1 Automata theory and Formal languages

1.1 Introduction

Example: Arithmetical Expressions: EXPR

$$\Sigma = \{a, +, *, (,)\}$$

a is a variable for constants or variables

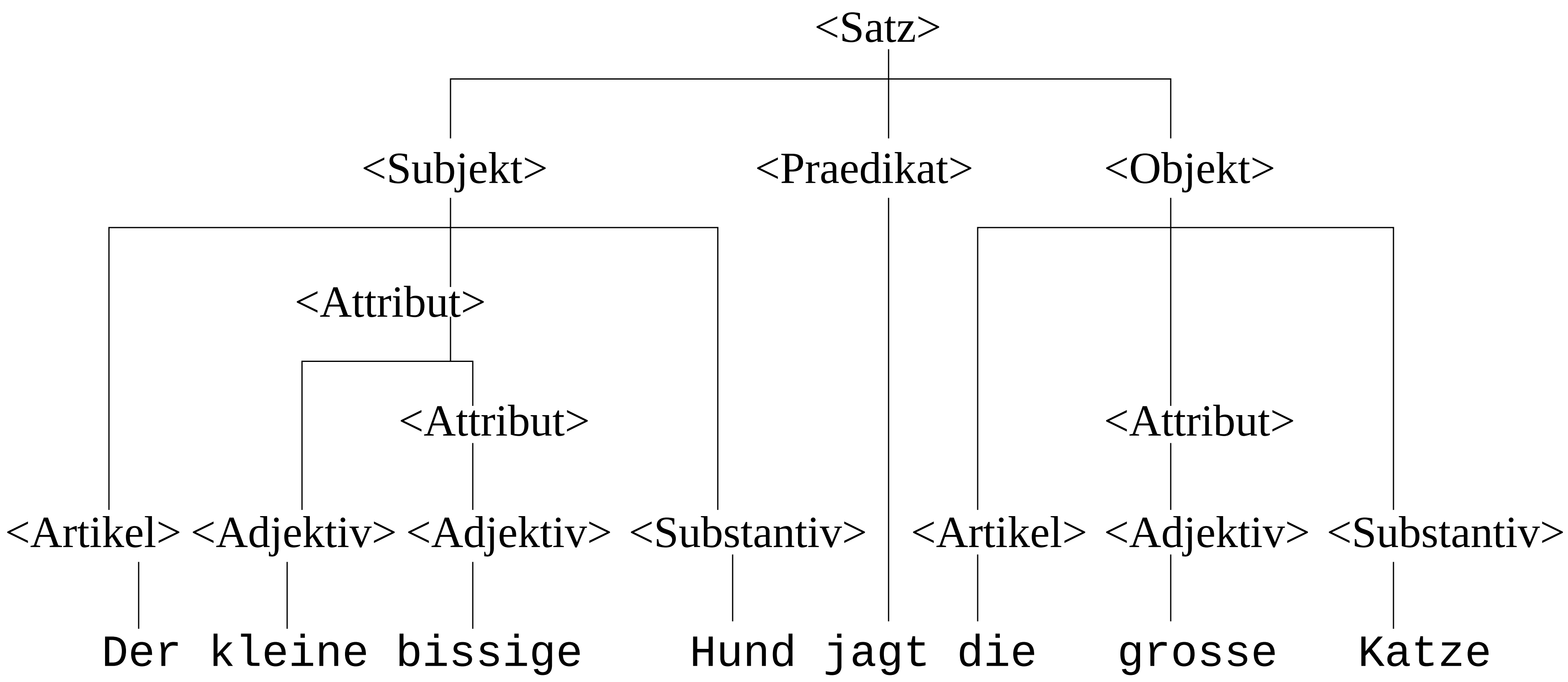
$$(a-a)*a+a/(a+a)-1 \in \mathtt{EXPR}$$

$$(((a))) \in \mathtt{EXPR}$$

$$((a+)-a(\notin \mathtt{EXPR}$$

How we can formalize this?

Example: The German grammar



At least a part of the structure we could do by context-free grammars (stay tuned).

