Automatentheorie

Reguläre Grammatiken

Prof. Dr. Franz-Karl Schmatzer
schmatzf@dhbw-loerrach.de

Literatur

- C.Wagenknecht, M.Hielscher; Formale Sprachen, abstrakte Automaten und Compiler; 2.Aufl. Springer Vieweg 2014;
- A.V.Aho, M.S.Lam,R.Savi,J.D.Ullman, *Compiler – Prinzipien,Techniken und Werkzeuge*. 2. Aufl., Pearson Studium, 2008.
- Güting, Erwin; *Übersetzerbau –Techniken, Werkzeuge, Anwendungen*, Springer Verlag 1999
- Sipser M.; Introduction to the Theory of Computation; 2.Aufl.; Thomson Course Technology 2006
- Hopecroft, T. et al; Introduction to Automata Theory, Language, and Computation; 3. Aufl. Pearson Verlag 2006

Agenda

- Einführung in Grammatiken
- Rechts und linkslineare Grammatiken
- Einteilung der Grammatiken
- Typ-3 Grammatiken
- Aufbau von Typ-3 Grammatiken

Grammatiken

Einführung

- Ein Satz einer gesprochen Sprache ist ein recht komplexes Gebilde, zeigt aber bei genauerem Hinsehen regelmäßige Strukturen auf, die die Bildung von Wörtern nach Regeln erlaubt.

- Im Deutschen z.B. findet man folgende Struktur (nur ein Ausschnitt):
  - Ein Satz <S> kann aus einer Nominalphase <NP> und einer Verbalphase <VP> zusammengesetzt sein.
  - In der Nominalphase herrscht das Nomen <N> vor. Ein Nomen kann ein Determinator <D> haben oder durch Konjunktionen <K> verbunden sein.
  - In der der Verbalphase <VP> herrscht das Verb <V> vor. Das Verb kann wieder mit einer Nominalphase oder mit einer Präpositionalphase <PP> verbunden sein.
  - Die Präpositionalphase <PP> setzt sich aus einer Präposition <P> und einer Nominalphase zusammen.

Aus @ Judit Macheiner, Das grammatische Varieté; Eichborn Verlag

Aus @ Judit Macheiner, Das grammatische Varieté; Eichborn Verlag

Grammatiken

Einführung Regeln

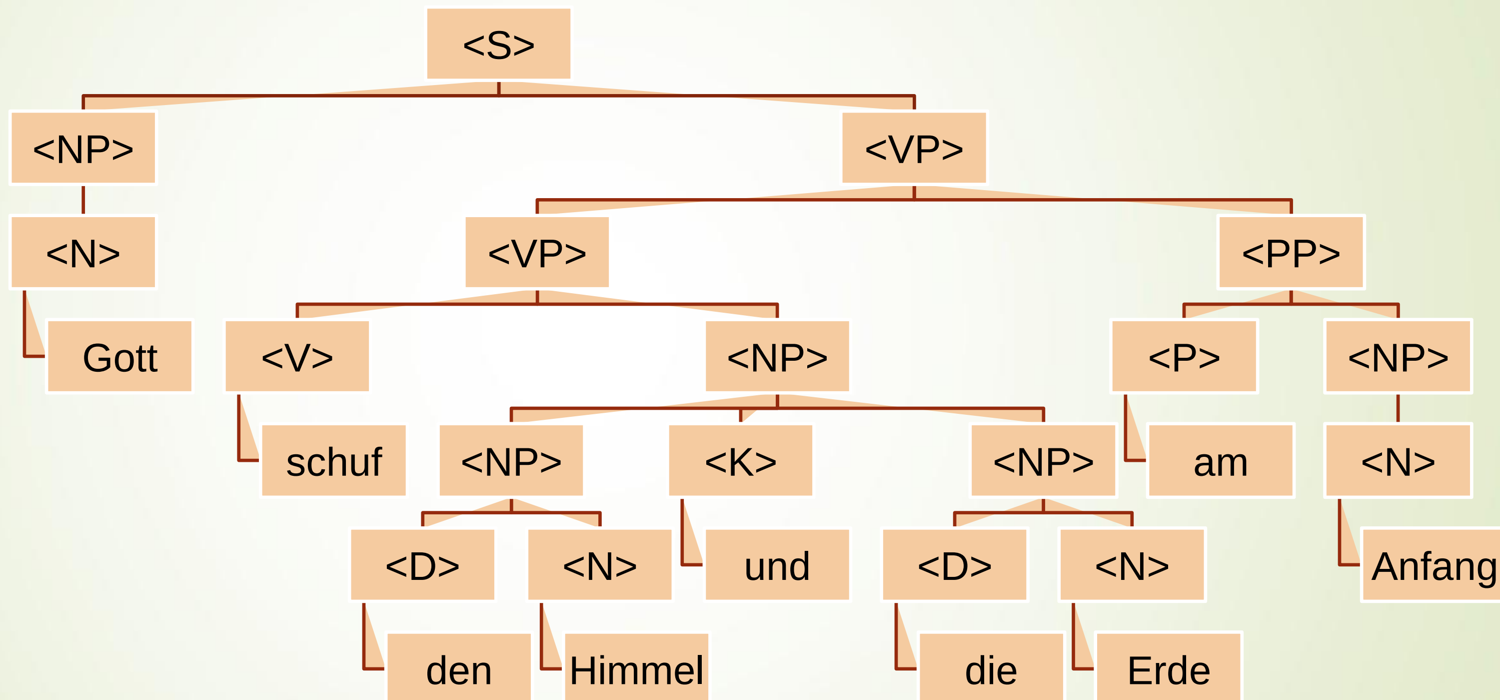
- Ein möglicher Ausschnitt aus der deutschen Grammatik

