Computerlinguistik I

Vorlesung im WiSe 2018/19
(M-GSW-09)

Prof. Dr. Udo Hahn

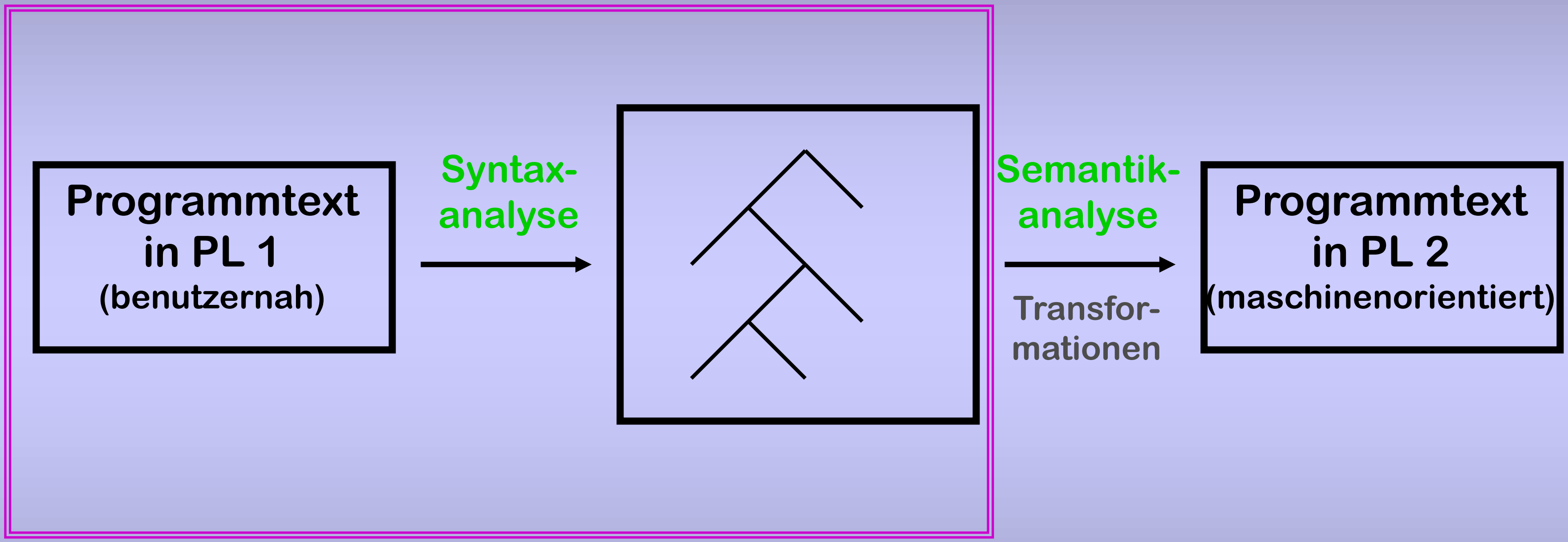
Lehrstuhl für Computerlinguistik
Institut für Germanistische Sprachwissenschaft
Friedrich-Schiller-Universität Jena

http://www.julielab.de

Syntaxanalyse

- Formale Analyse von Ausdrücken einer Sprache
  - Computerlinguistik
    - Formale Analyse von Wörtern oder Sätzen einer natürlichen Sprache (z.B. des Deutschen)
  - Informatik
    - Formale Analyse von Ausdrücken einer forma-len Sprache (z.B. einer Programmiersprache)

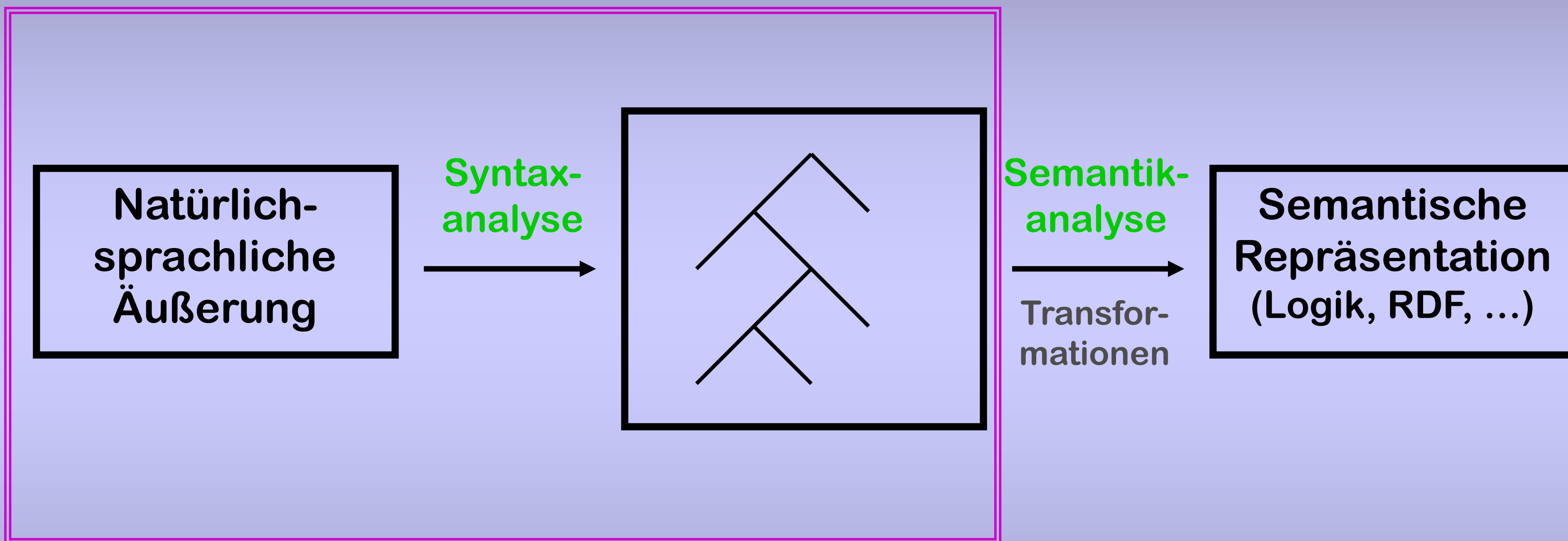
Analyse von Programmen



Aufgaben der Syntaxanalyse:

1. Syntaktisch korrekte Programme werden als korrekt erkannt
2. Syntaktisch unkorrekte Programme werden zurück gewiesen: Fehlererkennung und -diagnose

Analyse von natürlichsprachlichen Äußerungen



Aufgaben der Syntaxanalyse:

1. Syntaktisch korrekte Äußerungen werden als korrekt erkannt
2. Syntaktisch unkorrekte Äußerungen werden zurück gewiesen:

   Aber: Robustheit im Umgang mit paragrammatischen Äußerungen ist wünschenswert !

Beziehung zwischen Informatik und Computerlinguistik

- Informatik besitzt umfangreichen Methodenfundus
  - präzise beschriebene Analyseverfahren
  - Charakterisierung der formalen Eigenschaften dieser Verfahren (Entscheidbarkeit, Berechnungskomplexität)
  - mathematische Beschreibung der „Hintergrundtheorie“ (formale Grammatiken, formale Sprachen, Automaten)
- Übernahme und Adaption an NL in CL

Grundbegriffe zu formalen Sprachen (Wh.)

- Potenzen von Wörtern ω bzw. Wort-mengen $\mathcal{F}$ sind gegeben durch:
  - $\omega^0 := \varepsilon$ $\quad \omega^1 := \omega$ $\quad \omega^i := \omega^{i-1}\,\omega$, für $i \geq 1$
  - $\mathcal{F}^0 := \{\varepsilon\}$ $\quad \mathcal{F}^1 := \mathcal{F}$ $\quad \mathcal{F}^i := \mathcal{F}^{i-1}\,\mathcal{F}$, für $i \geq 1$

Beispiele für Potenzen von Wortmengen:

- DSilbe := {ba, di, ko}
- $\text{DSilbe}^0 = \{\varepsilon\}$, $\text{DSilbe}^1 = \{\text{ba}, \text{di}, \text{ko}\}$
- $\text{DSilbe}^2 = \text{DSilbe}^1\ \text{DSilbe}$
  = { baba, badi, bako, diba, didi, diko , koba, kodi, koko}
- $\text{DSilbe}^3 = \text{DSilbe}^2\ \text{DSilbe}$

Grundbegriffe zu formalen Sprachen (Wh.)

- Die Plushülle bzw. Sternhülle einer Wortmenge $\mathcal{F}$ werden definiert durch:

$$\mathcal{F}^{+} := \bigcup_{i \geq 1} \mathcal{F}^{i} \qquad\qquad \mathcal{F}^{*} := \bigcup_{i \geq 0} \mathcal{F}^{i}$$

- Ist Σ ein Alphabet, dann ist Σ^* die Gesamtheit aller Wörter über Σ. Jede Teilmenge dieser Sternhülle, $\mathcal{L} \subseteq \Sigma^*$, heißt formale Sprache über Σ .

Grundbegriffe zu formalen Grammatiken

- Eine formale Grammatik $\mathcal{G}$ ist ein 4-tupel

$$\mathcal{G} = (N, T, P, S)$$

mit

  - N: das Alphabet der Nicht-Terminalsymbole
  - T: das Alphabet der Terminalsymbole
  - P: eine endliche Menge von Produktionen der Form

    $\alpha \rightarrow \gamma$ (gesprochen: „α *produziert* γ“) mit

    $\alpha \in (N \cup T)^* \, N \, (N \cup T)^*$ und

    $\gamma \in (N \cup T)^*$

  - S: das Startsymbol, $S \in N$

$\mathcal{V} = N \cup T$, bezeichnet das Gesamtalphabet ($N \cap T = \emptyset$)

Beziehung zwischen formalen Grammatiken & formalen Sprachen

- Eine formale Grammatik $\mathcal{G}$ erzeugt eine formale Sprache. Der Erzeugungsprozess ist festgelegt durch eine auf $\mathcal{V}^*$ definierte Relation „$\Rightarrow$“ (gesprochen: *„ist direkt ableitbar nach“*). Für u, v, $\gamma \in \mathcal{V}^*$ und $\alpha \in \mathcal{V}^* N \mathcal{V}^*$ gilt:

$u\ \alpha\ v \Rightarrow u\ \gamma\ v$ genau dann, wenn $\alpha \rightarrow \gamma \in P$

Grundbegriffe zu formalen Grammatiken

- Die transitive Hülle der Relation „$\Rightarrow$“ schreibt man

  $\overset{+}{\Rightarrow}$ (gesprochen: „ist nichttrivial ableitbar nach“)
- Die reflexive und transitive Hülle der Relation „$\Rightarrow$“ schreibt man

  $\overset{*}{\Rightarrow}$ (gesprochen: „ist ableitbar nach“)

Grundbegriffe zu formalen Grammatiken

- Man schreibt $s \overset{n}{\Rightarrow} z$, um auszudrücken, dass $s_0, s_1, \ldots, s_n$ existieren mit

  $s = s_0,\ s_i \Rightarrow s_{i+1}$ für $0 \leq i < n$ und $s_n = z$

- Damit ist also

  $s \overset{+}{\Rightarrow} z$ g.d.w. $s \overset{n}{\Rightarrow} z$ für ein $n \geq 1$ und

  $s \overset{*}{\Rightarrow} z$ g.d.w. entweder $s = z$ oder $s \overset{+}{\Rightarrow} z$

Grundbegriffe zu formalen Grammatiken

- Die von der formalen Grammatik $\mathcal{G}$ = (N, T, P, S) erzeugte formale Sprache $\mathcal{L}(\mathcal{G})$ ist wie folgt definiert:

$$\mathcal{L}(\mathcal{G}) := \{\ \tau \mid \tau \in T^*,\ S \overset{*}{\Rightarrow} \tau\ ,\ \text{S ist Startsymbol von } \mathcal{G}\ \}$$

τ heißt auch Wort der Sprache $\mathcal{L}(\mathcal{G})$.

