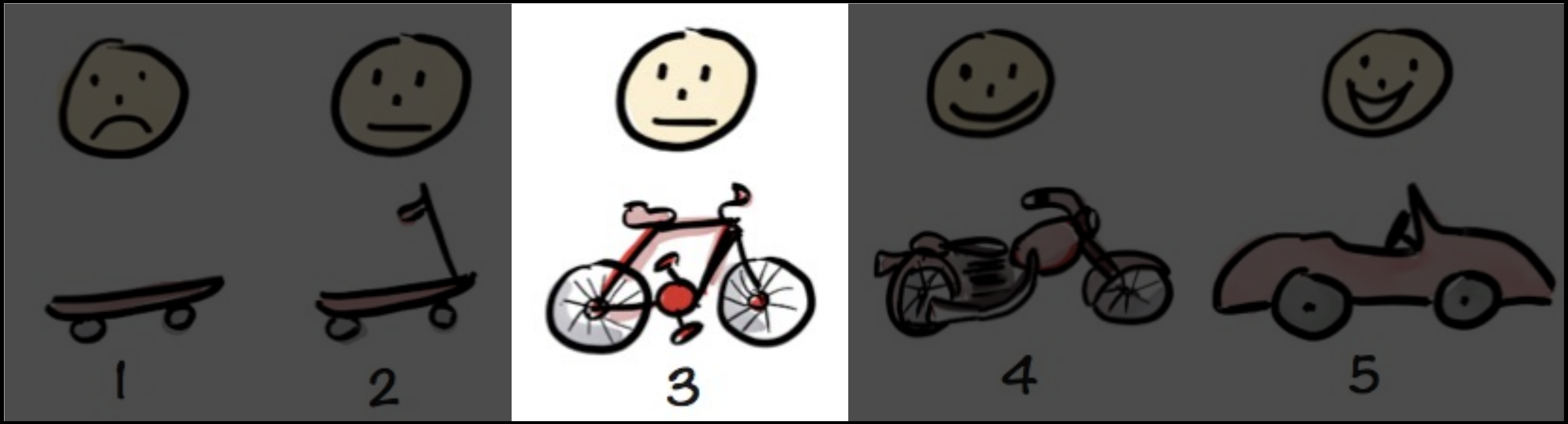
Errungenschaften der letzten Vorlesung



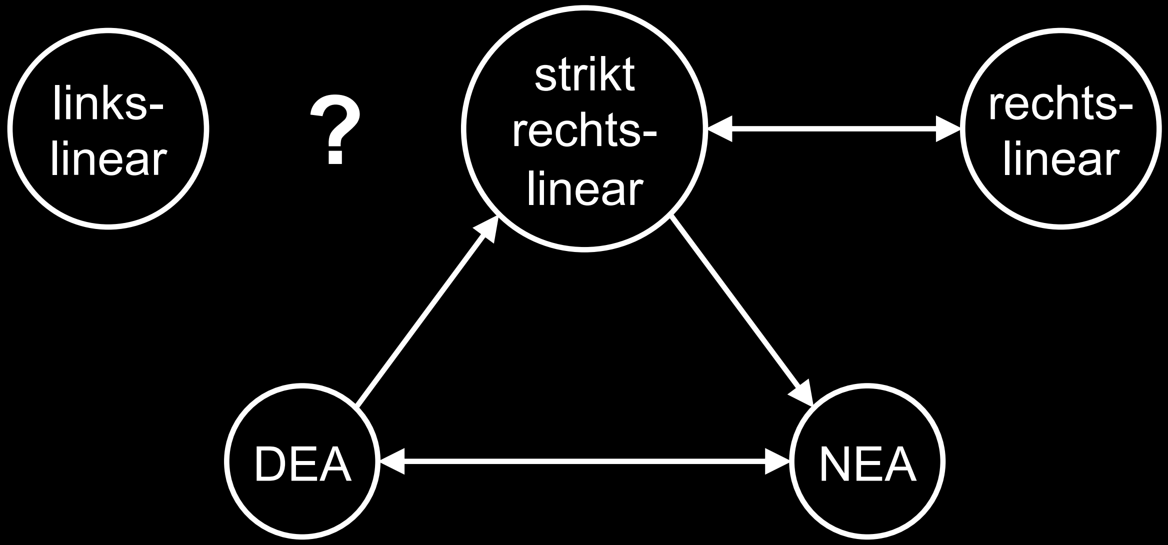
Die Syntax einer formalen Sprache wird mit **formalen Grammatiken** spezifiziert. **Rechtslineare Grammatiken** lassen sich mit endlichen Automaten verarbeiten. Für andere Grammatiken muss der **Syntaxbaum** noch manuell ermittelt werden. Alle Token, für die ein **endlicher Akzeptor** existiert, können automatisch zu einem **Tokenizer / Lexer** kombiniert werden. Zu jedem NEA kann mithilfe der **Potenzmengenkonstruktion** ein äquivalenter DEA berechnet werden.