<S>	→	<NP><VP>
<NP>	→	<N> | <D><N>| <NP> <K> <NP>
<VP>	→	<V> <NP> | <VP<PP>
<PP>	→	<P> <NP>
<K>	→	und
<N>	→	Gott | Himmel | Erde | Anfang
<V>	→	schuf
<D>	→	die | den
<P>	→	am

- Man findet hier Variablen (<S>, <D>, usw,.) und Terminale (am, Gott, ….) mit denen Sätze tatsächlich gebildet werden.
- Ein möglicher Satz wird gebildet indem man bei dem Startsymbol <S> beginnend die einzelnen Regeln solange anwendet bis alle Variablen durch Terminale ersetzt werden.

Grammatiken

Einführung Ableitung



- Eine Ableitung des Satzes:
- Gott schuf den Himmel und die Erde am Anfang

Aus @ Judit Macheiner, Das grammatische Varieté; Eichborn Verlag

Grammatiken

Formale Definition

Grammatiken werden formal definieren als:

- Eine Grammatik G besteht aus 4 Komponenten (N, Σ, P, S) mit:
  - N eine endliche Menge von Variablen (Nichtterminale).
  - Σ ist ein Alphabet aus Terminalen mit $N \cap \Sigma = \emptyset$.
  - Statt Σ schreibt man oft T (Terminale), d.h. (N, T, P, S)
  - P ist eine Menge von Produktionen (Regeln).
    - Eine Produktion ist eine Element $P=(L,R) \in (N \cup \Sigma)^* \, N \, (N \cup \Sigma)^* \times (N \cup \Sigma)^*$.
    - Mit $l \in L$ und $r \in R$ schreibt man statt (l, r) besser: $l \rightarrow_P r$ bzw. $l \rightarrow r$
  - $S \in N$ ist eine Startvariable

Grammatiken

Definition Ableitung, Sprache Äquivalenz

- Die Ableitung eines Wortes:

  Sei $G = (N, \Sigma, P,S)$ eine Grammatik, und seine $v,w \in (N\cup \Sigma)^*$, so gilt $v \Rightarrow w$ (d.h v geht in w über), falls gilt:

  $\exists\, x,y \in (N\cup \Sigma)^*$ und eine Produktion $P \rightarrow Q$ so dass $v=xPy$ und $w=xQy$ ist.