Grundbegriffe zu formalen Grammatiken

- Zwei formale Grammatiken $\mathcal{G}_1$ und $\mathcal{G}_2$, $\mathcal{G}_1 \neq \mathcal{G}_2$, heißen **äquivalent**, wenn sie dieselbe Sprache erzeugen, d.h.:

$$\mathcal{L}(\mathcal{G}_1) = \mathcal{L}(\mathcal{G}_2)$$

Grundbegriffe zu formalen Grammatiken

- Abhängig von der Form der zugelassenen Produktionen definiert man vier Typen von formalen Grammatiken:
  - Eine Grammatik $\mathcal{G}$ heißt Typ-0-Grammatik, wenn die Gestalt der Produktionen nicht weiter eingeschränkt ist. D.h., sie haben die Form

    $$\alpha \to \gamma$$

    mit

    $$\alpha \in (N \cup T)^* \, N \, (N \cup T)^* \text{ und}$$
    $$\gamma \in (N \cup T)^*$$

Grundbegriffe zu formalen Grammatiken

– Eine Grammatik $\mathcal{G}$ heißt **Typ-1-Grammatik** (**kontextsensitive Grammatik**), wenn P nur Produktionen der Gestalt

$$\alpha \rightarrow \gamma$$

mit

$$\alpha \in (N \cup T)^* \, N \, (N \cup T)^* \text{ und}$$

$$\gamma \in (N \cup T)^*$$

$$|\alpha| \leq |\gamma|$$

(sog. *non-shrinking rules*) und eventuell die Produktion $S \rightarrow \varepsilon$ enthält (wobei letztere nur zugelassen ist, wenn das Startsymbol S in keiner Produktion auf der rechten Seite auftritt).

Grundbegriffe zu formalen Grammatiken

– Eine Grammatik $\mathcal{G}$ heißt **Typ-2-Grammatik** (**kontextfreie Grammatik**), wenn P nur Produktionen enthält der Gestalt

$$A \to \gamma \quad \text{mit } A \in N \text{ und } \gamma \in (N \cup T)^*$$

Grundbegriffe zu formalen Grammatiken

- Eine Grammatik $\mathcal{G}$ heißt Typ-3-Grammatik (reguläre Grammatik), wenn P nur Produktionen der Gestalt

$$A \to \gamma \quad \text{mit } A \in \mathbf{N} \text{ und } \gamma \in \mathbf{N}\,\mathbf{T}^* \cup \mathbf{T}^*$$

  (sog. linkslineare Produktionen) oder nur Produktionen der Gestalt

$$A \to \gamma \quad \text{mit } A \in \mathbf{N} \text{ und } \gamma \in \mathbf{T}^*\,\mathbf{N} \cup \mathbf{T}^*$$

  (sog. rechtslineare Produktionen) enthält.
- Man spricht dann auch entsprechend von linkslinearen bzw. rechtslinearen Grammatiken.
- Eine reguläre Grammatik darf nicht Regeln nach beiden Produktionsregelmustern mischen.

Beispiel einer rechtslinearen Grammatik

$\mathcal{G}$-3 = (N, T, P, S) mit

N = { S, A, B }

T = { a, b }

P = { $S \longrightarrow aA$,

$A \longrightarrow aA$,

$A \longrightarrow bbB$,

$B \longrightarrow bB$,

$B \longrightarrow b$ }

$\mathcal{L}(\mathcal{G}$-3) = { abbb, aabbb, aaabbb, aaaabbb, abbbb, ... }

= $a^n b^m$, $n \geq 1, m \geq 3$

Beispiel einer rechtslinearen Grammatik

$\mathcal{G}$-3 = (N, T, P, S) mit

N = { S, A, B }

T = { a, b }

P = { $S \longrightarrow aA$,

$A \longrightarrow aA$,

$A \longrightarrow bbB$,

$B \longrightarrow bB$,

$B \longrightarrow b$ }

$S \Rightarrow aA$	mit	$S \rightarrow aA \in P$
$aA \Rightarrow abbB$	mit	$A \rightarrow bbB \in P$
$abbB \Rightarrow abbb$	mit	$B \rightarrow b \in P$

$\mathcal{L}(\mathcal{G}$-3) = { abbb, aabbb, aaabbb, aaaabbb, abbbb, ... }

$= a^n b^m$, $n,m \geq 1$

Beispiel einer kontextfreien Grammatik

$\mathcal{G}$-2 = (N, T, P, S) mit

N = { S }

T = { a, b }

P = { $S \longrightarrow aSb$,

$S \longrightarrow ab$ }

$\mathcal{L}(\mathcal{G}\text{-}2)$ = {ab, aabb, aaabbb, aaaabbbb, ... }

= $a^n b^n$, $n \geq 1$

Beispiel einer kontextfreien Grammatik

$\mathcal{G}$-2 = (N, T, P, S) mit

N = { S }

T = { a, b }

P = { S $\longrightarrow$ aSb,

S $\longrightarrow$ ab }

$\mathcal{L}(\mathcal{G}$-2) = {ab, aabb, aaabbb, aaaabbbb, … }

= $a^n b^n$, $n \geq 1$

S $\Rightarrow$ aSb mit S $\rightarrow$ aSb $\in$ P

aSb $\Rightarrow$ aabb mit S $\rightarrow$ ab $\in$ P

Beispiel einer kontextfreien Grammatik

$\mathcal{G}$-2 = (N, T, P, S) mit

N = { S }

T = { a, b }

P = { S $\longrightarrow$ aSb,

S $\longrightarrow$ ab }

$\mathcal{L}(\mathcal{G}\text{-}2)$ = {ab, aabb, aaabbb, aaaabbbb, ... }

= $a^n b^n$, $n \geq 1$

S $\overset{4}{\Rightarrow}$ aaaabbbb S $\overset{*}{\Rightarrow}$ aaaabbbb

$\underset{s_0}{\boxed{S}} \Rightarrow \underset{s_1}{\boxed{aSb}} \Rightarrow \underset{s_2}{\boxed{aaSbb}} \Rightarrow \underset{s_3}{\boxed{aaaSbbb}} \Rightarrow \underset{s_4}{\boxed{aaaabbbb}}$

Beispiel einer kontextsensitiven Grammatik

$\mathcal{G}$-1 = (N, T, P, S) mit

N = { S, B, C, X }

T = { a, b, c }

P = { $S \longrightarrow aSBC$, $S \longrightarrow aBC$,

$CB \longrightarrow XB$, $XB \longrightarrow XC$, $XC \longrightarrow BC$,

$aB \longrightarrow ab$,

$bB \longrightarrow bb$,

$C \longrightarrow c$ }

$\mathcal{L}(\mathcal{G}\text{-}1)$ = { abc, aabbcc, aaabbbccc, ... }

= $a^n b^n c^n$, $n \geq 1$

Beispiel einer kontextsensitiven Grammatik

$\mathcal{G}$-1 = (N, T, P, S) mit

N = { S, B, C, X }

T = { a, b, c }

P = { $S \longrightarrow aSBC$, $S \longrightarrow aBC$,

$CB \longrightarrow XB$, $XB \longrightarrow XC$, $XC \longrightarrow BC$,

$aB \longrightarrow ab$, $bB \longrightarrow bb$, $C \longrightarrow c$ }

$S \overset{*}{\Rightarrow} aaabbbccc$

Beispiel einer kontextsensitiven Grammatik

$\mathcal{G}$-1 = (N, T, P, S) mit

N = { S, B, C, X }

T = { a, b, c }

P = { $S \longrightarrow aSBC$, $S \longrightarrow aBC$,

$CB \longrightarrow XB$, $XB \longrightarrow XC$, $XC \longrightarrow BC$,

$aB \longrightarrow ab$, $bB \longrightarrow bb$, $C \longrightarrow c$ }

$S \Rightarrow aSBC$

Beispiel einer kontextsensitiven Grammatik

$\mathcal{G}$-1 = (N, T, P, S) mit

N = { S, B, C, X }

T = { a, b, c }

P = { S $\longrightarrow$ aSBC, S $\longrightarrow$ aBC,

CB $\longrightarrow$ XB, XB $\longrightarrow$ XC, XC $\longrightarrow$ BC,

aB $\longrightarrow$ ab, bB $\longrightarrow$ bb, C $\longrightarrow$ c }

S $\Rightarrow$ aSBC $\Rightarrow$ aaSBCBC

Beispiel einer kontextsensitiven Grammatik

$\mathcal{G}$-1 = (N, T, P, S) mit

N = { S, B, C, X }

T = { a, b, c }

P = { S $\longrightarrow$ aSBC, S $\longrightarrow$ aBC,

CB $\longrightarrow$ XB, XB $\longrightarrow$ XC, XC $\longrightarrow$ BC,

aB $\longrightarrow$ ab, bB $\longrightarrow$ bb, C $\longrightarrow$ c }

S $\Rightarrow$ aSBC $\Rightarrow$ aaSBCBC $\Rightarrow$ aaaBCBCBC

Beispiel einer kontextsensitiven Grammatik

$\mathcal{G}$-1 = (N, T, P, S) mit

N = { S, B, C, X }

T = { a, b, c }

P = { $S \longrightarrow aSBC$, $S \longrightarrow aBC$,

$CB \longrightarrow XB$, $XB \longrightarrow XC$, $XC \longrightarrow BC$,

$aB \longrightarrow ab$, $bB \longrightarrow bb$, $C \longrightarrow c$ }

$S \Rightarrow aSBC \Rightarrow aaSBCBC \Rightarrow aaaBCBCBC \Rightarrow \ldots \Rightarrow aaaBBCCBC$

Beispiel einer kontextsensitiven Grammatik

$\mathcal{G}$-1 = (N, T, P, S) mit

N = { S, B, C, X }

T = { a, b, c }

P = { S $\longrightarrow$ aSBC, S $\longrightarrow$ aBC,

CB $\longrightarrow$ XB, XB $\longrightarrow$ XC, XC $\longrightarrow$ BC,

aB $\longrightarrow$ ab, bB $\longrightarrow$ bb, C $\longrightarrow$ c }

S $\Rightarrow$ aSBC $\Rightarrow$ aaSBCBC $\Rightarrow$ aaaBCBCBC $\Rightarrow$ … $\Rightarrow$ aaaBBCCBC $\Rightarrow$ … $\Rightarrow$ aaaBBCBCC

