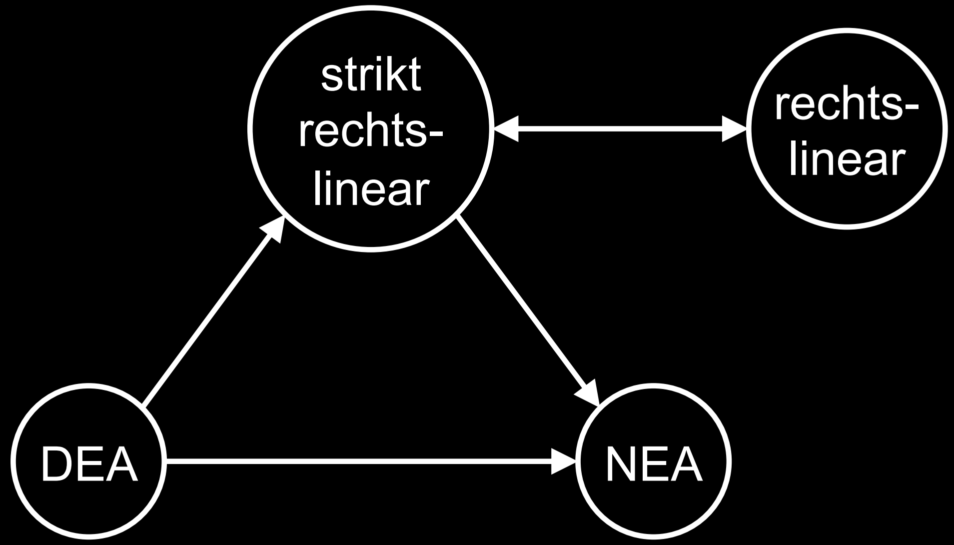
Let $M = (Q, \Sigma, \delta, q_0, F)$ be a finite acceptor.

Q	is a finite set of states,
Σ	is a finite (input) alphabet,
$\delta: Q \times \Sigma \to Q$	is the transition function if it is deterministic,
$\delta: Q \times \Sigma_\varepsilon \to P(Q)$	is the transition function if it is non-deterministic,
$q_0 \in Q$	is the initial state,
$F \subseteq Q$	is the set of accepting states.

Since $\Sigma \subset \Sigma_\varepsilon$ and $Q \subset P(Q)$, any deterministic finite acceptor satisfies the definition of a non-deterministic acceptor.

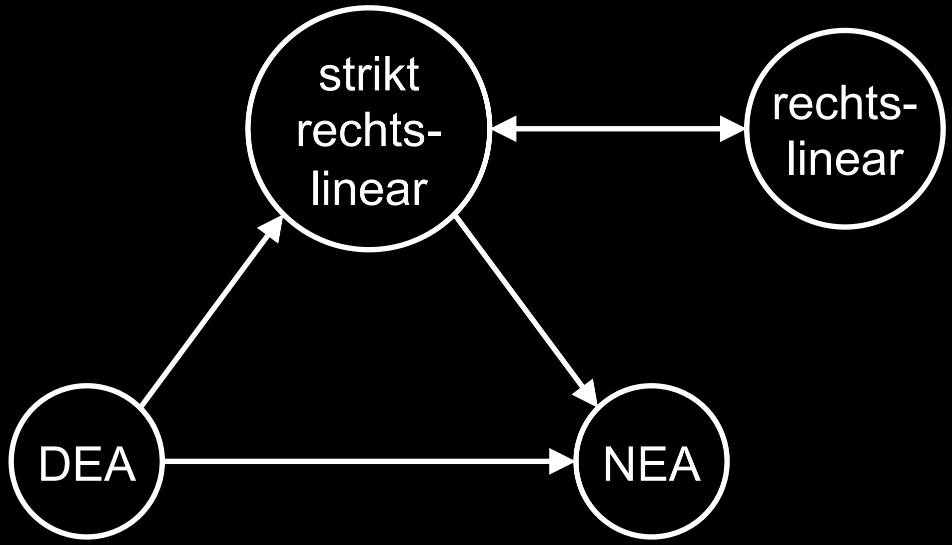
Abkürzung

Wir verzichten auch hier wieder auf den Beweis, weil er konzeptionell ähnlich erfolgt wie zuvor. Es stimmt aber tatsächlich. Strikt rechtslineare Grammatiken erzeugen genau die Sprachen, welche nichtdeterministische endliche Automaten akzeptieren. Somit ergibt sich folgendes Schaubild:



Abkürzung

Interessant: Wenn sich jetzt noch unsere Annahme bestätigen würde, dass sich alle **NFAs in DFAs transformieren** lassen, hätten wir einen **Zirkelschluss** erzielt. Damit wäre bewiesen, dass alle drei Konzepte die gleichen Möglichkeiten bieten.



Offene Fragen

1. ~~Nicht jedes Token darf an jeder Stelle stehen. Wie lässt sich das regulieren?~~
2. ~~Sind die von endlichen Automaten akzeptierten Sprachen identisch zu den von Grammatiken erzeugten Sprachen?~~
3. Lassen sich alle NEAs in DEAs umwandeln?
4. Sind rechtslineare und linkslineare Grammatiken verschieden?
5. Sind unbeschränkte Grammatiken mächtiger als die linearen?
6. Kann man die Erstellung eines endlichen Automaten automatisieren?

Lassen sich alle NEAs in DEAs umwandeln?

Bevor wir die Frage beantworten können, benötigen wir ein besseres Verständnis von nichtdeterministischen Automaten. Geben Sie einen Berechnungspfad für die Wörter 100, 0010 und 10101 an. Werden die Wörter akzeptiert?