  Gibt es eine Folge von Produktionen, die eine Ableitung v nach w implizieren schreibt man $v \Rightarrow^* w$

- Die Sprache L(G)

  Die von einer Grammatik G erzeugte Sprache L(G) ist die Menge aller terminalen Wörter, die durch G vom Startsymbol aus erzeugt werden können.

  $$L(G) = \{ w \in \Sigma^* \mid S \Rightarrow^*_G w \}$$

- Äquivalenz $L(G_1) = L(G_2)$

  Zwei Grammatiken G_1, G_2 heißen äquivalent $\Leftrightarrow L(G_1) = L(G_2)$

Beispiel einer Grammatik
Wortableitung

Grammatik G = (N,T,P,s) mit

N = {S, A, B}

T = {a,b,c}

P = {S → aS | cS | bA, A→ bA | cB, B → Bc | c}

s = S

Wortableitung

- w= aabbbcc
- S=>aS=>aaS =>aabA =>aabbA =>aabbbA=>aabbbcB =>aabbbcc (fertig)
- w= ccbba
- S=>cS=>ccS =>ccbA =>ccbbA ? Wort nicht ableitbar

Grammatiken

Einteilung I

- Die Grammatiken mit den allgemeinen Produktionen P sind sehr komplex und schwer zu durchschauen.
- Daher teilt man die Grammatiken weiter ein, indem man die Möglichkeiten für die Produktionen einschränkt. (Chomsky-Hierarchie)
- Eine Grammatik $G = (N, \Sigma, P, S)$ mit $P = (L,R)$ heißt
  - Regulär $\Leftrightarrow \forall\, L \to R$ gilt: $L \in N$ und $R \in (\{\varepsilon\} \cup \Sigma \cup \Sigma N)$ oder $R \in (\{\varepsilon\} \cup \Sigma \cup N\Sigma)$
    - Rechtslinear $\Leftrightarrow \forall\, L \to R$ gilt: $L \in N$ und $R \in (\{\varepsilon\} \cup \Sigma \cup \Sigma N)$
    - Linkslinear $\Leftrightarrow \forall\, L \to R$ gilt: $L \in N$ und $R \in (\{\varepsilon\} \cup \Sigma \cup N\Sigma)$

Grammatiken

Einteilung II

- Eine Grammatik $G = (N, \Sigma, P, S)$ mit $P = (L, R)$ heißt
  - kontextfrei $\Leftrightarrow \forall\, L \to R$ gilt: $L \in N$ und $R \in (\Sigma \cup N)^*$
  - kontextsensitiv $\Leftrightarrow \forall\, L \to R$ gilt: $L \in (N \cup \Sigma)^*\, N\, (N \cup \Sigma)^*$ und $R \in (\Sigma \cup N)^*$
    - Entweder $\exists\, u, v, w \in (N \cup \Sigma)^*$ und $\exists\, A \in N$, so dass $L = uAv$ und $R = uwv$ und $|w| \geq 1$ oder die Produktion hat die Form $S \to \varepsilon$
    - S kommt nicht in R (das heißt auf der rechten Seite) vor.
  - beschränkt $\Leftrightarrow \forall\, L \to R$ gilt:
    - Entweder $|L| \leq |R|$ oder die Produktion hat die Form $S \to \varepsilon$.
    - S kommt nicht in R (das heißt auf der rechten Seite) vor.

Grammatiken

Formale Einteilung

Sprachklasse	definiert	Name der Klasse
L_3	$\{L(G) \mid G$ ist regulär$\}$	Regulär, Typ 3
L_2	$\{L(G) \mid G$ ist kontextfrei$\}$	Kontextfrei, Typ 2
L_1	$\{L(G) \mid G$ ist kontextsensitiv$\}$	Kontextsensitiv, Typ 1
	$\{L(G) \mid G$ ist beschränkt$\}$	
L_0	$\{L(G) \mid G$ ist eine Grammatik$\}$	Rekursiv aufzählbar, Typ 0
L	$\{L \subseteq T^* \mid T$ ist ein Alphabet$\}$	Sprache