1.1.1 Grammars

Grammar $G = (V, \Sigma, P, S)$

- V, Variables
- Σ, Terminal alphabet ($V \cap \Sigma = \emptyset$)
- $P \subseteq (V \cup \Sigma)^+ \times (V \cup \Sigma)^*$, Rules, $|P| < \infty$

  Every left part of the rule has at least one variable
- S, Start variable

Example: Balanced parentheses

$G = (\{E,T,F\},\{a,+,*,(,)\},P,E)$, where

$$
\begin{aligned}
P = \{&E \to T,\\
&E \to E + T,\\
&T \to F,\\
&T \to T * F,\\
&F \to a,\\
&F \to (E)\}
\end{aligned}
$$

Transition relation $\Rightarrow$

Given a grammar $G = (V, \Sigma, P, S)$.

$u \Rightarrow_G v$ holds if

$u = xyz \in (V \cup \Sigma)^*$,

$v = xy'z \in (V \cup \Sigma)^*$,

$y \rightarrow y' \in P$.

„u goes directly to v“ or v is derivable directly from u by G.

Subscript $_G$ is dropped, when it is clear which grammar is assumed.

Transition relation $\overset{*}{\Rightarrow}, \overset{n}{\Rightarrow}$

The length of the derivation :

$\forall u \in (V \cup \Sigma)^* : u \overset{0}{\Rightarrow} u$

$\forall u, v, w \in (V \cup \Sigma)^* : u \Rightarrow v \wedge v \overset{n}{\Rightarrow} w \longrightarrow u \overset{n+1}{\Rightarrow} w$

Derivation:

$\exists n \geq 0 : u \overset{n}{\Rightarrow} v \longrightarrow u \overset{*}{\Rightarrow} v$

Observation: $\overset{*}{\Rightarrow}$ is an reflexive and a transitive closure of $\Rightarrow$.

$u \overset{*}{\Rightarrow}_G v$ means „v is derivable from u“

The language generated by $G = (V, \Sigma, P, S)$

$$L(G) := \left\{ w \in \Sigma^* : S \overset{*}{\Rightarrow} w \right\}$$

Derivation

A sequence of words,

$$(\underbrace{w_1}_{=S}, \underbrace{w_2}_{\in(\Sigma\cup V)^*}, \ldots, \underbrace{w_{n-1}}_{\in(\Sigma\cup V)^*}, \underbrace{w_n}_{\in\Sigma^*})$$

is called a derivation of w_n if

$w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_n$.

Example:

$E \Rightarrow$	$E \rightarrow E+T$
$\Rightarrow E+T$	$E \rightarrow T$
$\Rightarrow T+T$	$T \rightarrow T*F$
$\Rightarrow T*F+T$	$T \rightarrow T*F$
$\Rightarrow T*F*F+T$	$T \rightarrow F$
$\Rightarrow F*F*F+T$	$F \rightarrow a$
$\Rightarrow a*F*F+T$	$F \rightarrow a$
$\Rightarrow a*a*F+T$	$F \rightarrow (E)$
$\Rightarrow a*a*(E)+T$	$E \rightarrow E+T$
$\Rightarrow a*a*(E+T)+T$	$E \rightarrow T$
$\Rightarrow a*a*(T+T)+T$	$T \rightarrow F$
$\Rightarrow a*a*(F+T)+T$	$F \rightarrow a$
$\Rightarrow a*a*(a+T)+T$	$T \rightarrow F$
$\Rightarrow a*a*(a+F)+T$	$F \rightarrow a$
$\Rightarrow a*a*(a+a)+T$	$T \rightarrow F$
$\Rightarrow a*a*(a+a)+F$	$F \rightarrow a$
$\Rightarrow a*a*(a+a)+a$	

1.1.2 Chomsky-Hierarchy

- An elegant specification for the languages
- A classification for languages

Classification for grammars

$G=(V,\Sigma,P,S)$ **[Noam Chomsky, 1956]**

Let $G=(V,\Sigma,P,S)$.

$\forall \ell \longrightarrow r \in P$:

