

Computerlinguistik I

Vorlesung im WiSe 2016/17
(M-GSW-09)

Prof. Dr. Udo Hahn

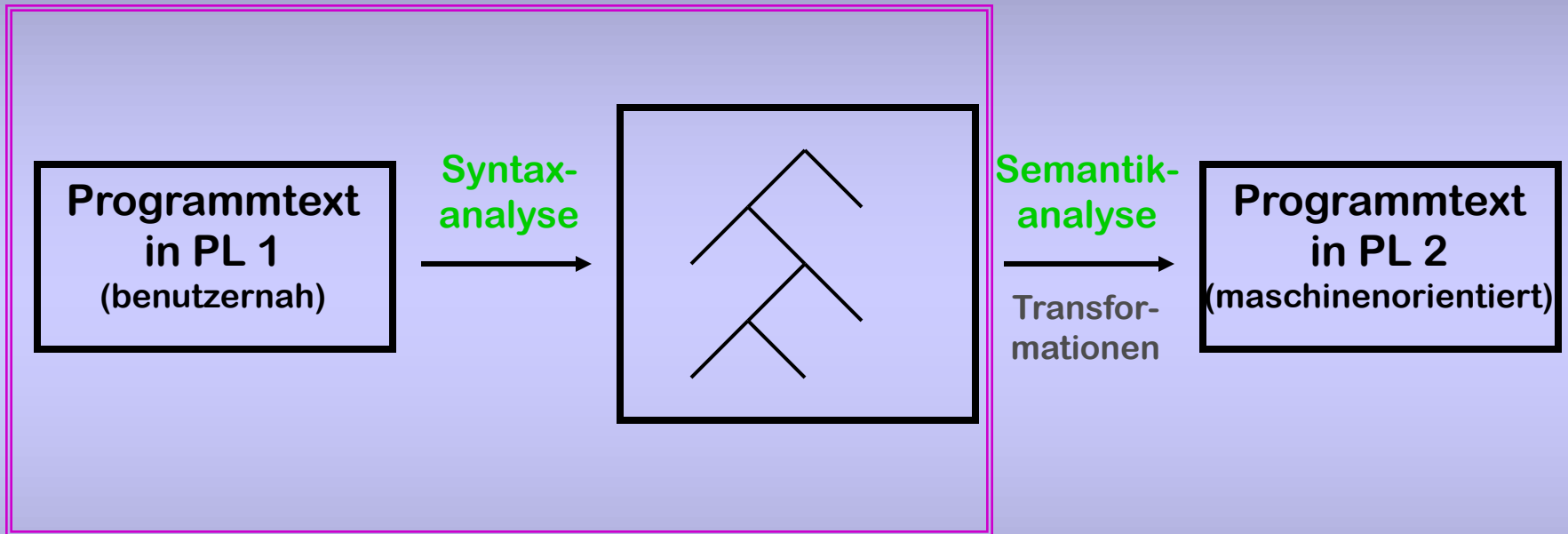
Lehrstuhl für Computerlinguistik
Institut für Germanistische Sprachwissenschaft
Friedrich-Schiller-Universität Jena

<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 **formalen** Sprache (z.B. einer Programmiersprache)

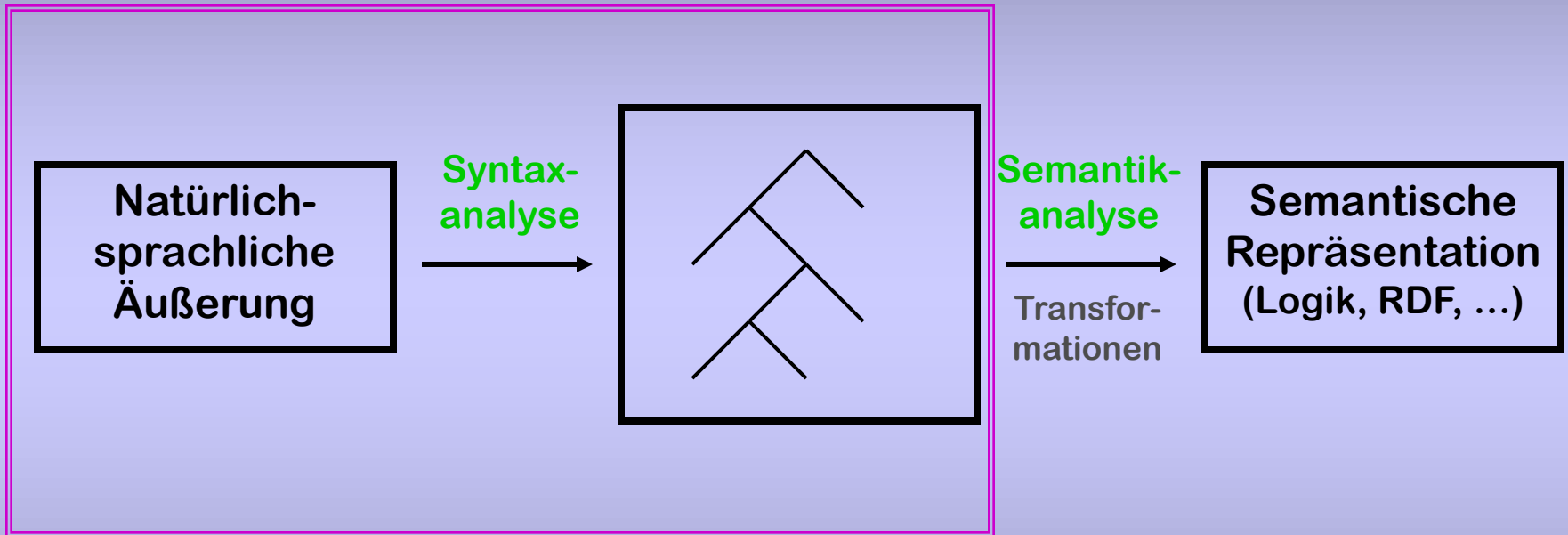
Analyse von Programmen



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

Grundlagen formaler Sprachen: Wörter

- Seien **Wörter** (**Sätze**, **Strings**, **Ketten**) über einem Alphabet Σ in der folgenden Weise definiert:
 1. ε ist ein Wort über Σ (ε ist das Leerwort, das keine Symbole hat)
 2. falls χ ein Wort über Σ und $\alpha \in \Sigma$ ist, dann ist $\chi \alpha$ ein Wort über Σ
 3. γ ist ein Wort über Σ genau dann, wenn sein Bildung aus (1) oder (2) folgt

Formale Sprache

- Eine (**formale**) **Sprache** \mathcal{L} (oder **Wortmenge**) über einem Alphabet Σ ist eine Menge von Wörtern über Σ :

$$\mathcal{L} := \{ \omega \mid \omega \text{ ist Wort über } \Sigma \}$$

- Sei ferner Σ^* (bzw. Σ^+) die Menge *aller* Wörter über Σ unter Einschluss (bzw. Ausschluss) von ε .
- Dann gilt für jede Sprache \mathcal{L} über Σ :

$$\mathcal{L} \subseteq \Sigma^*$$

Grundlagen formaler Sprachen

- Seien ω , τ , α und γ Wörter über Σ mit $\omega = \tau \alpha \gamma$. α heißt **Teilwort (Substring)** von ω , im Falle $\tau = \varepsilon$ auch **Präfix**, im Falle $\gamma = \varepsilon$ auch **Suffix** von ω .

