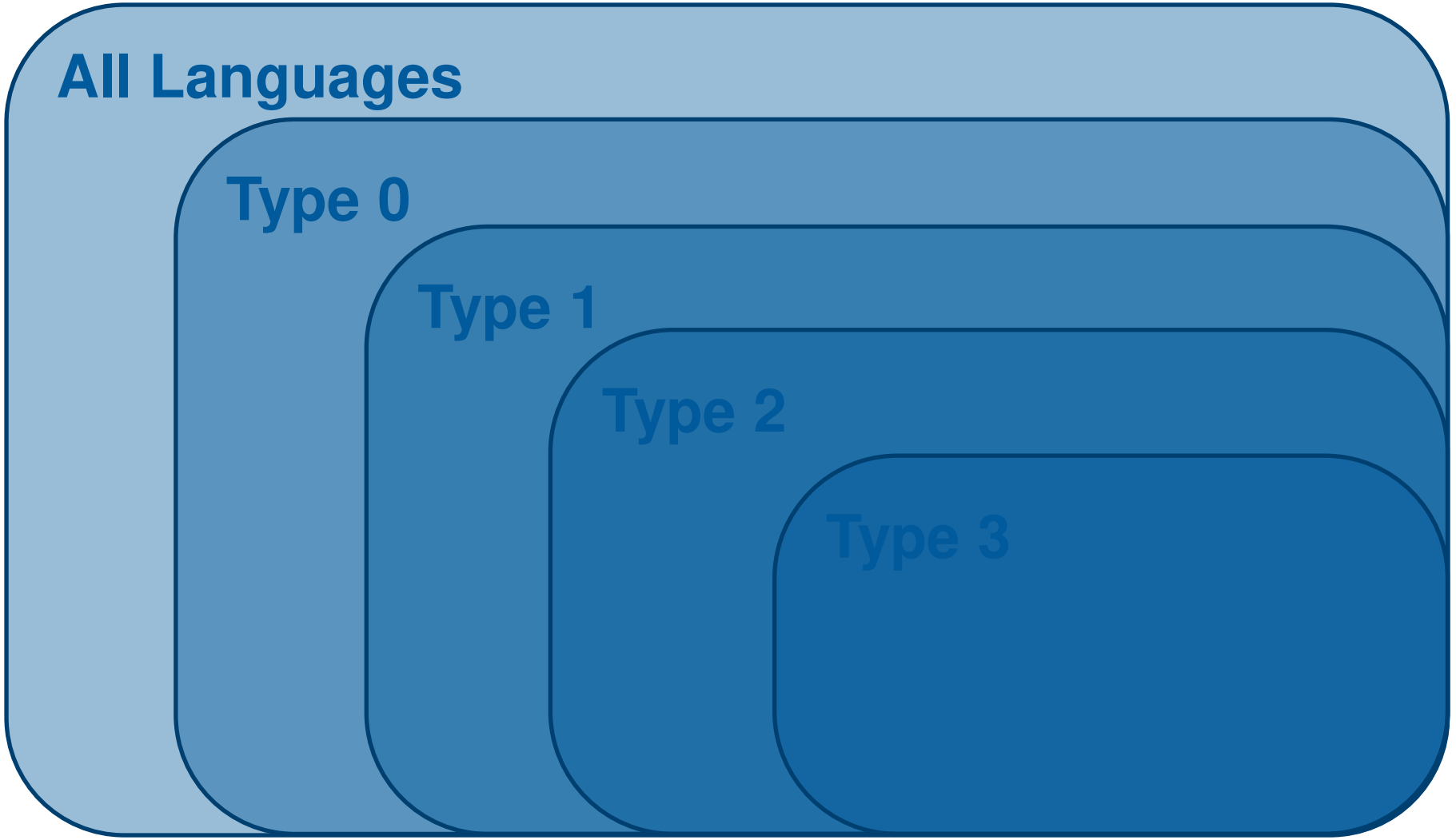
Chomsky hierarchy

Classification of Formal Languages

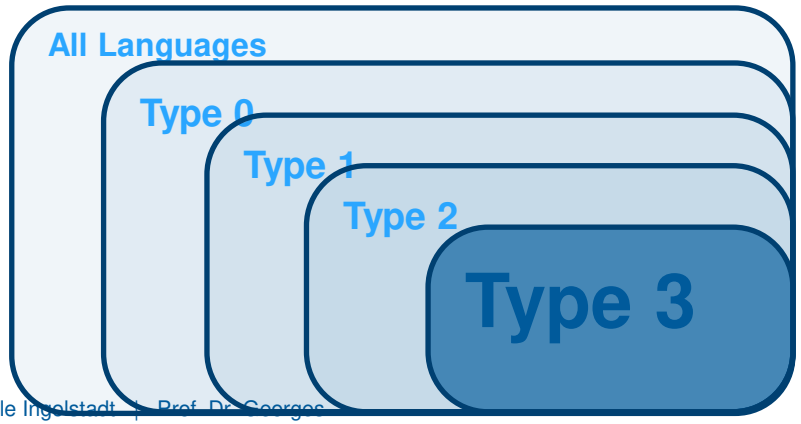
Chomsky hierarchy



https://de.wikipedia.org/wiki/Chomsky-Hierarchie
https://studyflix.de/informatik/chomsky-hierarchie-1442?topic_id=46

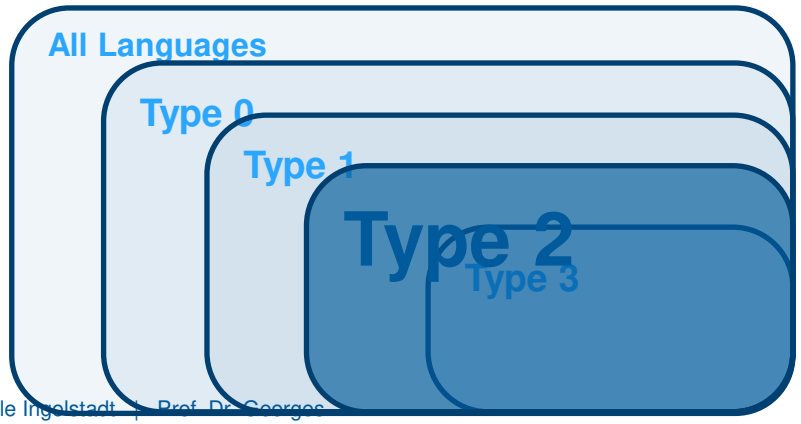
Type3 Languages

Regular Languages



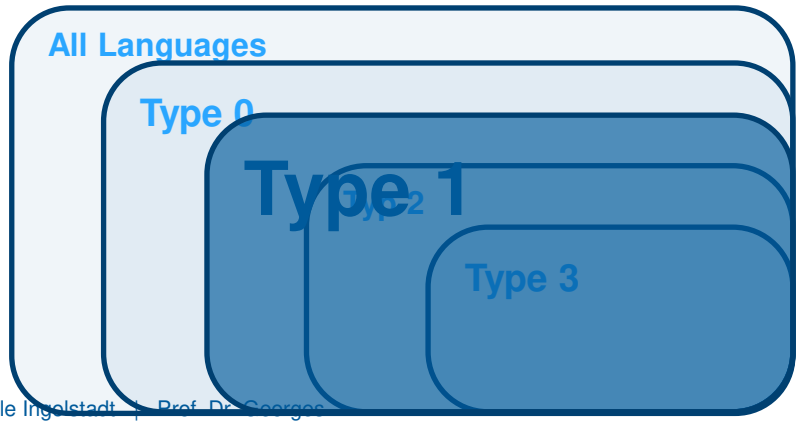
Type2 Languages

Context free languages



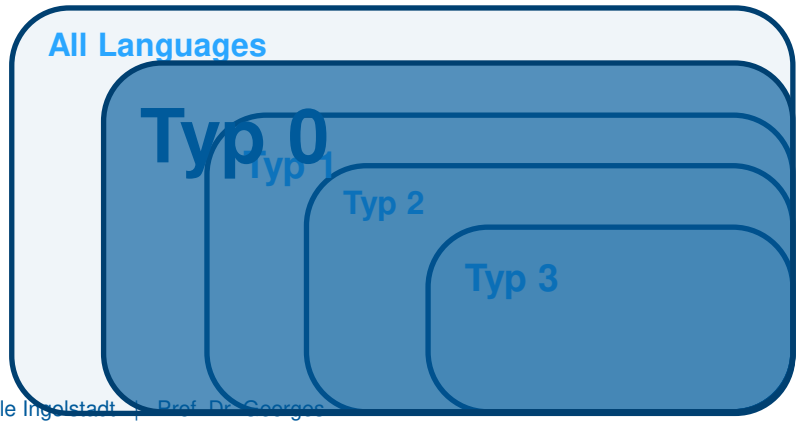
Type1 Languages

Context sensitive languages



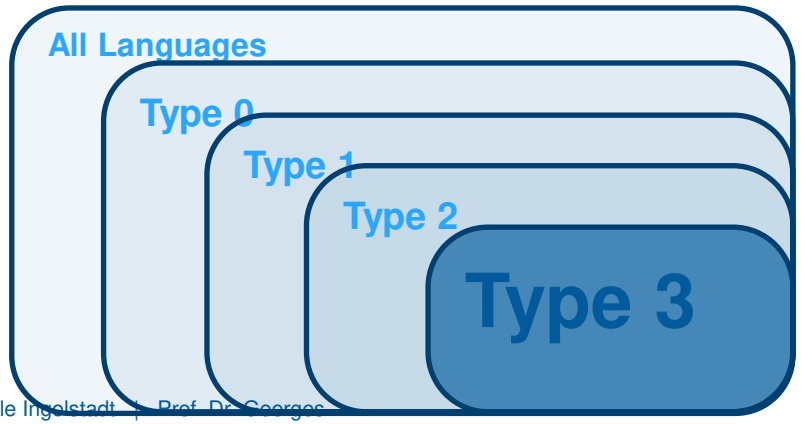
Type0 Languages

„ unlimited grammars“



Type3 Languages

Regular Languages



Definition 3.3.4. A linear grammar $\mathcal{G} = (N, \Sigma, P, S)$ is said to be *right linear* if the nonterminal symbol in the righthand side of each production rule, if any, occurs at the right end. That is, righthand side of each production rule is of the form – a terminal string followed by at most one nonterminal symbol – as shown below.

$$A \to x \quad \text{or} \quad A \to xB$$

for some $x \in \Sigma^*$ and $B \in N$.

Note:
a right-linear grammar is also called **regular**

And for what else?