Type 0: any

Type 1, context-sensitive: $|\ell| \leq |r|$

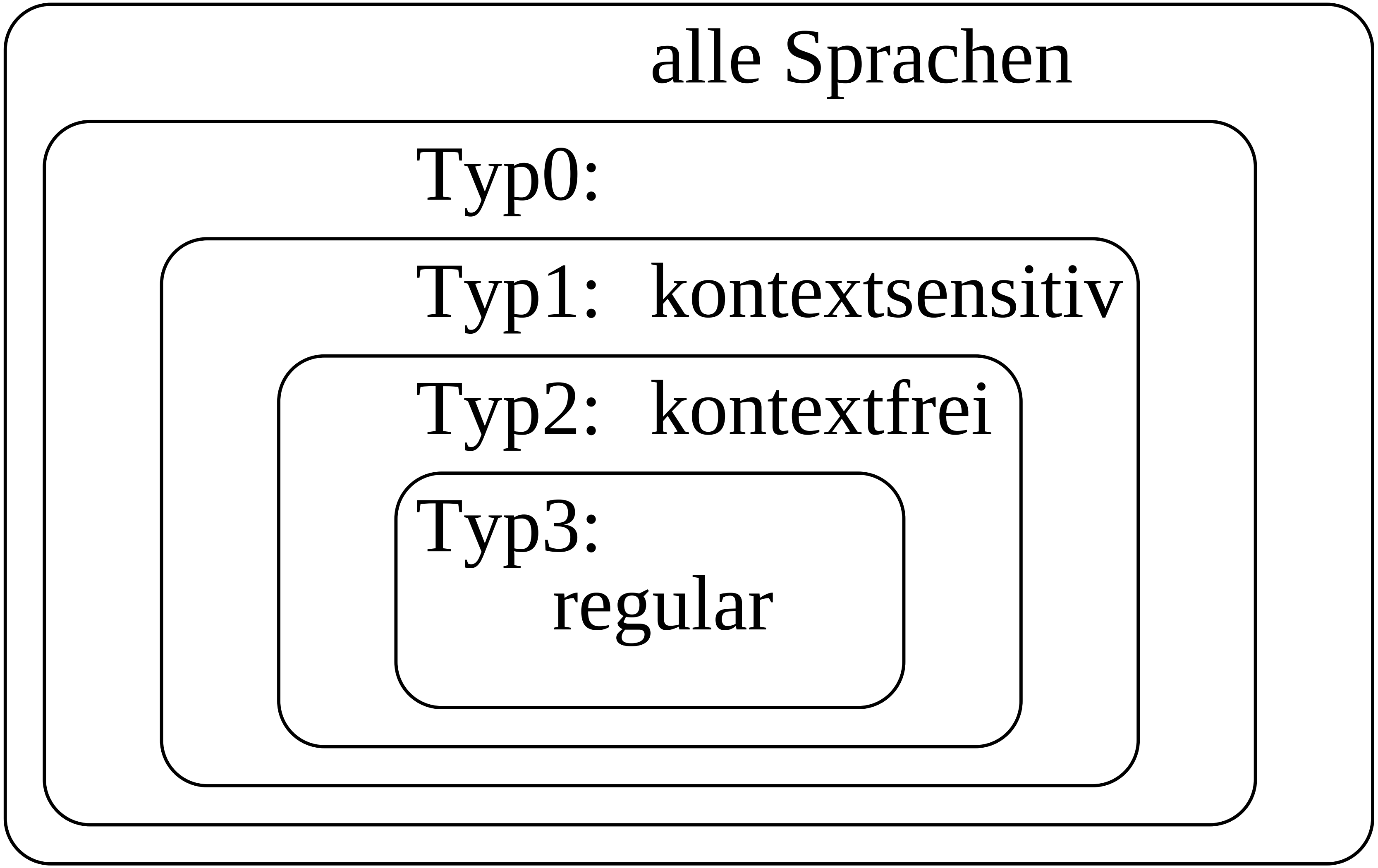
Special rules: $S \longrightarrow \varepsilon$ allowed if $S \notin r$,

Attention: in the literature not uniformly handled!

Type 2, context-free: Type 1 and $\ell \in V$ $A \longrightarrow \varepsilon$ allowed

Type 3, regular: Type 2 and $r \in \Sigma \cup \Sigma V$

Chomsky-Hierarchy



Example: Type 3

$G = (\{A, B\}, \{a, b\}, P, A)$, where

$$
\begin{aligned}
P = \{A &\rightarrow aA, \\
A &\rightarrow aB, \\
B &\rightarrow bB, \\
B &\rightarrow b\}
\end{aligned}
$$

Attention: $L(G) = \{a^n b^m : n \geq 1, m \geq 1\}$

Proof — basic approach:

1. $L(G) \supseteq \{a^n b^m : n \geq 1, m \geq 1\}$
2. $L(G) \subseteq \{a^n b^m : n \geq 1, m \geq 1\}$

Always by complete induction.

$$\boxed{G = (\{A, B\}, \{a, b\}, \{A \to aA, A \to aB, B \to bB, B \to b\}, A)}$$