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 rechtslinearen?
6. Kann man die Erstellung eines endlichen Automaten automatisieren?

Sind rechtslineare und linkslineare Grammatiken verschieden?

Die Frage ist schlecht gestellt. Verschieden sind sie natürlich, schon allein gemäß Definition. Inzwischen wissen wir, dass wir ermitteln müssen, ob rechtslineare und linkslineare Grammatiken die gleichen Sprachen erzeugen.



reversal

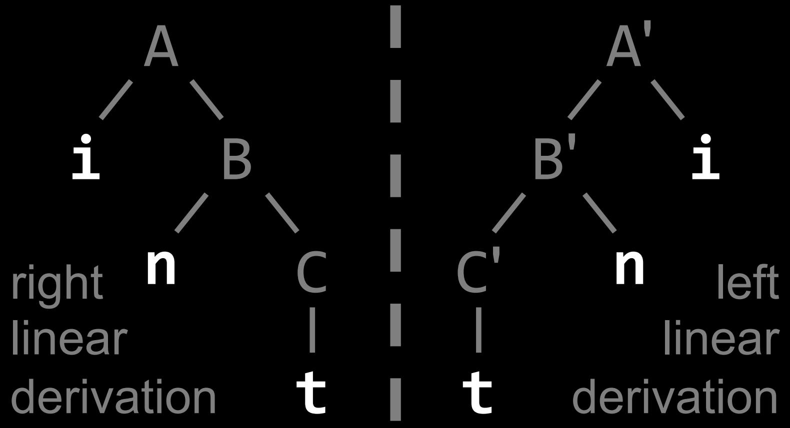
We define the reversal of a string $a_1 a_2 \dots a_n$ as $\boldsymbol{(a_1 a_2 \dots a_n)^R = a_n \dots a_2 a_1}$

The reversal of a language L as $\boldsymbol{L^R = \{\alpha^R : \alpha \in L\}}$

The reversal of a right-linear grammar G as $\boldsymbol{G^R}$ being the result of replacing every rule in G of the form $\boldsymbol{A \to aB}$ **by** $\boldsymbol{A \to Ba}$.

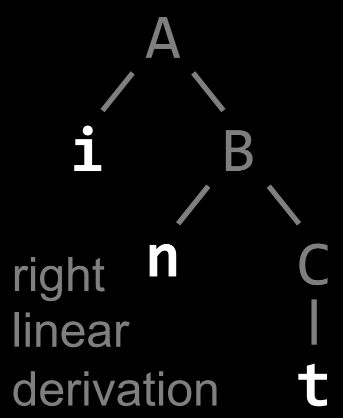
Fact: If G is a right-linear grammar, then $\boldsymbol{L(G^R) = (L(G))^R}$.

