Theoretische Informatik

Prof.Dr.Christian Wagenknecht

Hochschule Zittau/Görlitz
Fachbereich Informatik

3. Mai 2007

Einordnung und Motivationen

Kerninformatik: Theoretische, praktische, technische Informatik
Angewandte Informatik

Theoretische Informatik:

- Theorie der formalen Sprachen
- Automatentheorie
- Berechenbarkeitstheorie
- Komplexitätstheorie

Einordnung und Motivationen

Kerninformatik: Theoretische, praktische, technische Informatik
Angewandte Informatik

Theoretische Informatik:

- Theorie der formalen Sprachen
- Automatentheorie
- Berechenbarkeitstheorie
- Komplexitätstheorie

Einordnung und Motivationen

Kerninformatik: Theoretische, praktische, technische Informatik
Angewandte Informatik

Theoretische Informatik:

- Theorie der formalen Sprachen
- Automatentheorie
- Berechenbarkeitstheorie
- Komplexitätstheorie

Einordnung und Motivationen

Kerninformatik: Theoretische, praktische, technische Informatik
Angewandte Informatik

Theoretische Informatik:

- Theorie der formalen Sprachen
- Automatentheorie
- Berechenbarkeitstheorie
- Komplexitätstheorie

Einordnung und Motivationen

Kerninformatik: Theoretische, praktische, technische Informatik
Angewandte Informatik

Theoretische Informatik:

- Theorie der formalen Sprachen
- Automatentheorie
- Berechenbarkeitstheorie
- Komplexitätstheorie

Fehlerhaftes Programm

```
pprogram test;

begin
  x:=3/0
end.
```

Syntaktische und logische Fehler
Interesse beschränkt sich auf Syntax!
Gegenstand sind *formale Sprachen*: NICHT Semantik, NICHT natürliche Sprachen, NICHT PS (PS = FS + Semantik)

Fehlerhaftes Programm

```
pprogram test;

begin
  x:=3/0
end.
```

Syntaktische und logische Fehler
Interesse beschränkt sich auf Syntax!
Gegenstand sind *formale Sprachen*: NICHT Semantik, NICHT natürliche Sprachen, NICHT PS (PS = FS + Semantik)

Fehlerhaftes Programm

```
pprogram test;
var x: real;
begin
  x:=3/0
end.
```

Syntaktische und logische Fehler
Interesse beschränkt sich auf Syntax!
Gegenstand sind *formale Sprachen*: NICHT Semantik, NICHT natürliche Sprachen, NICHT PS (PS = FS + Semantik)

Fehlerhaftes Programm

```
pprogram test;
var x: real;
begin
  x:=3/0
end.
```

Syntaktische und logische Fehler
Interesse beschränkt sich auf Syntax!
Gegenstand sind *formale Sprachen*: NICHT Semantik, NICHT natürliche Sprachen, NICHT PS (PS = FS + Semantik)

Grundbegriffe

- Alphabet
- Verkettung von Zeichen (Schreibweise!)
- Wort – auch leeres Wort
- Verkettung von Wörtern
- Wortmenge – Wortlänge, Aufbauschema, KLEENE-Stern $A^* = \bigcup_{i=0}^{\infty} A^i$, abzählbar unendliche Menge (längenlexikografische Ordnung)
- Wortmenge und $(A^*, \circ)$ – Halbgruppe mit neutralem Element
- Sprache

Grundbegriffe

- Alphabet
- Verkettung von Zeichen (Schreibweise!)
- Wort – auch leeres Wort
- Verkettung von Wörtern
- Wortmenge – Wortlänge, Aufbauschema, KLEENE-Stern $A^* = \bigcup_{i=0}^{\infty} A^i$, abzählbar unendliche Menge (längenlexikografische Ordnung)
- Wortmenge und $(A^*, \circ)$ – Halbgruppe mit neutralem Element
- Sprache

Grundbegriffe

- Alphabet
- Verkettung von Zeichen (Schreibweise!)
- Wort – auch leeres Wort
- Verkettung von Wörtern
- Wortmenge – Wortlänge, Aufbauschema, KLEENE-Stern $A^* = \bigcup_{i=0}^{\infty} A^i$, abzählbar unendliche Menge (längenlexikografische Ordnung)
- Wortmenge und $(A^*, \circ)$ – Halbgruppe mit neutralem Element
- Sprache

