# Mathematische Grundlagen der Computerlinguistik

## Eigenschaften formaler Sprachen

Michael Staniek

ICL, Universität Heidelberg, SoSe 2019

22.05.2019

# Inhalte der heutigen Vorlesung

- Eigenschaften kontextfreier Sprachen
- Die Chomsky-Hierarchie der Sprachen
- Komplexität von Sprachen
  - Das Wortproblem für reguläre und kontextfreie Sprachen
  - Das asymptotische Verhalten von Funktionen
- Das Pumping-Lemma
- Formale Sprachen und natürliche Sprachen

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

- Letzte Sitzung:
  - NEAs/ DEAs/ reguläre Grammatiken definieren reguläre Sprachen

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

- Letzte Sitzung:
  - NEAs/ DEAs/ reguläre Grammatiken definieren reguläre Sprachen
    - lösen *Wortproblem* für reguläre Sprachen

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

- Letzte Sitzung:
  - NEAs/ DEAs/ reguläre Grammatiken definieren reguläre Sprachen
    - lösen *Wortproblem* für reguläre Sprachen
  - Kontextfreie Grammatiken definieren kontextfreie Sprachen

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

- Letzte Sitzung:
  - NEAs/ DEAs/ reguläre Grammatiken definieren reguläre Sprachen
    - lösen *Wortproblem* für reguläre Sprachen
  - Kontextfreie Grammatiken definieren kontextfreie Sprachen
  - Erlaubte Regelformen in beiden Grammatiktypen

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

- Letzte Sitzung:
  - NEAs/ DEAs/ reguläre Grammatiken definieren reguläre Sprachen
    - lösen *Wortproblem* für reguläre Sprachen
  - Kontextfreie Grammatiken definieren kontextfreie Sprachen
  - Erlaubte Regelformen in beiden Grammatiktypen
  - Mehrdeutigkeit

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

## Wortproblem

Sei $L$ eine Sprache. Das Wortproblem für $L$ besteht darin, festzustellen, ob ein beliebiges Wort $w$ endlicher Länge in $L$ enthalten ist oder nicht.

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

## Wortproblem

Sei $L$ eine Sprache. Das Wortproblem für $L$ besteht darin, festzustellen, ob ein beliebiges Wort $w$ endlicher Länge in $L$ enthalten ist oder nicht.

- Ist das Wortproblem für eine Sprache (für alle möglichen Wörter) in endlicher Rechenzeit lösbar, so heißt die Sprache *entscheidbar*.

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

- Worin genau besteht der Unterschied zwischen regulären und kontextfreien Sprachen?

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

- Worin genau besteht der Unterschied zwischen regulären und kontextfreien Sprachen?
- Warum sind kontextfreie Sprachen mächtiger/komplexer?

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

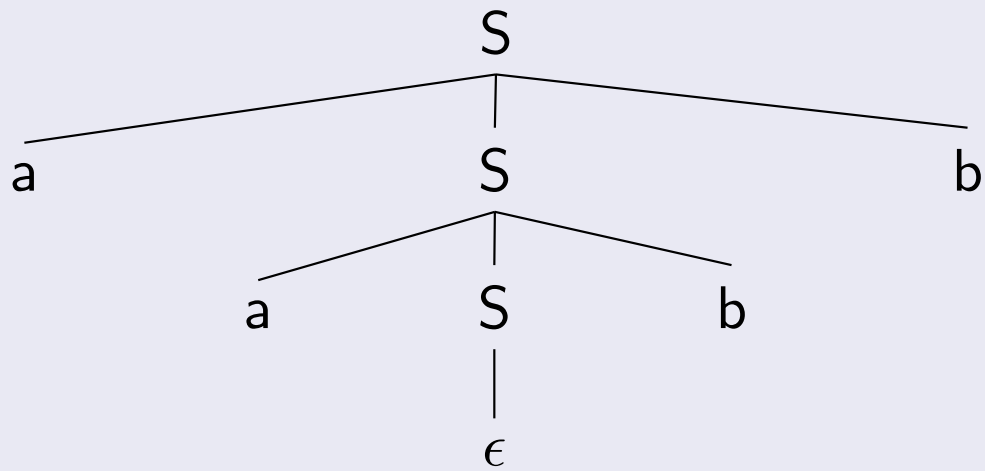
- Worin genau besteht der Unterschied zwischen regulären und kontextfreien Sprachen?
- Warum sind kontextfreie Sprachen mächtiger/komplexer?
- Was sind Vorteile/Nachteile regulärer/kontextfreier Sprachen?