Chomsky-Hierarchie

Es gilt: $L_3 \subset L_2 \subset L_1 \subset L_0 \subset L$

d.h. jeder der Sprachen L_i ist eine echte Obermenge zu der nächsten Sprache L_{i+1}

Grammatiken

Beispiele

- $L(A) = \{a^n \mid n > 0\}$ (Typ3 Grammatik)
  - Sie wird erzeugt von der Grammatik $G = \{\{S\},\{a\},\{S \rightarrow aS \mid a\},S)$
- $L(A) = \{a^n b^n \mid n > 0\}$ (Typ2 Grammatik)
  - Sie wird erzeugt von der Grammatik $G = \{\{S\},\{a,b\},\{S \rightarrow aSb \mid ab\},S)$
- $L(A) = \{a^n b^n c^n \mid n > 0\}$ (Typ1 Grammatik)
  - Sie wird erzeugt von der Grammatik $G = \{\{S, A, B, C\}, \{a, b, c\}, P, S)$ mit $P =$

$$
\{ \quad
\begin{aligned}
& S \rightarrow aSBC \mid aBC, \\
& CB \rightarrow BC, \\
& aB \rightarrow ab, \\
& bB \rightarrow bb, \\
& bC \rightarrow bc, \\
& cC \rightarrow cc \}
\end{aligned}
$$

Aufgabe Grammatiktypen

Von welchem Typ sind folgende Grammatiken G=(N, {0,1}, P, S). Geben Sie dazu auch Worte der Sprache an und überprüfen anhand einer Ableitung, ob $w \in L(G)$ ist

1. $N=\{S,A\}$, $P=\{S \to 0A,\ A \to 0A \mid 1\}$ $w=001$
2. $N=\{S,B\}$, $P=\{S \to B1,\ B \to B0 \mid 0\}$ $w=001$
3. $N=\{S,A,B,C\}$, $P=\{S \to 0S \mid 1S \mid 0A,\ A \to 1B,\ B \to 0C,\ C \to \varepsilon\}$ $w=11010$
4. $N=\{S,A\}$, $P=\{S \to 0A,\ A \to 0A1 \mid 1\}$ $w=000111$
5. $N=\{S,A,B\}$, $P=\{S \to 0A0 \mid 1B1,\ A \to 1A0 \mid \varepsilon,\ B \to 0B1 \mid \varepsilon\}$ $w=011000$
6. $N=\{S,A,B\}$, $P=\{S \to 0AB \mid 1BA, 0A \to 01B0 \mid \varepsilon, 1B \to 00A1 \mid 1, 0B \to 1\}$ $w=00101$

Grammatiken

Wort-Ableitung

- Leiten Sie das Wort $w = a^4$ für vorherige Typ3 Grammatik ab.
- Leiten Sie das Wort $w = a^4 b^4$ für vorherige Typ2 Grammatik ab.
- Leiten Sie das Wort $w = a^4 b^4 c^4$ für vorherige Typ1 Grammatik ab.

Typ-3-Grammatiken

Einführung

- Man unterscheidet
  - rechtslineare und
  - linkslineare Grammatiken
- Beispiel: $G = (\{S, A, B\}, \{0, 1\}, P, S)$ mit
  - $P = \{ S \to 0S, S \to 1S, S \to 0A, A \to 0B, A \to 1B, B \to 0, B \to 1\}$
- Wie werden nun Wörter gebildet?
  - Anwenden der Regeln $uA \Rightarrow uw$ genau dann, wenn $A \to w \in P$
  - Ableiten solange möglich, bis nur noch Terminalsymbole übrig sind.
  - Alle möglichen Ableitungen (reflexive Hülle) definiert die Sprache L(G)

    $L(G) = \{ w \in \Sigma^* \mid S \Rightarrow^* w \}$
- Zwei Grammatiken G_1 und G_2 heißen äquivalent wenn $L(G_1) = L(G_2)$