Beispiel einer kontextsensitiven Grammatik

$\mathcal{G}$-1 = (N, T, P, S) mit

N = { S, B, C, X }

T = { a, b, c }

P = { S $\longrightarrow$ aSBC, S $\longrightarrow$ aBC,

CB $\longrightarrow$ XB, XB $\longrightarrow$ XC, XC $\longrightarrow$ BC,

aB $\longrightarrow$ ab, bB $\longrightarrow$ bb, C $\longrightarrow$ c }

S $\Rightarrow$ aSBC $\Rightarrow$ aaSBCBC $\Rightarrow$ aaaBCBCBC $\Rightarrow$
… $\Rightarrow$ aaaBBCCBC $\Rightarrow$ … $\Rightarrow$ aaaBBCBCC $\Rightarrow$
… $\Rightarrow$ aaaBBBCCC

Beispiel einer kontextsensitiven Grammatik

$\mathcal{G}$-1 = (N, T, P, S) mit

N = { S, B, C, X }

T = { a, b, c }

P = { S $\longrightarrow$ aSBC, S $\longrightarrow$ aBC,

CB $\longrightarrow$ XB, XB $\longrightarrow$ XC, XC $\longrightarrow$ BC,

aB $\longrightarrow$ ab, bB $\longrightarrow$ bb, C $\longrightarrow$ c }

S $\Rightarrow$ aSBC $\Rightarrow$ aaSBCBC $\Rightarrow$ aaaBCBCBC $\Rightarrow$
… $\Rightarrow$ aaaBBCCBC $\Rightarrow$ … $\Rightarrow$ aaaBBCCBC $\Rightarrow$
… $\Rightarrow$ aaaBBBCCC $\Rightarrow$ aaabBBCCC

Beispiel einer kontextsensitiven Grammatik

$\mathcal{G}$-1 = (N, T, P, S) mit

N = { S, B, C, X }

T = { a, b, c }

P = { S $\longrightarrow$ aSBC, S $\longrightarrow$ aBC,

CB $\longrightarrow$ XB, XB $\longrightarrow$ XC, XC $\longrightarrow$ BC,

aB $\longrightarrow$ ab, bB $\longrightarrow$ bb, C $\longrightarrow$ c }

S $\Rightarrow$ aSBC $\Rightarrow$ aaSBCBC $\Rightarrow$ aaaBCBCBC $\Rightarrow$ … $\Rightarrow$ aaaBBCCBC $\Rightarrow$ … $\Rightarrow$ aaaBBCCBC $\Rightarrow$ … $\Rightarrow$ aaaBBBCCC $\Rightarrow$ aaabBBCCC $\Rightarrow$ aaabbBCCC

Beispiel einer kontextsensitiven Grammatik

$\mathcal{G}$-1 = (N, T, P, S) mit

N = { S, B, C, X }

T = { a, b, c }

P = { S $\longrightarrow$ aSBC, S $\longrightarrow$ aBC,

CB $\longrightarrow$ XB, XB $\longrightarrow$ XC, XC $\longrightarrow$ BC,

aB $\longrightarrow$ ab, bB $\longrightarrow$ bb, C $\longrightarrow$ c }

S $\Rightarrow$ aSBC $\Rightarrow$ aaSBCBC $\Rightarrow$ aaaBCBCBC $\Rightarrow$ … $\Rightarrow$ aaaBBCCBC $\Rightarrow$ … $\Rightarrow$ aaaBBCCBC $\Rightarrow$ … $\Rightarrow$ aaaBBBCCC $\Rightarrow$ aaabBBCCC $\Rightarrow$ aaabbBCCC $\Rightarrow$ aaabbbCCC

Beispiel einer kontextsensitiven Grammatik

$\mathcal{G}$-1 = (N, T, P, S) mit

N = { S, B, C, X }

T = { a, b, c }

P = { $S \longrightarrow aSBC$, $S \longrightarrow aBC$,

$CB \longrightarrow XB$, $XB \longrightarrow XC$, $XC \longrightarrow BC$,

$aB \longrightarrow ab$, $bB \longrightarrow bb$, $C \longrightarrow c$ }

$S \Rightarrow aSBC \Rightarrow aaSBCBC \Rightarrow aaaBCBCBC \Rightarrow \ldots \Rightarrow aaaBBCCBC \Rightarrow \ldots \Rightarrow aaaBBCCBC \Rightarrow \ldots \Rightarrow aaaBBBCCC \Rightarrow aaabBBCCC \Rightarrow aaabbBCCC \Rightarrow aaabbbCCC \Rightarrow \ldots \Rightarrow aaabbbccc$

Beispiel einer kontextsensitiven Grammatik

$\mathcal{G}$-1 = (N, T, P, S) mit

N = { S, B, C, X }

T = { a, b, c }

P = { $S \longrightarrow aSBC$, $S \longrightarrow aBC$,
$CB \longrightarrow XB$, $XB \longrightarrow XC$, $XC \longrightarrow BC$,
$aB \longrightarrow ab$, $bB \longrightarrow bb$, $C \longrightarrow c$ }

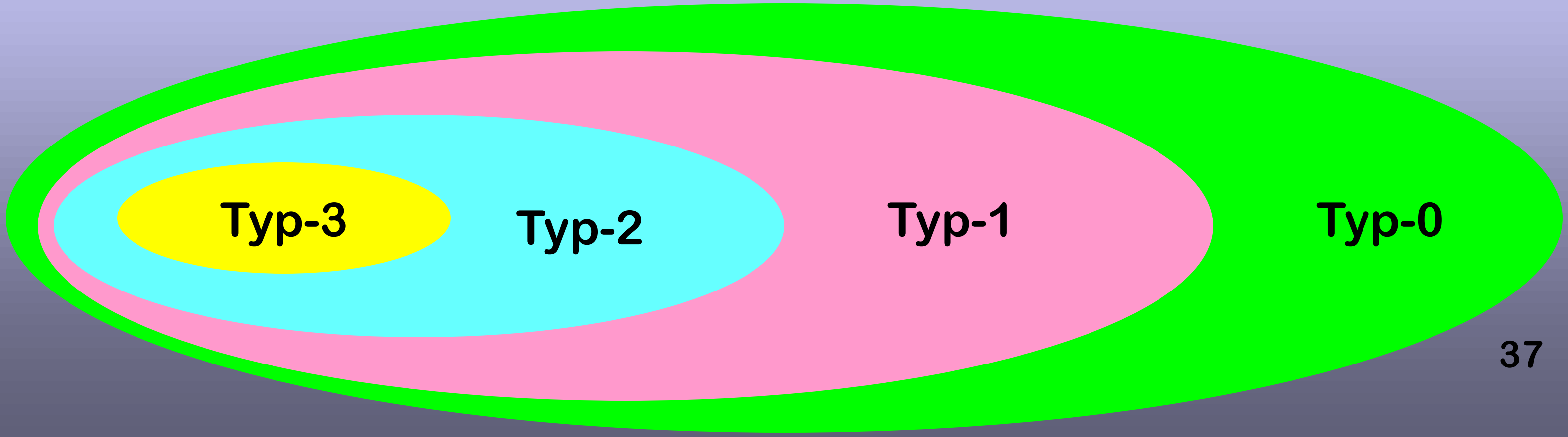
S $\Rightarrow$ aSBC $\Rightarrow$ aaSBCBC $\Rightarrow$ aaaBCBCBC $\Rightarrow$ … $\Rightarrow$ aaaBBCCBC $\Rightarrow$ … $\Rightarrow$ aaaBBCCBC $\Rightarrow$ … $\Rightarrow$ aaaBBBCCC $\Rightarrow$ aaabBBCCC $\Rightarrow$ aaabbBCCC $\Rightarrow$ aaabbbCCC $\Rightarrow$ … aaabbbccc

Typen formaler Sprachen (und ihr Bezug zu Grammatik-Typen)

– Eine formale Sprache heißt vom Typ 0, 1, 2 oder 3, wenn sie von einer Grammatik des entsprechenden Typs erzeugt werden kann.
  - Eine Typ-1-Sprache heißt auch kontextsensitive Sprache.
  - Eine Typ-2-Sprache heißt auch kontextfreie Sprache.
  - Eine Typ-3-Sprache heißt auch reguläre Sprache.

Chomsky-Hierarchie formaler Sprachen

– **Für jedes Alphabet Σ (mit mindestens zwei Symbolen) ist die Menge der Typ-i-Sprachen über Σ für $i = 0, 1, 2$ jeweils (echte) Obermenge der Typ-[i+1]-Sprachen über Σ.**
Die damit gegebene Hierarchie von formalen Sprachen heißt Chomsky-Hierarchie.



Formale Eigenschaften natürlicher Sprachen

- Natürliche Sprachen werden im Folgenden als formale Sprachen (Mengen von Wörtern) betrachtet
- Problemstellung:

  Welcher Typ formaler Sprachen charakterisiert natürliche Sprachen ?

Natürliche Sprachen als reguläre Sprachen

- Natürliche Sprachen sind ausdrucksstärker als reguläre Sprachen (Typ-3)
  - <u>Beweisbar</u> durch Pumping-Theorem und die Tatsache, dass reguläre Sprachen unter Mengen-schnitt abgeschlossen sind ($L_{r1} \cap L_{r2} = L_{r3}$)
  - Illustration durch Sprachdaten

Sind Natürliche Sprachen regulär ?

Satzbeispiele:

(1) The cat waited.

(2) The cat the dog admired waited.

(3) The cat the dog the ant bit admired waited.

Satzformel:

(„the“ N)^{n} (V$_{transitiv}$)$^{n-1}$ („waited“ [V$_{intransitiv}$])^{1}

bzw. $a^n b^{n-1} x$

Beweisidee (Abschluss unter Schnitt):

Englisch$_{reg}$ [als Menge von Ketten] $\cap$ $N^* V^* x$ = $N^n V^{n-1} x$

Aber: $a^n b^n$ ist eine CFL (pumping lemma)!

Natürliche Sprachen als kontextfreie Sprachen

- Einige natürliche Sprachen sind (etwas) ausdrucksstärker als kontextfreie Sprachen
  - Beweisbar durch Pumping Theorem und die Tatsache, dass kontextfreie Sprachen unter Mengenschnitt mit regulären Sprachen abgeschlossen sind ($L_{cf1} \cap L_{r2} = L_{cf3}$)
  - Illustration durch Sprachdaten

Sind alle Natürlichen Sprachen kontextfrei ?

Satzbeispiele (cross-serial dependencies):

(1) Jan säit.

(2) Jan säit das mer em Hans hälfed.

(3) Jan säit das mer em Hans es huus hälfed aastriiche.

(4) Jan säit das mer d'chind em Hans es huus lönd hälfe aastriiche.