„ Replication of a subset of natural languages.“

Example:

NP → the **N**

N → car **|** ship

correct nominal phrases can be formed :

„the Auto“, „the ship“

Comments

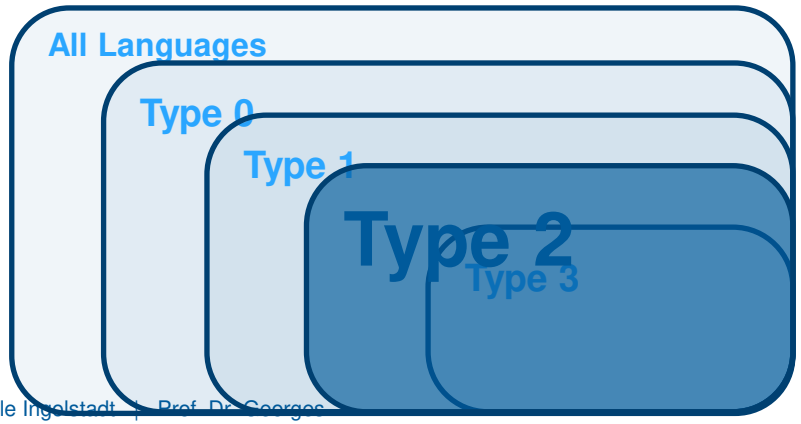
FST based speech recognition can only recognize type 3 languages.

German texts are usually type 3

Natural language is often not type 3

Type2 Languages

Context free languages



Type 2: Context-free

- **On the left is exactly one terminal :**

  A -> BC

  B -> abCabA

  C -> c

- **In the industry: Backus-Naur Form**

Type 2: Context-free

- **On the left is exactly one terminal :**

  A -> BC

  B -> abCabA

  C -> c

- **In the industry: Backus-Naur Form**

Question: why is such a grammar called *context-free*?

(Hint: see Definition 3.1.2. [1])

[1] „Formal Languages and Automata Theory”, D. Goswami and K. V. Krishna, Nov 5, 2010

Type 2: Example C-Parser

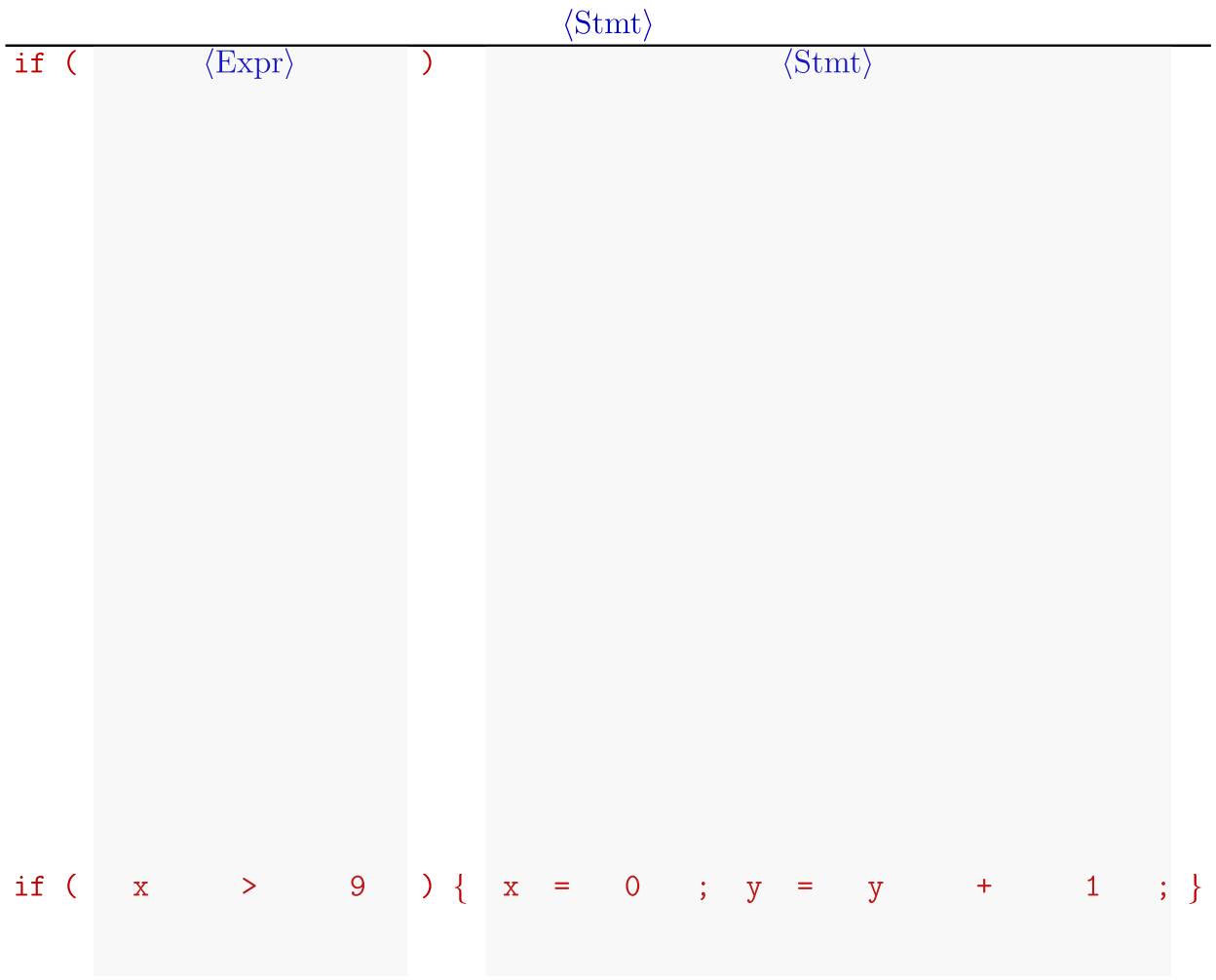
⟨Stmt⟩ → ⟨Id⟩ = ⟨Expr⟩ ;
⟨Stmt⟩ → { ⟨StmtList⟩ }
⟨Stmt⟩ → if (⟨Expr⟩) ⟨Stmt⟩
⟨StmtList⟩ → ⟨Stmt⟩
⟨StmtList⟩ → ⟨StmtList⟩ ⟨Stmt⟩
⟨Expr⟩ → ⟨Id⟩
⟨Expr⟩ → ⟨Num⟩
⟨Expr⟩ → ⟨Expr⟩ ⟨Optr⟩ ⟨Expr⟩
⟨Id⟩ → x
⟨Id⟩ → y
⟨Num⟩ → 0
⟨Num⟩ → 1
⟨Num⟩ → 9
⟨Optr⟩ → >
⟨Optr⟩ → +

```
if ( x > 9 ) { x = 0 ; y = y + 1 ; }
```


https://en.wikipedia.org/wiki/Context-free_grammar

Type 2: Example C-Parser

⟨Stmt⟩ → ⟨Id⟩ = ⟨Expr⟩ ;
⟨Stmt⟩ → { ⟨StmtList⟩ }
⟨Stmt⟩ → if (⟨Expr⟩) ⟨Stmt⟩
⟨StmtList⟩ → ⟨Stmt⟩
⟨StmtList⟩ → ⟨StmtList⟩ ⟨Stmt⟩
⟨Expr⟩ → ⟨Id⟩
⟨Expr⟩ → ⟨Num⟩
⟨Expr⟩ → ⟨Expr⟩ ⟨Optr⟩ ⟨Expr⟩
⟨Id⟩ → x
⟨Id⟩ → y
⟨Num⟩ → 0
⟨Num⟩ → 1
⟨Num⟩ → 9
⟨Optr⟩ → >
⟨Optr⟩ → +



https://en.wikipedia.org/wiki/Context-free_grammar

Type 2: Example C-Parser

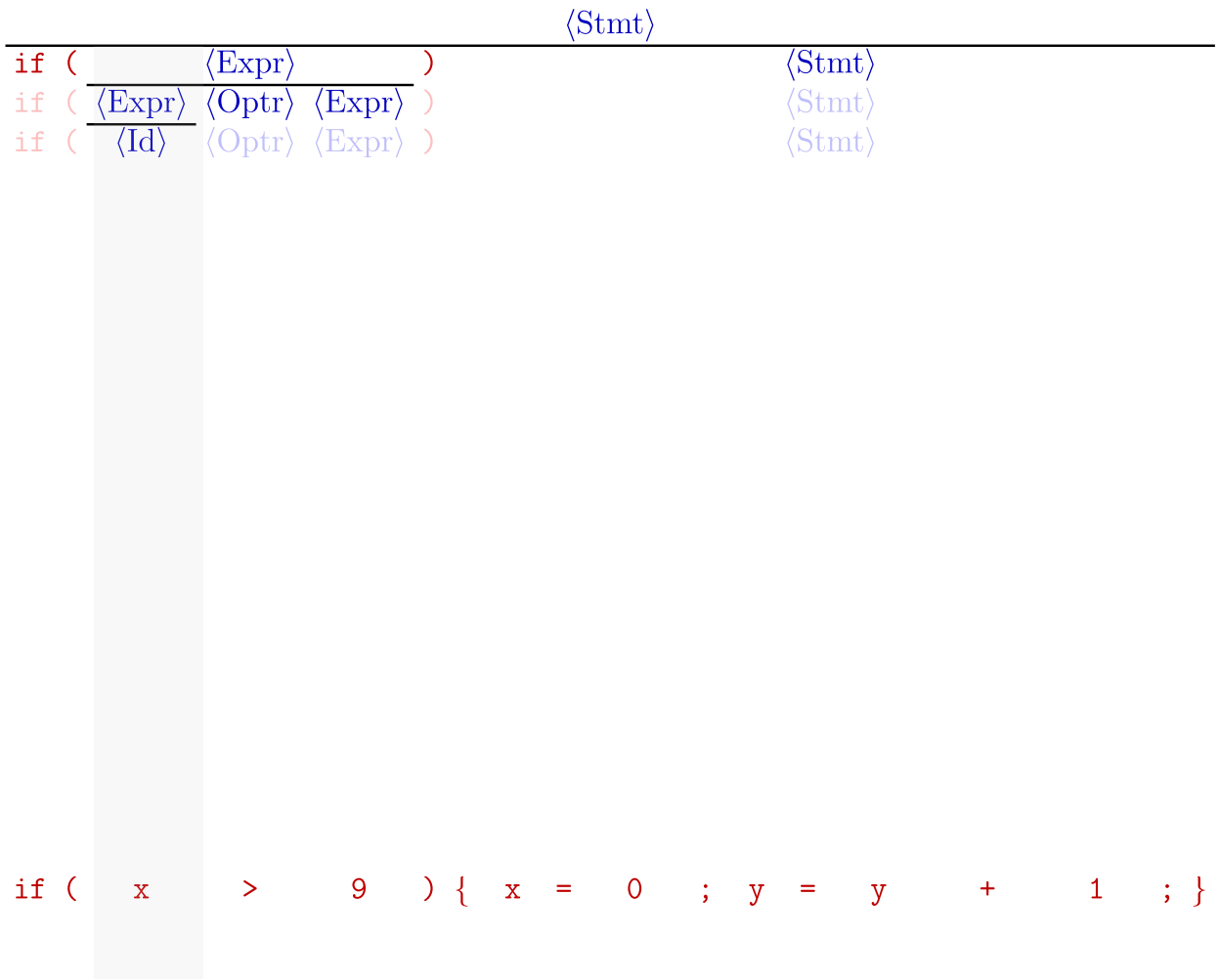
$$\frac{\text{if (} \frac{\langle \text{Expr} \rangle \; \langle \text{Optr} \rangle \; \langle \text{Expr} \rangle}{\langle \text{Expr} \rangle} \text{) } \langle \text{Stmt} \rangle}{\langle \text{Stmt} \rangle}$$