Typ-3-Grammatiken

formale Definition

- Eine rechtslineare Grammatik $G = (N,\Sigma,P,S)$ wird definiert als
  - N eine Menge von Nicht-Terminalsymbolen, die zu Σ disjunkt ist.
  - Σ eine Menge von Terminalsymbolen
  - P eine Relation $\subseteq N \times (\Sigma N \cup \Sigma \cup \{\varepsilon\})$
  - $S \in N$, dem Startsymbol
  - Eine Element $p \in P$ mit $P=(L,R)$ und $p = (l, r)$ heißt Produktion oder Regel mit

    $l \in N$ und $r \in \{\Sigma N \cup \Sigma \cup \{\varepsilon\}\}$ und der Notation: $l \rightarrow r$

    (l geht über in r oder l wird durch r ersetzt)
  - D.h. auf bei einer Produktion $l \rightarrow r$ stehen alle Nicht-Terminale N nur rechts von den Terminalen Σ in r.

Typ-3-Grammatiken

Beispiel rechtslineare Grammatik: Ableitung

- $G = (\{S, A, B\}, \{0, 1\}, P, S)$ mit
  - $P = \{ S \rightarrow 0S, S \rightarrow 1S, S \rightarrow 0A, A \rightarrow 0B, A \rightarrow 1B, B \rightarrow 0, B \rightarrow 1\}$
- Ableiten eines Wortes w = 01001
  1. Starten mit S
  2. Schauen welche Regeln mit S beginnen, dann Anwenden einer dieser Regeln.
  3. Solange fortfahren, bis nur noch Terminalsymbolen übrig sind.
  - $S \Rightarrow 0S$ (Regel $S \rightarrow 0S$)
  - $S \Rightarrow 01S$ (Regel $S \rightarrow 1S$)
  - $S \Rightarrow 010A$ (Regel $S \rightarrow 0A$)
  - $S \Rightarrow 0100B$ (Regel $A \rightarrow 0B$)
  - $S \Rightarrow 01001$ (Regel $B \rightarrow 1$)
  - **$S \Rightarrow^* 01001$ und damit $01001 \in L(G)$**

Typ-3-Grammatiken

formale Definition

- Eine linkslineare Grammatik $G = (N,\Sigma,P,S)$ wird genauso wie eine rechtslineare Grammatik definiert außer
  - P eine Relation $\subseteq N \times (N\Sigma \cup \Sigma \cup \{\varepsilon\})$, d.h $l \in N$ wie oben aber r wird zu $r \in (N\Sigma \cup \Sigma \cup \{\varepsilon\})$
  - D.h. auf bei einer Produktion $l \rightarrow r$ stehen alle Nicht-Terminale N nur links von den Terminalen Σ in r.

Typ-3-Grammatiken

Beispiel linkslineare Grammatik: Ableitung

- $G = (\{S, A, B\}, \{0, 1\}, P, S)$ mit
  - $P = \{ S \to S0, S \to S1, S \to A0, A \to B0, A \to B1, B \to 0, B \to 1\}$
- Ableiten eines Wortes
  1. Starten mit S
  2. Schauen welche Regeln mit S beginnen, dann Anwenden einer dieser Regeln.
  3. Solange fortfahren, bis nur noch Terminalsymbolen übrig sind.
  - $S \Rightarrow S0$ (Regel $S \to S0$)
  - $S \Rightarrow S10$ (Regel $S \to S1$)
  - $S \Rightarrow A010$ (Regel $S \to A0$)
  - $S \Rightarrow B$ (Regel $A \to B0$)
  - $S \Rightarrow$ (Regel $B \to 1$)
  - **$S \Rightarrow^* 10010$ und damit $10010 \in L(G)$**

Typ-3-Grammatiken

rechts- und linkslineare Grammatik

- Eine rechtslineare Grammatik hat auf der linken Seite immer ein nicht Terminalsymbol und auf der rechten Seite ein Wort aus $\Sigma N \cup \Sigma \cup \{\varepsilon\}$, d.h. $(A,B \in N,\ a \in \Sigma)$
  - $A \to aB$ (1)
  - $A \to a$ (2)
  - $A \to \varepsilon$ (3)
- Eine linkslineare Grammatik hat auf der linken Seite immer ein nicht Terminalsymbol und auf der rechten Seite ein Wort aus $N\Sigma \cup \Sigma \cup \{\varepsilon\}$, d.h. $(A,B \in N,\ a \in \Sigma)$
  - $A \to Ba$ (1)
  - $A \to a$ (2)
  - $A \to \varepsilon$ (3)