Proof: $L(G) \supseteq \{a^n b^m : n \geq 1, m \geq 1\}$ in details

Lemma 1: $\forall n \geq 1 : A \overset{*}{\Rightarrow} a^n B$

$n = 1$**:** $A \rightarrow aB \in P$

$n \rightsquigarrow n+1$**:** $A \rightarrow aA \underbrace{\overset{*}{\Rightarrow}}_{IH} aa^n B = a^{n+1} B$

$$\boxed{G = (\{A, B\}, \{a, b\}, \{A \rightarrow aA, A \rightarrow aB, B \rightarrow bB, B \rightarrow b\}, A)}$$

Proof: $L(G) \supseteq \{a^n b^m : n \geq 1, m \geq 1\}$ in details

Lemma 1: $\forall n \geq 1 : A \overset{*}{\Rightarrow} a^n B$

Lemma 2: $\forall m \geq 1 : B \overset{*}{\Rightarrow} b^m$

$m = 1$**:** $B \rightarrow b \in P$

$m \rightsquigarrow m+1$**:** $B \rightarrow bB \underbrace{\overset{*}{\Rightarrow}}_{IH} bb^m = b^{m+1}$

$$\boxed{G = (\{A, B\}, \{a, b\}, \{A \rightarrow aA, A \rightarrow aB, B \rightarrow bB, B \rightarrow b\}, A)}$$

Proof: $L(G) \supseteq \{a^n b^m : n \geq 1, m \geq 1\}$ in details

Lemma 1: $\forall n \geq 1 : A \overset{*}{\Rightarrow} a^n B$

Lemma 2: $\forall m \geq 1 : B \overset{*}{\Rightarrow} b^m$

Proof $\supseteq$: $\forall n \geq 1, m \geq 1 : A \underbrace{\overset{*}{\Rightarrow}}_{\text{Lemma 1}} a^n B \underbrace{\overset{*}{\Rightarrow}}_{\text{Lemma 2}} a^n b^m$

so $a^n b^m \in L(G)$ (Def. $L(G)$)

$$\boxed{G = (\{A, B\}, \{a, b\}, \{A \to aA, A \to aB, A \to a, B \to bB, B \to b\}, A)}$$

Sketch of the proof: $L(G) \supseteq \{a^n b^m : n \geq 1, m \geq 1\}$

$$A \underbrace{\overset{n-1}{\Rightarrow}}_{A \to aA} a^{n-1}A \Rightarrow a^n B \underbrace{\overset{m-1}{\Rightarrow}}_{B \to bB} a^n b^{m-1} B \Rightarrow a^n b^m$$

$$\boxed{G = (\{A, B\}, \{a, b\}, \{A \to aA, A \to aB, B \to bB, B \to b\}, A)}$$

Proof: $L(G) \subseteq \{a^n b^m : n \geq 1, m \geq 1\}$ in details

Induction on the derivation length ℓ:

(Stronger) Induction assumptions : $\forall \alpha \in (V \cup \Sigma)^* : A \overset{\leq \ell}{\Rightarrow} \alpha \longrightarrow$
$\alpha \in \{a\}^* \cdot A \cup \{a\}^+ \cdot \{b\}^* \cdot B \cup \{a\}^+ \cdot \{b\}^+$

$\ell = 0$: $A \in \{a\}^* \cdot A$

$\ell \rightsquigarrow \ell + 1$: Consider the derivation $A \overset{*}{\Rightarrow} \alpha' \overbrace{\Rightarrow}^{C \to \beta} \alpha$

α'	$C \to \beta$	α	$\longrightarrow \alpha \in$
$a^n A$	$A \to aA$	$a^{n+1}A$	$\{a\}^+ \cdot A$
$a^n A$	$A \to aB$	$a^{n+1}B$	$\{a\}^+ \cdot \{b\}^* \cdot B$
$a^n b^m B$	$B \to bB$	$a^n b^{m+1} B$	$\{a\}^+ \cdot \{b\}^* \cdot B$
$a^n b^m B$	$B \to b$	$a^n b^{m+1}$	$\{a\}^+ \cdot \{b\}^+$

■

Sketch of the proof: $L(G) \subseteq \{a^n b^m : n \geq 1, m \geq 1\}$

If $A \stackrel{*}{\Rightarrow} \alpha$ then $\alpha \in \{a\}^* \cdot A \cup \{a\}^+ \cdot \{b\}^* \cdot B \cup \{a\}^+ \cdot \{b\}^+$.