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

- Lassen sich kontextfreie Sprachen auch durch DEAs /NEAs beschreiben?

## Kontextfreie Grammatik (Beispiel):

$$G := (\{S\}, \{a, b\}, R, S)\ R := \{S \to aSb, S \to \epsilon\}$$

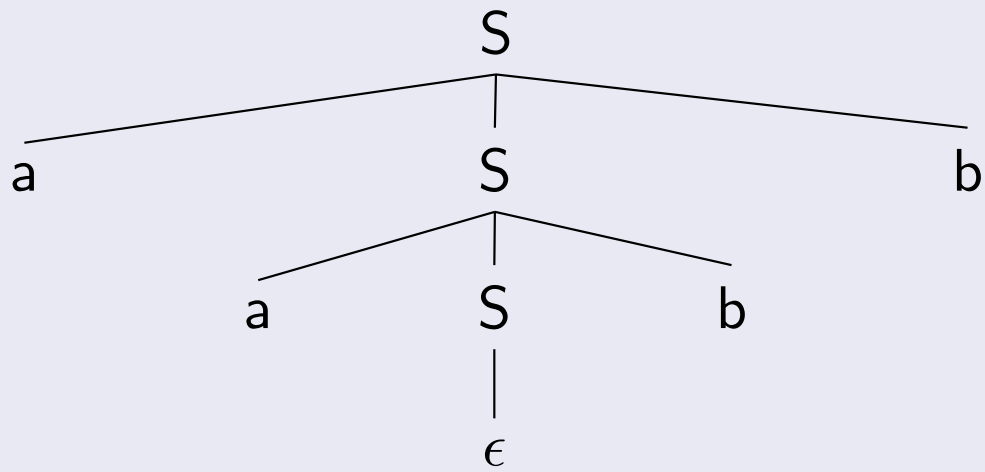


# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

- Lassen sich kontextfreie Sprachen auch durch DEAs /NEAs beschreiben?

## Kontextfreie Grammatik (Beispiel):

$$G := (\{S\}, \{a, b\}, R, S)\ R := \{S \to aSb, S \to \epsilon\}$$



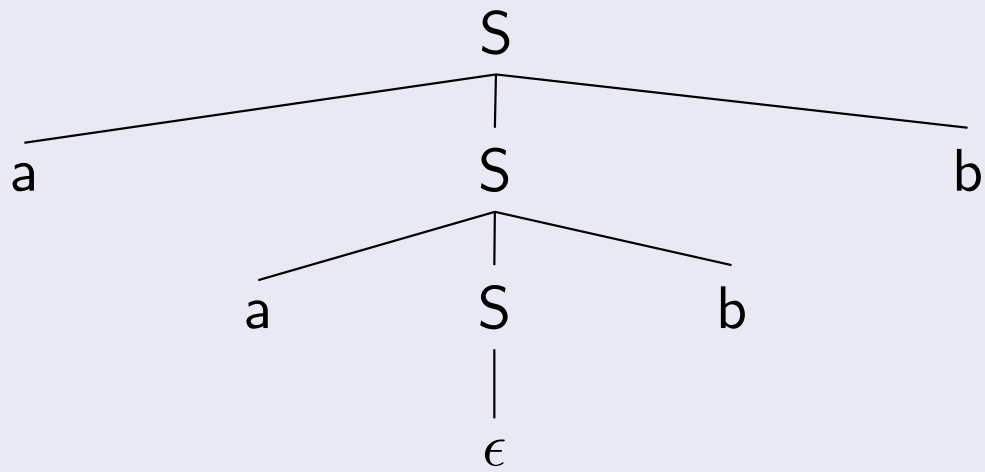

- Beschreibt die kontextfreie Sprache $a^n b^n$

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

- Lassen sich kontextfreie Sprachen auch durch DEAs /NEAs beschreiben?

## Kontextfreie Grammatik (Beispiel):

$$G := (\{S\}, \{a, b\}, R, S)\ R := \{S \to aSb, S \to \epsilon\}$$




- Beschreibt die kontextfreie Sprache $a^n b^n$
- Wie sieht der entsprechende Automat aus?