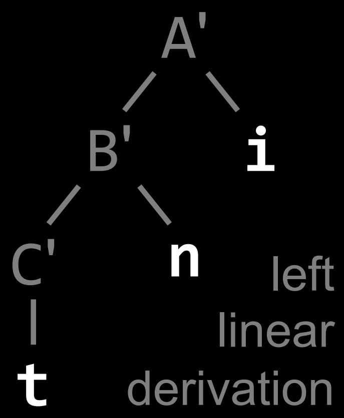
$\{\text{int}\} =$



$= \{\text{tni}\} = \{\text{int}\}^R$

Beweisidee: rechtslineare Sprachen sind auch linkslinear



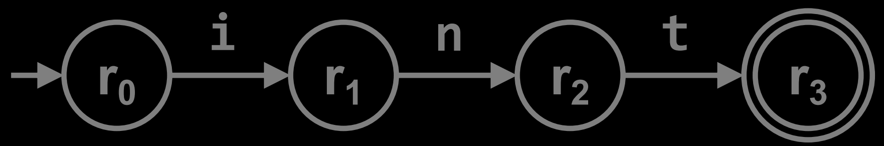


Let L be a right linear language.

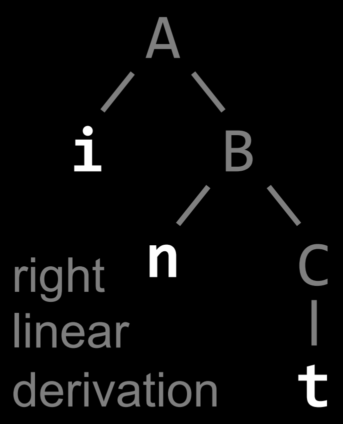
right linear grammar G

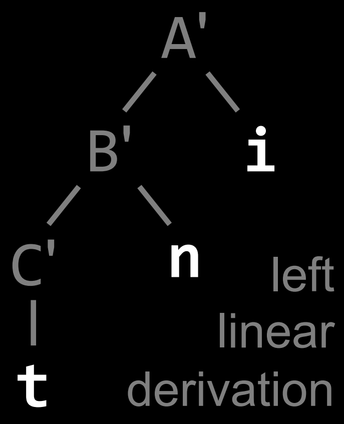
left linear grammar G^R

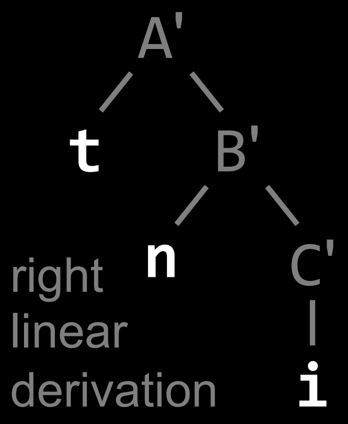
Then L is a finite state language.



Beweisidee: rechtslineare Sprachen sind auch linkslinear







Let L be a right linear language.

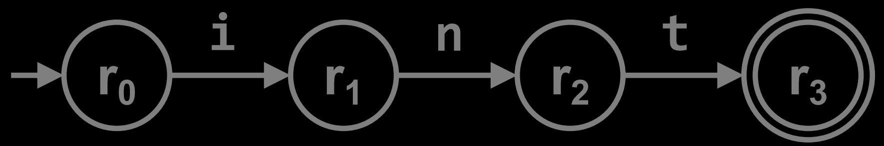
right linear grammar G

A reverted right linear grammar is left and right linear…

left linear grammar G^R

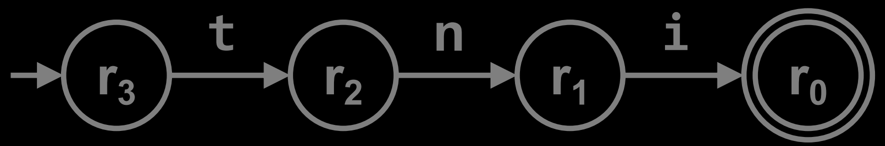
Take a right linear grammar G' for L^R.

Then L is a finite state language.