$$\begin{aligned}
\langle \text{Stmt} \rangle &\rightarrow \langle \text{Id} \rangle \texttt{ = } \langle \text{Expr} \rangle \texttt{ ;} \\
\langle \text{Stmt} \rangle &\rightarrow \texttt{\{ } \langle \text{StmtList} \rangle \texttt{ \}} \\
\langle \text{Stmt} \rangle &\rightarrow \texttt{if (} \langle \text{Expr} \rangle \texttt{)} \; \langle \text{Stmt} \rangle \\
\langle \text{StmtList} \rangle &\rightarrow \langle \text{Stmt} \rangle \\
\langle \text{StmtList} \rangle &\rightarrow \langle \text{StmtList} \rangle \; \langle \text{Stmt} \rangle \\
\langle \text{Expr} \rangle &\rightarrow \langle \text{Id} \rangle \\
\langle \text{Expr} \rangle &\rightarrow \langle \text{Num} \rangle \\
\langle \text{Expr} \rangle &\rightarrow \langle \text{Expr} \rangle \; \langle \text{Optr} \rangle \; \langle \text{Expr} \rangle \\
\langle \text{Id} \rangle &\rightarrow \texttt{x} \\
\langle \text{Id} \rangle &\rightarrow \texttt{y} \\
\langle \text{Num} \rangle &\rightarrow \texttt{0} \\
\langle \text{Num} \rangle &\rightarrow \texttt{1} \\
\langle \text{Num} \rangle &\rightarrow \texttt{9} \\
\langle \text{Optr} \rangle &\rightarrow \texttt{>} \\
\langle \text{Optr} \rangle &\rightarrow \texttt{+}
\end{aligned}$$

```
if ( x > 9 ) { x = 0 ; y = y + 1 ; }
```

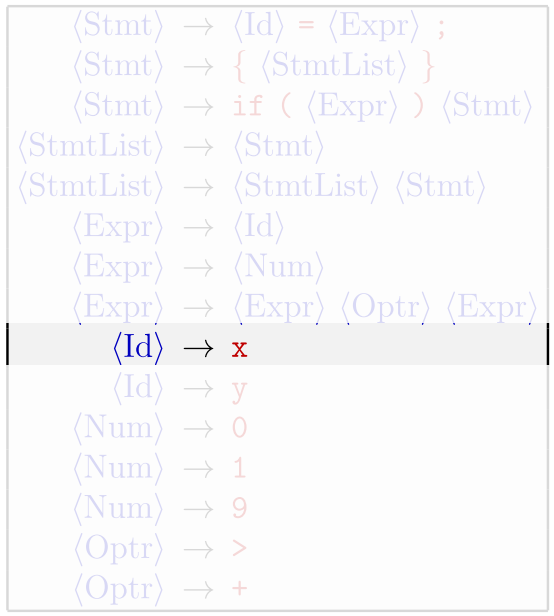
https://en.wikipedia.org/wiki/Context-free_grammar