Beispiele:

- **tst** ist Teilwort in **wortstamm**
- **wort** ist Präfix in **wortstamm**
- **mm** ist Suffix in **wortstamm**

Grundlagen formaler Sprachen

- Als **Länge** eines Wortes ω definiert man die Anzahl der Symbole in ω und bezeichnet diese mit $|\omega|$. Es gilt:
 - $|\varepsilon| = 0$
 - $|\sigma| = 1$ für alle $\sigma \in \Sigma$
 - $|\omega\tau| = |\omega| + |\tau|$ für alle Wörter ω, τ über Σ .

Beispiele:

- $|\text{ABCX}| = 4$
- $|\text{wort o stamm}| = |\text{wort}| + |\text{stamm}| = 4 + 5 = 9$

Grundbegriffe zu formalen Sprachen

- Eine **Wortmenge (Sprache)** \mathcal{F} bezüglich eines Alphabets Σ ist gegeben durch

$$\mathcal{F} := \{ \omega \mid \omega \text{ ist Wort über } \Sigma \}$$

- Das **Spiegelbild** eines **Wortes** $\omega := \omega_1 \dots \omega_m$ ($\omega_i \in \Sigma$) ist das Wort $\omega^r := \omega_m \dots \omega_1$, das **Spiegelbild** einer **Wortmenge** \mathcal{F} ist

$$\mathcal{F}^r := \{ \omega^r \mid \omega \in \mathcal{F} \}.$$

Beispiele für Spiegelbilder:

- $2424^r = 4242$, $\text{wort}^r = \text{trow}$, $a^r = a$
- $\text{MIR}^r = \{2424^r, \text{wort}^r, a^r\}$
= $\{4242, \text{trow}, a\}$

Grundbegriffe zu formalen Sprachen

- Die **Zusammensetzung (Konkatenation)** von **Wortmengen** \mathcal{F} und \mathcal{M} ist gegeben durch

$$\mathcal{F} \circ \mathcal{M} := \mathcal{F} \mathcal{M} := \{ \omega\tau \mid \omega \in \mathcal{F}, \tau \in \mathcal{M} \}$$

- Dabei gilt:

$$\mathcal{F} \{ \varepsilon \} = \{ \varepsilon \} \mathcal{F} = \mathcal{F}$$

$$\mathcal{F} \emptyset = \emptyset \mathcal{F} = \emptyset$$

Beispiele für Konkatenationen:

- $\text{DLex} := \{\text{Fisch, Fest, Fleck}\}$; $\text{DEnd} := \{\text{e, es, er}\}$
- $\text{DLex} \circ \text{DEnd} = \{\text{Fische, Fisches, Fischer, Feste, Festes, Fester, Flecke, Fleckes, Flecker}\}$

Grundbegriffe zu formalen Sprachen

- **Potenzen** von **Wörtern** ω bzw. **Wortmengen** \mathcal{F} sind gegeben durch:

$$- \omega^0 := \varepsilon \quad \omega^1 := \omega \quad \omega^i := \omega^{i-1} \omega, \text{ für } i \geq 1$$

$$- \mathcal{F}^0 := \{\varepsilon\} \quad \mathcal{F}^1 := \mathcal{F} \quad \mathcal{F}^i := \mathcal{F}^{i-1} \mathcal{F}, \text{ für } i \geq 1$$

Beispiele für Potenzen von Wortmengen:

- **DSilbe** := {ba, di, ko}

- **DSilbe**⁰ = { ε }, **DSilbe**¹ = {ba, di, ko}

- **DSilbe**² = **DSilbe**¹ **DSilbe**

$$= \{ \text{baba, badi, bako, diba, didi, diko, koba, kodi, koko} \}$$

- **DSilbe**³ = **DSilbe**² **DSilbe**

Grundbegriffe zu formalen Sprachen

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

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

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

- 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** G ist ein 4-tupel

$$G = (N, T, P, S)$$

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

Grundbegriffe zu formalen Grammatiken

- Man schreibt $s \stackrel{n}{\Rightarrow} z$, um auszudrücken, dass s_0, s_1, \dots, s_n existieren mit
$$s = s_0, s_i \Rightarrow s_{i+1} \text{ für } 0 \leq i < n \text{ und } s_n = z$$
- Damit ist also
$$s \stackrel{+}{\Rightarrow} z \text{ g.d.w. } s \stackrel{n}{\Rightarrow} z \text{ für ein } n \geq 1 \quad \text{und}$$
$$s \stackrel{*}{\Rightarrow} z \text{ g.d.w. entweder } s = z \text{ oder } s \stackrel{+}{\Rightarrow} z$$

Grundbegriffe zu formalen Grammatiken

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

$$\mathcal{L}(G) := \{ \tau \mid \tau \in T^*, S \stackrel{*}{\Rightarrow} \tau,$$

$S \text{ ist Startsymbol von } G \}$

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

Grundbegriffe zu formalen Grammatiken

- Zwei formale Grammatiken G_1 und G_2 , $G_1 \neq G_2$, heißen **äquivalent**, wenn sie dieselbe Sprache erzeugen, d.h.:

$$\mathcal{L}(G_1) = \mathcal{L}(G_2)$$

Grundbegriffe zu formalen Grammatiken

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

$$\alpha \rightarrow \gamma$$

mit

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

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

Grundbegriffe zu formalen Grammatiken

- Eine Grammatik 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)₁

Grundbegriffe zu formalen Grammatiken

- Eine Grammatik G heißt **Typ-2-Grammatik** (**kontextfreie Grammatik**), wenn P nur Produktionen enthält der Gestalt

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

Grundbegriffe zu formalen Grammatiken

- Eine Grammatik G heißt **Typ-3-Grammatik** (**reguläre Grammatik**), wenn P nur Produktionen der Gestalt

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

(sog. **links**lineare Produktionen) oder nur Produktionen der Gestalt

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

(sog. **rechts**lineare Produktionen) enthält.

- Man spricht dann auch entsprechend von **linkslinaren** bzw. **rechtslinaren Grammatiken**.
- Eine reguläre Grammatik darf nicht Regeln nach beiden Produktionsregelmustern mischen.

