

Arten von formalen Sprachen

Herbert Bauer Matr.Nr: 0303779

Chomsky-Grammatik

Laut Chomsky Grammatik gibt es 4 Typen von Grammatiken

- ▶ Type 0 Grammatik - Turingmaschine.
- ▶ Type 1 Grammatik - linear beschränkte Turingmaschine.
- ▶ Type 2 Grammatik - Kellerautomat.
- ▶ Type 3 Grammatik - Endlicher Automat.

Chomsky-Grammatik

Type 0 Grammatik - allgemeine Grammatik - Turingmaschine.

$$G = (N, \Sigma, P, S)$$

N : Menge der Nichtterminalsymbole, endlich

Σ : Menge der Terminalsymbole, endlich

$P \subseteq \Gamma^* N \Gamma^* \times \Gamma^*$: Produktionsregeln, endlich

$S \in N$: Startsymbol, $N \cap \Sigma = \emptyset$

$\Gamma := N \cup \Sigma$: Gesamtalphabet

Chomsky-Grammatik

Type 1 Grammatik - linear beschränkte Turingmaschine.

$$G = (N, \Sigma, P, S)$$

Produktionsregeln

P der Form

$$\alpha X \beta \rightarrow \alpha \gamma \beta$$

oder der Form

$$S \rightarrow \varepsilon$$

S kommt nicht auf keiner rechten Seite einer Produktionsregel vor

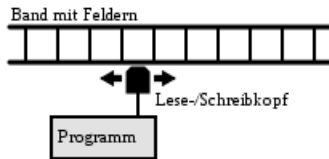


Abbildung: Turingmaschine

Von der Turing Maschine zum Smartphone-wie geht das?

Präsentation von Kollitsch, Lintner und Penker

Sie erzählten uns vom Halte-Problem, einer Maschine, die unendlich lange lief und nur 1er schriebe, aber noch aufhört. Ich hätte euch gern mehr in Erinnerung gerufen, aber das Handout war noch nicht am Server.

Chomsky-Grammatik

Type 2 Grammatik - Kellerautomat

$$G = (N, \Sigma, P, S)$$

Regeln

$$P \subseteq N \times (N \cup \Sigma)^*$$

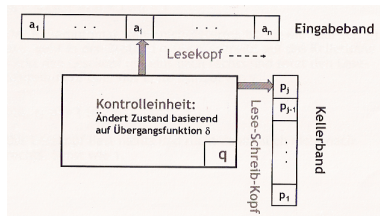


Abbildung: Turingmaschine

Chomsky-Grammatik

Type 3 Grammatik - Endlicher Automat

$$G = (N, \Sigma, P, S)$$

rechtslineare Grammatik

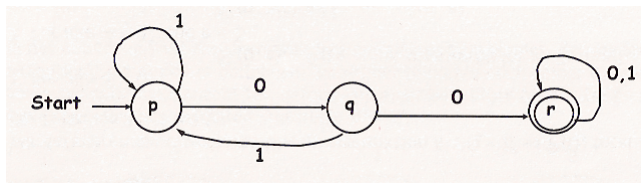
- ▶ $A \rightarrow xB$, mit $x \in \Sigma^*$ und $B \in N$
- ▶ $A \rightarrow x$, mit $x \in \Sigma^*$

linkslineare Grammatik

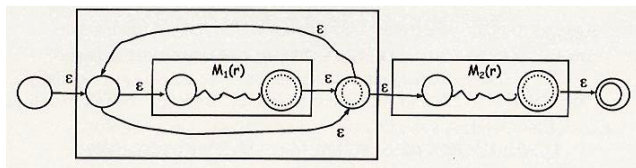
- ▶ $A \rightarrow Bx$, mit $x \in \Sigma^*$ und $B \in N$
- ▶ $A \rightarrow x$, mit $x \in \Sigma^*$

Chomsky-Grammatik

Type 3 Grammatik - Endlicher Automat



(a) Endlicher Automat



(b) mehrere Endliche Automaten

Verwendung

Compiler

Ein Compiler übersetzt ein Programm von menschlich lesbarer Programmiersprache in eine computerverständliche Maschinsprache.

- ▶ Syntaxprüfung: Es wird geprüft, ob der Quellcode ein gültiges Programm darstellt.
- ▶ Lexikalische Analyse: Der Code wird in logische Grundbausteine zerteilt. (Endliche Automaten)
- ▶ Semantische Analyse: Variablen müssen deklariert sein, und zu einander kompatibel sein.