Typ-3-Grammatiken

Vereinfachte Grammatik

- Statt der 3 Regeln (1) bis (3) lässt sich die Grammatik vereinfachen und nur mit Regeln (1) und (3) erstellen
- Jede Regel (2) $A \rightarrow a$ wird ersetzt durch:
  - $A \rightarrow aT$ bzw. $A \rightarrow Ta$ (1) T ein neues nicht Terminalsymbol
  - $T \rightarrow \varepsilon$ (3)
- Beispiel: $G = (\{S, A, B\}, \{0, 1\}, P, S)$ mit
  - $P = \{ S \rightarrow 0S, S \rightarrow 1S, S \rightarrow 0A, A \rightarrow 0B, A \rightarrow 1B, B \rightarrow 0, B \rightarrow 1\}$
  - Ersetzen von P durch P_R
  - $P_R = \{ S \rightarrow 0S, S \rightarrow 1S, S \rightarrow 0A, A \rightarrow 0B, A \rightarrow 1B, B \rightarrow 0T, B \rightarrow 1T, T \rightarrow \varepsilon \}$

Erzeugen von Grammatiken

Reguläre Ausdrücke

- Der Ausdruck L=a* ergibt die Produktion S->aS | ε
- Der Ausdruck L =(a+b)B ergibt die Produktion S->aB | bB

Aufgabe Typ3 Grammatiken

reguläre Ausdrücke

- Erstellen Sie eine rechtslineare Grammatik zu folgenden regulären Ausdrücken
  - R = (a+b)*c
  - R = (0+1)*11
  - R = 0*11*011*
  - R = (a+b)c*b*a
  - R = (0+1)(11)*0
  - R = (0+1)*01(0+1)(0+1)

Typ-3-Grammatiken

Äquivalenz Typ-3-Grammatik und endliche Automaten

- Zu jeder Typ-3 Grammatik G über das Alphabet Σ existiert ein endlicher Automat A über S mit L(G) = L(A)
- Beweis in 2 Schritten
  - Sei G = (N,Σ,P,S) eine rechtslineare Grammatik dann existiert ein Automat A = (Q, Σ, δ, s_0, F) mit L(G) = L(A)
  - Sei A = (Q, Σ, δ, s_0, F) ein endlicher Automat dann existiert ein Grammatik G = (N,Σ,P,S) mit L(G) = L(A)

Typ-3-Grammatiken

Äquivalenz $G \Rightarrow A$

- Beweis durch Konstruktion:
  - Sei $G = (N,\Sigma,P,S)$ eine rechtslineare Grammatik, die nur die Regeln (1) und (3) enthält.
  - $A = (Q, \Sigma, \delta, s_0, F)$ sei definiert als:
  - Die Zustandsmenge $S \in A$ besteht aus den nicht Terminalsymbolen N
  - Das Startsymbol S von G wird Startzustand S_0 von A
  - Die Endzustandsmenge F von A enthält alle nicht Terminalsymbolen, zu denen eine ε-Regel existiert.
  - Die Zustandsübergänge ergeben sich aus den Produktionsregeln
    - Regeln $B \rightarrow aC$ ergeben $(B,a,C) \in \delta$,
    - Regeln $B \rightarrow \varepsilon$ ergeben keine Zustandsübergänge
    - Formal $\delta = \{(B,a,C) \mid B \rightarrow aC \in P\}$