(5) Jan sagte, dass wir - die Kinder$_{AKK}$ – dem Hans$_{DAT}$ – das Haus lassen$_{AKK}$ - helfen$_{DAT}$ - anzustreichen

Satzformel:

Jan säit das mer („d'chind"$_{AKK}$)^n („em Hans"$_{DAT}$)^m es huus („lönd"$_{AKK}$)^n („hälfe"$_{DAT}$)^m aastriiche.

bzw. $w\ a^n\ b^m\ x\ c^n\ d^m\ y$

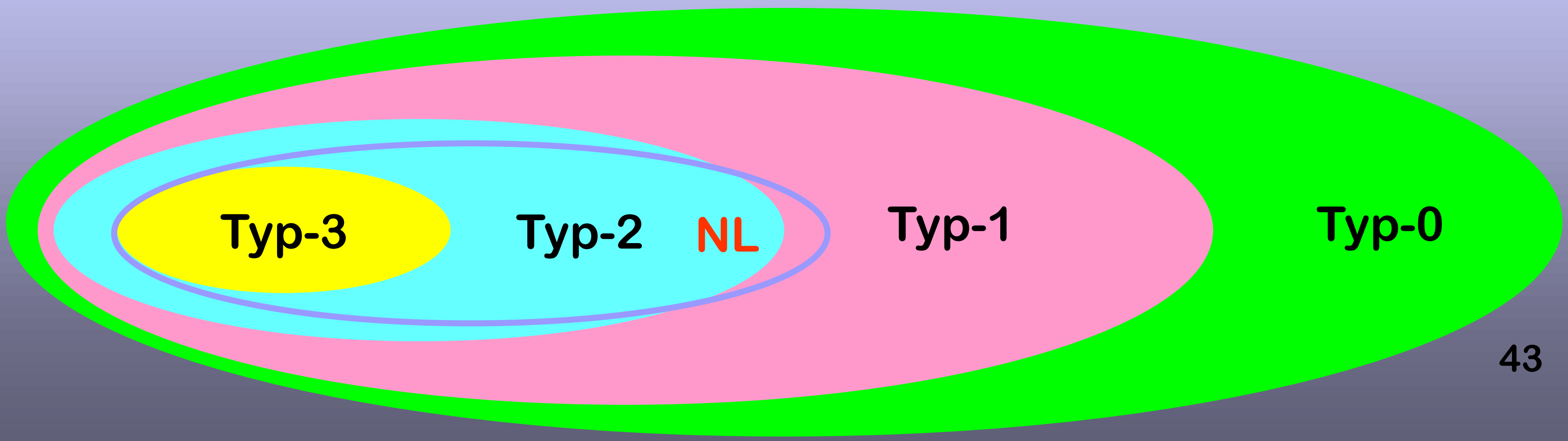
Beweisidee (Abschluss unter Schnitt):

SchweizerDeutsch$_{cf}$ [als Menge von Ketten] $\cap$ Jan säit das mer $(N_{AKK})^*$ $(N_{DAT})^*$ es huus $(V_{AKK})^*$ $(V_{DAT})^*$ aastriiche
= Jan säit das mer $(N_{AKK})^n$ $(N_{DAT})^m$ es huus $(V_{AKK})^n$ $(V_{DAT})^m$ aastriiche.

Aber: $w\ a^n\ b^m\ x\ c^n\ d^m\ y$ ist eine CSL (pumping lemma)!

Natürliche Sprachen als formale Sprachen

- NLs sind keine Typ-3-Sprachen
- NLs sind überwiegend (Englisch, Deutsch, Französisch, Spanisch, …) Typ-2-Sprachen
- Einige wenige NLs sind sicher (Schweizer Deutsch) bzw. vermutlich (Niederländisch, Bambara [Mali]) milde Typ-1-Sprachen



Deterministischer FSA-Erkennungsalgorithmus

- Band (für die zu testende Kette)
  - in n Zellen aufgeteilt
  - jede Zelle hält ein Symbol der Kette
- Zustandstransitionstabelle (2-D Matrix)
  - Zeilen: Zustandsmarken des FSA
  - Spalten: Symbole des Alphabets
  - Zelle: Folgezustand

Erkennungsalgorithmus für deterministischen FSA

Funktion D-Erkenner(↓Band,↓FSA) = „accept“ oder **„reject“**

Index ⇚ Bandanfang
AktualZustand ⇚ Anfangszustand des FSA
LOOP
IF Ende der Eingabekette ist erreicht THEN
    IF AktualZustand ist ein Endzustand THEN return **„accept“**
    ELSE return **„reject“**
ELSE-IF Zustandstransitionstabelle[AktualZustand, Band(Index)] = 0 THEN
    return **„reject“**
ELSE AktualZustand ⇚ Zustandstransitionstabelle[AktualZustand, Band(Index)]
    Index ⇚ Index + 1
LOOPEND

Funktion Band(Index) ermittelt das Symbol, das auf der Position **Index** im **Band** steht

Kostenrechnung für det. FSA-Erkennungsalgorithmus

Funktion D-Erkenner(↓Band,↓FSA) = „accept“ oder **„reject“**

Index ⇚ Bandanfang 1
AktualZustand ⇚ Anfangszustand des FSA 1
LOOP
IF Ende der Eingabekette ist erreicht THEN n
    IF AktualZustand ist ein Endzustand THEN return **„accept“** 1
    ELSE return **„reject“**
ELSE-IF Zustandstransitionstabelle[AktualZustand, Band(Index)] = 0 THEN 1
    return **„reject“** 1
ELSE AktualZustand ⇚ Zustandstransitionstabelle[AktualZustand, Band(Index)] 1
    Index ⇚ Index + 1 1
LOOPEND

Funktion Band(Index) ermittelt das Symbol, das auf der Position **Index** im **Band** steht

Komplexitätsklassen

- $O(1)$ konstant ☺
- $O(\log n)$ logarithmisch ☺
- $O(n)$ linear 😐
- $O(n^k)$ polynomial ($k \in [2,4]$) ☹
- $O(n^k)$ polynomial ($k > 4$) ☺
- $O(k^n)$ exponentiell ☹

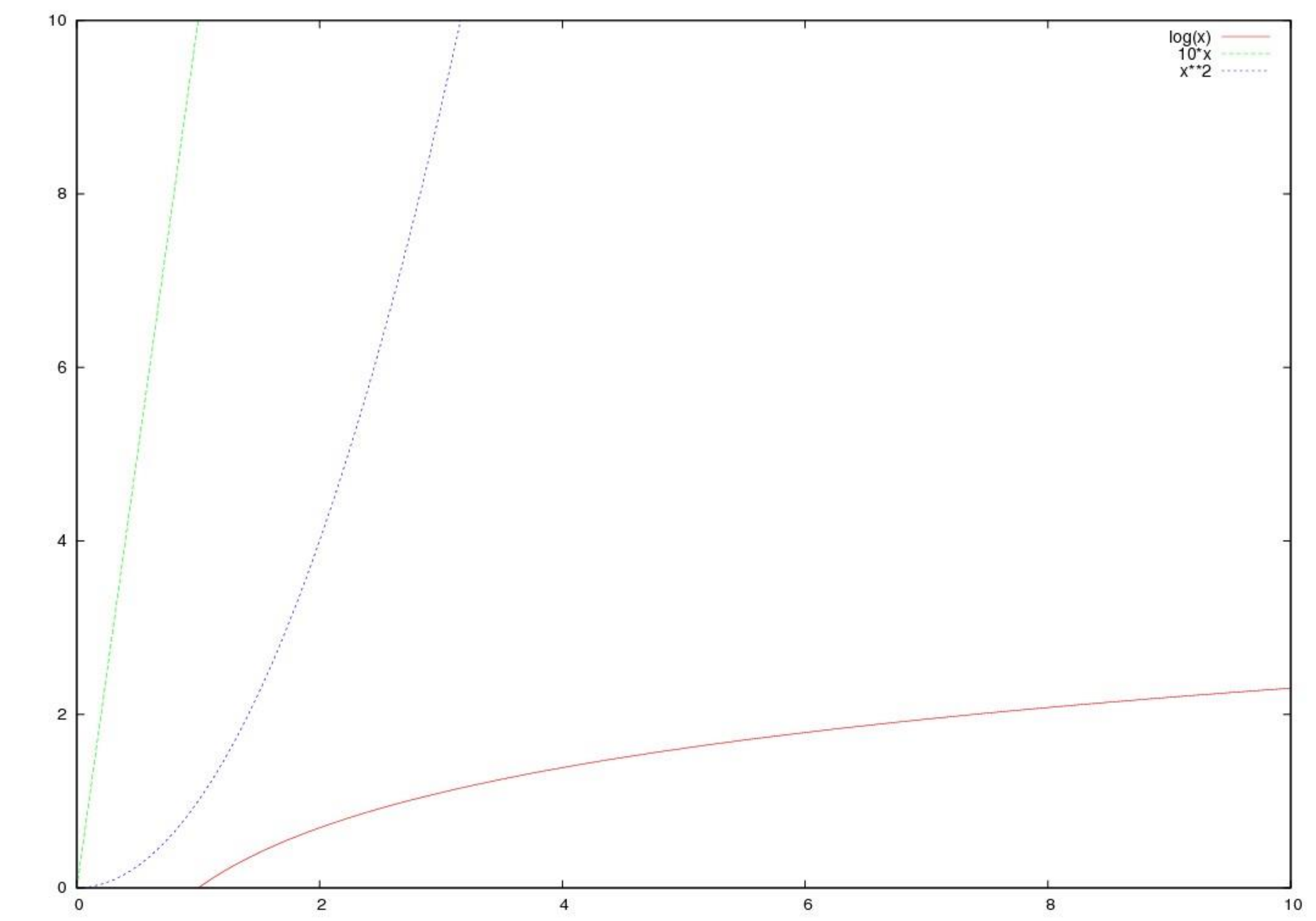
- wobei *n* die Problemgröße ist

Charakteristische Problemgrößen beim Parsing

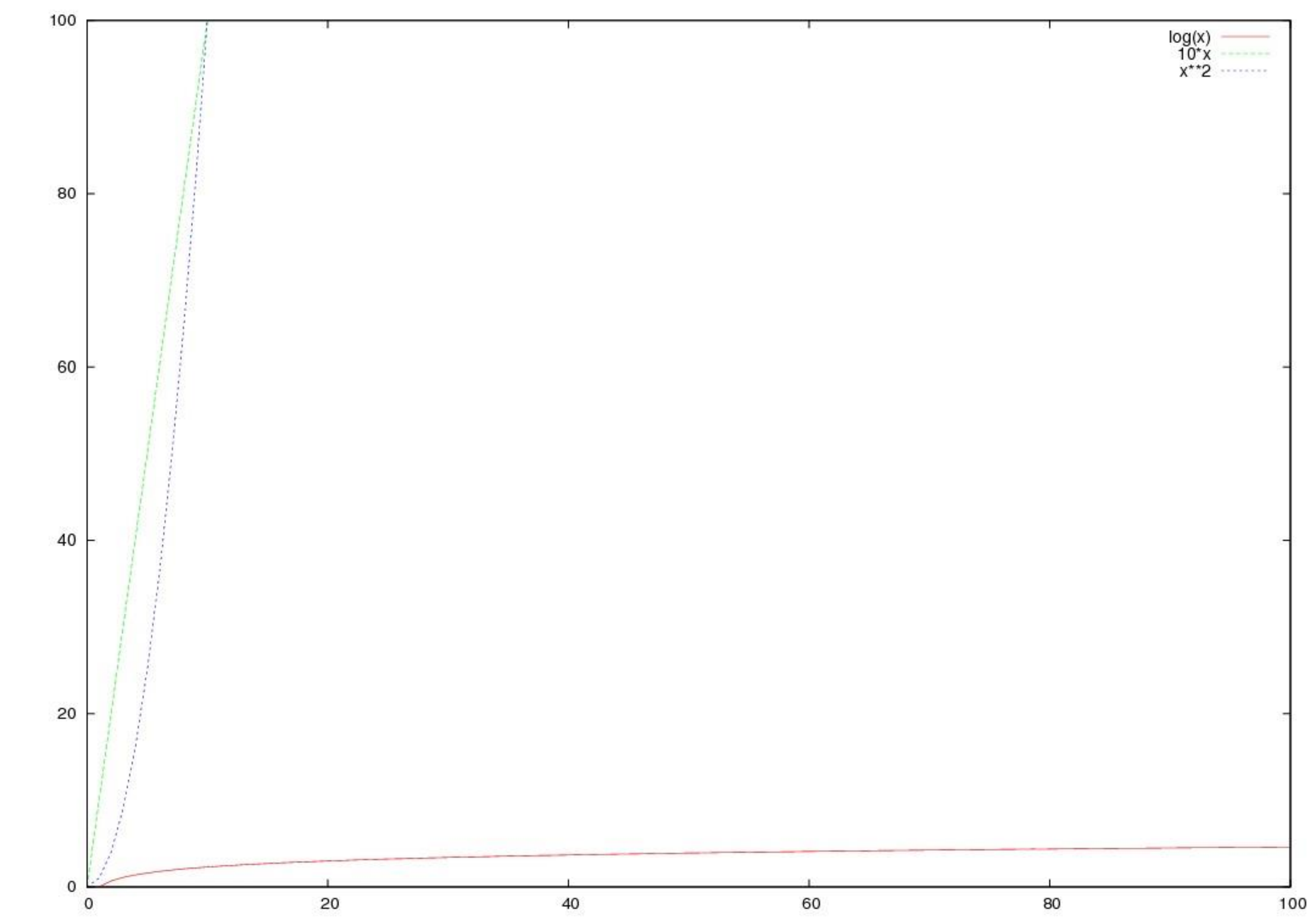
- *n* ist die Länge der Eingabe (Wort, Satz)
- *n* ist die Kardinalität der Produktionsregel-menge