Grundbegriffe

- Alphabet
- Verkettung von Zeichen (Schreibweise!)
- Wort – auch leeres Wort
- Verkettung von Wörtern
- Wortmenge – Wortlänge, Aufbauschema, Kleene-Stern $A^* = \bigcup_{i=0}^{\infty} A^i$, abzählbar unendliche Menge (längenlexikografische Ordnung)
- Wortmenge und $(A^*, \circ)$ – Halbgruppe mit neutralem Element
- Sprache

Grundbegriffe

- Alphabet
- Verkettung von Zeichen (Schreibweise!)
- Wort – auch leeres Wort
- Verkettung von Wörtern
- Wortmenge – Wortlänge, Aufbauschema, KLEENE-Stern $A^* = \bigcup_{i=0}^{\infty} A^i$, abzählbar unendliche Menge (längenlexikografische Ordnung)
- Wortmenge und $(A^*, \circ)$ – Halbgruppe mit neutralem Element
- Sprache

Grundbegriffe

- Alphabet
- Verkettung von Zeichen (Schreibweise!)
- Wort – auch leeres Wort
- Verkettung von Wörtern
- Wortmenge – Wortlänge, Aufbauschema, KLEENE-Stern $A^* = \bigcup_{i=0}^{\infty} A^i$, abzählbar unendliche Menge (längenlexikografische Ordnung)
- Wortmenge und $(A^*, \circ)$ – Halbgruppe mit neutralem Element
- Sprache

Grundbegriffe

- Alphabet
- Verkettung von Zeichen (Schreibweise!)
- Wort – auch leeres Wort
- Verkettung von Wörtern
- Wortmenge – Wortlänge, Aufbauschema, KLEENE-Stern
  $A^* = \bigcup_{i=0}^{\infty} A^i$, abzählbar unendliche Menge
  (längenlexikografische Ordnung)
- Wortmenge und $(A^*, \circ)$ – Halbgruppe mit neutralem Element
- Sprache

Sprache

Problem

Definition unendlicher Mengen (Sprachen sind im Allg. unendliche Mengen)

Lösung

1. *formale Grammatiken*
2. *abstrakte Automaten (wird noch thematisiert)*
3. *Schablonen (nicht allgemeingültig)*

Sprache

Problem

Definition unendlicher Mengen (Sprachen sind im Allg. unendliche Mengen)

Lösung

1. *formale Grammatiken*
2. *abstrakte Automaten (wird noch thematisiert)*
3. *Schablonen (nicht allgemeingültig)*

Sprache

Problem

Definition unendlicher Mengen (Sprachen sind im Allg. unendliche Mengen)

Lösung

1. *formale Grammatiken*
2. *abstrakte Automaten (wird noch thematisiert)*
3. *Schablonen (nicht allgemeingültig)*

Sprache

Problem

Definition unendlicher Mengen (Sprachen sind im Allg. unendliche Mengen)

Lösung

1. *formale Grammatiken*
2. *abstrakte Automaten (wird noch thematisiert)*
3. *Schablonen (nicht allgemeingültig)*

Sprache

Problem

Definition unendlicher Mengen (Sprachen sind im Allg. unendliche Mengen)

Lösung

1. *formale Grammatiken*
2. *abstrakte Automaten (wird noch thematisiert)*
3. *Schablonen (nicht allgemeingültig)*

Syntax

Definition

Die *Syntax* einer Sprache ist die Lehre vom Satzbau. Sie umfasst sämtliche grammatikalische Regeln, deren Anwendung zu korrekt gebauten Sätzen führt.

Beispiel: Satz der engl. Sprache – SPO
Ableitungsbaum (Parsebaum)

Syntax

Definition

Die *Syntax* einer Sprache ist die Lehre vom Satzbau. Sie umfasst sämtliche grammatikalische Regeln, deren Anwendung zu korrekt gebauten Sätzen führt.

Beispiel: Satz der engl. Sprache – SPO
Ableitungsbaum (Parsebaum)

Syntax

Definition

Die *Syntax* einer Sprache ist die Lehre vom Satzbau. Sie umfasst sämtliche grammatikalische Regeln, deren Anwendung zu korrekt gebauten Sätzen führt.

