

# Formale Sprachen

---

- Ein endlicher Akzeptor akzeptiert eine bestimmte Menge von Wörtern
- Die Menge aller von einem Automaten akzeptierten Worte bilden eine formale Sprache
- Automaten sind nicht die einzige Möglichkeit, formale Sprachen zu definieren

- Eine **formale Sprache** ist die Menge zulässiger Wörter über einem bestimmten (endlichen) Alphabet
- Die **Syntax** einer formalen Sprache beinhaltet die Regeln, nach denen erlaubte Wörter gebildet werden
- Die **Semantik** einer formalen Sprache sagt etwas über die Bedeutung der gebildeten Wörter aus

## Beispiel: Java als formale Sprache

- Eingabealphabet: Schlüsselwörter (**if**, **while** etc.) und erlaubte Zeichen (Klammern, Rechenzeichen etc.)
- Syntax: Welche Abfolgen von Eingabezeichen sind erlaubt?
- Was sind die Wörter der formalen Sprache Java?
  - Alle syntaktisch korrekten Programmcodes
- Was ist die Semantik eines solchen “Wortes”?
  - Das, was das Programm tut

- Eine formale **Grammatik** definiert die Regeln, nach denen die Wörter einer Sprache gebildet werden
- Formal ist eine solche Grammatik ein 4-Tupel  $G = \{V, \Sigma, P, S\}$  mit
  - $V$ : Menge der Nichtterminalsymbole (Variablen)<sup>1</sup>
  - $\Sigma$ : Menge der Terminalsymbole (Alphabet)
  - $P$ : Menge von Produktionsregeln
  - $S$ : Ein Startsymbol aus der Menge  $V$
- Übliche Festlegung:
  - Großbuchstaben für Nichtterminalsymbole
  - Kleinbuchstaben für Terminalsymbole

---

<sup>1</sup>In der Literatur findet sich auch eine alternative Definition, bei der  $V$  Terminalsymbole **und** Nichtterminalsymbole enthält

## Beispiel 1

- Wir betrachten eine Grammatik  $G = \{V, \Sigma, P, S\}$  mit
  - $V = \{S, A, B\}$  (mit  $S$  als Startsymbol)
  - $\Sigma = \{a, b\}$
  - $P = \{S \rightarrow aA, S \rightarrow bB, A \rightarrow aA, A \rightarrow b, B \rightarrow bB, B \rightarrow \varepsilon\}$
- Das Symbol  $\varepsilon$  steht für das **leere Wort**, also ein Wort der Länge 0
- Produktionsregeln mit identischer linker Seite lassen sich zusammenfassen, zum Beispiel:
  - Statt  $S \rightarrow aA, S \rightarrow bB$  schreibt man  $S \rightarrow aA|bB$
- Um ein Wort der Sprache zu erzeugen, beginnt man mit  $S$  und wendet so lange passende Regeln an, bis keine Nichtterminalsymbole mehr enthalten sind, zum Beispiel:
  - $S \rightarrow aA \rightarrow aaA \rightarrow aab$
  - $S \rightarrow bB \rightarrow bbB \rightarrow bbbB \rightarrow bbb$

## Beispiel 2

- Grammatik  $G = \{V, \Sigma, P, S\}$  zur Erzeugung natürlicher Zahlen mit
  - $V = \{\text{ZAHL}, \text{ZIFFER}\}$
  - $\Sigma = \{0, 1, \dots, 9\}$
  - Startsymbol ZAHL
  - $P = \{R_1, R_2\}$  mit
    - $R_1 : \text{ZAHL} \rightarrow \text{ZIFFER} | \text{ZAHL ZIFFER}$
    - $R_2 : \text{ZIFFER} \rightarrow 0 | 1 | \dots | 9$
- Wie kann die Ziffernfolge 053 abgeleitet werden?