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

- Beschreibung von kontextfreien Sprachen durch NEAs/ DEAs im allgemeinen nicht möglich

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

- Beschreibung von kontextfreien Sprachen durch NEAs/ DEAs im allgemeinen nicht möglich
- Automaten müssten mit einem "Gedächtnis" ausgestattet werden

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

- Beschreibung von kontextfreien Sprachen durch NEAs/ DEAs im allgemeinen nicht möglich
- Automaten müssten mit einem "Gedächtnis" ausgestattet werden
  - Ein solcher Formalismus existiert (Kellerautomat)
  - löst Wortproblem für kontextfreie Sprachen

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

- Beschreibung von kontextfreien Sprachen durch NEAs/ DEAs im allgemeinen nicht möglich
- Automaten müssten mit einem "Gedächtnis" ausgestattet werden
  - Ein solcher Formalismus existiert (Kellerautomat)
  - löst Wortproblem für kontextfreie Sprachen
- Jede reguläre Grammatik lässt sich (durch einige Umformungen) als kontextfreie Grammatik formulieren (ohne Beweis)

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

- Beschreibung von kontextfreien Sprachen durch NEAs/ DEAs im allgemeinen nicht möglich
- Automaten müssten mit einem "Gedächtnis" ausgestattet werden
  - Ein solcher Formalismus existiert (Kellerautomat)
  - löst Wortproblem für kontextfreie Sprachen
- Jede reguläre Grammatik lässt sich (durch einige Umformungen) als kontextfreie Grammatik formulieren (ohne Beweis)
  - Reguläre Sprachen $\subset$ kontextfreie Sprachen

# Die Chomsky-Hierarchie der Sprachen

- Allgemeine Komplexitätshierarchie der Sprachen

**Chomsky-Hierarchie der formalen Sprachen**

reguläre Sprachen $\subset$

# Die Chomsky-Hierarchie der Sprachen

- Allgemeine Komplexitätshierarchie der Sprachen

**Chomsky-Hierarchie der formalen Sprachen**

reguläre Sprachen $\subset$
kontextfreie Sprachen $\subset$

# Die Chomsky-Hierarchie der Sprachen

- Allgemeine Komplexitätshierarchie der Sprachen

**Chomsky-Hierarchie der formalen Sprachen**

reguläre Sprachen $\subset$
kontextfreie Sprachen $\subset$
kontextsensitive Sprachen $\subset$

# Die Chomsky-Hierarchie der Sprachen

- Allgemeine Komplexitätshierarchie der Sprachen

**Chomsky-Hierarchie der formalen Sprachen**

reguläre Sprachen $\subset$
kontextfreie Sprachen $\subset$
kontextsensitive Sprachen $\subset$
rekursiv aufzählbare Sprachen $\subset$

# Die Chomsky-Hierarchie der Sprachen

- Allgemeine Komplexitätshierarchie der Sprachen

**Chomsky-Hierarchie der formalen Sprachen**

reguläre Sprachen $\subset$
kontextfreie Sprachen $\subset$
kontextsensitive Sprachen $\subset$
rekursiv aufzählbare Sprachen $\subset$
Sprachen (Megen von Zeichenketten)

# Die Chomsky-Hierarchie der Sprachen

| Typ | Sprachen | Grammatiken | Automaten |
|---|---|---|---|
| 3 | Reguläre Sprachen | Reguläre Grammatiken | Deterministische Endliche Automaten |

Tabelle: Chomsky-Hierarchie der Sprachen

# Die Chomsky-Hierarchie der Sprachen

| Typ | Sprachen | Grammatiken | Automaten |
|---|---|---|---|
| 3 | Reguläre Sprachen | Reguläre Grammatiken | Deterministische Endliche Automaten |
| 2 | Kontextfreie Sprachen | Kontextfreie Grammatiken | Kellerautomaten |

Tabelle: Chomsky-Hierarchie der Sprachen

# Die Chomsky-Hierarchie der Sprachen

| Typ | Sprachen | Grammatiken | Automaten |
|---|---|---|---|
| 3 | Reguläre Sprachen | Reguläre Grammatiken | Deterministische Endliche Automaten |
| 2 | Kontextfreie Sprachen | Kontextfreie Grammatiken | Kellerautomaten |
| 1 | Kontextsensitive Sprachen | Kontextsensitive Grammatiken | Nichtdeterministische linear beschränkte Automaten |