Beispiel: Satz der engl. Sprache – SPO
Ableitungsbaum (Parsebaum)

Syntax

Definition

Die *Syntax* einer Sprache ist die Lehre vom Satzbau. Sie umfasst sämtliche grammatikalische Regeln, deren Anwendung zu korrekt gebauten Sätzen führt.

Beispiel: Satz der engl. Sprache – SPO

Ableitungsbaum (Parsebaum)

Syntax

Definition

Die *Syntax* einer Sprache ist die Lehre vom Satzbau. Sie umfasst sämtliche grammatikalische Regeln, deren Anwendung zu korrekt gebauten Sätzen führt.

Beispiel: Satz der engl. Sprache – SPO
Ableitungsbaum (Parsebaum)

Grammatik

Definition

Eine Grammatik G ist ein 4-Tupel $G = (N, T, P, s)$ mit

$N \ldots$ Menge der **Nichtterminale** (grammatikalische Variablen)

$T \ldots$ Menge der **Terminale** (Alphabetzeichen)

N, T sind nichtleere, endliche und disjunkte Mengen.

$P \ldots$ endliche Menge von **Regeln** oder **Produktionen**

$P = \{\alpha \rightarrow \beta \mid \alpha \in (N \cup T)^* \setminus T^* \text{ und } \beta \in (N \cup T)^*\}$

$s \ldots$ **Start**- oder **Satz**- oder **Spitzensymbol**, wobei $s \in N$.

Einführungsbeispiel

Beispiel

$G = (N, T, P, s)$, mit $N = \{X, Y, Z\}$, $T = \{a, b, c\}$, $s = X$,
$P = \{X \rightarrow aYZcc, Y \rightarrow b \mid bY, Z \rightarrow a \mid aZ\}$

Welche Sprache $L(G)$ wird mit G beschrieben?

Ableitung eines Wortes aus $L(G)$: $s \Rightarrow \ldots \Rightarrow w$

Beispiel

$G_{Kfz} = (N, T, P, s)$, mit
$N = \{\text{Kennzeichen}, \text{Ortskennung}, \text{Buchstabe}, \text{Zahl}, \text{Ziffer}\}$,
$T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \_, -, A, B, C, D, E, G, I, L, R, Z\}$,
$P = \{$

$$
\begin{array}{lcl}
\text{Kennzeichen} & \rightarrow & \text{Ortskennung} - \text{Buchstabe} \_ \text{Zahl} \\
 & \rightarrow & \text{Ortskennung} - \text{Buchstabe Buchstabe} \_ \text{Zahl} \\
\text{Ortskennung} & \rightarrow & \text{DD} \mid \text{LB} \mid \text{ZI} \mid \text{GR} \\
\text{Buchstabe} & \rightarrow & \text{A} \mid \text{B} \mid \text{C} \mid \text{D} \mid \text{E} \mid \text{G} \mid \text{I} \mid \text{L} \mid \text{R} \mid \text{Z} \\
\text{Zahl} & \rightarrow & \text{Ziffer} \mid \text{Ziffer Ziffer} \mid \text{Ziffer Ziffer Ziffer} \\
\text{Ziffer} & \rightarrow & 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \qquad \}
\end{array}
$$

(in BNF – für kontextfreie Grammatiken)

$s = \text{Kennzeichen};$

Aufgabe: Ableitungsbäume für zwei korrekte Kennzeichen

Satzform und Ableitung

Definition

Eine Folge von *Satzformen* $(\alpha_0, \alpha_1, \ldots, \alpha_n)$, mit $\alpha_0 = s$, $\alpha_n = w$, $w \in T^*$ und $s \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow w$, heißt Ableitung von w.

Erzeugbare Sprache

Definition

Die von G erzeugbare Sprache $L(G)$ ist

$$L(G) = \{w \mid w \in T^* \text{ und } s \overset{*}{\Rightarrow}_G w\}.$$

Relation „geht unmittelbar über in"

Definition

Seien $u \in (N \cup T)^* \setminus T^*$ und $v \in (N \cup T)^*$.

Dann gilt die Relation $u \Rightarrow_G v$ und man sagt: „u geht (unter G) unmittelbar über in v", falls u und v die Form haben:
$u = xyz, \quad v = xy'z, \quad x, z \in (N \cup T)^*$ und $(y \rightarrow y') \in P$.