The derivations preserve this

invariant. ■

$$\boxed{G = (\{A, B\}, \{a, b\}, \{A \to aA, A \to aB, B \to bB, B \to b\}, A)}$$

Example: Type 2 (Balanced parentheses)

$G = (\{E,T,F\},\{a,+,*,(,)\},P,E)$, where

$$
\begin{aligned}
P = \{E &\to T,\\
E &\to E+T,\\
T &\to F,\\
T &\to T*F,\\
F &\to a,\\
F &\to (E)\}
\end{aligned}
$$

Example: Type 2

$$G = (\{S\}, \{a, b\}, \{S \to ab, S \to aSb\}, S).$$
$$L(G) = \{a^n b^n : n \geq 1\}.$$

Proof (sketch) $L(G) \supseteq \{a^n b^n : n \geq 1\}$:

$$S \overset{n-1}{\Rightarrow} a^{n-1} S b^{n-1} \Rightarrow a^n b^n.$$ ■

Proof (sketch) $L(G) \subseteq \{a^n b^n : n \geq 1\}$:

$$S \overset{*}{\Rightarrow} \alpha \longrightarrow \alpha \in \left\{a^k S b^k : k \geq 0\right\} \cup \{a^n b^n : n \geq 1\}$$

(Invariant) ■

Example: Type 1

$$G = (\{S, B, C\}, \{a, b, c\}, P, S)$$

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

Statement: $L(G) = \{a^n b^n c^n : n \in \mathbb{N}\}$

Example:

$$\underline{S} \Rightarrow a\underline{S}BC \Rightarrow aa\underline{S}BCBC \Rightarrow aaaB\underline{CB}CBC$$

$$\Rightarrow aaaBB\underline{CB}C \Rightarrow aaaBB\underline{CB}CC \Rightarrow aa\underline{aB}BBCCC$$

$$\Rightarrow aaa\underline{bB}BCCC \Rightarrow aaab\underline{bB}CCC \Rightarrow aaabb\underline{bC}CC$$

$$\Rightarrow aaabbb\underline{cC}C \Rightarrow aaabbbc\underline{cC} \Rightarrow aaabbbccc$$

$$\boxed{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc}$$

Proof (sketch) $a^n b^n c^n \subseteq L(G)$

$$S \overset{n-1}{\Rightarrow} a^{n-1} S (BC)^{n-1} \qquad (S \rightarrow aSBC)$$

$$\Rightarrow a^n (BC)^n \qquad (S \rightarrow aBC)$$

$$\underbrace{\overset{*}{\Rightarrow}}_{\text{Lemma S}} a^n B^n C^n \qquad (CB \rightarrow BC)$$

$$\Rightarrow a^n b B^{n-1} C^n \qquad (aB \rightarrow ab)$$

$$\overset{n-1}{\Rightarrow} a^n b^n C^n \qquad (bB \rightarrow bb)$$

$$\Rightarrow a^n b^n c C^{n-1} \qquad (bC \rightarrow bc)$$

$$\overset{n-1}{\Rightarrow} a^n b^n c^n \qquad (cC \rightarrow cc)$$

Exercise: Prove all parts.

Lexicographical order

Consider $\alpha, \beta \in \Sigma^*$

$\forall \alpha \in \Sigma^* : \varepsilon \leq \alpha$

$a\alpha \leq b\beta$ iff $a < b$ or $a = b$ and $\alpha \leq \beta$ $\qquad (a, b \in \Sigma;\ \alpha, \beta \in \Sigma^*)$

Observation: $\leq$ defines a total (linear) order

Proof: Exercise?

Example: $\varepsilon < \mathtt{a} < \mathtt{aa} < \mathtt{ab} < \mathtt{b} < \mathtt{ba} < \mathtt{bb}$

- An analogue for tuples
- We could do some proofs by induction on a total order on the finite sequences of words.

To do: (Example)

Lemma S: $(BC)^n \overset{*}{\Rightarrow} B^n C^n$ by means of $CB \rightarrow BC$

Proof by induction on the lexicographical order of

$\left\{ w \in \{B, C\}^{2n} : w \text{ contains exactly the same number of } B \text{ and } C \right\}$

IA: α minimal $\longrightarrow$ $\alpha = B^n C^n$

IS: α not minimal $\longrightarrow$

$$\alpha = \gamma CB\beta$$

$$\Rightarrow \gamma BC\beta \qquad \text{is less than!}$$

$$\overset{*}{\Rightarrow} B^n C^n \qquad \text{IH}$$

■

Exercise: Show that there is a not minimal word α of the form $\gamma CB\beta$.