Beispiel einer rechtslinearen Grammatik

$G-3 = (N, T, P, S)$ mit

$$N = \{ S, A, B \}$$

$$T = \{ a, b \}$$

$$P = \{ S \rightarrow aA,$$

$$A \rightarrow aA,$$

$$A \rightarrow bbB,$$

$$B \rightarrow bB,$$

$$B \rightarrow b \}$$

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

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

Beispiel einer rechtslinearen Grammatik

$G-3 = (N, T, P, S)$ mit

$N = \{ S, A, B \}$

$T = \{ a, b \}$

$P = \{ S \rightarrow aA,$

$A \rightarrow aA,$

$A \rightarrow bbB,$

$B \rightarrow bB,$

$B \rightarrow b \}$

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

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

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

Beispiel einer kontextfreien Grammatik

$G-2 = (N, T, P, S)$ mit

$$N = \{ S \}$$

$$T = \{ a, b \}$$

$$P = \{ S \rightarrow aSb, \\ S \rightarrow ab \}$$

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

Beispiel einer kontextfreien Grammatik

$G-2 = (N, T, P, S)$ mit

$$N = \{ S \}$$

$$T = \{ a, b \}$$

$$P = \{ S \rightarrow aSb, \\ S \rightarrow ab \}$$

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

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

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

Beispiel einer kontextfreien Grammatik

$G-2 = (N, T, P, S)$ mit

$$N = \{ S \}$$

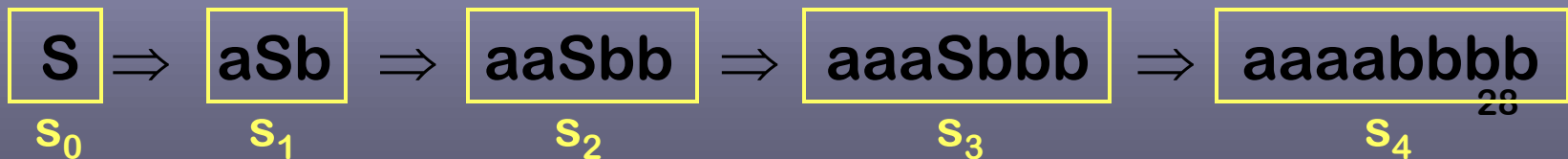
$$T = \{ a, b \}$$

$$P = \{ S \rightarrow aSb, \\ S \rightarrow ab \}$$

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

$$S \stackrel{4}{\Rightarrow} aaaabbbb$$

$$S^* \Rightarrow aaaabbbb$$



Beispiel einer kontextsensitiven Grammatik

$G-1 = (N, T, P, S)$ mit

$N = \{ S, B, C, X \}$

$T = \{ a, b, c \}$

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

$\mathcal{L}(G-1) = \{ abc, aabbcc, aaabbbccc, \dots \}$
 $= a^n b^n c^n, n \geq 1$

Beispiel einer kontextsensitiven Grammatik

$G-1 = (N, T, P, S)$ mit

$N = \{ S, B, C, X \}$

$T = \{ a, b, c \}$

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

$S^* \Rightarrow aaabbbccc$

Beispiel einer kontextsensitiven Grammatik

$G-1 = (N, T, P, S)$ mit

$N = \{ S, B, C, X \}$

$T = \{ a, b, c \}$

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

$S \Rightarrow aSBC$

Beispiel einer kontextsensitiven Grammatik

$G-1 = (N, T, P, S)$ mit

$N = \{ S, B, C, X \}$

$T = \{ a, b, c \}$

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

$S \Rightarrow aSBC \Rightarrow aaSBCBC$

Beispiel einer kontextsensitiven Grammatik

$G-1 = (N, T, P, S)$ mit

$N = \{ S, B, C, X \}$

$T = \{ a, b, c \}$

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

$S \Rightarrow aSBC \Rightarrow aaSBCBC \Rightarrow aaBCBCBC$

Beispiel einer kontextsensitiven Grammatik

$G-1 = (N, T, P, S)$ mit

$N = \{ S, B, C, X \}$

$T = \{ a, b, c \}$

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

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

Beispiel einer kontextsensitiven Grammatik

$G-1 = (N, T, P, S)$ mit

$N = \{ S, B, C, X \}$

$T = \{ a, b, c \}$

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

$CB \rightarrow XB, XB \rightarrow XC, XC \rightarrow BC,$

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

$S \Rightarrow aSBC \Rightarrow aaSBCBC \Rightarrow aaaBCBCBC \Rightarrow$
 $\dots \Rightarrow aaaBBCBC \Rightarrow \dots \Rightarrow aaaBBCBC$

Beispiel einer kontextsensitiven Grammatik

$G-1 = (N, T, P, S)$ mit

$N = \{ S, B, C, X \}$

$T = \{ a, b, c \}$

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

$CB \rightarrow XB, XB \rightarrow XC, XC \rightarrow BC,$

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

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

$\dots \Rightarrow aaaBBCBC \Rightarrow \dots \Rightarrow aaaBBBCBC \Rightarrow$

$\dots \Rightarrow aaaBBBCCC$

Beispiel einer kontextsensitiven Grammatik

$G-1 = (N, T, P, S)$ mit

$N = \{ S, B, C, X \}$

$T = \{ a, b, c \}$

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

$CB \rightarrow XB, XB \rightarrow XC, XC \rightarrow BC,$

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

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

$\dots \Rightarrow aaaBBCCBC \Rightarrow \dots \Rightarrow aaaBBCCBC \Rightarrow$

$\dots \Rightarrow aa**aB**BBCCC \Rightarrow aa**ab**BBCCC$

Beispiel einer kontextsensitiven Grammatik

$G-1 = (N, T, P, S)$ mit

$N = \{ S, B, C, X \}$

$T = \{ a, b, c \}$

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

$S \Rightarrow aSBC \Rightarrow aaSBCBC \Rightarrow aaaBCBCBC \Rightarrow$
 $\dots \Rightarrow aaaBBCCBC \Rightarrow \dots \Rightarrow aaaBBCCBC \Rightarrow$
 $\dots \Rightarrow aaaBBBCCC \Rightarrow aaa\mathbf{bB}BCCC \Rightarrow$
 $aaa\mathbf{bb}BCCC$

Beispiel einer kontextsensitiven Grammatik

$G-1 = (N, T, P, S)$ mit

$N = \{ S, B, C, X \}$

$T = \{ a, b, c \}$

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

$S \Rightarrow aSBC \Rightarrow aaSBCBC \Rightarrow aaaBCBCBC \Rightarrow$
 $\dots \Rightarrow aaaBBCCBC \Rightarrow \dots \Rightarrow aaaBBCCBC \Rightarrow$
 $\dots \Rightarrow aaaBBBCCC \Rightarrow aaabBBCCC \Rightarrow$
 $aaab**b**BCCC \Rightarrow aaab**bb**CCC$