Für $d \Rightarrow_G e \Rightarrow_G \ldots \Rightarrow_G f$ schreibt man verkürzend $d \overset{*}{\Rightarrow}_G f$.

Es ist üblich, das an den Doppelpfeil angehängte G wegzulassen.

Entscheidungen für kfG

s. Einführungsbeispiel

- Welches Nichtterminal soll zuerst ersetzt werden?
  Linksableitung $\longrightarrow$ gleicher Baum, bestimmte Aufbaureihenfolge
- Welche Regel für X verwenden, wenn es mehr als eine gibt?
  erste im Aufschrieb $\longrightarrow$ Gefahr von *Sackgassen* $\longrightarrow$ Backtracking
  Bei manchen Grammatiken gibt es für ein Wort mehrere verschiedene Ableitungen, d.h. unterschiedlich strukturierte Ableitungsbäume – *Mehrdeutigkeit*

Entscheidungen für kfG

s. Einführungsbeispiel

- Welches Nichtterminal soll zuerst ersetzt werden?
  Linksableitung $\rightarrow$ gleicher Baum, bestimmte Aufbaureihenfolge
- Welche Regel für X verwenden, wenn es mehr als eine gibt?
  erste im Aufschrieb $\rightarrow$ Gefahr von *Sackgassen* $\rightarrow$ Backtracking
  Bei manchen Grammatiken gibt es für ein Wort mehrere verschiedene Ableitungen, d.h. unterschiedlich strukturierte Ableitungsbäume – *Mehrdeutigkeit*

Entscheidungen für kfG

s. Einführungsbeispiel

- Welches Nichtterminal soll zuerst ersetzt werden?
  Linksableitung $\rightarrow$ gleicher Baum, bestimmte Aufbaureihenfolge
- Welche Regel für X verwenden, wenn es mehr als eine gibt?
  erste im Aufschrieb $\rightarrow$ Gefahr von *Sackgassen* $\rightarrow$ Backtracking
  Bei manchen Grammatiken gibt es für ein Wort mehrere verschiedene Ableitungen, d.h. unterschiedlich strukturierte Ableitungsbäume – *Mehrdeutigkeit*

Entscheidungen für kfG

s. Einführungsbeispiel

- Welches Nichtterminal soll zuerst ersetzt werden?
  Linksableitung $\rightarrow$ gleicher Baum, bestimmte Aufbaureihenfolge
- Welche Regel für X verwenden, wenn es mehr als eine gibt?
  erste im Aufschrieb $\rightarrow$ Gefahr von *Sackgassen* $\rightarrow$ Backtracking
  Bei manchen Grammatiken gibt es für ein Wort mehrere verschiedene Ableitungen, d.h. unterschiedlich strukturierte Ableitungsbäume – *Mehrdeutigkeit*

Entscheidungen für kfG

s. Einführungsbeispiel

- Welches Nichtterminal soll zuerst ersetzt werden?
  Linksableitung $\longrightarrow$ gleicher Baum, bestimmte Aufbaureihenfolge
- Welche Regel für X verwenden, wenn es mehr als eine gibt?
  erste im Aufschrieb $\longrightarrow$ Gefahr von *Sackgassen* $\longrightarrow$ Backtracking
  Bei manchen Grammatiken gibt es für ein Wort mehrere verschiedene Ableitungen, d.h. unterschiedlich strukturierte Ableitungsbäume – *Mehrdeutigkeit*

Linksableitung und Rechtsableitung

Beispiel

Gegeben sei die Grammatik $G = (N, T, P, s)$, mit $s = S$, $T = \{a, b, c, -\}$, $N = \{A, S\}$ und $P = \{S \rightarrow A\ (1),\ S \rightarrow S - A\ (2),\ A \rightarrow a\ (3),\ A \rightarrow b\ (4),\ A \rightarrow c\ (5)\}$.

Wegen $S \Rightarrow S - A \Rightarrow S - A - A \Rightarrow A - A - A \Rightarrow a - A - A \Rightarrow a - b - A \Rightarrow a - b - b$ gilt $a - b - b \in L(G)$.
Regelanwendungen: $(2, 2, 1, 3, 4, 4)$ aber auch $(2, 4, 2, 1, 4, 3)$.