- Hinweis: *n* ist vom (Parsing-)Algorithmus abhängig

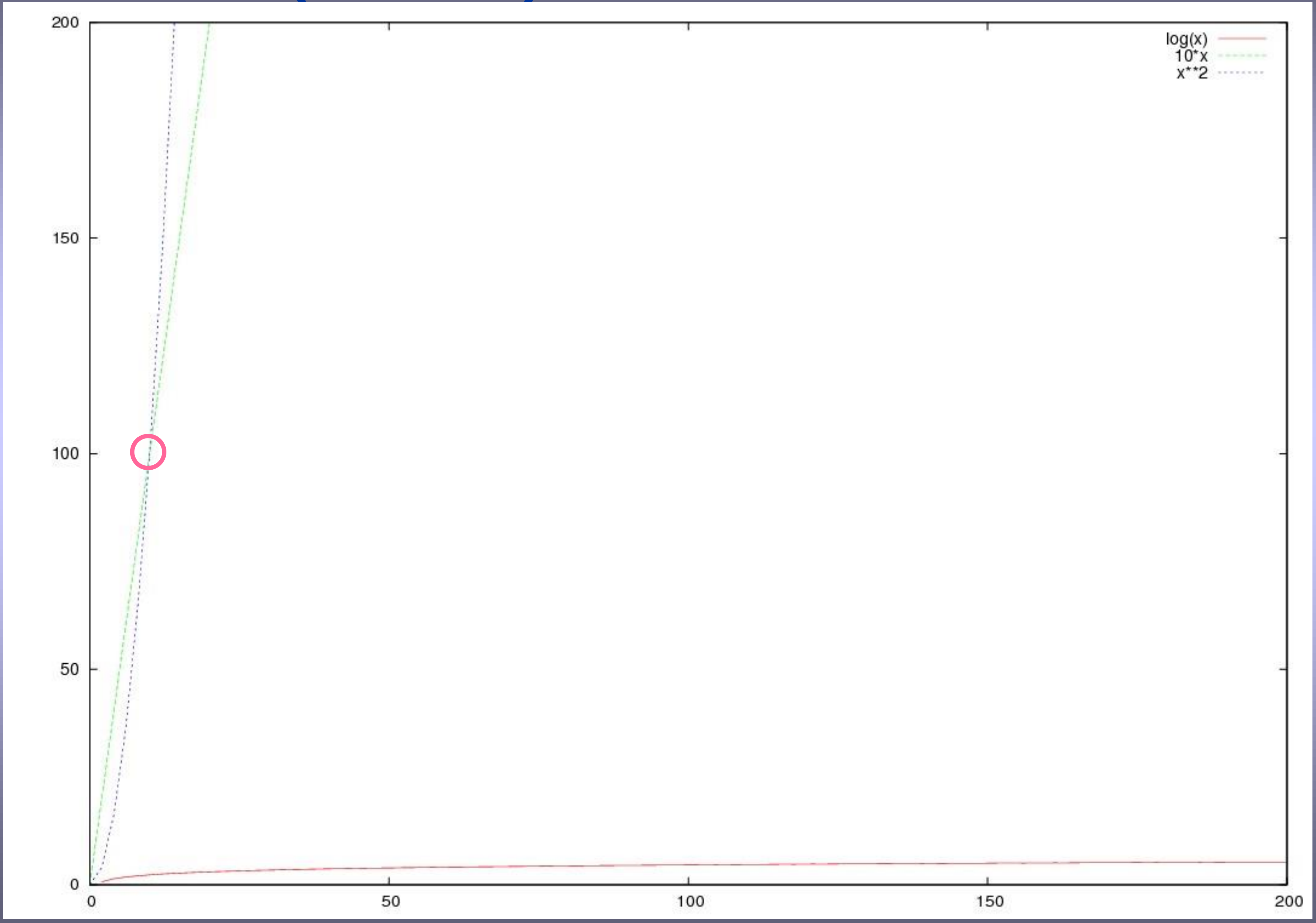
(Lauf-)Zeit-Klassen



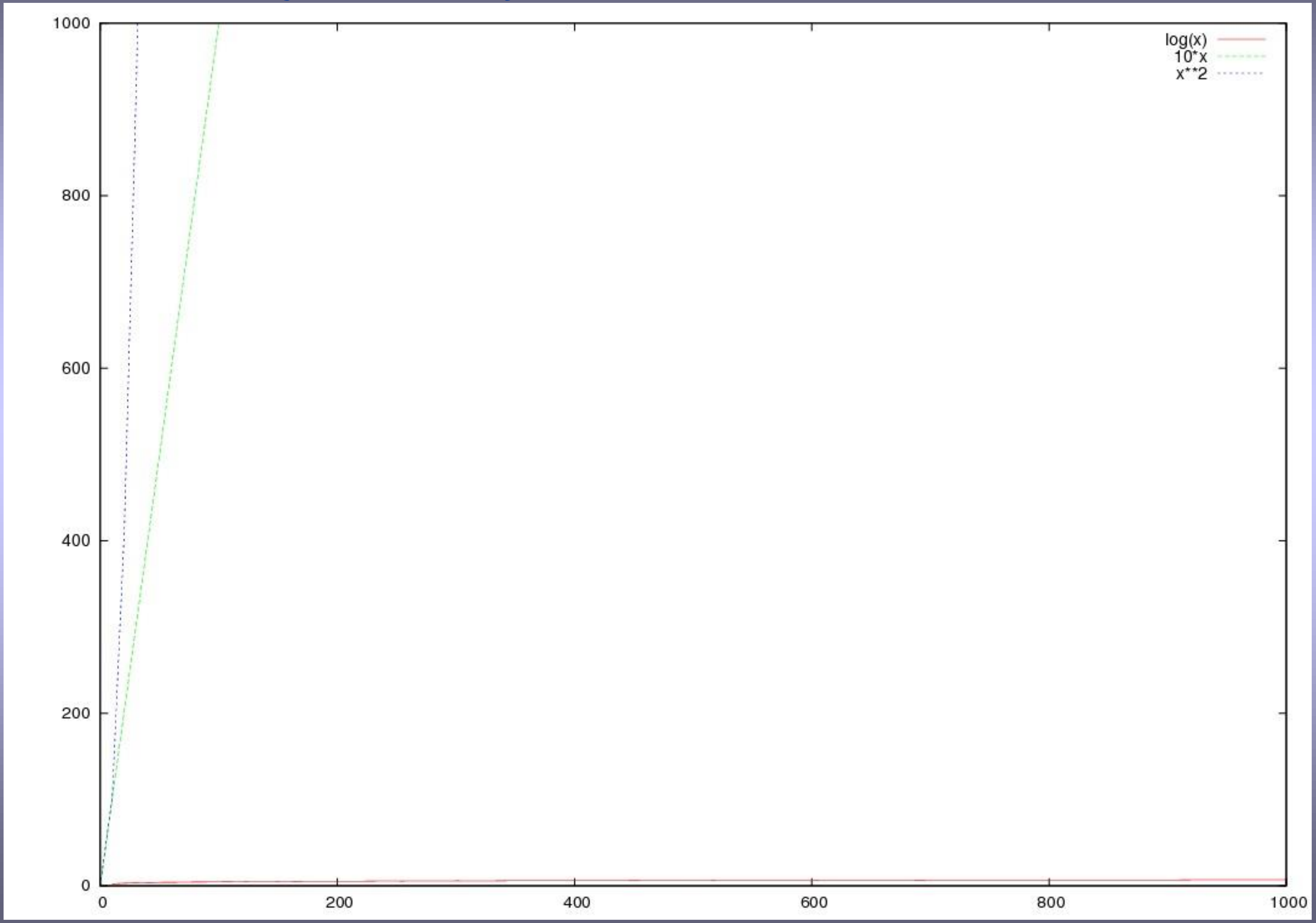
(Lauf-)Zeit-Klassen



(Lauf-)Zeit-Klassen



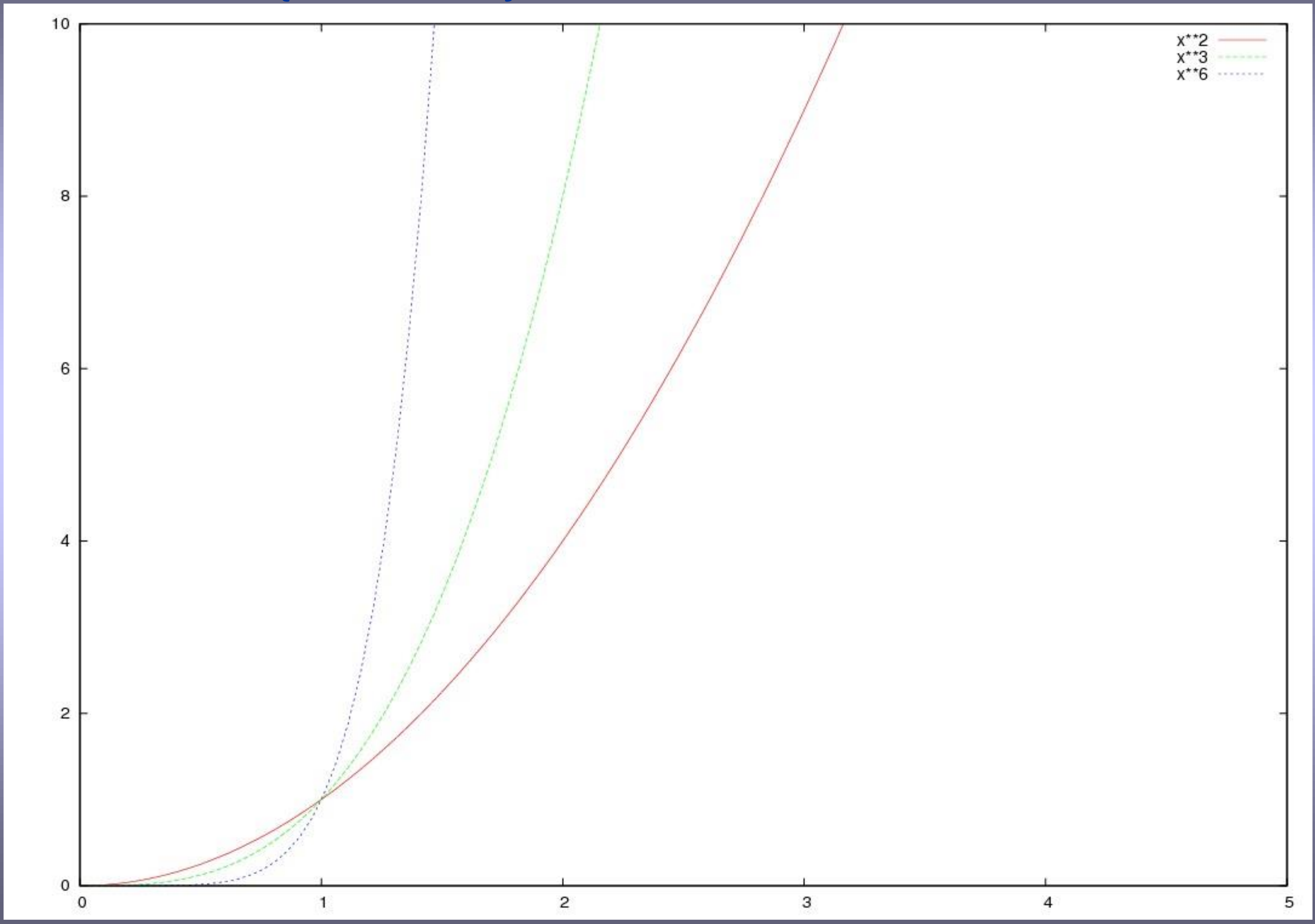
(Lauf-)Zeit-Klassen



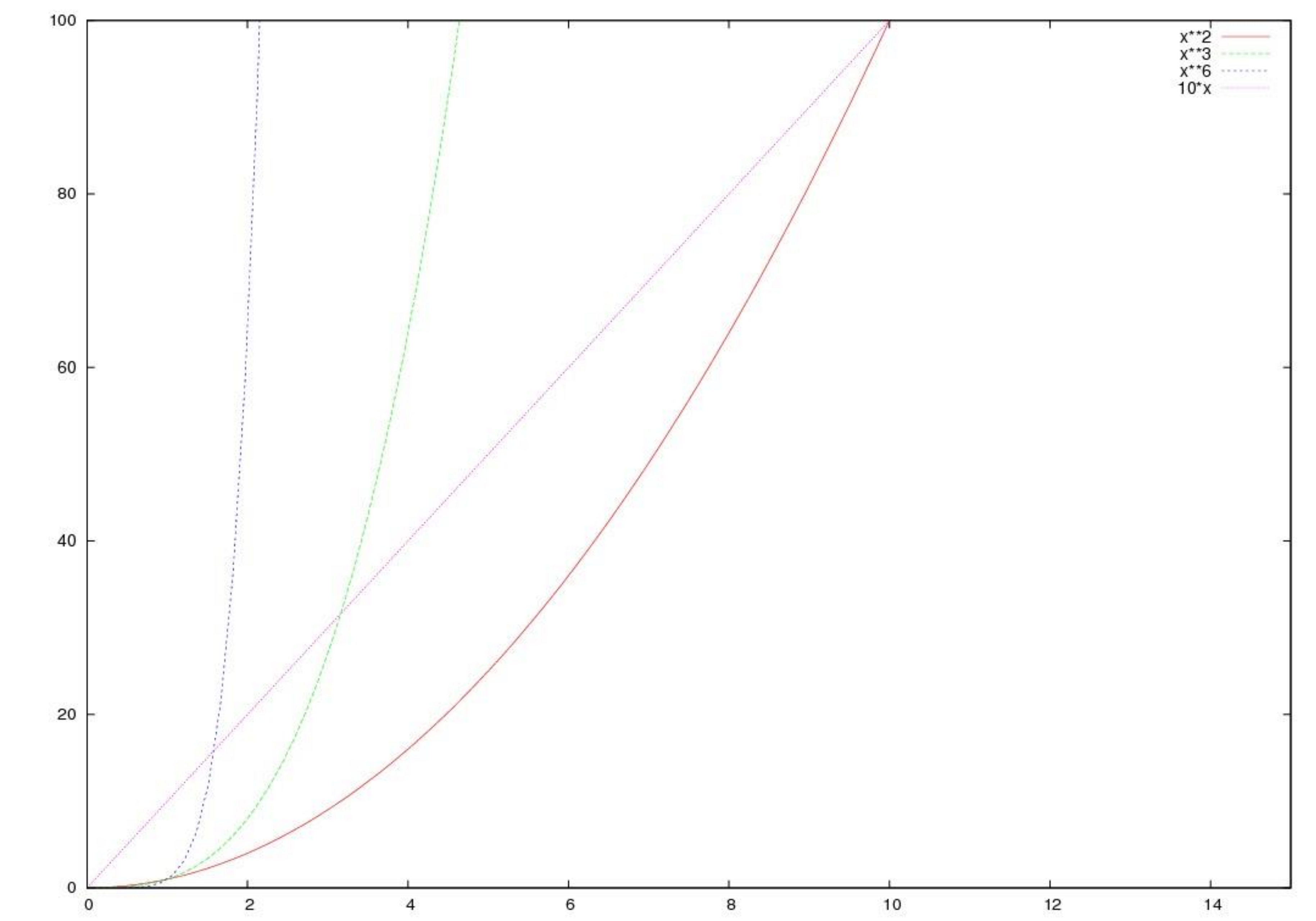
Implikationen für die automatische Sprachanalyse

- NLs sind keine Typ-3-Sprachen
  - Trotzdem werden endliche Automaten (FSA) für NLP-Analytik eingesetzt
  - Lineare Laufzeit ($O(n)$)
- NLs sind überwiegend (Englisch, Deutsch, Französisch, Spanisch, …) Typ-2-Sprachen
  - Kellerautomaten als Basismodell
  - Syntaxanalyse in max. kubischer Laufzeitkomplexität ($O(n^3)$)
- Einige wenige NLs sind sicher (Schweizer Deutsch) bzw. vermutlich (Niederländisch, Bambara [Mali]) milde Typ-1-Sprachen
  - Syntaxanalyse in max. $O(n^6)$ Laufzeitkomplexität