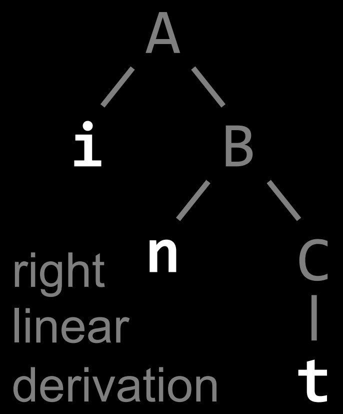


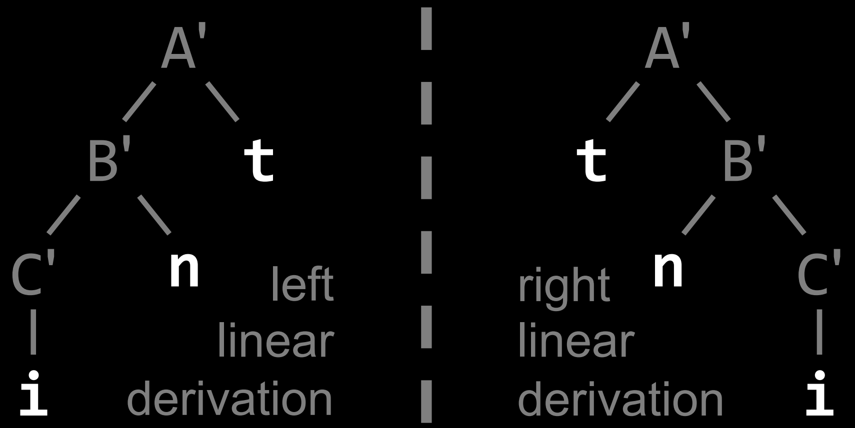
Can we reverse every automaton?

Then L^R is a finite state language.



Beweisidee: rechtslineare Sprachen sind auch linkslinear





$$L = L^{RR}$$

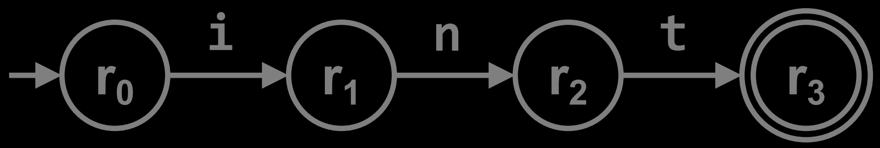
Let L be a right linear language.

right linear grammar G

G'^R is a left linear grammar with $L(G'^R) = L^{RR}$.

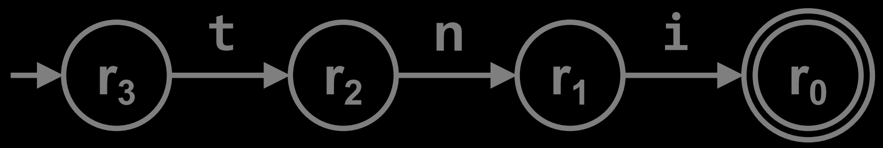
Take a right linear grammar G' for L^R.

Then L is a finite state language.



Can we reverse every automaton?

Then L^R is a finite state language.

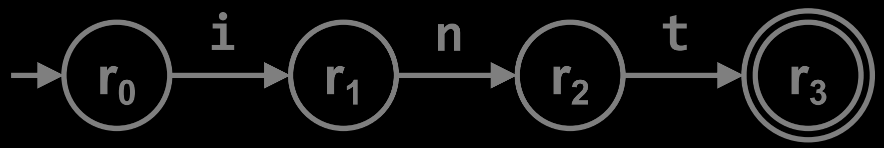


Beweisidee: linkslineare Sprachen sind auch rechtslinear

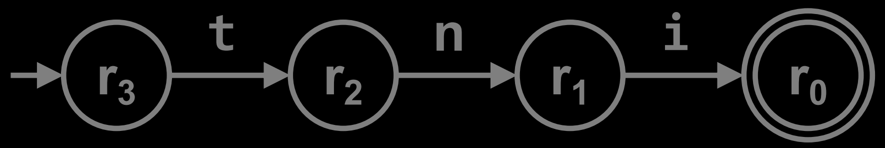
Let L be a left linear language.