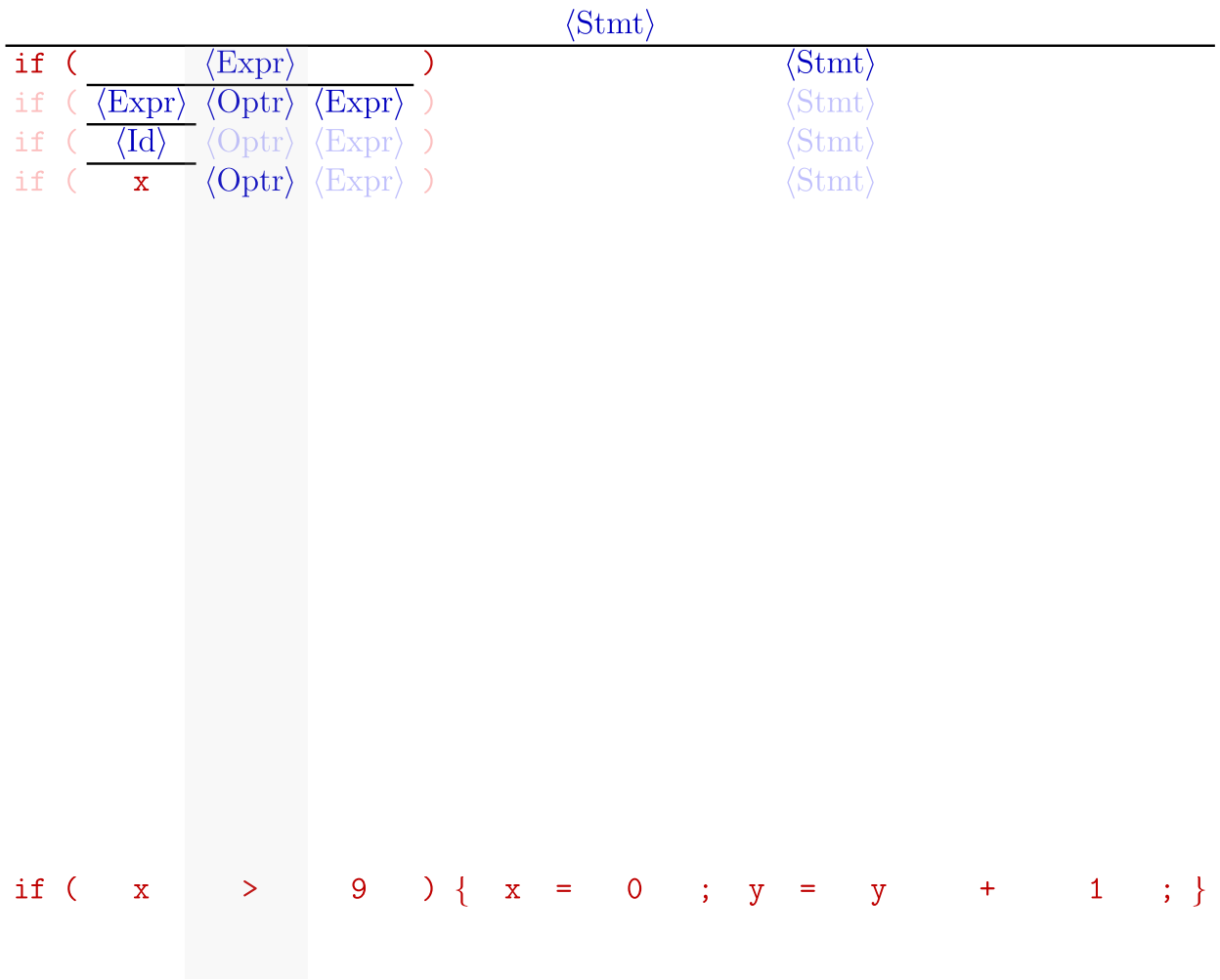
Type 2: Example C-Parser



https://en.wikipedia.org/wiki/Context-free_grammar

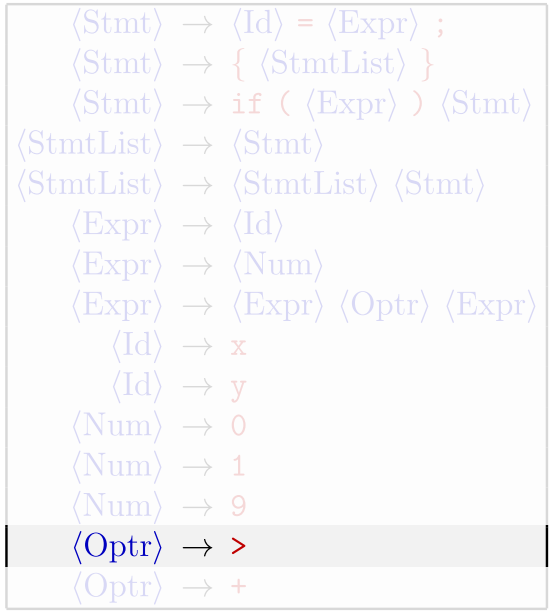
Type 2: Example C-Parser

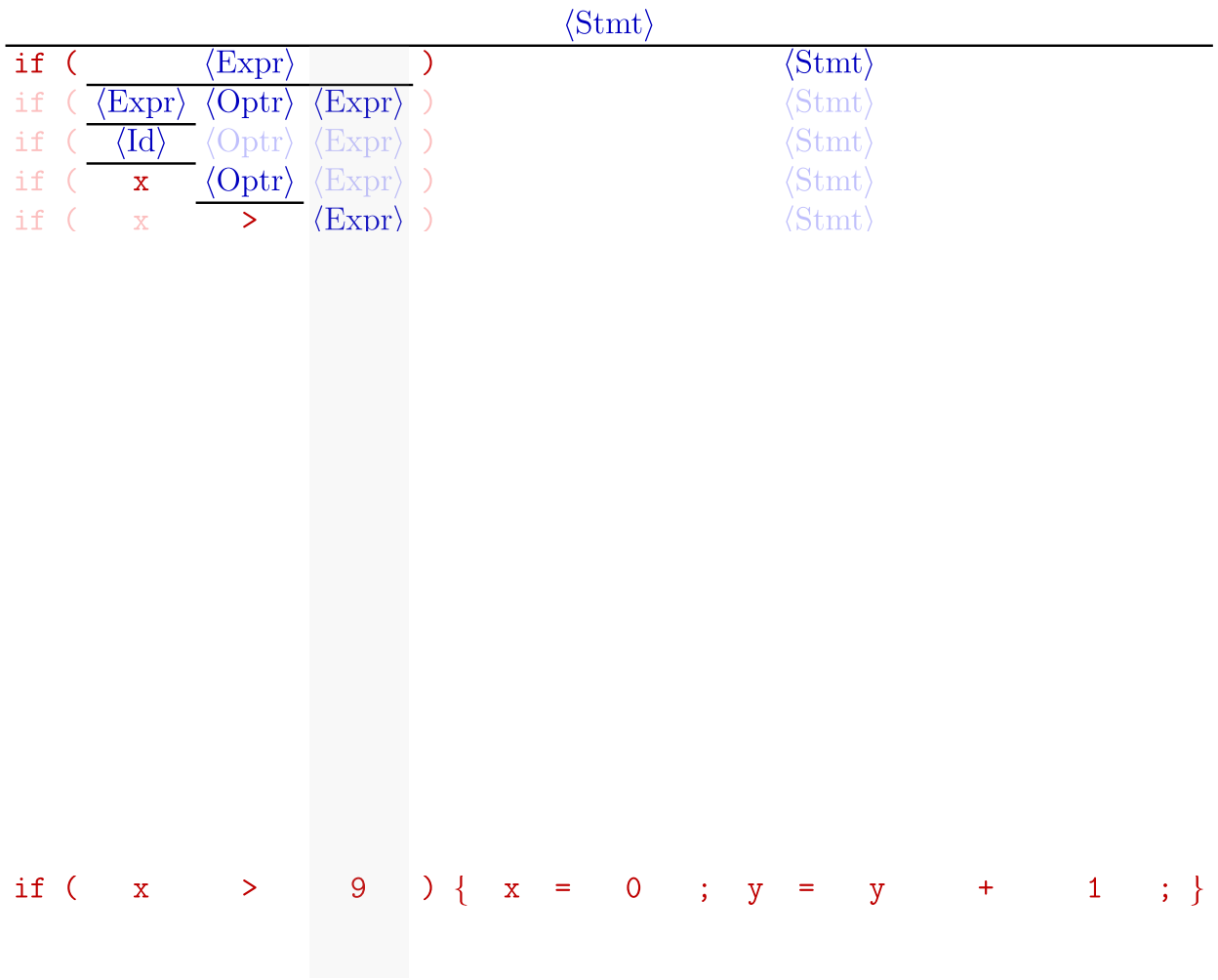




https://en.wikipedia.org/wiki/Context-free_grammar

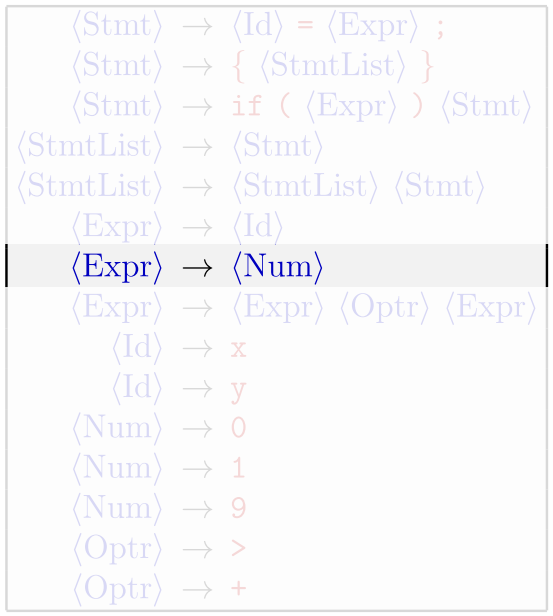
Type 2: Example C-Parser

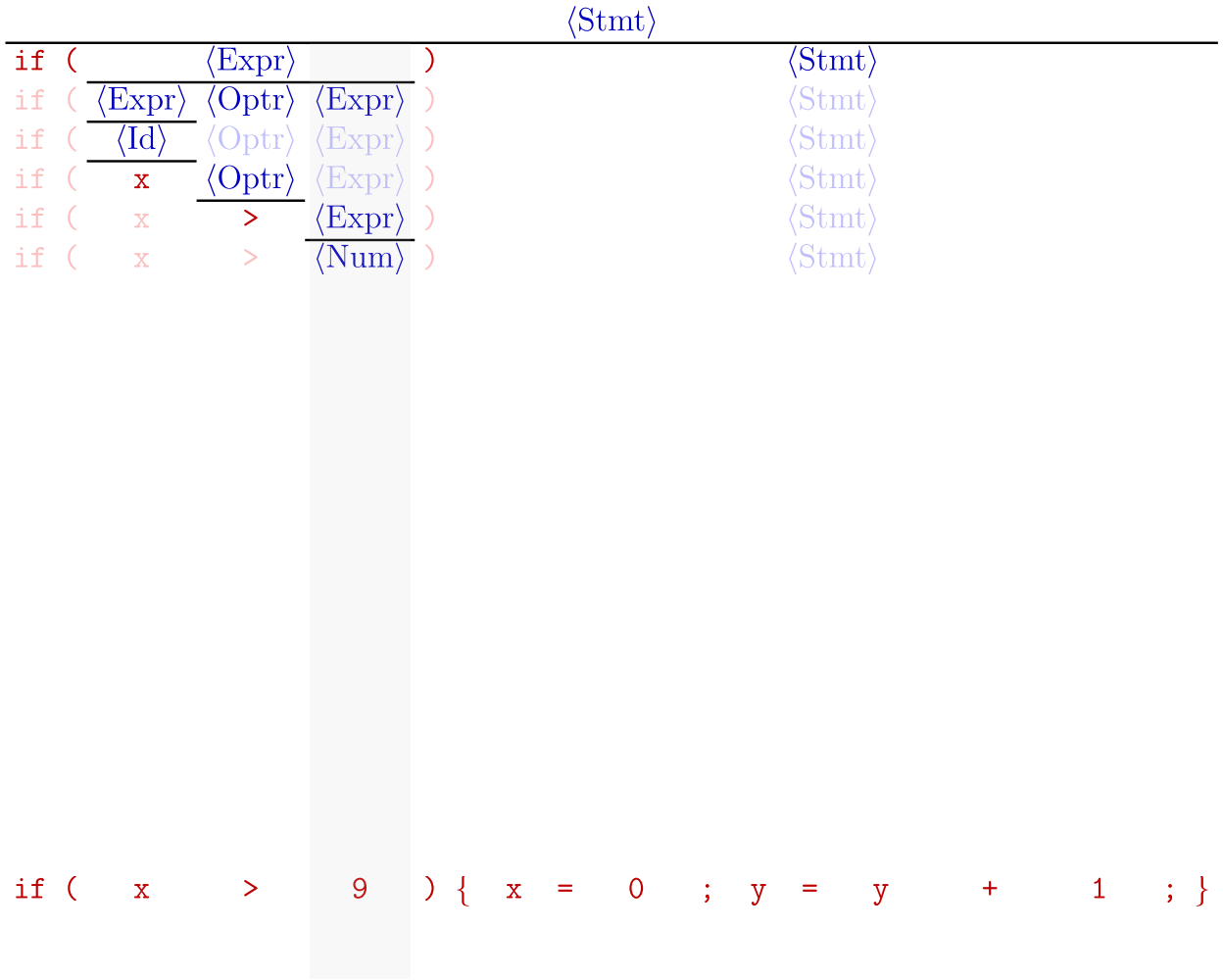




https://en.wikipedia.org/wiki/Context-free_grammar

Type 2: Example C-Parser

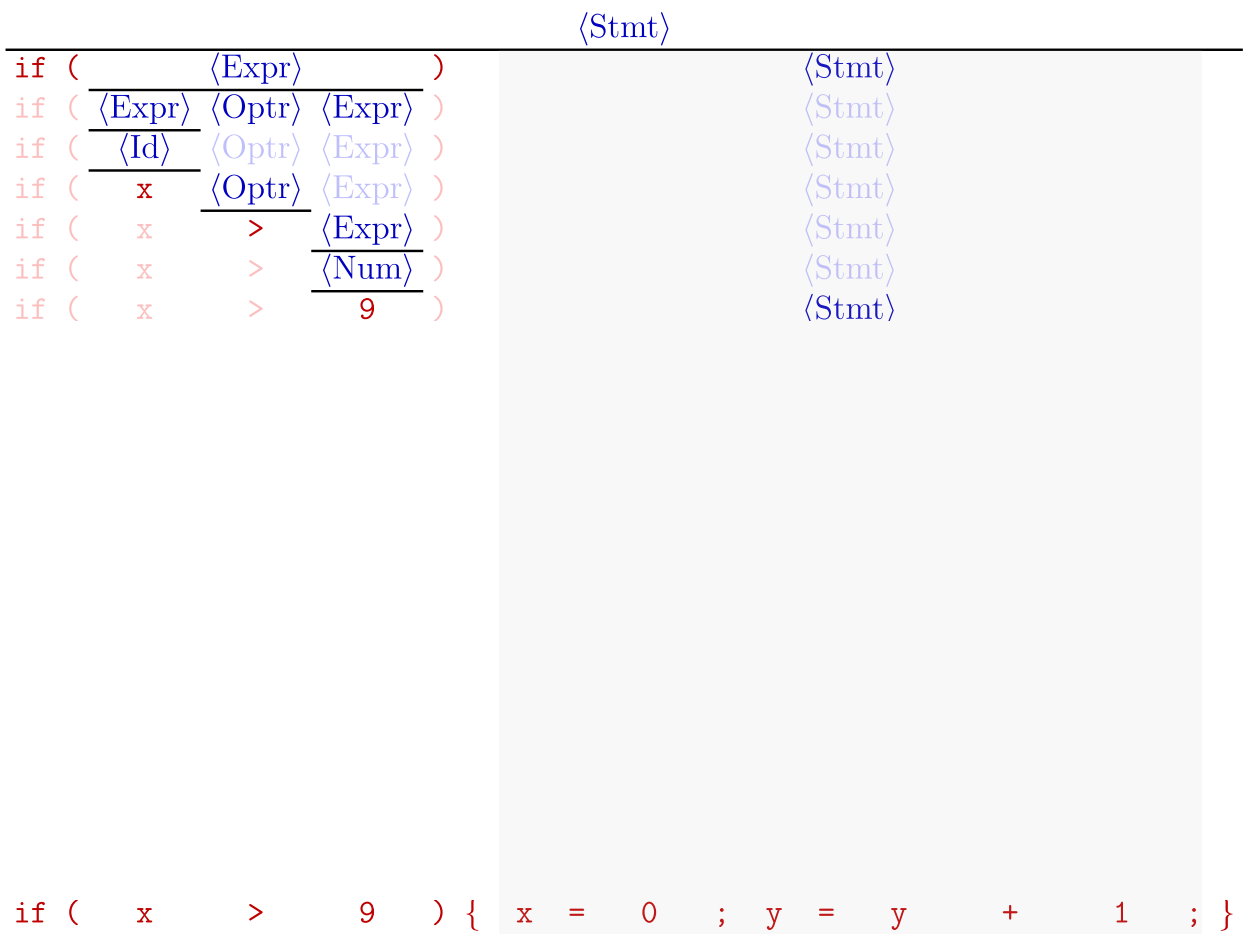




https://en.wikipedia.org/wiki/Context-free_grammar

Type 2: Example C-Parser

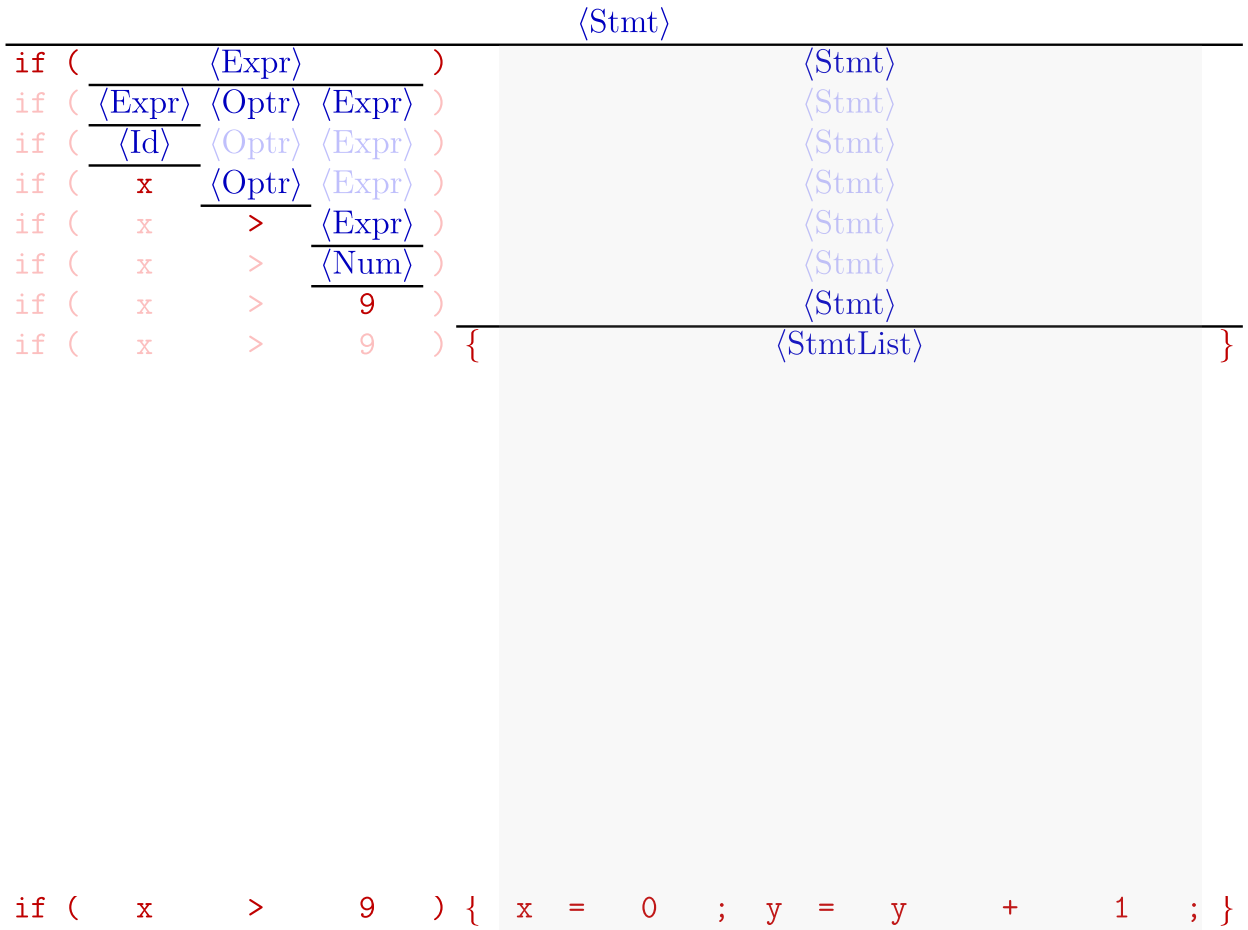
⟨Stmt⟩ → ⟨Id⟩ = ⟨Expr⟩ ;
⟨Stmt⟩ → { ⟨StmtList⟩ }
⟨Stmt⟩ → if (⟨Expr⟩) ⟨Stmt⟩
⟨StmtList⟩ → ⟨Stmt⟩
⟨StmtList⟩ → ⟨StmtList⟩ ⟨Stmt⟩
⟨Expr⟩ → ⟨Id⟩
⟨Expr⟩ → ⟨Num⟩
⟨Expr⟩ → ⟨Expr⟩ ⟨Optr⟩ ⟨Expr⟩
⟨Id⟩ → x
⟨Id⟩ → y
⟨Num⟩ → 0
⟨Num⟩ → 1
⟨Num⟩ → 9
⟨Optr⟩ → >
⟨Optr⟩ → +



https://en.wikipedia.org/wiki/Context-free_grammar

Type 2: Example C-Parser

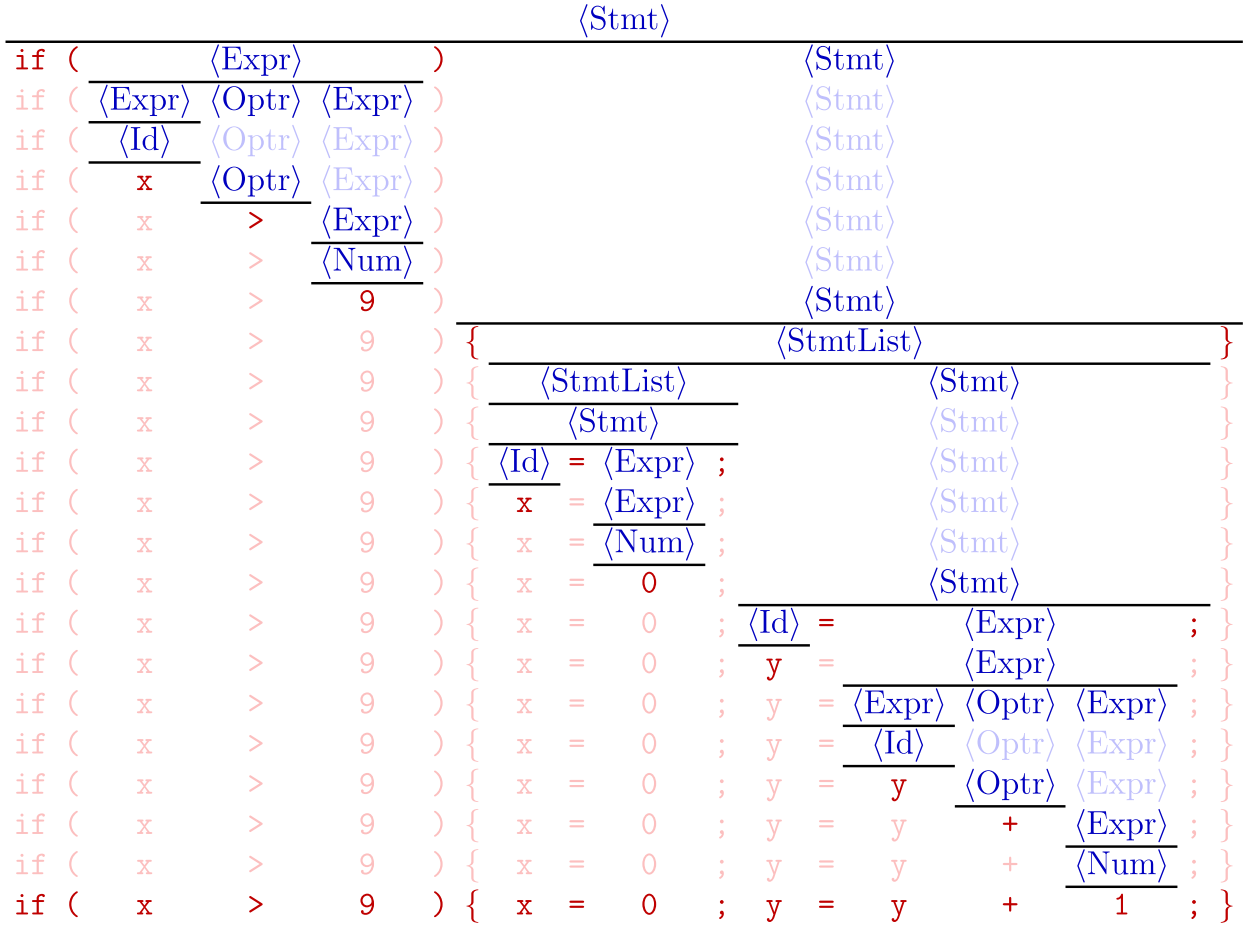
⟨Stmt⟩ → ⟨Id⟩ = ⟨Expr⟩ ;
⟨Stmt⟩ → { ⟨StmtList⟩ }
⟨Stmt⟩ → if (⟨Expr⟩) ⟨Stmt⟩
⟨StmtList⟩ → ⟨Stmt⟩
⟨StmtList⟩ → ⟨StmtList⟩ ⟨Stmt⟩
⟨Expr⟩ → ⟨Id⟩
⟨Expr⟩ → ⟨Num⟩
⟨Expr⟩ → ⟨Expr⟩ ⟨Optr⟩ ⟨Expr⟩
⟨Id⟩ → x
⟨Id⟩ → y
⟨Num⟩ → 0
⟨Num⟩ → 1
⟨Num⟩ → 9
⟨Optr⟩ → >
⟨Optr⟩ → +



https://en.wikipedia.org/wiki/Context-free_grammar