Beispiel einer kontextsensitiven Grammatik

$G-1 = (N, T, P, S)$ mit

$N = \{ S, B, C, X \}$

$T = \{ a, b, c \}$

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

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

Beispiel einer kontextsensitiven Grammatik

$G-1 = (N, T, P, S)$ mit

$N = \{ S, B, C, X \}$

$T = \{ a, b, c \}$

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

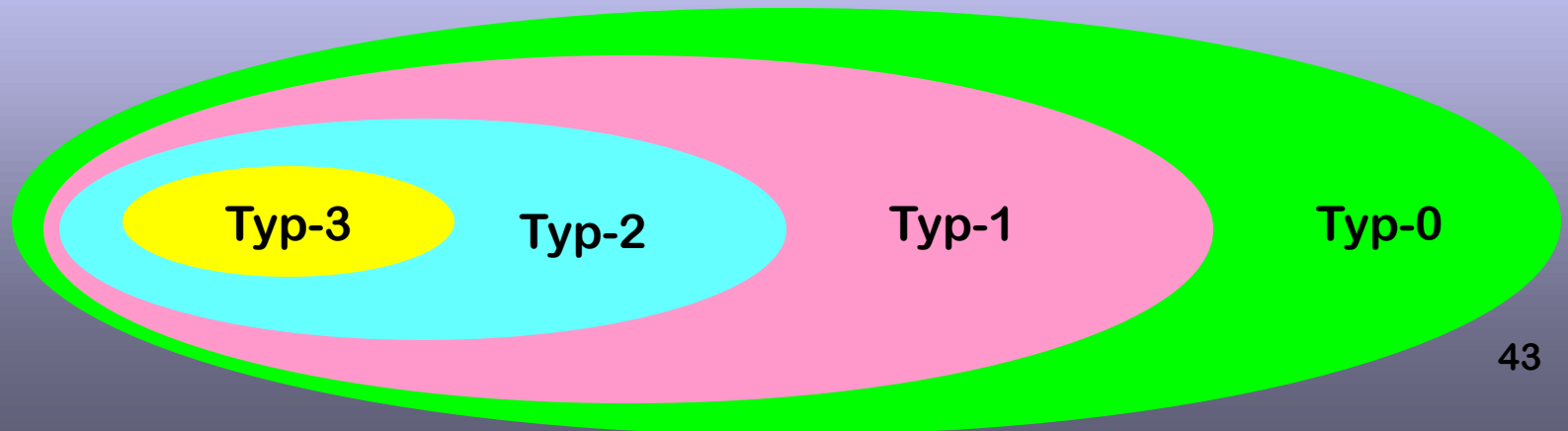
$S \Rightarrow aSBC \Rightarrow aaSBCBC \Rightarrow aaaBCBCBC \Rightarrow$
 $\dots \Rightarrow aaaBBCBC \Rightarrow \dots \Rightarrow aaaBBCBC \Rightarrow$
 $\dots \Rightarrow aaaBBBCCC \Rightarrow aaabBBCCC \Rightarrow$
 $aaabbBCCC \Rightarrow aaabbbCCC \Rightarrow \dots \text{aaabbbccc}$

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)
 - Beweisbar durch Pumping-Theorem und die Tatsache, dass reguläre Sprachen unter Mengenschnitt 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 ($V_{\text{transitiv}}$)ⁿ⁻¹ („waited“ [$V_{\text{intransitiv}}$])¹

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

Beweisidee (Abschluss unter Schnitt):

Englisch_{reg} [~~a~~ 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 aⁿ b^m x cⁿ d^m y**

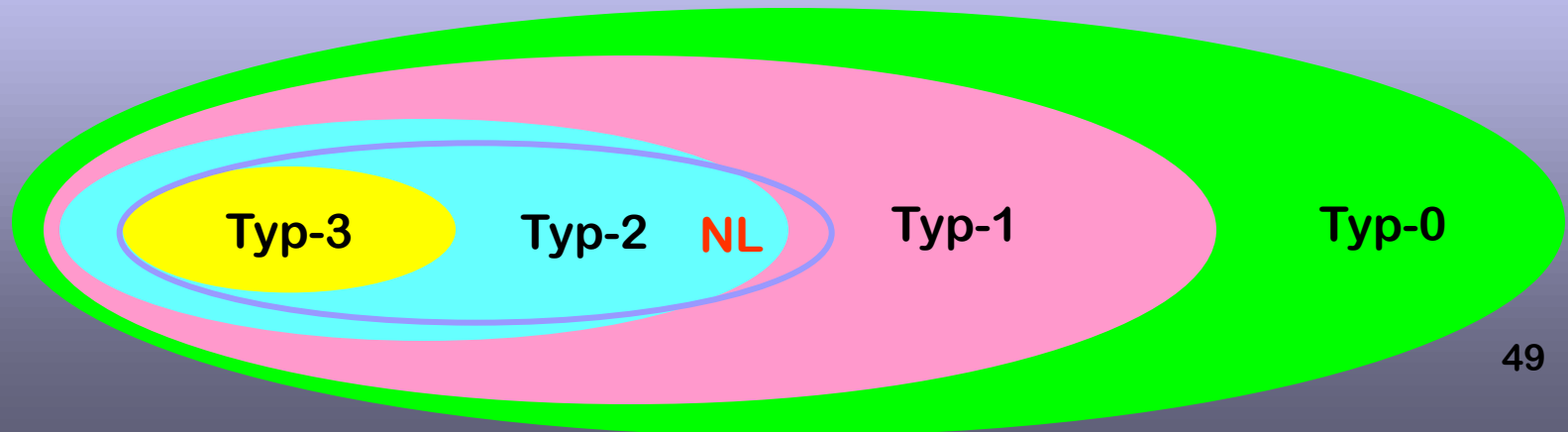
Beweisidee (Abschluss unter Schnitt):

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 aⁿ b^m x cⁿ 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



Kostenrechnung für det. FSA- Erkennungsalgorithmus

Funktion D-Erkennen(\downarrow Band, \downarrow FSA) = „accept“ oder „reject“

Index \Leftarrow Bandanfang

AktualZustand \Leftarrow 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 \Leftarrow Zustandstransitionstabelle[AktualZustand, Band(Index)]