  - Grammatikmodell: Tree Adjoining Grammar (TAG)

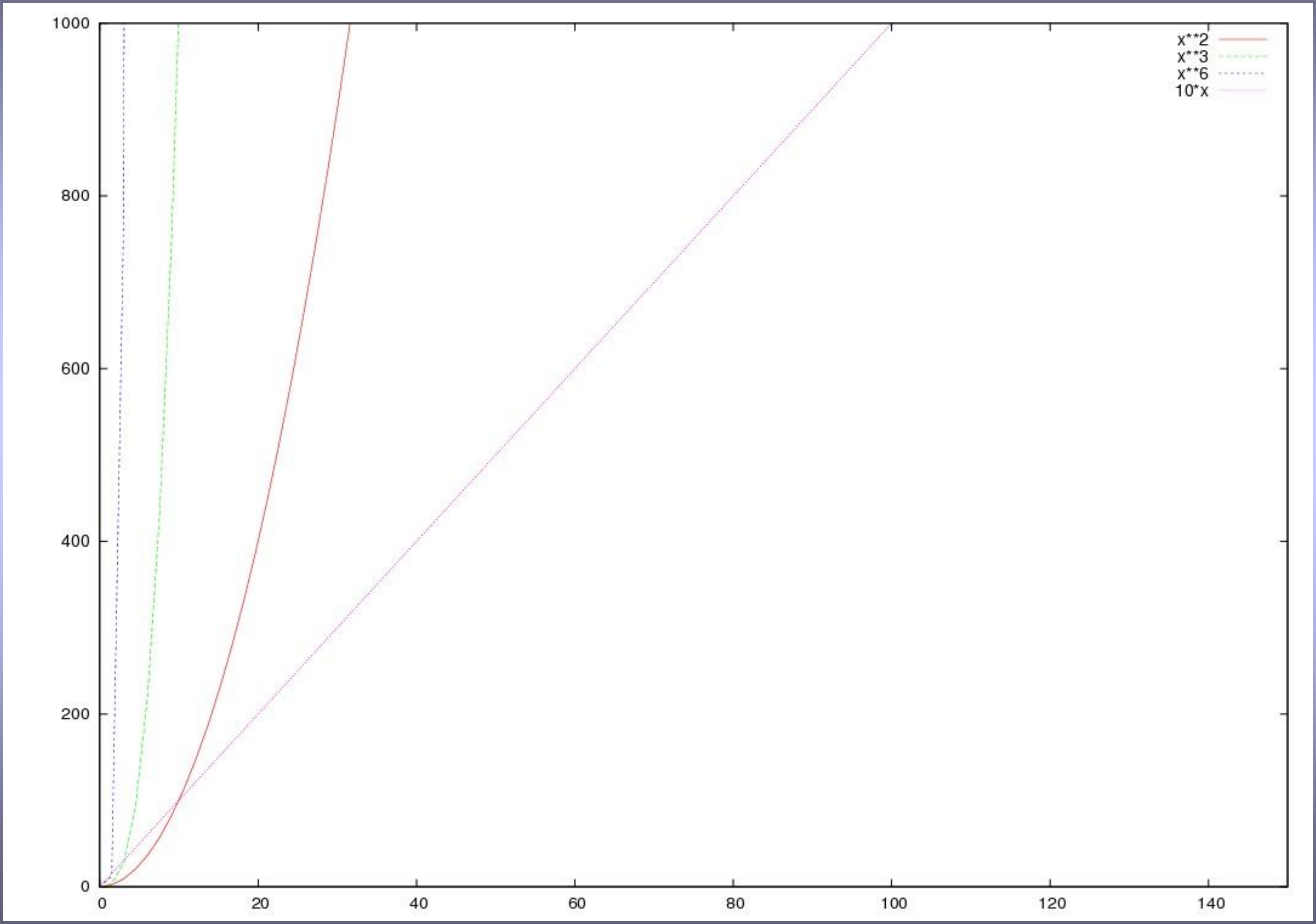
(Lauf-)Zeit-Klassen



(Lauf-)Zeit-Klassen



(Lauf-)Zeit-Klassen



Chomsky-Hierarchie formaler Sprachen und Automaten

Sprache	*Automat*	*Grammatik*	*Erkennung*	*Abhängigkeit*
rekursiv aufzählbar	Turing Maschine	unbeschränkt $Baa \rightarrow \varepsilon$	unentscheidbar	beliebig
kontext-sensitiv	Linear gebunden	kontext-sensitiv $At \rightarrow aA$	NP-vollständig	überkreuzt
kontext-frei	Kellerautomat (Stapel)	kontextfrei $S \rightarrow gSc$	polynomiell	eingebettet
regulär	Endlicher Automat	regulär $A \rightarrow cA$	linear	strikt lokal

Quelle: D. Searls

Ableitungen in kontextfreien Grammatiken

Die folgenden Festlegungen zu Ableitungen beruhen auf der Annahme, dass die zugrundeliegende Grammatik kontextfrei ist!