Linksableitung und Rechtsableitung

Beispiel

Gegeben sei die Grammatik $G = (N, T, P, s)$, mit $s = S$, $T = \{a, b, c, -\}$, $N = \{A, S\}$ und $P = \{S \rightarrow A\ (1),\ S \rightarrow S - A\ (2),\ A \rightarrow a\ (3),\ A \rightarrow b\ (4),\ A \rightarrow c\ (5)\}$.

Wegen $S \Rightarrow S - A \Rightarrow S - A - A \Rightarrow A - A - A \Rightarrow a - A - A \Rightarrow a - b - A \Rightarrow a - b - b$ gilt $a - b - b \in L(G)$.
Regelanwendungen: $(2, 2, 1, 3, 4, 4)$ aber auch $(2, 4, 2, 1, 4, 3)$.

Mehrdeutigkeit

Beispiel

Untersuchen Sie die folgende Grammatik auf Mehrdeutigkeit.
$G = (N, T, P, s)$, mit $s = X$, $T = \{a, b\}$, $N = \{X\}$ und
$P = \{X \to a \mid b \mid XX\}$.

Definition

Eine (kontextfreie) Grammatik ist mehrdeutig, wenn es in $L(G)$ ein Wort gibt, das zwei (verschiedene) Linksableitungen besitzt. Andernfalls ist die Grammatik eindeutig.

Eine Sprache L ist **eindeutig**, wenn es (mindestens) eine eindeutige kfG G, mit $L = L(G)$, gibt. Ansonsten ist L (inhärent) mehrdeutig.

Man kann versuchen, zu einer mehrdeutigen Grammatik eine eindeutige zu finden, die die gleiche Sprache beschreibt. Wenn dies nicht gelingt, ist die Sprache „hoffnungslos" mehrdeutig.

Äquivalenz von Grammatiken

Definition

Zwei Grammatiken G_1 und G_2 heißen **äquivalent**, geschrieben: $G_1 \cong G_2$, wenn die zugehörigen erzeugbaren Sprachen übereinstimmen, d.h. wenn $L(G_1) = L(G_2)$.

Aufgabe 1

Untersuchen Sie die folgende Grammatik auf Mehrdeutigkeit. $G = (N, T, P, s)$, mit $s = S$, $T = \{a, b, c\}$, $N = \{A, B, A\}$ und $P = \{S \rightarrow aB \mid Ab,\ A \rightarrow ac,\ B \rightarrow cb\}$.

Hinweis: Betrachten Sie beispielsweise das Wort *acb*.

Aufgabe 2

Stellen Sie fest, welche Sprache durch die folgende Grammatik G definiert wird.

$G = (N, T, P, s)$, mit $s = Z$, $T = \{a, b, c\}$, $N = \{Z, M, C\}$ und $P = \{Z \rightarrow abc \mid aMbc,\ M \rightarrow aMbC \mid abC,\ Cb \rightarrow bC,\ Cc \rightarrow cc\}$.

Aufgabe 2

Stellen Sie fest, welche Sprache durch die folgende Grammatik G definiert wird.

$G = (N, T, P, s)$, mit $s = Z$, $T = \{a, b, c\}$, $N = \{Z, M, C\}$ und $P = \{Z \rightarrow abc \mid aMbc,\ M \rightarrow aMbC \mid abC,\ Cb \rightarrow bC,\ Cc \rightarrow cc\}$.

$Z \overset{*}{\Rightarrow} a^n b^n c^n,\ n \geq 1$

Aufgabe 3

Geben Sie eine formale Grammatik G für die Sprache der natürlichen Zahlen (ohne führende Nullen) an.

Weisen Sie nach, dass das Zahlwort 873 zu $L(G)$ gehört, 0873 hingegen nicht.

Aufgabe 4

Geben Sie eine Grammatik für arithmetische Ausdrücke an. Die Ausdrücke sollen nur die Variablen a,b und c, sowie die Operatoren $+, -, *$ und $/$ für die entsprechenden zweistelligen Operationen enthalten dürfen. Außerdem können Klammern (und) verwendet werden, wobei bei deren Einsatz auf korrekte Klammerung geachtet werden muss. Beispiele: $a + b$, $a + (b * c)$, $a/(b - c * c)$.