Index \Leftarrow Index + 1

LOOPEND

1

1

n

1







1

1

1

1

Komplexitätsklassen

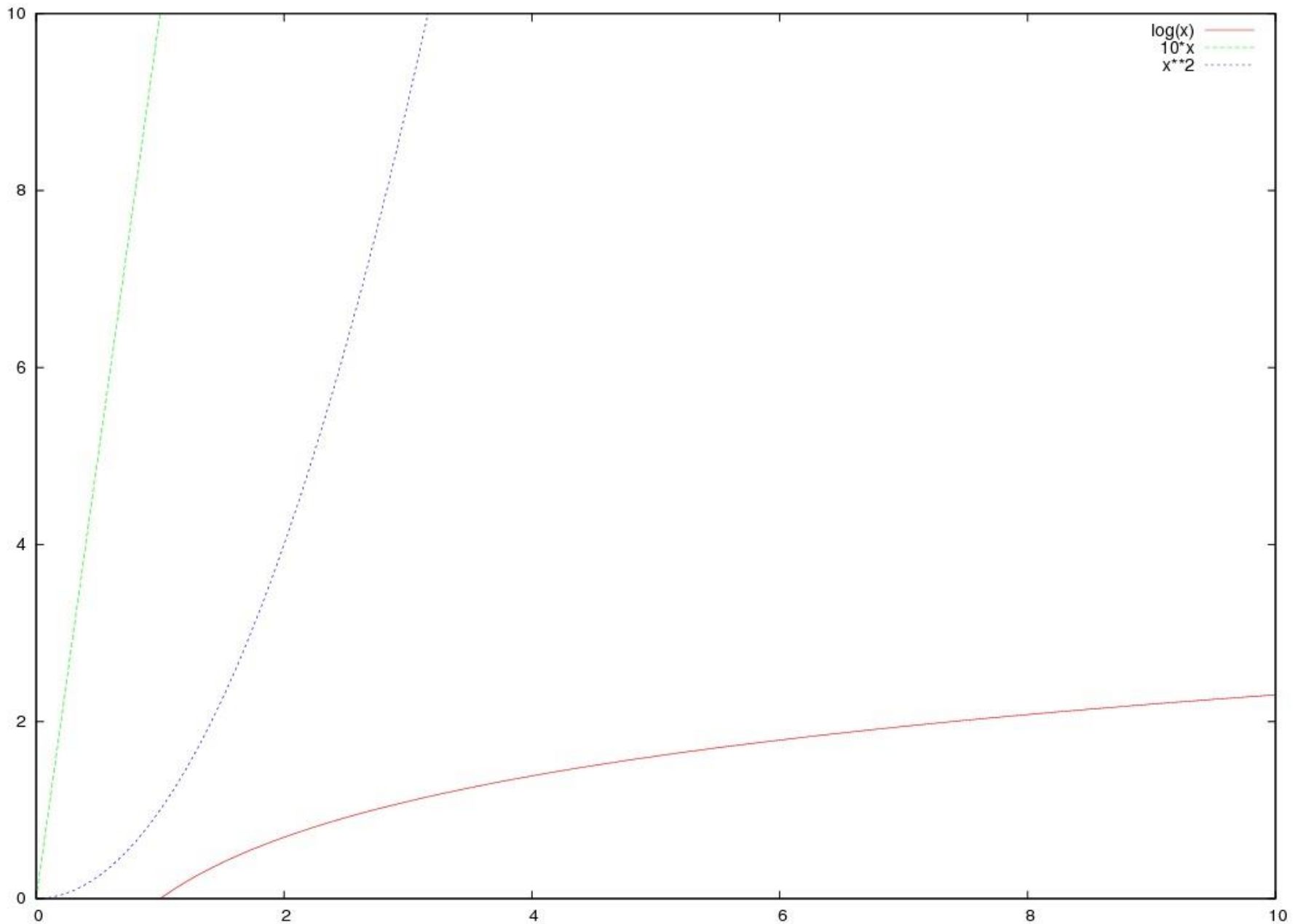
- $O(1)$ konstant 
- $O(\log n)$ logarithmisch 
- $O(n)$ linear 
- $O(n^k)$ polynomial ($k \in [2,4]$) 
- $O(n^k)$ polynomial ($k > 4$) 
- $O(k^n)$ exponentiell 

– wobei n die Problemgröße ist

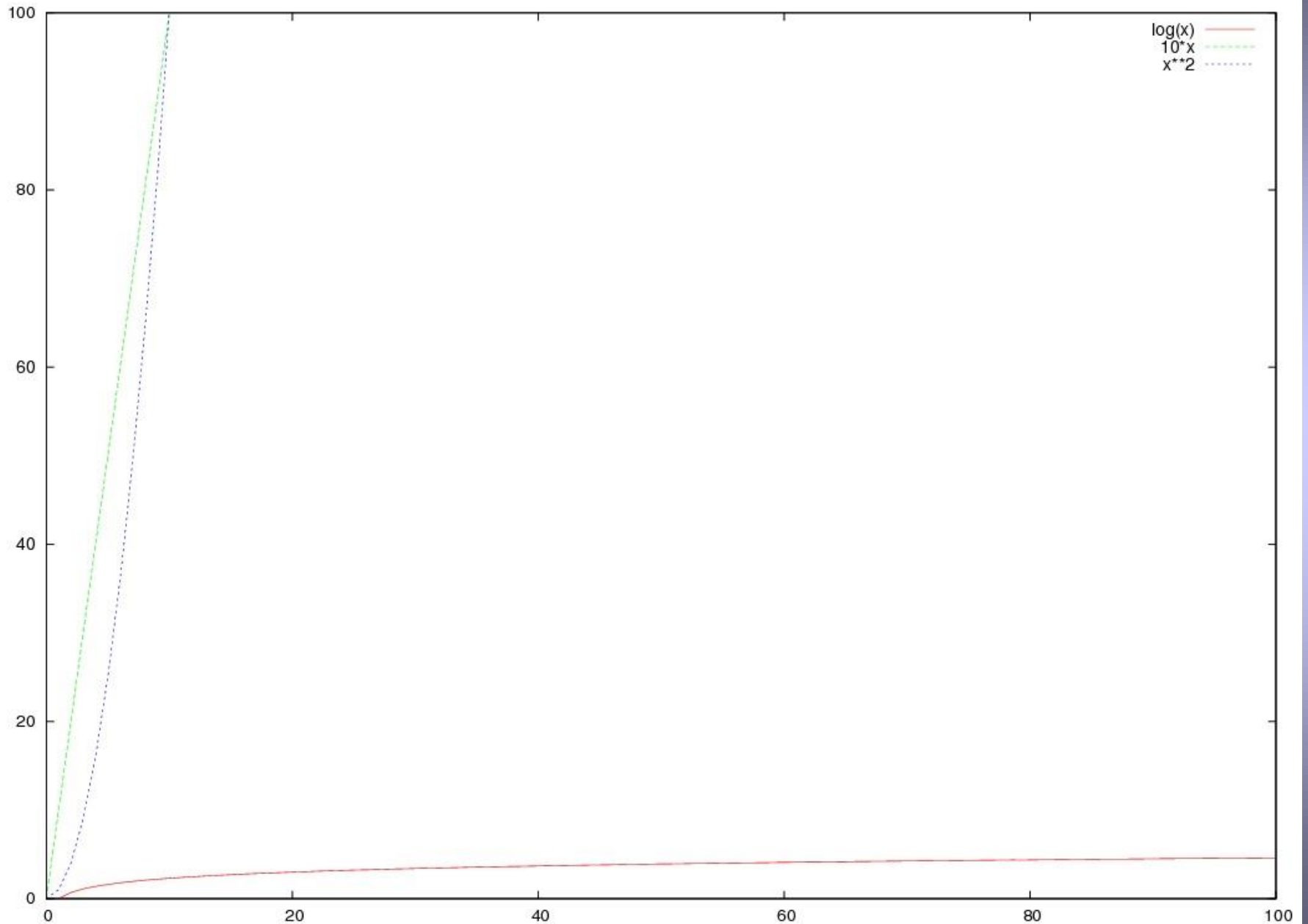
Charakteristische Problemgrößen beim Parsing