Then L^R is a right linear language.

Then L^R is a finite state language.

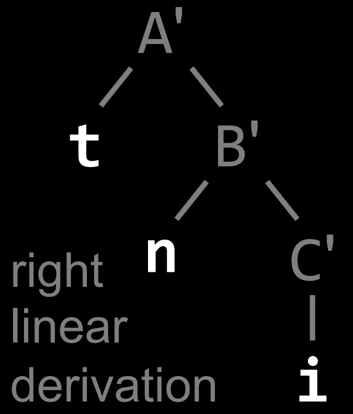


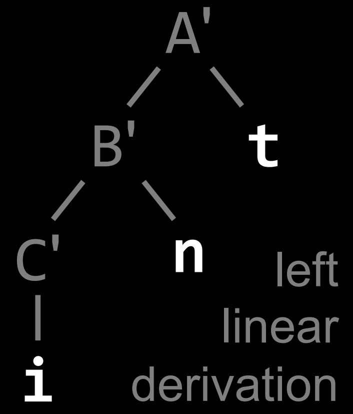
Can we reverse every automaton?



Then L^{RR} is a finite state language.

L is a finite state language.





L is a right linear language.

$$M' = (Q \cup \{q_0'\}, \Sigma, \delta', q_0', \{q_0\})$$

$$\boxed{\delta'(q_0', \varepsilon) = F}$$

$$\forall a \in \Sigma\colon \delta'(q_0', a) = \emptyset$$

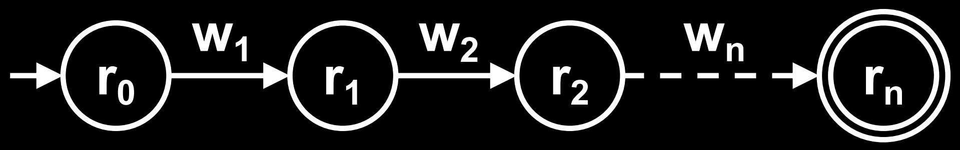
$$\forall q \in Q, a \in \Sigma\colon \delta'(p, a) = \{q \mid \delta(q, a) = p\}$$

Theorem If L is a finite state language, then L^R is a finite state language.

Let $\boldsymbol{M = (Q, \Sigma, \delta, q_0, F)}$ be a **DFA** such that $\boldsymbol{L(M) = L}$. Construct an NFA $\boldsymbol{M'}$ such that $\boldsymbol{L(M') = L^R}$ using the highlighted specification.

Claim 1 $\boldsymbol{w \in L(M) \to w^R \in L(M')}$

Since $\boldsymbol{w \in L(M)}$ we know that $\boldsymbol{w = w_1 w_2 \ldots w_n}$ and there exist states $\boldsymbol{r_0, r_1, \ldots, r_n}$ such that $\boldsymbol{r_0 = q_0}$, $\boldsymbol{r_n \in F}$ and $\boldsymbol{\forall i, 0 < i \le n\colon r_i = \delta(r_{i-1}, w_i)}$.



In this case M' is supposed to accept $\boldsymbol{w^R = \varepsilon w_n w_{n-1} \ldots w_1}$ with the state sequence $\boldsymbol{q_0', r_n, r_{n-1}, \ldots, r_1}$.

According to the specification, q_0' is the initial state and $r_1 = q_0$ is the final state for M', so to complete the proof, we only need to show that each transition is valid for M'.

The first transition satisfies $\boldsymbol{r_n \in \delta'(q_0', \varepsilon) = F}$.

$$M' = (Q \cup \{q_0'\}, \Sigma, \delta', q_0', \{q_0\})$$
$$\delta'(q_0', \varepsilon) = F$$
$$\forall a \in \Sigma: \delta'(q_0', a) = \emptyset$$
$$\boxed{\forall q \in Q, a \in \Sigma: \delta'(p, a) = \{q \mid \delta(q, a) = p\}}$$

Theorem

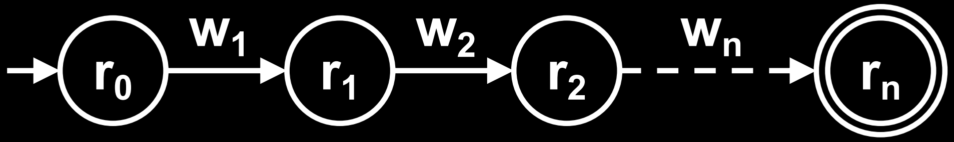
If L is a finite state language, then L^R is a finite state language.