Next exercise: how long is the derivation as a function of n?

Proof (sketch) $L(G) \subseteq a^n b^n c^n$

Invariant: $\#a = \#(b,B) = \#(c,C)$

In particular: $\forall w \in L(G) : \#a = \#b = \#c$.

It remains to see that $L(G) \subseteq a^* b^* c^*$.

All a-s come in front of all b-s and c-s. $(S \rightarrow aSBC, S \rightarrow aBC)$

The first b follows the last a. $(aB \rightarrow ab)$

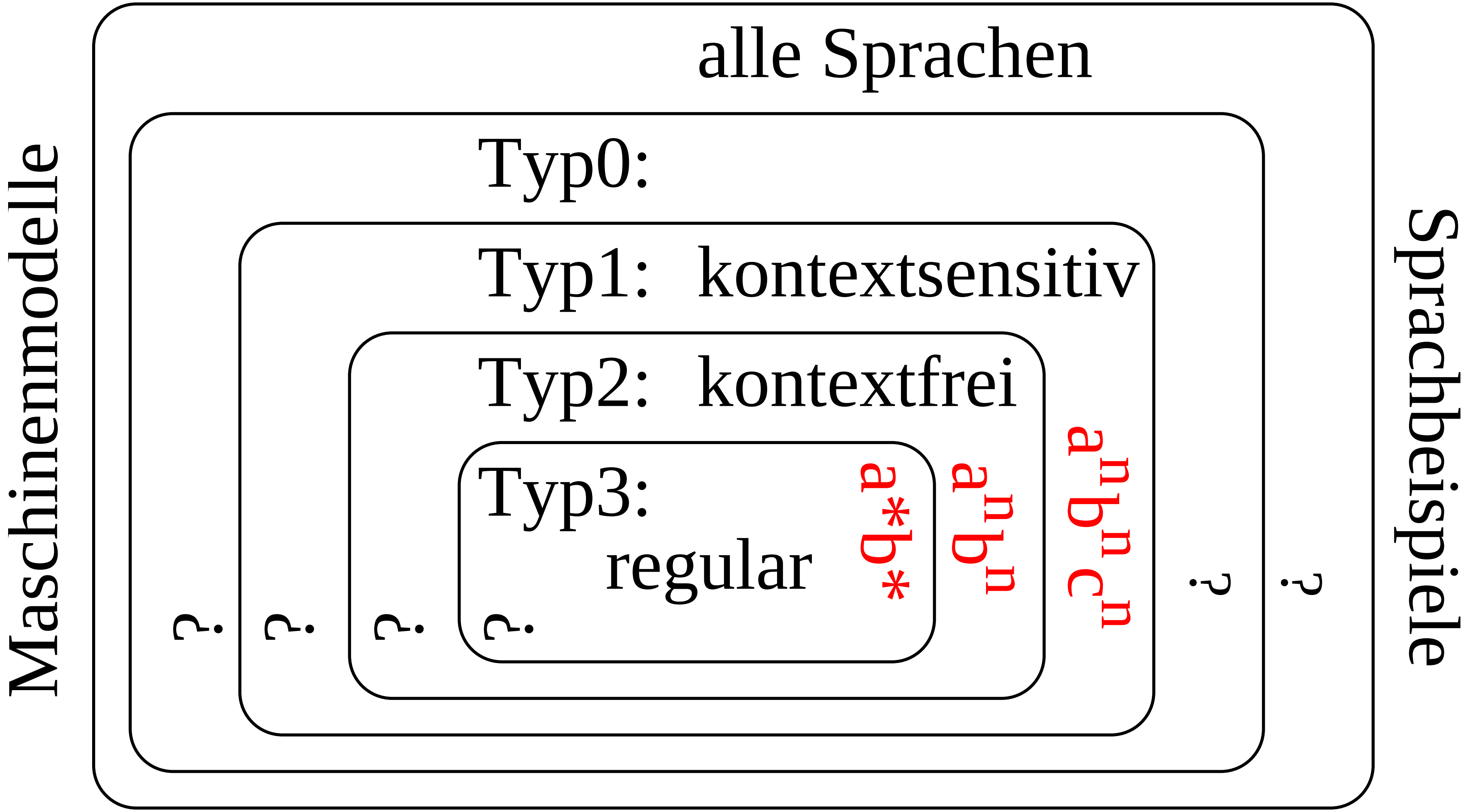
The next coming b follows the existing b-s. $(bB \rightarrow bb)$