- n ist die Länge der Eingabe (Wort, Satz)
- n ist die Kardinalität der Produktionsregelmenge
- Hinweis: n ist vom (Parsing-)Algorithmus abhängig

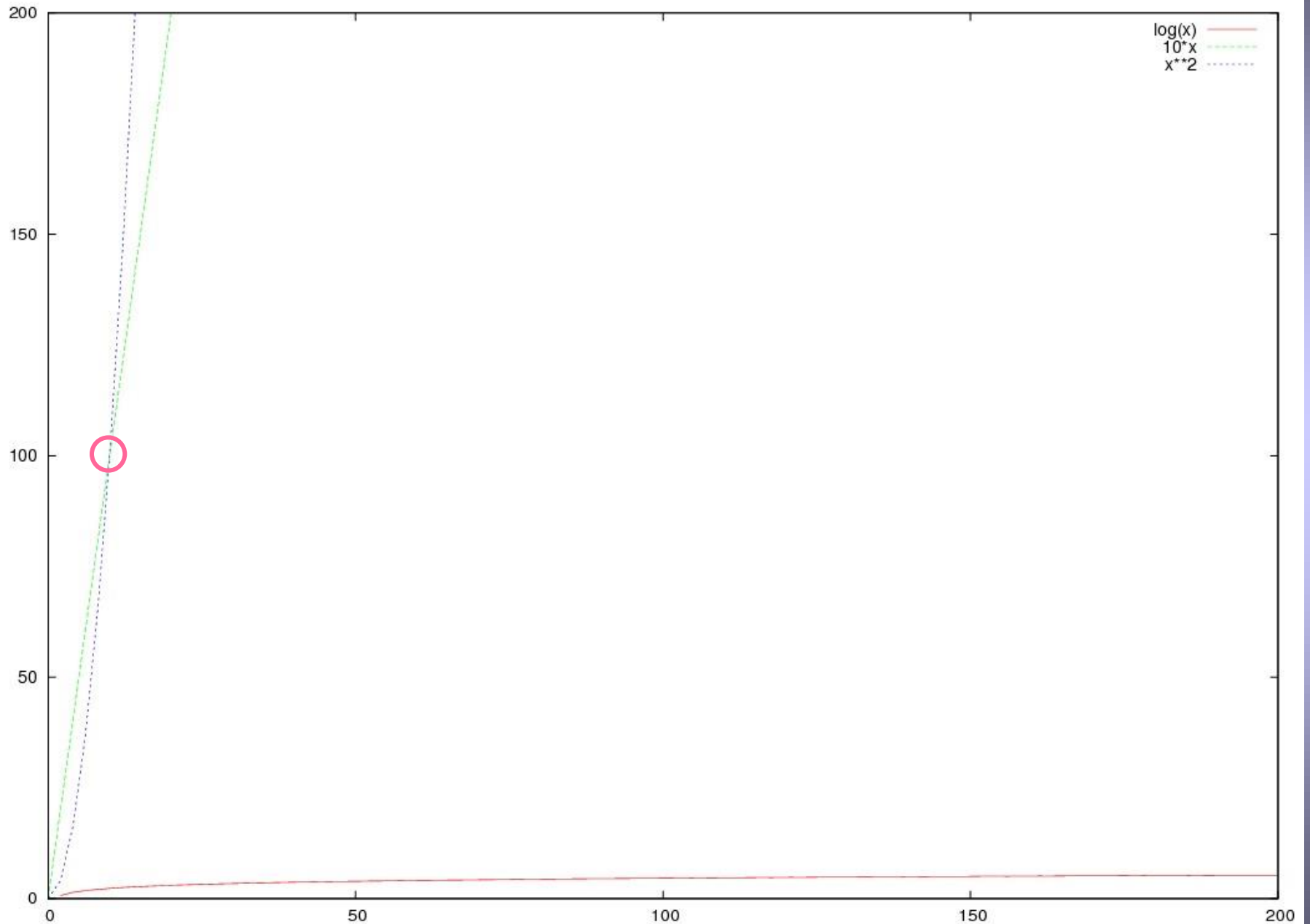
(Lauf-)Zeit-Klassen



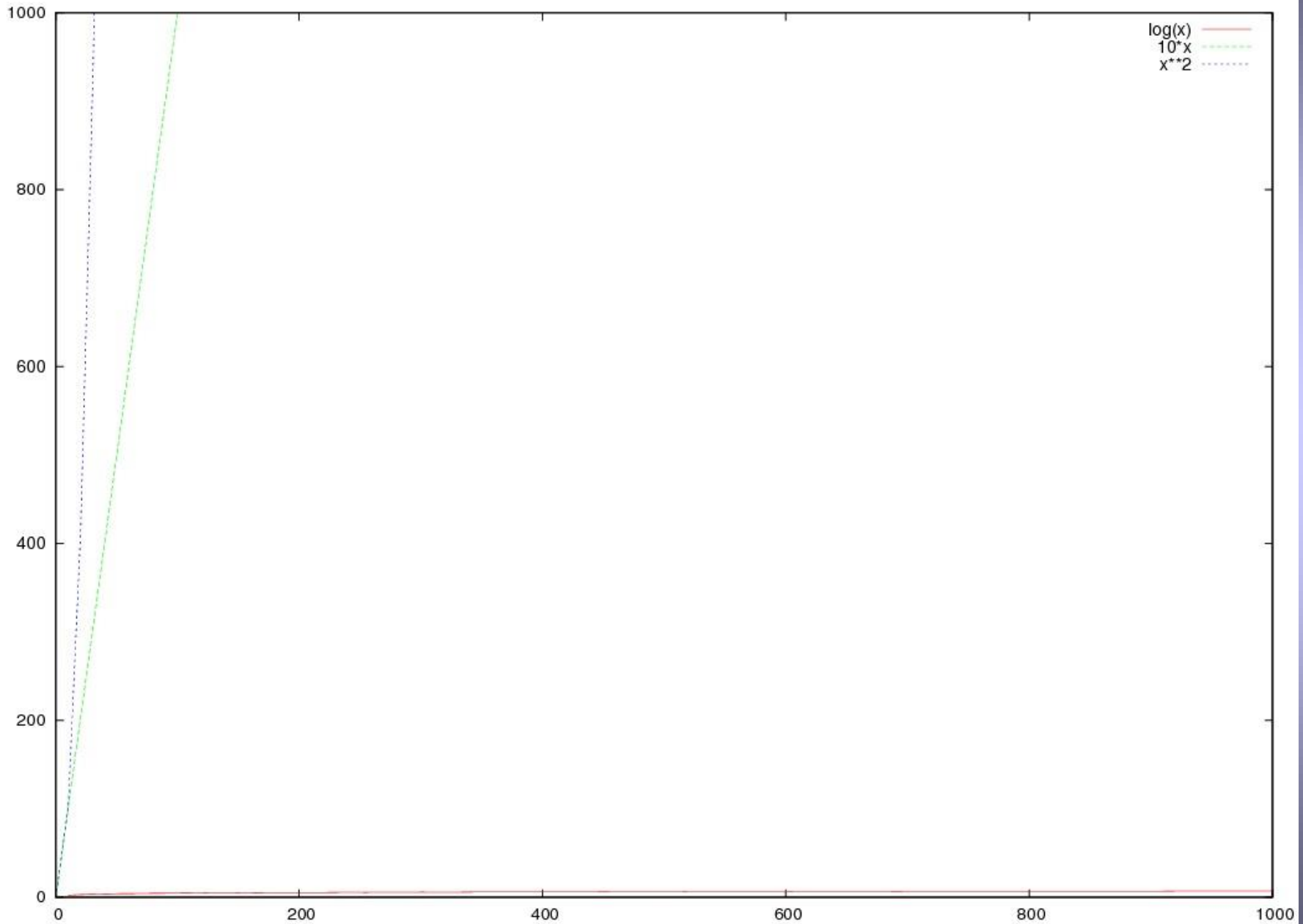
(Lauf-)Zeit-Klassen



(Lauf-)Zeit-Klassen



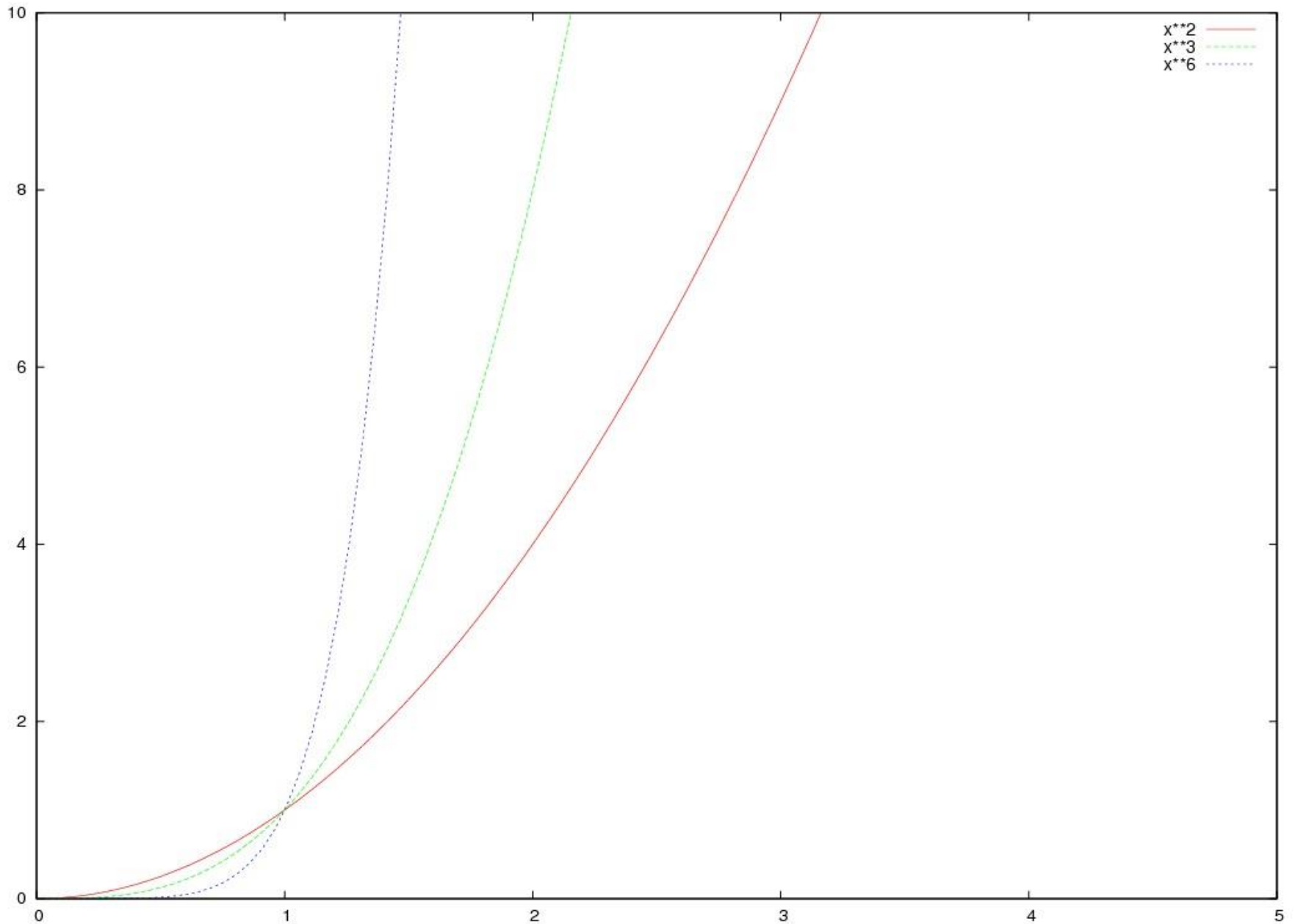