Let $\boldsymbol{M = (Q, \Sigma, \delta, q_0, F)}$ be a **DFA** such that $\boldsymbol{L(M) = L}$. Construct an NFA $\boldsymbol{M'}$ such that $\boldsymbol{L(M') = L^R}$ using the highlighted specification.

Claim 1

$\boldsymbol{w \in L(M) \to w^R \in L(M')}$

Since $\boldsymbol{w \in L(M)}$ we know that $\boldsymbol{w = w_1 w_2 \dots w_n}$ and there exist states $\boldsymbol{r_0, r_1, \dots, r_n}$ such that $\boldsymbol{r_0 = q_0}$, $\boldsymbol{r_n \in F}$ and $\boldsymbol{\forall i, 0 < i \leq n: r_i = \delta(r_{i-1}, w_i)}$.



In this case M' is supposed to accept $\boldsymbol{w^R = \varepsilon w_n w_{n-1} \dots w_1}$ with the state sequence $\boldsymbol{q_0', r_n, r_{n-1}, \dots, r_1}$.

According to the specification, q_0' is the initial state and $r_1 = q_0$ is the final state for M', so to complete the proof, we only need to show that each transition is valid for M'.

The first transition satisfies $\boldsymbol{r_n \in \delta'(q_0', \varepsilon) = F}$. The remainder of the transitions are of the form: $\boldsymbol{r_{i-1} \in \delta'(r_i, w_i)}$. This becomes $\boldsymbol{r_{i-1} \in \{q \mid \delta(q, w_i) = r_i\}}$.

This follows immediately from $\boldsymbol{\delta(r_{i-1}, w_i) = r_i}$ which was established by $\boldsymbol{w \in L(M)}$.

$$M' = (Q \cup \{q_0'\}, \Sigma, \delta', q_0', \{q_0\})$$

$$\boxed{\begin{array}{c}\delta'(q_0', \varepsilon) = F \\ \forall a \in \Sigma: \delta'(q_0', a) = \emptyset\end{array}}$$

$$\forall q \in Q, a \in \Sigma: \delta'(p, a) = \{q | \delta(q, a) = p\}$$

Theorem

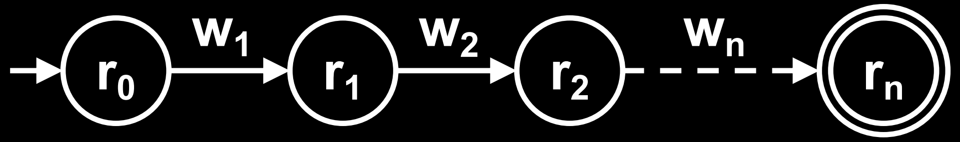
If L is a finite state language, then L^R is a finite state language.

Let $\boldsymbol{M = (Q, \Sigma, \delta, q_0, F)}$ be a **DFA** such that $\boldsymbol{L(M) = L}$. Construct an NFA $\boldsymbol{M'}$ such that $\boldsymbol{L(M') = L^R}$ using the highlighted specification.

Claim 2

$\boldsymbol{w \in L(M') \to w^R \in L(M)}$

Since $\boldsymbol{w \in L(M')}$ we know that $\boldsymbol{w = w_1 w_2 \dots w_n}$ and there exist states $\boldsymbol{r_0, r_1, \dots, r_n}$ such that $\boldsymbol{r_0 = q_0'}$, $\boldsymbol{r_n \in \{q_0\}}$ and $\boldsymbol{\forall i, 0 < i \leq n: r_{i+1} = \delta'(r_i, w_{i+1})}$.