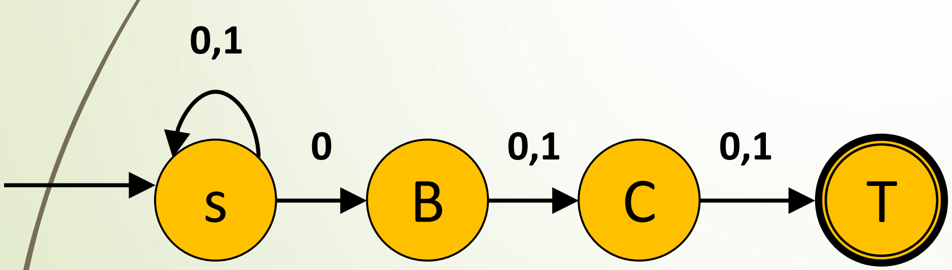
Typ-3-Grammatiken

Äquivalenz $G \Rightarrow A$ Beispiel

- Gegeben die Grammatik $G = (\{S, B, C, T\}, \{0, 1\}, P, S)$ mit

$P = \{S \to 0S,\ S \to 1S,\ S \to 0B,\ B \to 0C,\ B \to 1C,$
$C \to 0T,\ C \to 1T,\ T \to \varepsilon\}$

Der Automat $A = (\{S, B, C, T\}, \{0, 1\}, \delta, S, T)$



δ	0	1
S	{S,B}	{S}
B	{C}	{C}
C	{T}	{T}
T	∅	∅

Typ-3-Grammatiken

Äquivalenz A $\Rightarrow$ G

- Beweis durch Konstruktion:
  - Sei A = (Q, Σ, δ, s_0, F) ein endlicher Automat. Wir konstruieren einen Grammatik G = (N,Σ,P,S)
  - N = Q (Die Nicht-Terminale sind gerade die Zustände)
  - Σ der Grammatik und Σ des Automaten sind identisch.
  - Wir geben und die Produktionen P (Regeln) an:
    - Falls $\delta(s,a) = s'$ dann ist $s \to as'$ eine Regel,
    - Falls $s \in F$ dann ist $s \to \varepsilon$ eine Regel
    - Formal $P = \{s \to as' \mid \delta(s,a) = s'\} \{s \to \varepsilon \mid s \in F\}$

Typ-3-Grammatiken

Äquivalenz A $\Rightarrow$ G Beispiel

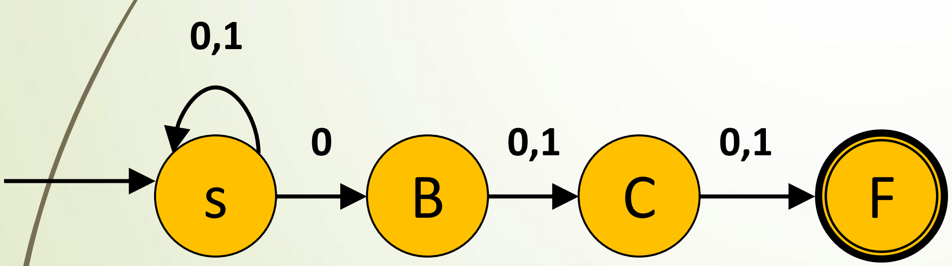
- Gegeben der Automat A =({S, B, C, F), {0, 1}, δ, S, F)

  Konstruktion nach vorherigem Satz

  P = {S → 0S, S → 1S, S → 0B, B → 0C, B → 1C,

  C → 0F, C → 1F, F → ε }

  Die Grammatik G =({S, B, C, F), {0, 1}, P, S)



δ	0	1
S	{S,B}	{S}
B	{C}	{C}
C	{F}	{F}
F	∅	∅

Aufgabe Typ3-Grammatik

Umwandeln in einen Automaten

Konstruieren Sie einen endlichen Automaten aus folgender regulären Grammatik G

1. $G=\{S,A\}$, $P=\{S \rightarrow 0A,\ A \rightarrow 0A \mid 1\}$
2. $G=\{S,A,B\}$, $P=\{S \rightarrow 1S \mid 0A,\ A \rightarrow 1A \mid 0B,\ B \rightarrow 1B \mid 0A \mid \varepsilon\}$
3. $G=\{S,A,B\}$, $P=\{S \rightarrow 0A \mid 1B,\ A \rightarrow 0S \mid 0,\ B \rightarrow 1S \mid 1\}$