(Lauf-)Zeit-Klassen



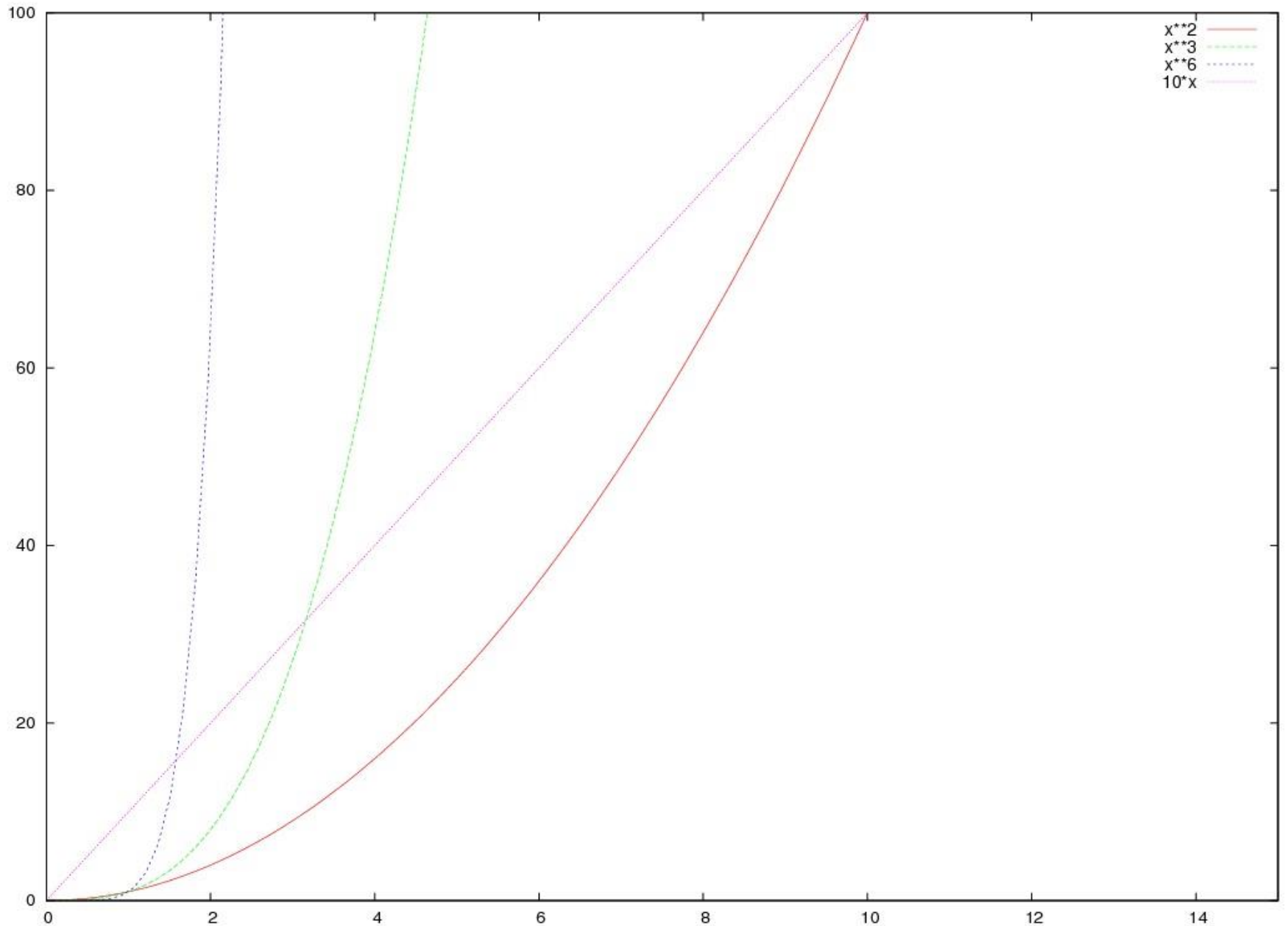
Implikationen für die automatische Sprachanalyse

- 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)

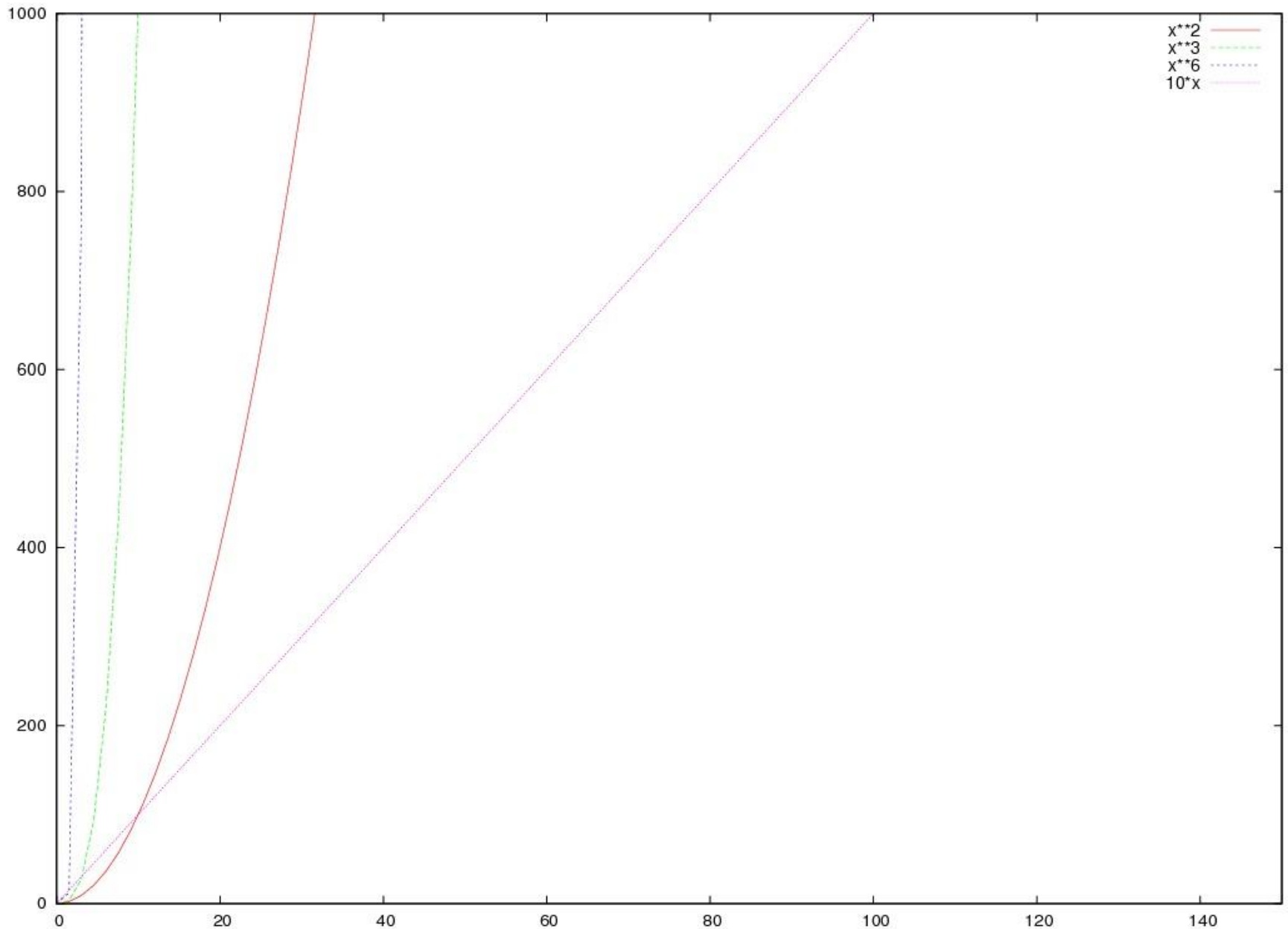
(Lauf-)Zeit-Klassen



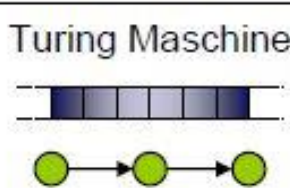
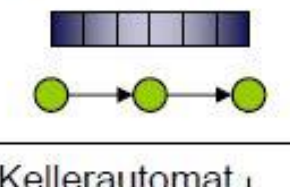


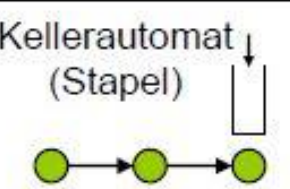





(Lauf-)Zeit-Klassen



(Lauf-)Zeit-Klassen



Chomsky-Hierarchie formaler Sprachen und Automaten

| Sprache | Automat | Grammatik | Erkennung | Abhängigkeit |
|------------------------|---|---|---|---|
| rekursiv aufzählbar | Turing Maschine  | unbeschränkt $Baa \rightarrow s$ | 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  |