The first coming c follows the last b. $(bC \rightarrow bc)$

The next c follows the existing c-s . $(cC \rightarrow cc)$

$$\boxed{S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc}$$

Chomsky Hierarchy



To do

- Assign to the each type of grammars $\leftrightarrow$ a machine model
- Show that there are some examples of languages not in the simpler types of grammars
- An example of a language of type 0
- Algorithms and proof technics for the standard problems

1.1.3 Word problem

The basic standard problem for the formal languages:

Given: $G = (V, \Sigma, P, S)$, $w \in \Sigma^*$

Question: $w \in L(G)$?

($\Leftrightarrow$ $S \overset{*}{\Rightarrow} w$?)

The word problem for type 1 languages:

Given: $G = (V, \Sigma, P, S)$, $w \in \Sigma^*$

Question: $w \in L(G)$?

Consider a finite graph $H = (U, E)$, where

$U = \{x \in (\Sigma \cup V)^* : |x| \leq |w|\}$ and

$E = \{(x, y) : x \Rightarrow_G y\}$.

$w \in L(G)$ iff w is in H and it is reachable from S.

Corollary:

The word problem for type 1 is in finite time algorithmic decidable.

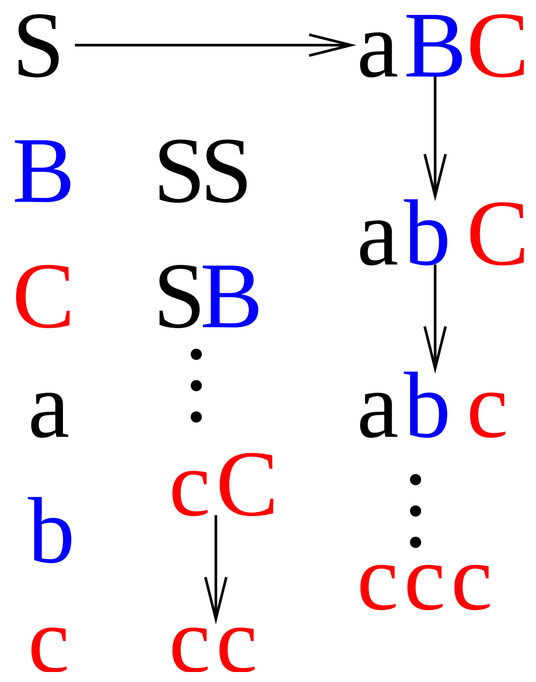
Question: Why this does not work for type 0?

Example

$abc \in L(G)$

$G = (\{S, B, C\}, \{a, b, c\}, P, S)$?

$$P = \{S \rightarrow aSBC,$$
$$S \rightarrow aBC,$$
$$CB \rightarrow BC,$$
$$aB \rightarrow ab,$$
$$bB \rightarrow bb,$$
$$bC \rightarrow bc,$$
$$cC \rightarrow cc\}$$



Run-time estimation

Given: $G = (V, \Sigma, P, S)$, $w \in \Sigma^*$

Question: $w \in L(G)$?

Consider the the finite graph $H = (U, E)$, where

$U = \{x \in (\Sigma \cup V)^* : |x| \leq |w|\}$ and

$E = \{(x, y) : x \Rightarrow_G y\}$.

The reachability is going in time $\mathcal{O}(|U| + |V|)$.

Dominating is the time for building the graph.

$(|V| + |\Sigma|)^{|w|}$ knots (!)

$\times |w|$ possible replacements

$\times |P|$ possible derivations

$\times \mathcal{O}(|w|)$ time for checking and replacements

1.1.4 Syntax (parse) tree (for type 2)

An ordered tree which describes the derivation (type 2) $S \overset{*}{\Rightarrow} w$ independently of the order of the replacements.

Construction by the derivation

$S = x_1 \Rightarrow x_2 \Rightarrow \cdots \Rightarrow x_n = x \in \Sigma^*$:

The root S.