Tabelle: Chomsky-Hierarchie der Sprachen

# Die Chomsky-Hierarchie der Sprachen

| Typ | Sprachen | Grammatiken | Automaten |
|---|---|---|---|
| 3 | Reguläre Sprachen | Reguläre Grammatiken | Deterministische Endliche Automaten |
| 2 | Kontextfreie Sprachen | Kontextfreie Grammatiken | Kellerautomaten |
| 1 | Kontextsensitive Sprachen | Kontextsensitive Grammatiken | Nichtdeterministische linear beschränkte Automaten |
| 0 | Rekursiv aufzählbare Sprachen | Allgemeine Regelgrammatik | Turingmaschinen |

Tabelle: Chomsky-Hierarchie der Sprachen

# Die Chomsky-Hierarchie der Sprachen

| Typ | Sprachen | Grammatiken | Automaten |
|---|---|---|---|
| 3 | Reguläre Sprachen | Reguläre Grammatiken | Deterministische Endliche Automaten |
| 2 | Kontextfreie Sprachen | Kontextfreie Grammatiken | Kellerautomaten |
| 1 | Kontextsensitive Sprachen | Kontextsensitive Grammatiken | Nichtdeterministische linear beschränkte Automaten |
| 0 | Rekursiv aufzählbare Sprachen | Allgemeine Regelgrammatik | Turingmaschinen |
| nicht typisiert | Sprachen (allgemein) | - | - |

Tabelle: Chomsky-Hierarchie der Sprachen

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

## Wortproblem für reguläre Sprachen

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

## Wortproblem für reguläre Sprachen

Um ein Wort der Länge $n$ zu erkennen, muss ein DEA $n+1$ Zustände durchlaufen, und in einen Endzustand gelangen. Das Wortproblem für reguläre Sprachen ist für ein Wort der Länge $n$ also in $n+1$ Rechenschritten lösbar.

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

## Wortproblem für reguläre Sprachen

Um ein Wort der Länge $n$ zu erkennen, muss ein DEA $n+1$ Zustände durchlaufen, und in einen Endzustand gelangen. Das Wortproblem für reguläre Sprachen ist für ein Wort der Länge $n$ also in $n+1$ Rechenschritten lösbar.

- Wortproblem für reguläre Sprachen hat lineare Laufzeit:

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

## Wortproblem für reguläre Sprachen

Um ein Wort der Länge $n$ zu erkennen, muss ein DEA $n+1$ Zustände durchlaufen, und in einen Endzustand gelangen. Das Wortproblem für reguläre Sprachen ist für ein Wort der Länge $n$ also in $n+1$ Rechenschritten lösbar.

- Wortproblem für reguläre Sprachen hat lineare Laufzeit:
  - $\mathcal{O}(n)$ (generischer Fall) bzw.

# Eigenschaften kontextfreier Grammatiken bzw. Sprachen

## Wortproblem für reguläre Sprachen

Um ein Wort der Länge $n$ zu erkennen, muss ein DEA $n+1$ Zustände durchlaufen, und in einen Endzustand gelangen. Das Wortproblem für reguläre Sprachen ist für ein Wort der Länge $n$ also in $n+1$ Rechenschritten lösbar.

- Wortproblem für reguläre Sprachen hat lineare Laufzeit:
  - $\mathcal{O}(n)$ (generischer Fall) bzw.
  - $\Theta(n)$ (schlechtester Fall).

# Asymptotisches Verhalten von Funktionen

- $\mathcal{O}$-Notation bzw. $\mathcal{O}$-Kalkül (auch "Landau-Notation" oder "Landau-Symbole") beschreibt das asymptotische Verhalten von Funktionen

# Asymptotisches Verhalten von Funktionen

- $\mathcal{O}$-Notation bzw. $\mathcal{O}$-Kalkül (auch "Landau-Notation" oder "Landau-Symbole") beschreibt das asymptotische Verhalten von Funktionen
- Gibt Auskunft über das Laufzeitverhalten von Algorithmen bei ansteigender Problemgröße

# Asymptotisches Verhalten von Funktionen

- $\mathcal{O}$-Notation bzw. $\mathcal{O}$-Kalkül (auch "Landau-Notation" oder "Landau-Symbole") beschreibt das asymptotische Verhalten von Funktionen
- Gibt Auskunft über das Laufzeitverhalten von Algorithmen bei ansteigender Problemgröße
- Definition von Mengen von Funktionen mit bestimmten asymptotischen Eigenschaften in Abhängigkeit von Funktion $g(n)$ durch:
  - $\Theta(g(n))$: asymptotisch scharfe Schranke
  - $\mathcal{O}(g(n))$: asymptotische obere Schranke
  - $\Omega(g(n))$ : asymptotische untere Schranke