Verwendung

CPU

CPU arbeiten als **deterministischer endlicher Automat** nach dem Prinzip: „Fetch-Decode-Execute“. Der Operator wird eingelesen, dieser bestimmt die DEA, als nächstes werden die Operanten eingelesen.

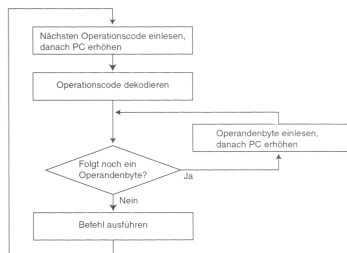


Abbildung:
Fetch-Decode-Execute

Registersatz: Register, um Daten innerhalb des Prozessors speichern zu können

Steuerwerk: Verantwortlichen für Ablaufsteuerung

Operationswerk (ALU): Eigentliche Datenverarbeitung

Adresswerk: Um auf Daten und Code im Hauptspeicher zugreifen zu können

Systembus (Schnittstelle): Datenverkehr mit dem Rest des Systems

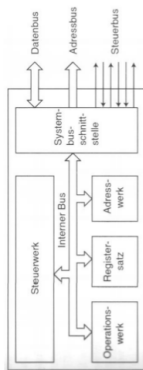


Abbildung:
Funktionsschema
eines CPU

- ▶ Registersatz - Kellerspeicher
- ▶ Steuerwerk - Regeln
- ▶ Operationswerk - Zustand
- ▶ Adresswerk - Band
- ▶ Systembus - Schreibkopf

Abweichungen von der Norm bei formalen Sprachen vs. Natürlichen Sprachen

Präsentation von Andert und Österreicher

In dieser Präsentation werden viele Esoterische Formal Sprachen aufgelistet

- ▶ OMGROFL
- ▶ Brainfuck
- ▶ Befunge

Normen für Programmiersprachen

- ▶ Ermöglichen Austausch zwischen den in dieser Sprache geschriebenen Programmen und den Datenverarbeitungsanlagen

ISO/IEC TR 10176	Informationstechnik – Richtlinien für die Erarbeitung von Programmiersprachen-Normen (Sep 1998)
ISO/IEC 10967	Informationstechnik – Sprachunabhängige Arithmetik – Teil 1: Ganzzahl- und Gleitpunkt-Arithmetik (Dez 1994)
ISO/IEC 11404	Informationstechnik – Programmiersprachen, ihre Umgebungen und System-Software-Schnittstellen – Sprachunabhängige Datentypen (Dez 1996)
ISO/IEC TR 10182	Informationstechnik – Programmiersprachen, ihre Umgebungen und System-Software-Schnittstellen – Richtlinien für Sprachbindungen (Dez 1993)

Plansprachen

Es gibt Plansprachen wie Esperanto, Interlingue oder Interlingua, die gesprochen und geschrieben werden, ein anderes Beispiel ist Lincos.

Lincos

Hans **Freudenthals** Ziel war es eine Sprache zu erschaffen, die von jedem intelligenten Wesen, d.h. auch außerirdischen Lebensformen, verstanden werden kann, obwohl keine direkte Kommunikation über Bilder oder mit einer beiden Seiten bekannten Drittsprache möglich ist.

Lincos	Bedeutung
X O X	$1 = 1$
X O XX A X O X \in Fal:	$(1 = 2)$ UND $(1 = 1)$ falsch
E1 Ant E2 \in Ver:	$(E1$ ZEITLICH VOR $E2)$ wahr

Übersetzungen

Die Translation von einer formalen Sprache in die andere Sprache erledigt ein Compiler einwandfrei und richtig.

Plansprachen

Die Übersetzungen einer Plansprache in die andere, würde nach Konzept ebenfalls nur einen Compiler benötigen, diese wäre einwandfrei und richtig, weil die Regeln endlich, eindeutig und ohne Ausnahmen sind.

Ich verwende die folgende Präsentation

- ▶ von Peischl und Pela „Wie realisieren Computer das Nachrichtenquadrat?“ (Handout fehlt)
- ▶ von Althammer und Schindelka in „Wie verstehen Computer natürliche Sprachen?“

um wegfallenden Schwierigkeiten zu zeigen.