Type 2: Example C-Parser

⟨Stmt⟩ → ⟨Id⟩ = ⟨Expr⟩ ;
⟨Stmt⟩ → { ⟨StmtList⟩ }
⟨Stmt⟩ → if (⟨Expr⟩) ⟨Stmt⟩
⟨StmtList⟩ → ⟨Stmt⟩
⟨StmtList⟩ → ⟨StmtList⟩ ⟨Stmt⟩
⟨Expr⟩ → ⟨Id⟩
⟨Expr⟩ → ⟨Num⟩
⟨Expr⟩ → ⟨Expr⟩ ⟨Optr⟩ ⟨Expr⟩
⟨Id⟩ → x
⟨Id⟩ → y
⟨Num⟩ → 0
⟨Num⟩ → 1
⟨Num⟩ → 9
⟨Optr⟩ → >
⟨Optr⟩ → +



https://en.wikipedia.org/wiki/Context-free_grammar

Context-free grammars used for structured and nested concepts, i.e.

"Class declaration" or "If statement"

$$L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$$

$$L_2 = \{vv^{\mathrm{R}} \mid v \quad \text{consists of any number of} \quad a\text{s} \text{ and } b\text{s}\}$$

- Programming languages
- arithmetic expressions
- general (correct) bracket structures

Context-free grammars used for structured and nested concepts, i.e.

“Class declaration” or “If statement”

Reverse order of v

$$L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$$

$$L_2 = \{vv^{\mathrm{R}} \mid v \quad \text{consists of any number of} \quad a\text{s} \text{ and } b\text{s}\}$$

- Programming languages
- arithmetic expressions
- general (correct) bracket structures

Can only be formalized context-free:

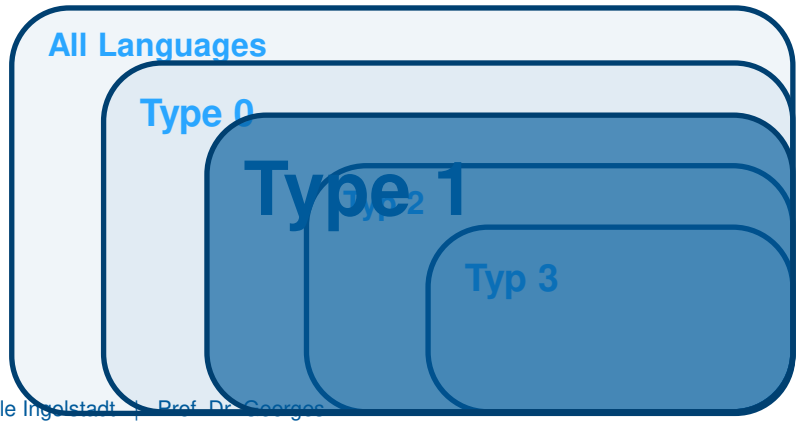
$$S \rightarrow \varepsilon|a|b|aSa|bSb$$

$$G(\{S, a, b\}, \{a, b\}, P, S)$$

A palindrome can only be recognized with a *pushdown* automaton!