# Asymptotisches Verhalten von Funktionen

**$f(n) \in \Theta(g(n))$:**

gdw. es positive Konstanten $c_1$ und $c_2$ sowie eine Zahl $n_0 \in \mathbb{R}$ gibt, so dass:

$$\forall n, n \geq n_0 : 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$$

# Asymptotisches Verhalten von Funktionen

## $f(n) \in \Theta(g(n))$:

gdw. es positive Konstanten $c_1$ und $c_2$ sowie eine Zahl $n_0 \in \mathbb{R}$ gibt, so dass:

$$\forall n, n \geq n_0 : 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$$

## $f(n) \in \mathcal{O}(g(n))$:

gdw. es eine positive Konstante $c$ und eine Zahl $n_0 \in \mathbb{R}$ gibt, so dass:

$$\forall n, n \geq n_0 : 0 \leq f(n) \leq c g(n)$$

# Asymptotisches Verhalten von Funktionen

## $f(n) \in \Theta(g(n))$:

gdw. es positive Konstanten $c_1$ und $c_2$ sowie eine Zahl $n_0 \in \mathbb{R}$ gibt, so dass:

$$\forall n, n \geq n_0 : 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$$

## $f(n) \in \mathcal{O}(g(n))$:

gdw. es eine positive Konstante $c$ und eine Zahl $n_0 \in \mathbb{R}$ gibt, so dass:

$$\forall n, n \geq n_0 : 0 \leq f(n) \leq c g(n)$$

## $f(n) \in \Omega(g(n))$:

gdw. es eine positive Konstante $c$ und eine Zahl $n_0 \in \mathbb{R}$ gibt, so dass:

$$\forall n, n \geq n_0 : 0 \leq c g(n) \leq f(n)$$

# Asymptotisches Verhalten von Funktionen

**Asymptotisch scharfe Schranke kombiniert obere und untere asymptotische Schranke:**

$$f(n) \in \Theta(g(n)) \Leftrightarrow f(n) \in \mathcal{O}(g(n)) \wedge f(n) \in \Omega(g(n))$$

# Wortproblem für reguläre Sprachen

- $t_a$: Rechenzeit, die zum Durchlaufen/Fehlschlag eines Übergangs in einem DEA maximal benötigt wird

# Wortproblem für reguläre Sprachen

- $t_a$: Rechenzeit, die zum Durchlaufen/Fehlschlag eines Übergangs in einem DEA maximal benötigt wird
- Zum Lösen der Wortproblems für $w$ der Länge $n$:
  - Minimum: $t_a$: Erstes Zeichen schlägt fehl

# Wortproblem für reguläre Sprachen

- $t_a$: Rechenzeit, die zum Durchlaufen/Fehlschlag eines Übergangs in einem DEA maximal benötigt wird
- Zum Lösen der Wortproblems für $w$ der Länge $n$:
  - Minimum: $t_a$: Erstes Zeichen schlägt fehl
  - Maximum: $t_a n$: Wort wird erkannt

# Wortproblem für reguläre Sprachen

- $t_a$: Rechenzeit, die zum Durchlaufen/Fehlschlag eines Übergangs in einem DEA maximal benötigt wird
- Zum Lösen der Wortproblems für $w$ der Länge $n$:
  - Minimum: $t_a$: Erstes Zeichen schlägt fehl
  - Maximum: $t_a n$: Wort wird erkannt
- Rechenzeit als Funktion: $r(n) \leq t_a n$:

# Wortproblem für reguläre Sprachen

- $t_a$: Rechenzeit, die zum Durchlaufen/Fehlschlag eines Übergangs in einem DEA maximal benötigt wird
- Zum Lösen der Wortproblems für $w$ der Länge $n$:
  - Minimum: $t_a$: Erstes Zeichen schlägt fehl
  - Maximum: $t_a n$: Wort wird erkannt
- Rechenzeit als Funktion: $r(n) \leq t_a n$:
  - $t_a$ ist positive Konstante