Grundbegriffe zu formalen Grammatiken

– **Eine Grammatik $\mathcal{G}$ heißt Typ-2-Grammatik, (kontextfreie Grammatik), wenn P nur Produktionen enthält der Gestalt**

$$A \rightarrow \gamma \quad \text{mit } A \in N \text{ und } \gamma \in (N \cup T)^*$$

Ableitungen in kontextfreien Grammatiken

- Ein nicht-identischer Ableitungsschritt ist definiert als Tripel

  δ = $[\mathcal{l}, \mathrm{A} \longrightarrow \gamma, \mathcal{r}]$ mit $\mathcal{l}, \mathcal{r} \in \mathcal{V}^*$ und $\mathrm{A} \longrightarrow \gamma \in$ P

  δ beschreibt die Beziehung: $\mathcal{l}\mathrm{A}\mathcal{r} \Rightarrow \mathcal{l}\gamma\mathcal{r}$
- Als Quelle bzw. Ziel des Ableitungsschrittes δ gelten $\mathcal{Q}(\delta) = \mathcal{l}\mathrm{A}\mathcal{r}$ bzw. $\mathcal{Z}(\delta) = \mathcal{l}\gamma\mathcal{r}$.

Ableitungen in kontextfreien Grammatiken

- Eine endliche nichtleere Folge

$$\Delta = \{ \delta_i \}^n_{i=1}$$

von Ableitungsschritten δ_i mit $\mathcal{Q}(\delta_{i+1}) = \mathcal{Z}(\delta_i)$ für $1 \leq i < n$ heißt **Ableitung** von $\mathcal{Q}(\delta_1)$ nach $\mathcal{Z}(\delta_n)$.

- **Quelle** von Δ ist $\mathcal{Q}(\Delta) = \mathcal{Q}(\delta_1)$, **Ziel** von Δ ist $\mathcal{Z}(\Delta) = \mathcal{Z}(\delta_n)$.

Ableitungen in kontextfreien Grammatiken

- Als Länge einer Ableitung $\Delta = \{ \delta_i \}^n_{i=1}$ ist die Anzahl von nicht-identischen Ableitungsschritten in Δ definiert.
  - Sei Δ eine Ableitung mit $\mathcal{Q}(\Delta) = s$ und $\mathcal{Z}(\Delta) = z$ der Länge ℓ. Dann gilt: $s \stackrel{\ell}{\Rightarrow} z$.
  - Umgekehrt folgt aus $s \stackrel{\ell}{\Rightarrow} z$ die Existenz einer Ableitung der Länge ℓ mit $\mathcal{Q}(\Delta) = s$ und $\mathcal{Z}(\Delta) = z$.

Ableitung für „aaaabbbb“

δ_1 = [ε, S → aSb, ε]

Beispiel einer kontextfreien Grammatik

$\mathcal{G}$-2 = (N, T, P, S) mit

N = { S }

T = { a, b }

P = { S → aSb,

S → ab }

$\mathcal{L}(\mathcal{G}\text{-}2)$ = {ab, aabb, aaabbb, aaaabbbb, ... }

= $a^n b^n$, $n \geq 1$

59

$\mathcal{Q}(\delta_1)$ = εSε , $\mathcal{Z}(\delta_1)$ = εaSbε

Ableitung für „aaaabbbb“

δ_1 = [ε, S $\rightarrow$ aSb, ε]

δ_2 = [a, S $\rightarrow$ aSb, b]

Beispiel einer kontextfreien Grammatik

$\mathcal{G}$-2 = (N, T, P, S) mit

N = { S }

T = { a, b }

P = { S $\rightarrow$ aSb,

S $\rightarrow$ ab }

$\mathcal{L}(\mathcal{G}\text{-}2)$ = {ab, aabb, aaabbb, aaaabbbb, ... }

= $a^n b^n$, $n \geq 1$

$\mathcal{Q}(\delta_1) = \varepsilon S \varepsilon$, $\mathcal{Z}(\delta_1) = \varepsilon aSb \varepsilon$

$\mathcal{Q}(\delta_2) = aSb$, $\mathcal{Z}(\delta_2) = aaSbb$

Ableitung für „aaaabbbb“

δ_1 = [ε, S → aSb, ε]
δ_2 = [a, S → aSb, b]
δ_3 = [aa, S → aSb, bb]

Beispiel einer kontextfreien Grammatik

$\mathcal{G}$-2 = (N, T, P, S) mit
N = { S }
T = { a, b }
P = { S → aSb,
S → ab }
$\mathcal{L}(\mathcal{G}\text{-}2)$ = {ab, aabb, aaabbb, aaaabbbb, ... }
= $a^n b^n$, $n \geq 1$

59

$\mathcal{Q}(\delta_1)$ = εSε , $\mathcal{Z}(\delta_1)$ = εaSbε
$\mathcal{Q}(\delta_2)$ = aSb , $\mathcal{Z}(\delta_2)$ = aaSbb
$\mathcal{Q}(\delta_3)$ = aaSbb , $\mathcal{Z}(\delta_3)$ = aaaSbbb

Ableitung für „aaaabbbb“

δ_1 = [ε, S → aSb, ε]

δ_2 = [a, S → aSb, b]

δ_3 = [aa, S → aSb, bb]

δ_4 = [aaa, S → ab, bbb]

Beispiel einer kontextfreien Grammatik

$\mathcal{G}$-2 = (N, T, P, S) mit

N = { S }

T = { a, b }

P = { S → aSb,

S → ab }

$\mathcal{L}(\mathcal{G}\text{-2})$ = {ab, aabb, aaabbb, aaaabbbb, ... }

= $a^n b^n$, $n \geq 1$

59

$\mathcal{Q}(\delta_1)$ = εSε , $\mathcal{Z}(\delta_1)$ = εaSbε

$\mathcal{Q}(\delta_2)$ = aSb , $\mathcal{Z}(\delta_2)$ = aaSbb

$\mathcal{Q}(\delta_3)$ = aaSbb , $\mathcal{Z}(\delta_3)$ = aaaSbbb

$\mathcal{Q}(\delta_4)$ = aaaSbbb , $\mathcal{Z}(\delta_4)$ = aaaabbbb

Ableitung für „aaaabbbb“

δ_1 = [ε, S $\longrightarrow$ aSb, ε]
δ_2 = [a, S $\longrightarrow$ aSb, b]
δ_3 = [aa, S $\longrightarrow$ aSb, bb]
δ_4 = [aaa, S $\longrightarrow$ ab, bbb]

Beispiel einer kontextfreien Grammatik

$\mathcal{G}$-2 = (N, T, P, S) mit
N = { S }
T = { a, b }
P = { S $\rightarrow$ aSb,
S $\rightarrow$ ab }
$\mathcal{L}(\mathcal{G}\text{-}2)$ = {ab, aabb, aaabbb, aaaabbbb, ... }
= $a^n b^n$, $n \geq 1$

59

$\mathcal{Q}(\Delta) = \mathcal{Q}(\delta_1) = S$,
$\mathcal{Z}(\Delta) = \mathcal{Z}(\delta_4) = aaaabbbb$

$\mathcal{Q}(\delta_1) = \varepsilon S \varepsilon$, $\mathcal{Z}(\delta_1) = \varepsilon aSb\varepsilon$
$\mathcal{Q}(\delta_2) = aSb$, $\mathcal{Z}(\delta_2) = aaSbb$
$\mathcal{Q}(\delta_3) = aaSbb$, $\mathcal{Z}(\delta_3) = aaaSbbb$
$\mathcal{Q}(\delta_4) = aaaSbbb$, $\mathcal{Z}(\delta_4) = aaaabbbb$

Ableitungen in kontextfreien Grammatiken

- Ein Ableitungsschritt $\delta = [\mathit{l}, \mathrm{A} \rightarrow \gamma, \mathit{r}]$ heißt Links- bzw. Rechtsableitungs-schritt, wenn $\mathit{l} \in \mathcal{T}^*$ bzw. $\mathit{r} \in \mathcal{T}^*$.
- Man schreibt $s \Rightarrow_{\mathrm{L}} z$ bzw. $s \Rightarrow_{\mathrm{R}} z$, wenn z aus s durch einen Links- bzw. Rechtsableitungsschritt hervor geht.
  - D.h.: z entsteht aus s durch Ersetzen des in s am weitesten links bzw.rechts stehen-den Nichtterminalsymbols gemäß einer Produktion $\mathrm{A} \rightarrow \gamma$.

Ableitungen in kontextfreien Grammatiken

- Eine Links- bzw. Rechtsableitung ist eine Ableitung, die aus einer endlichen nichtleeren Folge $\Delta = \{ \delta_i \}^n_{i=1}$ von nicht-identischen Ableitungsschritten $\delta_i = [l_i, A_i \longrightarrow \gamma_i\ , r_i]$ besteht mit $l_i \in \mathcal{T}^*$ bzw. $r_i \in \mathcal{T}^*$ für $1 \leq i \leq n$.
- Man schreibt $s \overset{*}{\Rightarrow}_R z$ bzw. $s \overset{l}{\Rightarrow}_R z$, um auszudrücken, dass eine Rechtsableitung von s nach z bzw. eine Rechtsableitung der Länge l von s nach z existiert (analog für Linksableitungen).

Ableitungen in kontextfreien Grammatiken

- **Jede Zeichenfolge s mit S $\overset{*}{\Rightarrow} s$ heißt Satzform; im Fall S $\overset{*}{\Rightarrow}_L s$ bzw. S $\overset{*}{\Rightarrow}_R s$ auch Links- bzw. Rechtssatzform.**

Ableitungen in kontextfreien Grammatiken

- **Sei Π eine Menge von Marken, mit denen die Produktionen P der zugrundeliegenden Grammatik $\mathcal{G}$ eindeutig identifiziert werden können.**

  Eine Folge $\pi = p_1,\ldots,p_n$ mit $p_i \in \Pi$, $1 \leq i \leq q$, heißt Kontrollwort (Parse) einer Ableitung $\Delta = \{ \delta_i \}^n_{i=1}$, wenn für $1 \leq i \leq q$ die im i-ten Ableitungsschritt angewandte Produktion durch die Marke p_i gekennzeichnet ist.

Ableitungen in kontextfreien Grammatiken

- Um auszudrücken, dass π Kontrollwort einer Ableitung von s nach z ist, schreibt man auch $s \overset{\pi}{\Rightarrow} z$.
- Falls $\omega \in \mathcal{L}(\mathcal{G})$ und $S \overset{\pi}{\Rightarrow} \omega$, heißt π auch Kontrollwort (Parse) für ω.
- Analog gibt es ein Linkskontrollwort (Links-Parse) bzw. ein Rechtskontrollwort (Rechts-Parse), wenn die dazugehörige Ableitung eine Links- bzw. Rechtsableitung ist, also: $s \overset{\pi}{\Rightarrow}_L z$ bzw. $s \overset{\pi}{\Rightarrow}_R z$.