Type1 Languages

Context sensitive languages



Type1: Context sensitive

- "individual non-terminal symbols can only be substituted in a given context“

  aN -> ac

  Nb -> cb

  N -> c
- Type0 + length constraint
- Non-deterministic linear constrained

https://www.geeksforgeeks.org/context-sensitive-grammar-csg-and-language-csl/
Definition 11.4 in *„An introduction to formal languages and automata“,* 5th ed., Linz, Peter.

„Context sensitive but not context free“

$$\mathbf{count}_3 := \{a^n b^n c^n \mid n \in \mathbb{N}\}$$

- Turing machine with linear band constraint
- Type checking in programming languages

e.g.

$$xAy \rightarrow x\upsilon y.$$

e.g.

$$xAy \rightarrow xvy.$$

Question: When can A be replaced?

Swiss German

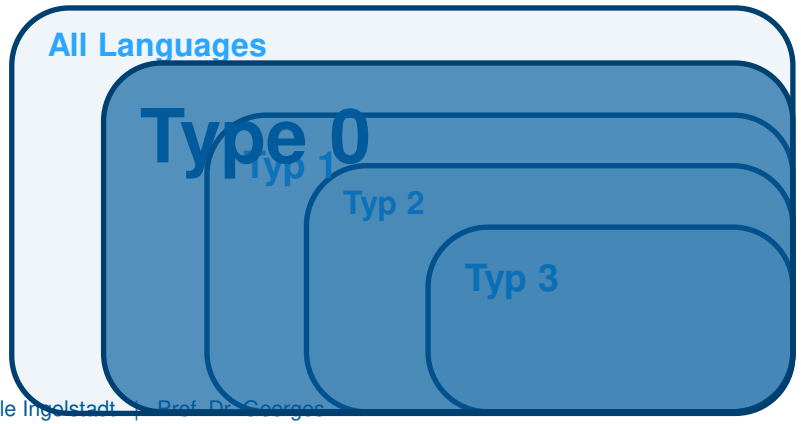
„It has been demonstrated for Swiss German that the language cannot be completely described by a context-free grammar.

Swiss German is probably a context sensitive language. “

Stuart M. Shieber; Evidence against the context-freeness of natural language; In Linguistics and Philosophy 8; 1985

Type0 Languages

„ unlimited grammars“



Phrase structure grammar

- **recursively enumerable languages or semi-decidable languages**
- **Turing machine without restrictions**

Not really relevant for natural languages except you are talking about the human ability to "debug" programs.

https://en.wikipedia.org/wiki/Phrase_structure_grammar

Language models

„Every time I fire a linguist, the performance of the speech recognizer goes up“

Frederick Jelinek

(why?)

https://en.wikipedia.org/wiki/Frederick_Jelinek

Applications

- Spelling Correction
- Suggestion in Messaging
- Machine Translation
- Speech Recogniton
- Handwritng recognition
- …

https://www.youtube.com/watch?v=GiyMGBuu45w

ASR and Language Models

Sequence of audio features

$$\underline{\boldsymbol{w}}' = \underset{\underline{\boldsymbol{w}} \in \underline{\boldsymbol{w}}^*}{\operatorname{argmax}} P(\underline{\boldsymbol{w}} | \underline{\boldsymbol{X}})$$

https://www.youtube.com/watch?v=GiyMGBuu45w

ASR and Language Models

Sequence of audio features

$$\underline{\boldsymbol{w}}' = \underset{\underline{\boldsymbol{w}} \in \underline{\boldsymbol{w}}^*}{\operatorname{argmax}} P(\underline{\boldsymbol{w}} | \underline{\boldsymbol{X}})$$

$\underline{\boldsymbol{w}}^*$ = { „Hallo“, „wie“, „geht“, „es“, „dir“, …
„Hallo wie“, „Hallo, wie geht es dir?“, … }

https://www.youtube.com/watch?v=GiyMGBuu45w

ASR and Language Models

Sequence of audio features

$$\underline{\boldsymbol{w}}' = \underset{\underline{\boldsymbol{w}} \in \underline{\boldsymbol{w}}^*}{\operatorname{argmax}} P(\underline{\boldsymbol{w}} | \underline{\boldsymbol{X}}) = \underset{\underline{\boldsymbol{w}} \in \underline{\boldsymbol{w}}^*}{\operatorname{argmax}} \frac{P(\underline{\boldsymbol{w}}) \cdot P(\underline{\boldsymbol{X}} | \underline{\boldsymbol{w}})}{P(\underline{\boldsymbol{X}})}$$

https://www.youtube.com/watch?v=GiyMGBuu45w

ASR and Language Models

Sequence of audio features

$$\underline{\boldsymbol{w}}' = \underset{\underline{\boldsymbol{w}} \in \underline{\boldsymbol{w}}^*}{\operatorname{argmax}} P(\underline{\boldsymbol{w}} | \underline{\boldsymbol{X}}) = \underset{\underline{\boldsymbol{w}} \in \underline{\boldsymbol{w}}^*}{\operatorname{argmax}} \frac{P(\underline{\boldsymbol{w}}) \cdot P(\underline{\boldsymbol{X}} | \underline{\boldsymbol{w}})}{P(\underline{\boldsymbol{X}})} = \underset{\underline{\boldsymbol{w}} \in \underline{\boldsymbol{w}}^*}{\operatorname{argmax}} P(\underline{\boldsymbol{w}}) \cdot P(\underline{\boldsymbol{X}} | \underline{\boldsymbol{w}})$$

https://www.youtube.com/watch?v=GiyMGBuu45w

ASR and Language Models

Sequence of audio features

$$\underline{\boldsymbol{w}}' = \underset{\underline{\boldsymbol{w}} \in \underline{\boldsymbol{w}}^*}{\operatorname{argmax}} P(\underline{\boldsymbol{w}} | \underline{\boldsymbol{X}}) = \underset{\underline{\boldsymbol{w}} \in \underline{\boldsymbol{w}}^*}{\operatorname{argmax}} \frac{P(\underline{\boldsymbol{w}}) \cdot P(\underline{\boldsymbol{X}} | \underline{\boldsymbol{w}})}{P(\underline{\boldsymbol{X}})} = \underset{\underline{\boldsymbol{w}} \in \underline{\boldsymbol{w}}^*}{\operatorname{argmax}} P(\underline{\boldsymbol{w}}) \cdot P(\underline{\boldsymbol{X}} | \underline{\boldsymbol{w}})$$

Language Model

https://www.youtube.com/watch?v=GiyMGBuu45w

ASR and Language Models

Sequence of audio features

$$\underline{\boldsymbol{w}}' = \underset{\underline{\boldsymbol{w}} \in \underline{\boldsymbol{w}}^*}{\operatorname{argmax}} P(\underline{\boldsymbol{w}} | \underline{\boldsymbol{X}}) = \underset{\underline{\boldsymbol{w}} \in \underline{\boldsymbol{w}}^*}{\operatorname{argmax}} \frac{P(\underline{\boldsymbol{w}}) \cdot P(\underline{\boldsymbol{X}} | \underline{\boldsymbol{w}})}{P(\underline{\boldsymbol{X}})} = \underset{\underline{\boldsymbol{w}} \in \underline{\boldsymbol{w}}^*}{\operatorname{argmax}} P(\underline{\boldsymbol{w}}) \cdot P(\underline{\boldsymbol{X}} | \underline{\boldsymbol{w}})$$

Acoustic Model

https://www.youtube.com/watch?v=GiyMGBuu45w

How likely is a sentence?

"This lecture is exciting."

How likely is a sentence?

"This lecture is exciting." => fits, often

How likely is a sentence?

"This lecture is exciting." => fits, often

"Exciting is this lecture." => fits, rarely

How likely is a sentence?

"This lecture is exciting." => fits, often

"Exciting is this lecture." => fits, rarely

"Is lecture exciting this." => does not fit, never

How likely is a sentence?

"This lecture is exciting." => fits, often

"Exciting is this lecture." => fits, rarely

"Is lecture exciting this." => does not fit, never

"This exciting is lecture." => does not fit, never

How likely is a sentence?

"This lecture is exciting." => fits, often

"Exciting is this lecture." => fits, rarely

"Is lecture exciting this." => does not fit, never

"This exciting is lecture." => does not fit, never

„This lecture is awesome.“ => fits, often

„Lecture this awesome is.“ => does not fit, never

How likely is a sentence?

"This lecture is exciting." => fits, often

"Exciting is this lecture." => fits, rarely

"Is lecture exciting this." => does not fit, never

"This exciting is lecture." => does not fit, never

Exception

"Languages with free-word-order"

Latin, Polish, Hungarian, Hindi, …

… well, a distinction is made:

Colloquial vs. Formal vs. Literary

https://en.wikipedia.org/wiki/Word_order#Pragmatic_word_order

Language freedom

- Morphological freedom
- Grammatical freedom

- Freedom of speech („ehm“, „umm“, …)
- Freedom in prosody

„ Language is direct, intuitive and error-prone“

Refresh: Probabilities

Let $P(a_i)$ be the probability for an event a_i

It always applies:

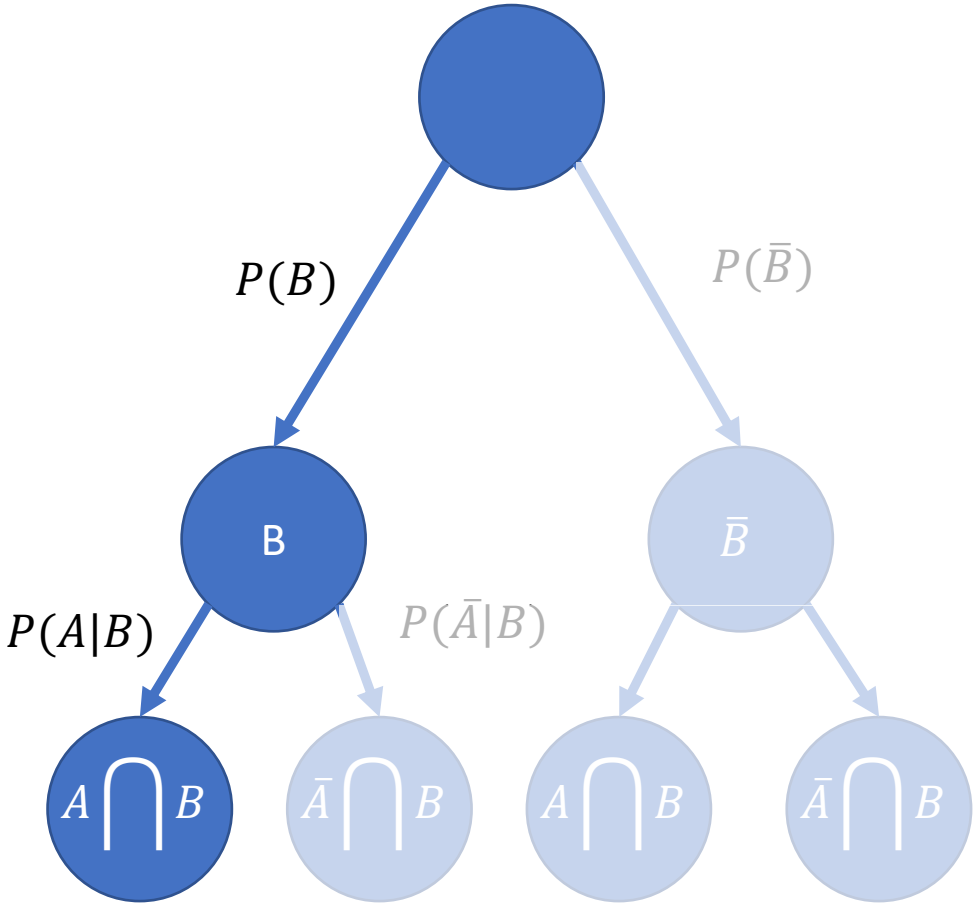
$P(a_i) \geq 0$

$0 \leq P(a_i) \leq 1$ and P assigns a real number

$\sum_i P(a_i) = 1$

https://www.uni-kassel.de/fb07/fileadmin/datas/fb07/5-Institute/IVWL/Kosfeld/lehre/statistikII/StatII_Kap4_.pdf