# Wortproblem für reguläre Sprachen

- $t_a$: Rechenzeit, die zum Durchlaufen/Fehlschlag eines Übergangs in einem DEA maximal benötigt wird
- Zum Lösen der Wortproblems für $w$ der Länge $n$:
  - Minimum: $t_a$: Erstes Zeichen schlägt fehl
  - Maximum: $t_a n$: Wort wird erkannt
- Rechenzeit als Funktion: $r(n) \leq t_a n$:
  - $t_a$ ist positive Konstante
  - $r(n) \in \Theta(n)$ (ganzes Wort wird erkannt)
  - $r(n) \in \mathcal{O}(n)$ (immer)

## CYK-Parsing-Algorithmus für Grammatiken in Chomsky-NF:

```
w = a1 ... an
N = R1 ... Rr
let P[n,n,r] be an array of booleans.
Initialize all elements of P to false.
for each i = 1 to n:
  for each rule Rj -> ai:
    set P[1,i,j] = true
for each i = 2 to n:
  for each j = 1 to n-i+1:
    for each k = 1 to i-1:
      for each rule RA -> RB RC:
        if P[k,j,B] and P[i-k,j+k,C]:
          set P[i,j,A] = true
if any of P[n,1,x] is true:
  S is member of language
else
  S is not member of language
```

# Wortproblem für kontextfreie Sprachen

- CYK gehört zu den effizientesten Parsing-Algorithmen für kontextfreie Sprachen

# Wortproblem für kontextfreie Sprachen

- CYK gehört zu den effizientesten Parsing-Algorithmen für kontextfreie Sprachen
- Enthält drei verschachtelte Schleifen über Teilbereiche des Eingabeworts $a_1 ... a_n$ sowie

# Wortproblem für kontextfreie Sprachen

- CYK gehört zu den effizientesten Parsing-Algorithmen für kontextfreie Sprachen
- Enthält drei verschachtelte Schleifen über Teilbereiche des Eingabeworts $a_1 ... a_n$ sowie
- innere Schleife über Regelmenge $R$

# Wortproblem für kontextfreie Sprachen

- CYK gehört zu den effizientesten Parsing-Algorithmen für kontextfreie Sprachen
- Enthält drei verschachtelte Schleifen über Teilbereiche des Eingabeworts $a_1 ... a_n$ sowie
- innere Schleife über Regelmenge $R$
- Rechenzeit:

# Wortproblem für kontextfreie Sprachen

- CYK gehört zu den effizientesten Parsing-Algorithmen für kontextfreie Sprachen
- Enthält drei verschachtelte Schleifen über Teilbereiche des Eingabeworts $a_1 ... a_n$ sowie
- innere Schleife über Regelmenge $R$
- Rechenzeit:
  - $\mathcal{O}(n^3|R|)$ in Abhängigkeit von $n$ und $|R|$

# Wortproblem für kontextfreie Sprachen

- CYK gehört zu den effizientesten Parsing-Algorithmen für kontextfreie Sprachen
- Enthält drei verschachtelte Schleifen über Teilbereiche des Eingabeworts $a_1 ... a_n$ sowie
- innere Schleife über Regelmenge $R$
- Rechenzeit:
  - $\mathcal{O}(n^3|R|)$ in Abhängigkeit von $n$ und $|R|$
  - bzw. $\mathcal{O}(n^3)$ in Abhängigkeit von $n$

# Wortproblem für kontextfreie Sprachen

- CYK gehört zu den effizientesten Parsing-Algorithmen für kontextfreie Sprachen
- Enthält drei verschachtelte Schleifen über Teilbereiche des Eingabeworts $a_1 ... a_n$ sowie
- innere Schleife über Regelmenge $R$
- Rechenzeit:
  - $\mathcal{O}(n^3|R|)$ in Abhängigkeit von $n$ und $|R|$
  - bzw. $\mathcal{O}(n^3)$ in Abhängigkeit von $n$
- Das Wortproblem für kontextfreie Sprachen ist nicht in linearer Rechenzeit lösbar

# Wortproblem für kontextfreie Sprachen

- CYK gehört zu den effizientesten Parsing-Algorithmen für kontextfreie Sprachen
- Enthält drei verschachtelte Schleifen über Teilbereiche des Eingabeworts $a_1 ... a_n$ sowie
- innere Schleife über Regelmenge $R$
- Rechenzeit:
  - $\mathcal{O}(n^3|R|)$ in Abhängigkeit von $n$ und $|R|$
  - bzw. $\mathcal{O}(n^3)$ in Abhängigkeit von $n$
- Das Wortproblem für kontextfreie Sprachen ist nicht in linearer Rechenzeit lösbar
- Lösung in $\mathcal{O}(n^3)$ ist allgemein möglich (Beweisskizze)