In this case M' is supposed to accept $\boldsymbol{w^R = \varepsilon w_n w_{n-1} \dots w_1}$ with the state sequence $\boldsymbol{q_0', r_n, r_{n-1}, \dots, r_1}$.

All transitions on q_0' are defined by the construction rules. Thus, we know that $\boldsymbol{w_1 = \varepsilon}$ and $\boldsymbol{r_1 \in F}$.

$$M' = (Q \cup \{q_0'\}, \Sigma, \delta', q_0', \{q_0\})$$

$$\delta'(q_0', \varepsilon) = F$$

$$\forall a \in \Sigma : \delta'(q_0', a) = \emptyset$$

$$\boxed{\forall q \in Q, a \in \Sigma : \delta'(p, a) = \{q \mid \delta(q, a) = p\}}$$

Theorem

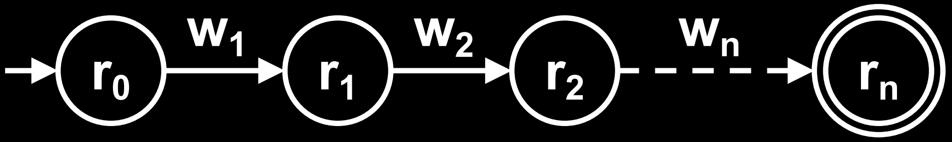
If L is a finite state language, then L^R is a finite state language.

Let $\boldsymbol{M = (Q, \Sigma, \delta, q_0, F)}$ be a **DFA** such that $\boldsymbol{L(M) = L}$. Construct an NFA $\boldsymbol{M'}$ such that $\boldsymbol{L(M') = L^R}$ using the highlighted specification.

Claim 2

$\boldsymbol{w \in L(M') \rightarrow w^R \in L(M)}$

Since $\boldsymbol{w \in L(M')}$ we know that $\boldsymbol{w = w_1 w_2 \dots w_n}$ and there exist states $\boldsymbol{r_0, r_1, \dots, r_n}$ such that $\boldsymbol{r_0 = q_0'}$, $\boldsymbol{r_n \in \{q_0\}}$ and $\boldsymbol{\forall i, 0 < i \leq n : r_{i+1} = \delta'(r_i, w_{i+1})}$.



In this case M' is supposed to accept $\boldsymbol{w^R = \varepsilon w_n w_{n-1} \dots w_1}$ with the state sequence $\boldsymbol{q_0', r_n, r_{n-1}, \dots, r_1}$.

All transitions on q_0' are defined by the construction rules. Thus, we know that $\boldsymbol{w_1 = \varepsilon}$ and $\boldsymbol{r_1 \in F}$.

M is supposed to accept $\boldsymbol{w^R = w_n w_{n-1} \dots w_2}$ with state sequence $\boldsymbol{r_n, r_{n-1}, \dots, r_1}$. Note that $\boldsymbol{r_n = q_0}$ and $\boldsymbol{r_1 \in F}$. It remains to show that $\boldsymbol{r_{i-1} = \delta(r_i, w_i)}$.

Since $\boldsymbol{w \in L(M')}$, we know that $\boldsymbol{r_i \in \delta'(r_{i-1}, w_i) = \{q \mid \delta(q, w_i) = r_{i-1}\}}$. So, $\boldsymbol{\delta(r_i, w_i) = r_{i-1}}$ as required.