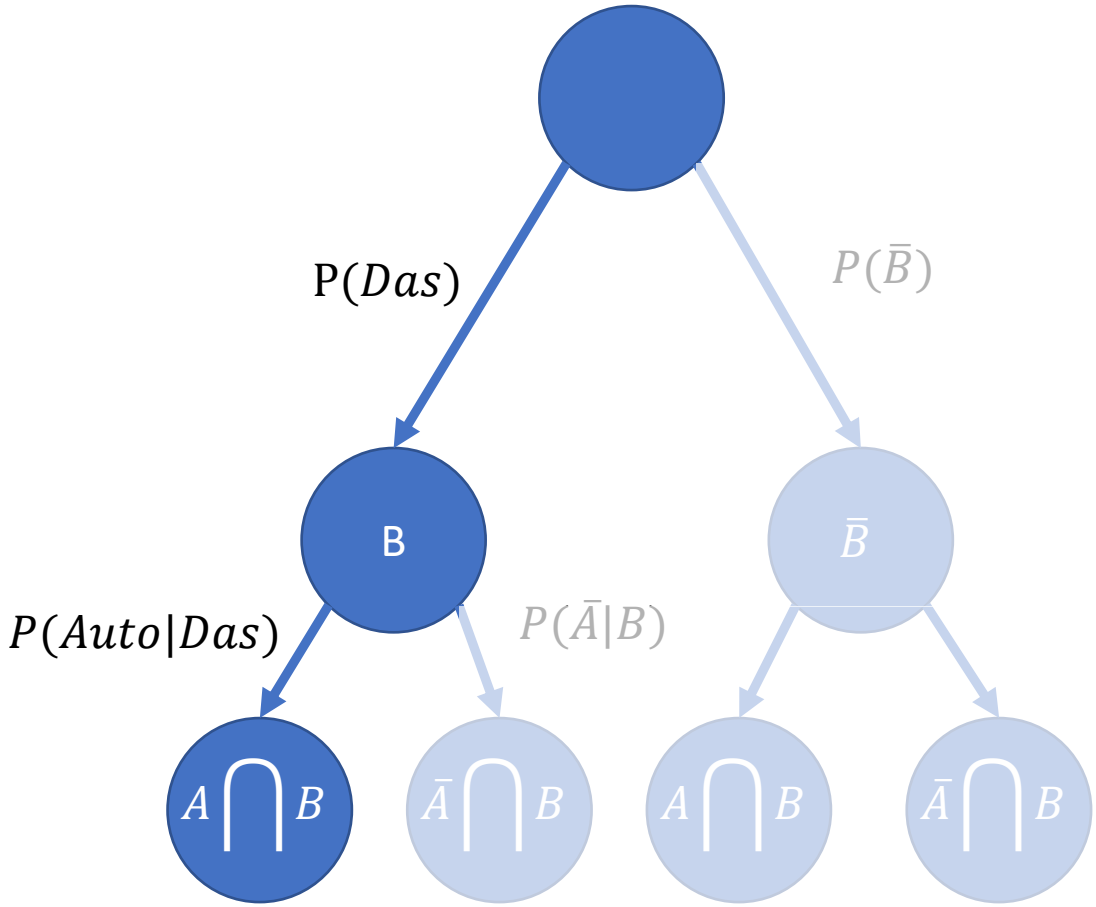
Axiom of probability



$$P(A \cap B) = P(A \mid B)\, P(B)$$ Bayes' theorem

https://de.wikipedia.org/wiki/Bedingte_Wahrscheinlichkeit#Multiplikationssatz

Axiom of probability



$$P(Das\ Auto) = P(Auto|Das)P(Das)$$

Generalize

$$P(A \cap B) = P(A \mid B)\, P(B)$$

https://de.wikipedia.org/wiki/Satz_von_Bayes

Generalize

$$P(A \cap B) = P(A \mid B)\, P(B)$$

$$P\left(\bigcap_{k=1}^{n} A_k\right) = P(A_1) \prod_{k=2}^{n} P\left(A_k \mid \bigcap_{j=1}^{k-1} A_j\right)$$

https://de.wikipedia.org/wiki/Satz_von_Bayes

Generalize

$$P\left(\bigcap_{k=1}^{n} A_k\right) = P(A_1) \prod_{k=2}^{n} P\left(A_k \mid \bigcap_{j=1}^{k-1} A_j\right)$$

$A_1 \cap A_2 \cap \cdots \cap A_n$

$$\frac{P(A_1 \cap A_2)}{P(A_1)}$$

Generalize

$$P\left(\bigcap_{k=1}^{n} A_k\right) = P(A_1) \prod_{k=2}^{n} P\left(A_k \mid \bigcap_{j=1}^{k-1} A_j\right)$$

$$P(A_1 \cap A_2 \cap \cdots \cap A_n) = P(A_1) \cdot \frac{P(A_1 \cap A_2)}{P(A_1)} \cdot \frac{P(A_1 \cap A_2 \cap A_3)}{P(A_1 \cap A_2)} \cdots \frac{P(A_1 \cap \cdots \cap A_n)}{P(A_1 \cap \cdots \cap A_{n-1})}$$

Generalize

$$P\left(\bigcap_{k=1}^{n} A_k\right) = P(A_1) \prod_{k=2}^{n} P\left(A_k \mid \bigcap_{j=1}^{k-1} A_j\right)$$

$$P(A_1 \cap A_2 \cap \cdots \cap A_n) = P(A_1) \cdot \frac{P(A_1 \cap A_2)}{P(A_1)} \cdot \frac{P(A_1 \cap A_2 \cap A_3)}{P(A_1 \cap A_2)} \cdots \frac{P(A_1 \cap \cdots \cap A_n)}{P(A_1 \cap \cdots \cap A_{n-1})}$$

$$= P(A_1) \cdot P(A_2 \mid A_1) \cdot P(A_3 \mid A_1 \cap A_2) \cdots P(A_n \mid A_1 \cap \cdots \cap A_{n-1})$$

How likely is a sentence?

This lecture is exciting

P(<s> Diese Vorlesung ist spannend </s>) =

https://en.wikipedia.org/wiki/Language_model

How likely is a sentence?

P(<s> Diese Vorlesung ist spannend </s>) =

https://en.wikipedia.org/wiki/Language_model

How likely is a sentence?

P(<s> Diese Vorlesung ist spannend </s>) =

P(<s>)

$$P(A_1) \cdot P(A_2 \mid A_1) \cdot P(A_3 \mid A_1 \cap A_2) \cdots P(A_n \mid A_1 \cap \cdots \cap A_{n-1})$$

https://en.wikipedia.org/wiki/Language_model

How likely is a sentence?

P(<s> Diese Vorlesung ist spannend </s>) =

P(<s>)

·P(Diese|<s>)

$$P(A_1) \cdot P(A_2 \mid A_1) \cdot P(A_3 \mid A_1 \cap A_2) \cdots P(A_n \mid A_1 \cap \cdots \cap A_{n-1})$$

https://en.wikipedia.org/wiki/Language_model

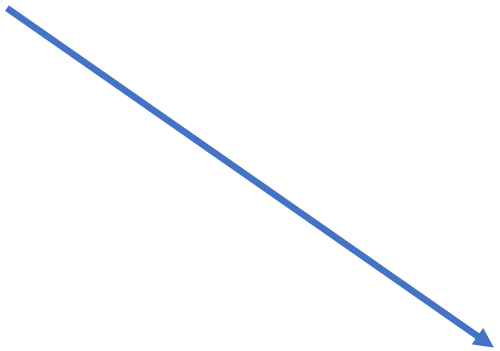
How likely is a sentence?

P(<s> Diese Vorlesung ist spannend </s>) =

P(<s>)

·P(Diese|<s>)

·P(Vorlesung|<s> Diese)



$$P(A_1) \cdot P(A_2 \mid A_1) \cdot P(A_3 \mid A_1 \cap A_2) \cdots P(A_n \mid A_1 \cap \cdots \cap A_{n-1})$$

https://en.wikipedia.org/wiki/Language_model

How likely is a sentence?

P(<s> Diese Vorlesung ist spannend </s>) =

P(<s>)

·P(Diese|<s>)

·P(Vorlesung|<s> Diese)

·P(ist|<s> Diese Vorlesung)

·P(spannend|<s> Diese Vorlesung ist)

·P(</s>|<s> Diese Vorlesung ist spannend)

Word history of 5

https://en.wikipedia.org/wiki/Language_model

How likely is a sentence?

P(<s> Diese Vorlesung ist spannend </s>) ≈

P(<s>)

·P(Diese|<s>)

·P(Vorlesung|<s> Diese)

·P(ist|<s> Diese Vorlesung)

·P(spannend|<s> Diese Vorlesung ist)

·P(</s>|<s> Diese Vorlesung ist spannend)

Word history of 4

Markov assumption

https://en.wikipedia.org/wiki/Markov_property

How likely is a sentence?

P(<s> Diese Vorlesung ist spannend </s>) ≈

P(<s>)

·P(Diese|<s>)

·P(Vorlesung|<s> Diese)

·P(ist|<s> Diese Vorlesung)

·P(spannend|<s> Diese Vorlesung ist)

·P(</s>|<s> Diese Vorlesung ist spannend)

Word history of 3

Markov assumption

How likely is a sentence?

P(<s> Diese Vorlesung ist spannend </s>) ≈

P(<s>)

·P(Diese|<s>)

·P(Vorlesung|<s> Diese)

·P(ist|<s> Diese Vorlesung)

·P(spannend|<s> Diese Vorlesung ist)

·P(</s>|<s> Diese Vorlesung ist spannend)

Word history of 2

Markov assumption

How likely is a sentence?