# Ableitungsbäume

- Die Ableitung eines Wortes ausgehend vom Startsymbol kann durch einen Ableitungsbaum verdeutlicht werden

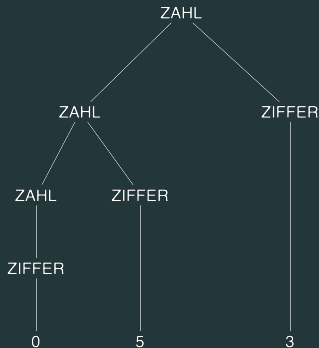


Abbildung 1: Ableitungsbaum für 053



## Aufgabe 1

- Der Nikolaus stellt entsetzt fest, dass es weniger als drei Monate bis zum Nikolaustag sind. Dabei hatte er sich vorgenommen, dieses Jahr die Lagerplätze in seinem Geschenkedept besser zu beschriften, und zwar nach folgendem Schema:
  - Lagerort (A bis D)
  - Regalreihe (A bis Z)
  - Feld (01 bis 99)
  - Fach (A bis Z)
  - Behälter (1 bis 9)
- Gib eine Grammatik an, die die Sprache der gültigen Lagerplatznummern erzeugt.

## Aufgabe 2

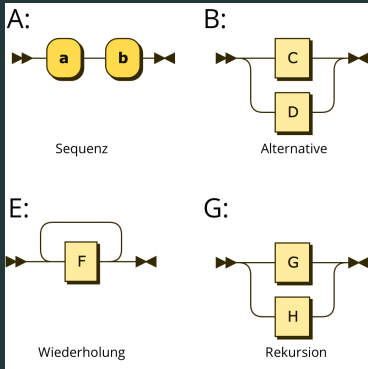
- Gib eine Grammatik an, die Datumsangaben in der Form `tt.mm.jjjj` erzeugt
  - Verwende als Terminalsymbole ausschließlich die Ziffern von 0 bis 9
  - Es sollen nur Jahreszahlen zwischen 1000 und 2999 erlaubt sein
  - Alle Monate dürfen 31 Tage haben

## Aufgabe 3

- Gib eine Grammatik an, die für  $\Sigma = \{a, b, c, d\}$  die Sprache aller Palindrome ungerader Länge erzeugt
- Erweitere die Grammatik so, dass auch Palindrome gerader Länge erlaubt sind

- Grafische Darstellung von Grammatiken bzw. deren Ableitungsregeln
  - Nichtterminalsymbole: Rechtecke/Quadrate
  - Terminalsymbole: Kreise/abgerundete Rechtecke
- Ein Teildiagramm für jedes Nichtterminal

# Syntaxdiagramm: Beispiele



Die Abbildung zeigt Ableitungen für die Nichtterminale A, B, E und G. Die dargestellte Grammatik ist nicht vollständig, da für die Nichtterminale C, D, F und H keine Ableitungen angegeben sind.

Die Syntaxdiagramme wurden erstellt mit dem Railroad Diagram Generator (<https://www.bottlecaps.de/rr/ui>)

# Erweiterte Backus-Naur-Form (EBNF)

- Standardisierte<sup>2</sup>, textuelle Beschreibung von Ableitungsregeln
- Festlegungen:

**Zeichenketten in Anführungszeichen** Terminale

**Zeichenketten ohne Anführungszeichen** Nichtterminale

- | Trennzeichen für Alternativen
- , Verbindung (Konkatenation) von Einzelzeichen
- { } Beliebige Wiederholung (auch 0 Mal)
- [ ] Option
- ( ) Gruppierung
- X \* X-malige Wiederholung des Nachfolgenden Elements

---

<sup>2</sup>Leider nicht so ganz, da es unterschiedliche Varianten der EBNF gibt...

`ZifferOhneNull = "1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"`

`Ziffer = "0"|ZifferOhneNull`

`Zahl = ["-"] Ziffer {Ziffer}`

`Einundvierzig = "4","1"`

`Weihnachtsmann = 3 * ("H","o","!")`