Modellübersicht

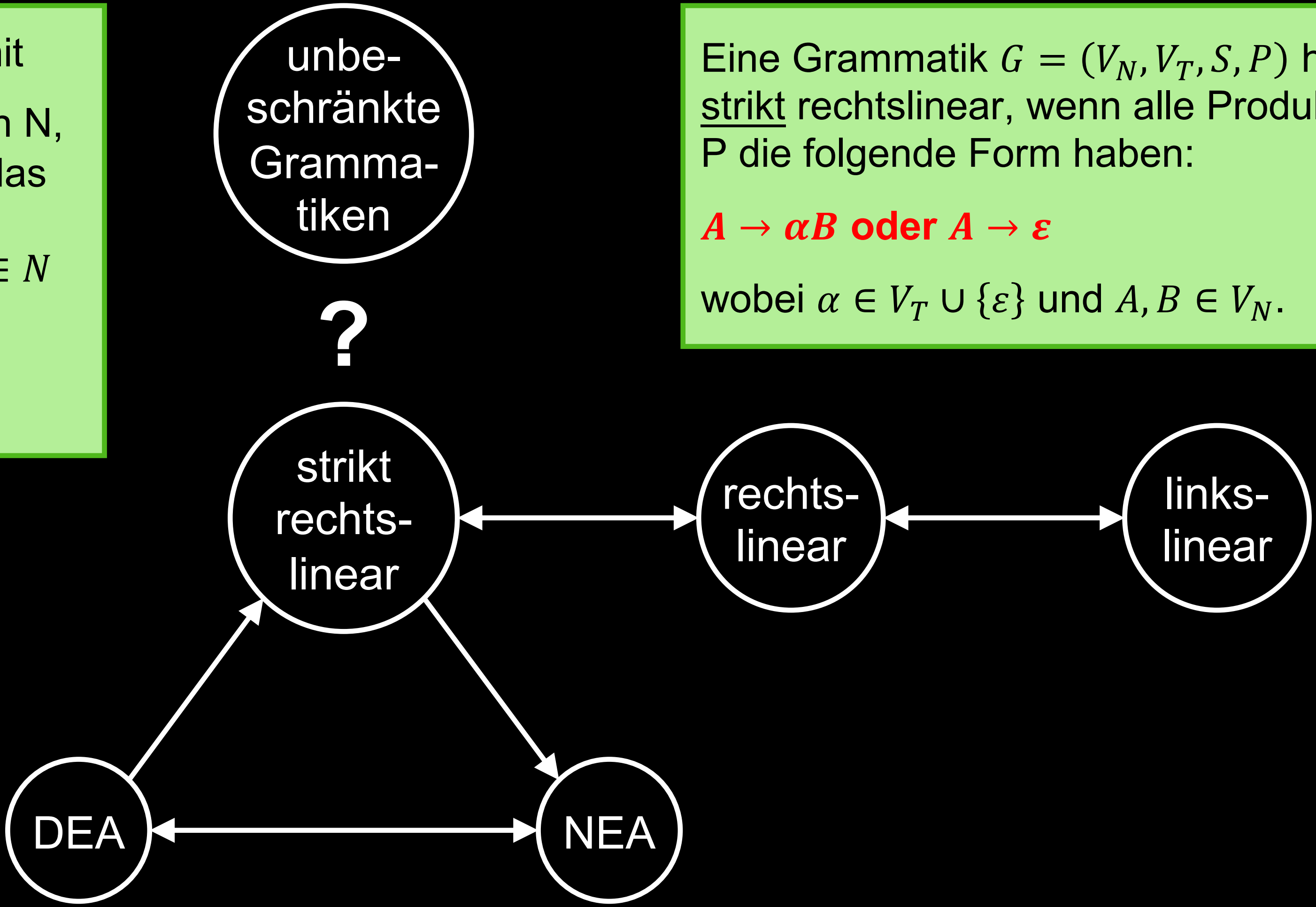
Eine Grammatik ist ein 4-Tupel (N,T,S,P) mit

- einer Menge von Nichtterminalsymbolen N,
- einer Menge von Terminalsymbolen T, das zu N disjunkt ist,
- einem ausgezeichnetes Startsymbol $S \in N$ und
- einer endlichen Menge $\boldsymbol{P \subset V^*NV^* \times V^*}$ sogenannter Produktionen.

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

$\boldsymbol{A \to \alpha B}$ **oder** $\boldsymbol{A \to \varepsilon}$

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



Benötigen wir die allgemeineren Produktionen überhaupt?