P(<s> Diese Vorlesung ist spannend </s>) ≈

P(<s>)

·P(Diese|<s>)

·P(Vorlesung|<s> Diese)

·P(ist|<s> Diese Vorlesung)

·P(spannend|<s> Diese Vorlesung ist)

·P(</s>|<s> Diese Vorlesung ist spannend)

Word history of 1

Markov assumption

How likely is a sentence?

P(<s> Diese Vorlesung ist spannend </s>) ≈

P(<s>)

·P(Diese|<s>)

·P(Vorlesung|<s> Diese)

·P(ist|<s> Diese Vorlesung)

·P(spannend|<s> Diese Vorlesung ist)

·P(</s>|<s> Diese Vorlesung ist spannend)

No word history

P(<s> Diese Vorlesung ist spannend </s>) =
P(<s>)·P(Diese)·P(Vorlesung)·P(ist)·P(spannend)·P(</s>)

Only applies if the words are independent!

Summary

$$P(\underline{w}) = P(w_0 w_1 \ldots w_n w_{n+1})$$

Summary

$$P(\underline{w}) = P(w_0 w_1 \ldots w_n w_{n+1})$$

…axiom of probability…

$$= P(w_0) P(w_1 | w_0) P(w_2 | w_0 w_1) \cdots P(w_n | w_1 w_2 \ldots w_{n-1})$$

$$= \prod_{i=1}^{|\underline{w}|} P(w_i | w_1 \ldots w_{i-1})$$

Summary

$$P(\underline{w}) = P(w_0 w_1 \dots w_n w_{n+1})$$ …axiom of probability…

$$= P(w_0) P(w_1|w_0) P(w_2|w_0 w_1) \cdots P(w_n|w_1 w_2 \dots w_{n-1})$$

$$= \prod_{i=1}^{|\underline{w}|} P(w_i|w_1 \dots w_{i-1})$$

$$\approx \prod_{i=1}^{|\underline{w}|} P(w_i|w_{i-n+1} \dots w_{i-1})$$

Markov assumption
"reduce history/context"
z.B. n=2 (this is a 2-gram)

$$P(\underline{w}) \approx \prod_{i=1}^{|\underline{w}|} P(w_i|w_{i-1})$$

Summary

$$P(\underline{w}) = P(w_0 w_1 \dots w_n w_{n+1})$$

$$= P(w_0) P(w_1|w_0) P(w_2|w_0 w_1) \cdots P(w_n|w_1 w_2 \dots w_{n-1})$$

$$= \prod_{i=1}^{|\underline{w}|} P(w_i|w_1 \dots w_{i-1})$$

$$\approx \prod_{i=1}^{|\underline{w}|} P(w_i|w_{i-n+1} \dots w_{i-1})$$

Markov assumption
"reduce history/context"
z.B. n=2 (this is a 2-gram)

$$P(\underline{w}) \approx \prod_{i=1}^{|\underline{w}|} P(w_i|w_{i-1})$$

$$\sum_{\underline{w}} P(\underline{w}) = 1$$

A language model should (if possible) be normalized!

3-gram

P(<s> Diese Vorlesung ist spannend </s>) ≈

P(<s>)

·P(Diese|<s>)

·P(Vorlesung|<s> Diese)

·P(ist| Diese Vorlesung)

·P(spannend| Vorlesung ist)

·P(</s>| ist spannend)

2-gram

P(<s> Diese Vorlesung ist spannend </s>) ≈

P(<s>)

·P(Diese|<s>)

·P(Vorlesung| Diese)

·P(ist| Vorlesung)

·P(spannend| ist)

·P(</s>| spannend)

$P(w_i|w_{i-1})$ = ?

Erinnerung:

$$P(A \cap B) = P(A \mid B)\, P(B)$$

2-gram

$$P(\text{<s> Diese Vorlesung ist spannend </s>}) \approx$$
$$P(\text{<s>})$$
$$\cdot P(\text{Diese} \mid \text{<s>})$$
$$\cdot P(\text{Vorlesung} \mid \text{Diese})$$
$$\cdot P(\text{ist} \mid \text{Vorlesung})$$
$$\cdot P(\text{spannend} \mid \text{ist})$$
$$\cdot P(\text{</s>} \mid \text{spannend})$$

$$P(w_i \mid w_{i-1}) = \frac{P(w_i w_{i-1})}{P(w_{i-1})}$$

2-gram

$$P(\text{<s> Diese Vorlesung ist spannend </s>}) \approx$$
$$P(\text{<s>})$$
$$\cdot P(\text{Diese} \mid \text{<s>})$$
$$\cdot P(\text{Vorlesung} \mid \text{Diese})$$
$$\cdot P(\text{ist} \mid \text{Vorlesung})$$
$$\cdot P(\text{spannend} \mid \text{ist})$$
$$\cdot P(\text{</s>} \mid \text{spannend})$$

Maximum Likelihood Estimate (MLE)

$|.|$:= Count occurrences of .

$$P(w_i \mid w_{i-1}) = \frac{P(w_i w_{i-1})}{P(w_{i-1})} = \frac{\frac{|w_i w_{i-1}|}{|corpus|}}{\frac{|w_{i-1}|}{|corpus|}} = \frac{|w_i w_{i-1}|}{|w_{i-1}|}$$

1-gram

P(<s> Diese Vorlesung ist spannend </s>) ≈

P(<s>)
·P(Diese)
·P(Vorlesung)
·P(ist)
·P(spannend)
·P(</s>)

$$P(w_i) = \frac{|wi|}{|Corpus|}$$

1-gram

P(<s> Diese Vorlesung ist spannend </s>) ≈

P(<s>)
·P(Diese)
·P(Vorlesung)
·P(ist)
·P(spannend)
·P(</s>)

This assumes that the words are independent of each other. This is obviously not correct...

...and yet it works surprisingly well.

0-gram

P(<s> Diese Vorlesung ist spannend </s>) ≈

1/|Vocabulary|

· 1/|Vocabulary|

· 1/|Vocabulary|

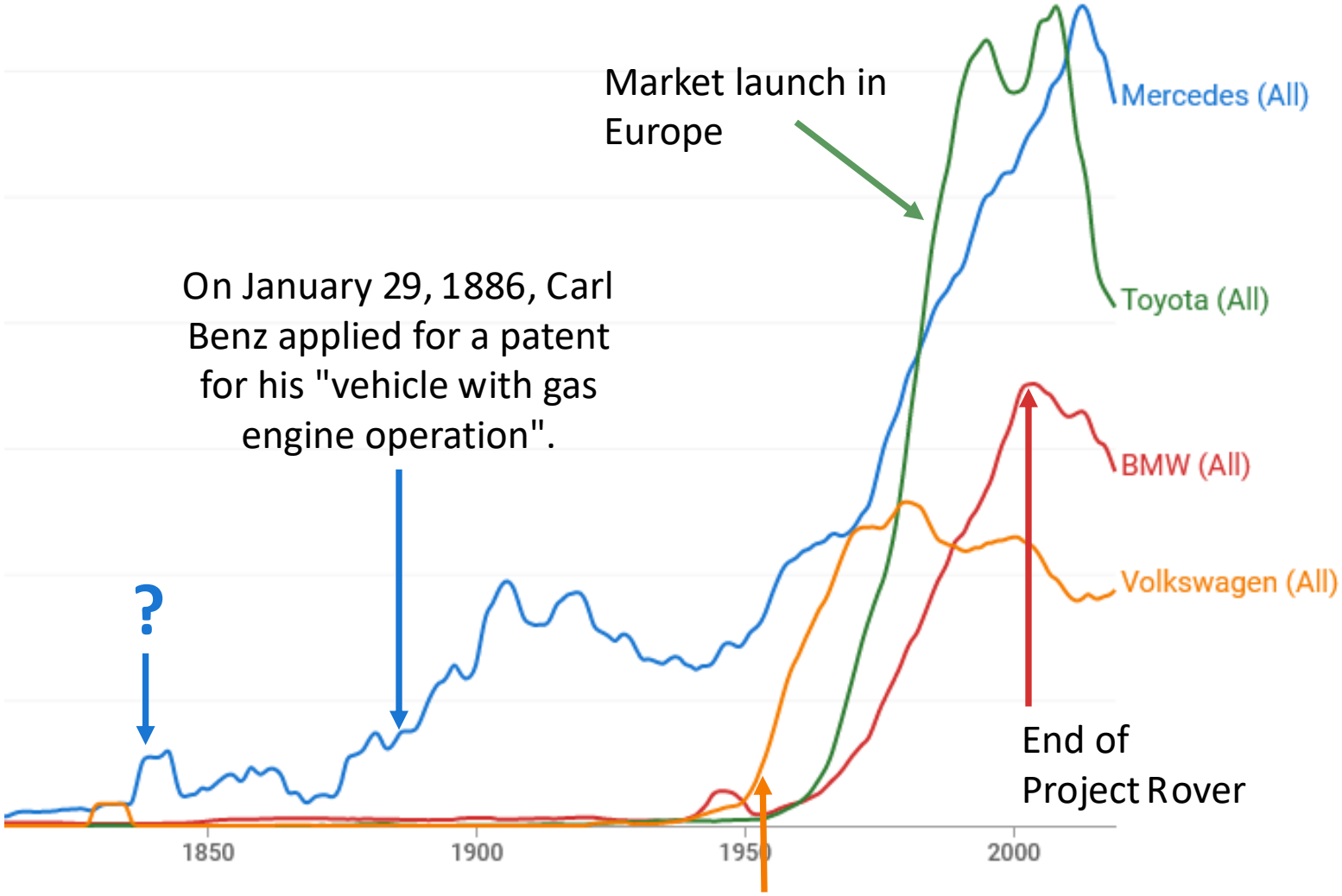
· 1/|Vocabulary|

· 1/|Vocabulary|

· 1/|Vocabulary|

= 1/|Vocabulary|^6 ⇨ The word itself doesn't even matter anymore...
....as long as it is in the .

Word frequencies



http://books.google.com/ngrams/

How many bits do you need?

Information content IC of a word, sentence, text w:

$$IC(w) = log_2 \frac{1}{P(w)} = -log_2 P(w)$$

How many bits do you need?

Information content IC of a word, sentence, text w:

$$IC(w) = log_2 \frac{1}{P(w)} = -log_2 P(w)$$

Probability of occurrence or how likely is w. Typically calculated with a language model P

How many bits do you need?

Information content IC of a word, sentence, text w:

$$IC(w) = log_2 \frac{1}{P(w)} = -log_2 P(w)$$

Example:

P(w) = ½ => IC(w) = ?

How many bits do you need?

Information content IC of a word, sentence, text w:

$$IC(w) = log_2 \frac{1}{P(w)} = -log_2 P(w)$$

Example:

P(w) = ½ => IC(w) = ?

P(v) = 1/16 => IC(v) = ?