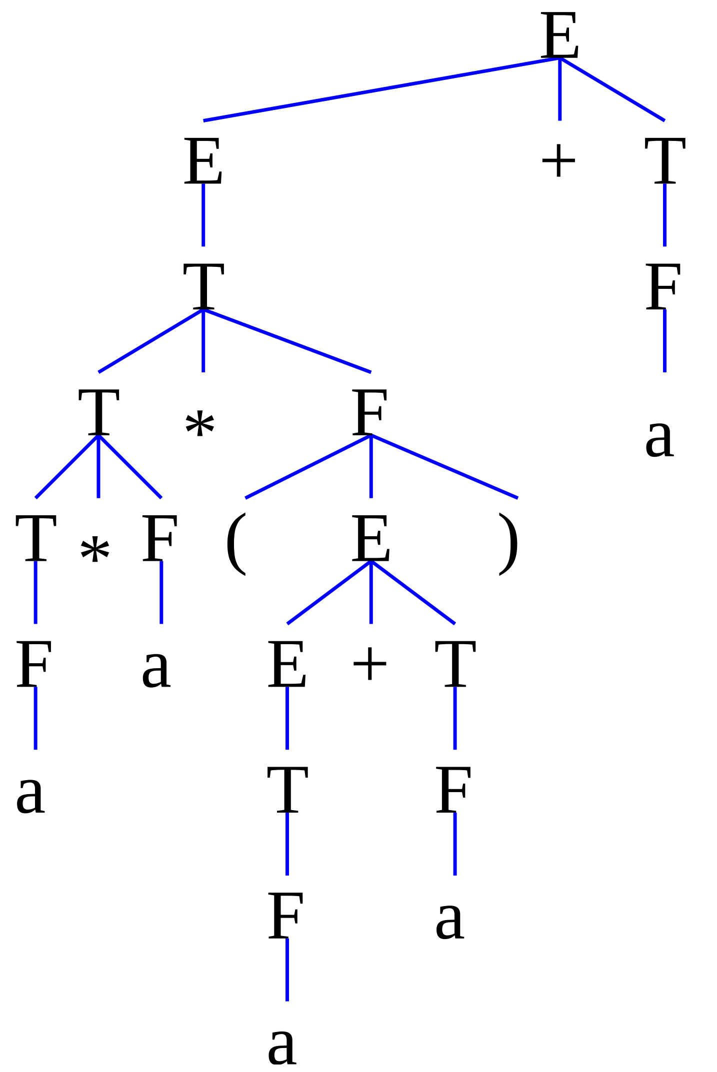
If on step i we do the replacement $A \longrightarrow z = z_1, \ldots, z_k$.

$\longrightarrow$ the successor knots for A are $z_1, \ldots, z_k$.

Observation: The leaves are the letters of x.

Example

$$
\begin{array}{l|l}
E \Rightarrow & E \rightarrow E+T \\
\Rightarrow E+T & E \rightarrow T \\
\Rightarrow T+T & T \rightarrow T*F \\
\Rightarrow T*F+T & T \rightarrow T*F \\
\Rightarrow T*F*F+T & T \rightarrow F \\
\Rightarrow F*F*F+T & F \rightarrow a \\
\Rightarrow a*F*F+T & F \rightarrow a \\
\Rightarrow a*a*F+T & F \rightarrow (E) \\
\Rightarrow a*a*(E)+T & E \rightarrow E+T \\
\Rightarrow a*a*(E+T)+T & E \rightarrow T \\
\Rightarrow a*a*(T+T)+T & T \rightarrow F \\
\Rightarrow a*a*(F+T)+T & F \rightarrow a \\
\Rightarrow a*a*(a+T)+T & T \rightarrow F \\
\Rightarrow a*a*(a+F)+T & F \rightarrow a \\
\Rightarrow a*a*(a+a)+T & T \rightarrow F \\
\Rightarrow a*a*(a+a)+F & F \rightarrow a \\
\Rightarrow a*a*(a+a)+a &
\end{array}
$$



The leftmost derivation

On every step of the derivation:

replace the leftmost Variable

Example: previous page

1-1 relation leftmost derivation $\leftrightarrow$ Syntax tree

Observation (Proposition) (for type 2)

$$
\begin{aligned}
x \in L(G) &\Leftrightarrow \exists \text{ derivation for } x \\
&\Leftrightarrow \exists \text{ Syntax tree with } x \text{ on the leaves} \\
&\Leftrightarrow \exists \text{ leftmost derivation for } x
\end{aligned}
$$

Task: Define the rightmost derivation with the respective properties.

Example for ambiguous syntax tree

$G = (\{E\}, \{a, +, *, (,)\}, P, E)$, where

$$\begin{aligned} P = \{ & E \to E + E, \\ & E \to E * E, \\ & E \to a, \\ & E \to (E)\} \end{aligned}$$