Kontrollwort für „aaaabbbb“

Beispiel einer kontextfreien Grammatik

$\mathcal{G}$-2 = (N, T, P, S) mit
N = { S }
T = { a, b }
P = { S ⟶ aSb, 1
S ⟶ ab } 2

$\mathcal{L}(\mathcal{G}\text{-}2)$ = {ab, aabb, aaabbb, aaaabbbb, ... }
= $a^n b^n$, $n \geq 1$

59

$\delta_1 = [\, \varepsilon, S \longrightarrow aSb, \varepsilon \,]$

$\delta_2 = [\, a, S \longrightarrow aSb, b \,]$

$\delta_3 = [\, aa, S \longrightarrow aSb, bb \,]$

$\delta_4 = [\, aaa, S \longrightarrow ab, aaa \,]$

$\mathcal{Q}(\Delta) = \mathcal{Q}(\delta_1) = S$,

$\mathcal{Z}(\Delta) = \mathcal{Z}(\delta_4) = aaaabbbb$

$\pi_{aaaabbbb} = 1\ 1\ 1\ 2$

$S \;{}^{\pi}\!\!\Rightarrow aaaabbbb$

$\mathcal{Q}(\delta_1) = \varepsilon S \varepsilon$, $\mathcal{Z}(\delta_1) = \varepsilon aSb \varepsilon$

$\mathcal{Q}(\delta_2) = aSb$, $\mathcal{Z}(\delta_2) = aaSbb$

$\mathcal{Q}(\delta_3) = aaSbb$, $\mathcal{Z}(\delta_3) = aaaSbbb$

$\mathcal{Q}(\delta_4) = aaaSbbb$, $\mathcal{Z}(\delta_4) = aaaabbbb$

Ableitungen in kontextfreien Grammatiken

- Eine Ableitung $\Delta = \{ \delta_i \}^n_{i=1}$ ist im Allgemeinen durch ihre Quelle und ihr Kontrollwort nicht eindeutig bestimmt.
- Dagegen ist jede Links- bzw. Rechtsableitung durch ihre Quelle und ihr Kontrollwort eindeutig bestimmt. Durch einen Links- bzw. Rechts-Parse für ein Wort $\omega \in \mathcal{L}(\mathcal{G})$ wird also die zugehörige Links- bzw. Rechtsableitung eindeutig beschrieben.

Ableitungen in kontextfreien Grammatiken

Eine wesentliche Aufgabe der Syntax-analyse besteht darin, für Wörter $\omega \in \mathcal{L}(\mathcal{G})$ eine Ableitung $S \overset{*}{\Rightarrow} \omega$ zu bestimmen. Dazu kann die Umkehrung einer Rechtsableitung genutzt werden:

- Sei $\Delta = \{ \delta_i \}^n_{i=1}$ eine Ableitung (Rechts-ableitung). Die aus $\Delta^a = [\delta_1 , \delta_2 ,\ldots, \delta_{n-1} , \delta_n]$ durch Inversion hervorgehende Folge $\Delta^r = [\delta_n , \delta_{n-1} ,\ldots, \delta_2 , \delta_1]$ heißt Reduktion (bzw. Rechtsreduktion oder kanonische Reduktion) von $\mathcal{Z}(\Delta)$ nach $\mathcal{Q}(\Delta)$.

Ableitungen in kontextfreien Grammatiken

- Sei s eine Rechtssatzform (d.h.: S $\overset{*}{\Rightarrow}_R s$). Ein Paar ($i$, p) mit $i \geq 0$ und einer Produktion p $\in$ P heißt **Henkel** von s, wenn

$$S \overset{*}{\Rightarrow}_R \ell A r \Rightarrow_R \ell \gamma r = s$$

mit (1) $i = | \ell |$

(2) $p = A \rightarrow \gamma$

Ableitungen in kontextfreien Grammatiken

- **Henkel einer Rechtssatzform s**
  - Beschreibt den letzten Ableitungsschritt einer Rechtsableitung von s.
  - Ist (i, $A \rightarrow \gamma$) Henkel von s, dann gibt es eine Rechtssatzform von s, deren letzter Ableitungsschritt durch [ℓ, $A \rightarrow \gamma$, r] gegeben ist, wobei ℓ aus den ersten i Symbolen und r aus den letzten $|s|$ - i - $|\gamma|$ Symbolen von s besteht.
  - Die Kenntnis des Henkels ermöglicht somit die Rechtsreduktion $s = \ell\gamma\, r$ auf die Rechtssatzform $\ell A r$
- **Folglich heißt der Rechtsableitungsschritt [ℓ, $A \rightarrow \gamma$, r] auch Rechtsreduktionsschritt (od. kanonischer Reduktionsschritt) zu $s = \ell\gamma\, r$**

Rechtsreduktionen zur Worterkennung

Soll für ein Wort $\omega \in \mathcal{L}(\mathcal{G})$ eine Rechtsreduktion berechnet werden, so erfolgt dies – ausgehend von ω – durch fortgesetzte Bestimmung eines Henkels und Durchführung der entsprechenden Rechtsreduktion, mit der eine Rechtssatzform aufgebaut wird, bis die Rechtssatzform „S“ aufgebaut ist.

Algorithmus für Rechtsreduktion (CFG-Rechtserkennung)

Funktion CFG-Rechtsreduktion(↓wort,↓G) = „accept“ oder **„reject“**

`rsatzform` $\Leftarrow$ wort
LOOP
IF ein Henkel (i, **A** $\rightarrow \gamma$) kann unter Verwendung einer Produktion (**A** $\rightarrow \gamma \in$ **P**) aus **G** für die `rsatzform` $l \gamma r$ konstruiert werden
THEN führe Rechtsreduktion mit dem gefundenen Henkel auf `rsatzform` aus, so dass `rsatzform` $\Leftarrow l$ **A** r
ELSE return **„reject“**
IF `rsatzform` = S
THEN return **„accept“**
LOOPEND

Reduktion von „aaaabbbb“

Beispiel einer kontextfreien Grammatik

$\mathcal{G}$-2 = (N, T, P, S) mit

N = { S }

T = { a, b }

P = { S $\rightarrow$ aSb,

S $\rightarrow$ ab }

$\mathcal{L}(\mathcal{G}\text{-}2)$ = {ab, aabb, aaabbb, aaaabbbb, ... }

= $a^n b^n$, $n \geq 1$

59

Sei aaaabbbb Rechtssatzform und Wort von $\mathcal{L}(\mathcal{G}\text{-}2)$

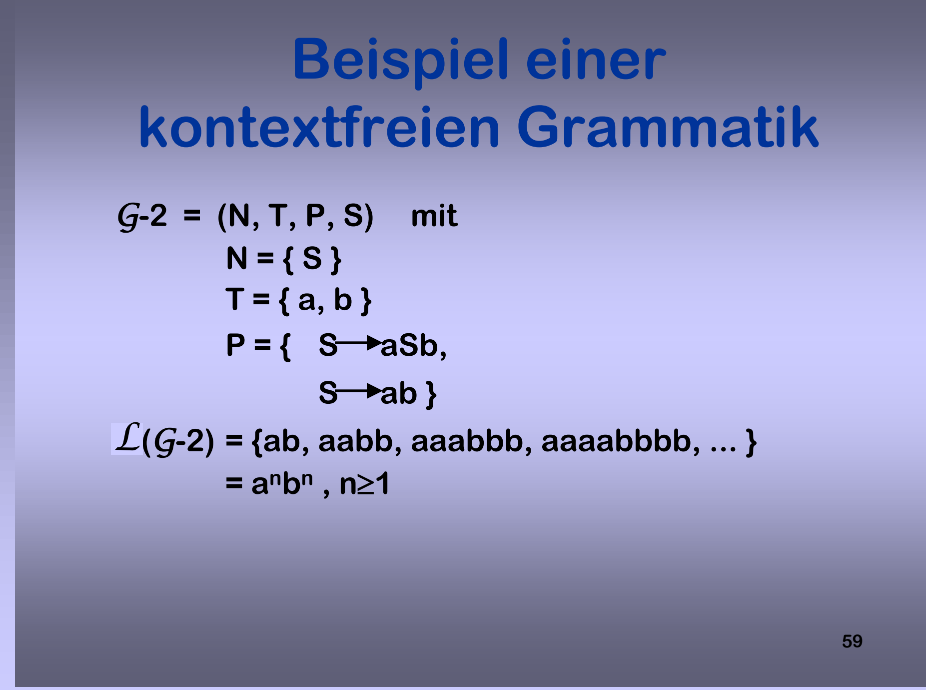
(3, S $\longrightarrow$ ab) ist Henkel für letzten Ableitungsschritt

$$\delta_4 = [\ \boxed{\text{aaa,}}\ S \longrightarrow ab,\ \boxed{\text{bbb}}\ ]$$

$|\ell| = 3$ $\quad$ $|r| = |s| - i - |\gamma| = 8\text{-}3\text{-}2 = 3$

Dann ist aaaSbbb neue Rechtssatzform.

Reduktion von „aaaabbbb“



aaaSbbb ist Rechtssatzform.

(2, S → aSb) ist Henkel für letzten Ableitungsschritt

$$\delta_3 = [\ \boxed{aa,}\ S \rightarrow aSb,\ \boxed{bb}\ ]$$

$|\ell| = 2$ $\qquad |r| = |s| - i - |\gamma| = 7\text{-}2\text{-}3 = 2$

Dann ist aaSbb neue Rechtssatzform.

Reduktion von „aaaabbbb“

aaSbb ist Rechtssatzform.

Beispiel einer kontextfreien Grammatik

$\mathcal{G}$-2 = (N, T, P, S) mit

N = { S }

T = { a, b }

P = { $S \rightarrow aSb$,

$S \rightarrow ab$ }

$\mathcal{L}(\mathcal{G}\text{-}2)$ = {ab, aabb, aaabbb, aaaabbbb, ... }

= $a^n b^n$, $n \geq 1$

59

(1, S → aSb) ist Henkel für letzten Ableitungsschritt

$$\delta_2 = [\ \boxed{a,}\ S \rightarrow aSb,\ \boxed{b}\ ]$$

$|\ell| = 1$

$|r| = |s| - i - |\gamma| = 5\text{-}1\text{-}3 = 1$

Dann ist aSb neue Rechtssatzform.

Reduktion von „aaaabbbb“

Beispiel einer kontextfreien Grammatik

$\mathcal{G}$-2 = (N, T, P, S) mit
N = { S }
T = { a, b }
P = { $S \rightarrow aSb$,
$S \rightarrow ab$ }
$\mathcal{L}(\mathcal{G}\text{-}2)$ = {ab, aabb, aaabbb, aaaabbbb, ... }
= $a^n b^n$, $n \geq 1$

59

aSb ist Rechtssatzform.

(0, S → aSb) ist Henkel für letzten Ableitungsschritt

$$\delta_1 = [\boxed{\varepsilon,}\ S \rightarrow aSb, \boxed{\varepsilon}\ ]$$

$|\mathcal{l}| = 0$ $\quad$ $|\mathcal{r}| = |\mathcal{s}| - \mathcal{i} - |\gamma| = 3\text{-}0\text{-}3 = 0$

Dann ist S neue Rechtssatzform.

Ausblick: Parsing