# Asymptotisches Verhalten von Funktionen

**Hilfssatz zum asymptotischen Verhalten von Polynomen (verallgemeinert):**

Ein asymptotisch positives Polynom des $r$-ten Grades ist immer Element in $\Theta(n^r)$ (Ohne Beweis):

$$f(n) = a_0 n^r \pm a_1 n^{r-1} \pm \ldots \pm a_n$$

$$\Rightarrow f(n) \in \Theta(n^r)$$

# Pumping-Lemma für reguläre Sprachen

- Wie lässt sich für eine beliebige Sprachen $L$ feststellen, ob sie regulär ist oder nicht?

# Pumping-Lemma für reguläre Sprachen

- Wie lässt sich für eine beliebige Sprachen $L$ feststellen, ob sie regulär ist oder nicht?
- Endliche Sprachen sind immer regulär

# Pumping-Lemma für reguläre Sprachen

- Wie lässt sich für eine beliebige Sprachen $L$ feststellen, ob sie regulär ist oder nicht?
- Endliche Sprachen sind immer regulär
- Wenn DEA, NEA oder reguläre Grammatik für $L$ gefunden werden kann, ist $L$ regulär

# Pumping-Lemma für reguläre Sprachen

- Wie lässt sich für eine beliebige Sprachen $L$ feststellen, ob sie regulär ist oder nicht?
- Endliche Sprachen sind immer regulär
- Wenn DEA, NEA oder reguläre Grammatik für $L$ gefunden werden kann, ist $L$ regulär
- Falls nicht, kann überprüft werden, ob $L$ das Pumping-Lemma für reguläre Sprachen erfüllt

# Pumping-Lemma für reguläre Sprachen

- Wie lässt sich für eine beliebige Sprachen $L$ feststellen, ob sie regulär ist oder nicht?
- Endliche Sprachen sind immer regulär
- Wenn DEA, NEA oder reguläre Grammatik für $L$ gefunden werden kann, ist $L$ regulär
- Falls nicht, kann überprüft werden, ob $L$ das Pumping-Lemma für reguläre Sprachen erfüllt
  - Erfüllt $L$ das Lemma nicht, ist $L$ nicht regulär

# Pumping-Lemma für reguläre Sprachen

- Wie lässt sich für eine beliebige Sprachen $L$ feststellen, ob sie regulär ist oder nicht?
- Endliche Sprachen sind immer regulär
- Wenn DEA, NEA oder reguläre Grammatik für $L$ gefunden werden kann, ist $L$ regulär
- Falls nicht, kann überprüft werden, ob $L$ das Pumping-Lemma für reguläre Sprachen erfüllt
  - Erfüllt $L$ das Lemma nicht, ist $L$ nicht regulär
  - Erfüllt $L$ das Lemma, ist $L$ möglicherweise (aber nicht notwendigerweise) regulär

# Reguläre Sprachen und natürliche Sprachen

- Kontextfreie Sprachen sind gute Annäherung an die Struktur natürlichen Sprachen

# Reguläre Sprachen und natürliche Sprachen

- Kontextfreie Sprachen sind gute Annäherung an die Struktur natürlichen Sprachen
- Natürliche Sprachen unterliegen allerdings weiteren morphosyntaktischen Einschränkungen

# Reguläre Sprachen und natürliche Sprachen

- Kontextfreie Sprachen sind gute Annäherung an die Struktur natürlichen Sprachen
- Natürliche Sprachen unterliegen allerdings weiteren morphosyntaktischen Einschränkungen
- Meist modelliert als Constraints in einem kontextfreien Parsingmodell

# Reguläre Sprachen und natürliche Sprachen

- Kontextfreie Sprachen sind gute Annäherung an die Struktur natürlichen Sprachen
- Natürliche Sprachen unterliegen allerdings weiteren morphosyntaktischen Einschränkungen
- Meist modelliert als Constraints in einem kontextfreien Parsingmodell
- Natürliche Sprachen sind “schwach kontextsensitive” Sprachen

Noch Fragen?

# Weiterführende Literatur

Dan Jurafsky und James H. Martin , *Speech and Language Processing*, dritte Ausgabe. 12. Kapitel:
`https://web.stanford.edu/~jurafsky/slp3/12.pdf`

Vielen Dank für die Aufmerksamkeit!