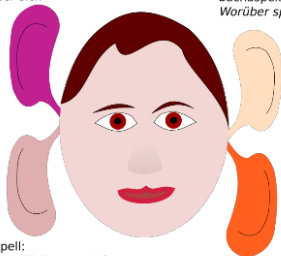
Interdisziplinäre Fachdidaktik: Natürliche Sprachen und Formale Sprachen (UE)

Grundüberlegungen zu „verstehen“ und „natürliche Sprache“

Schulz von Thun

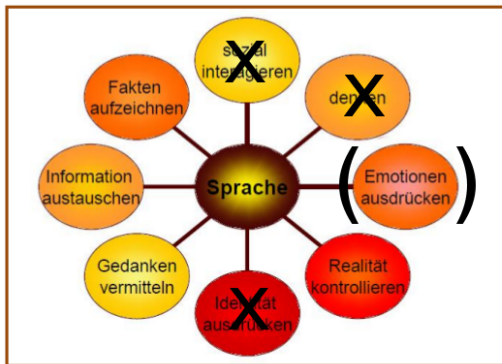
Selbstaussage:
Was offenbart sie über sich?

Sachaspekt:
Worüber spricht sie?



Appell:
Was will sie von mir?

Beziehungsaspekt:
Wie steht sie zu mir?



natürliche Sprachen

Bei der Übersetzung von natürlichen Sprachen treten Schwierigkeiten auf. Ich verwende die Präsentation von

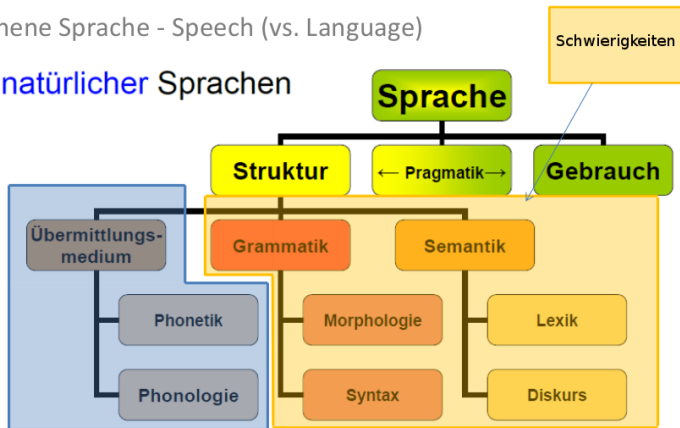
- ▶ Althammer und Schindelka in „Wie verstehen Computer natürliche Sprachen?“ um die Schwierigkeiten zu markieren,
- ▶ Cutura, Holzer A. und Sahling „Semantik in natürlichen Sprachen vs. Semantik in Computersprachen“ zu erklären,
- ▶ Berthold „Repräsentation von Semantik natürlicher Sprachen im Computer“ (Handout fehlt) zu zeigen,
- ▶ Mayer, Pointner M. „Automatisches Übersetzen“ (Handout fehlt) um Verwendung aufzuzeigen.

Interdisziplinäre Fachdidaktik: Natürliche Sprachen und Formale Sprachen (UE)

Fokus für heute

Gesprochene Sprache - Speech (vs. Language)

Modell natürlicher Sprachen



Semantik natürlicher Sprachen

Beim Hobeln fallen Späne.

- Ein Satz kann **mehrere Bedeutungen** haben
- Wird durch **unterschiedliche Semantiken** wiedergegeben
- **Wortwörtlich:** Beim Hobeln von Holz entstehen Späne
- **Übertragene Semantik:** Wenn man etwas erreichen will muss man auch etwas dafür opfern

Wie wir in „Automatisches Übersetzen“ gehört haben, wird nicht direkt in die Zielsprache übersetzt, sondern der Umweg über eine Interlingua Sprache gegangen. Es ist bekannt, dass das Übersetzen in die Plansprache verlustbehaftet ist, doch könnten Zusatzinformationen gespeichert werden.

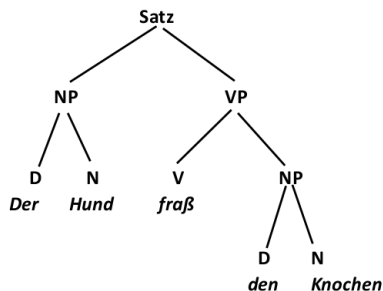


Abbildung: PART OF SPEECH Tagging

Was ich zeigen wollte

Die verschieden versteckten Anwendungen der Formalen Sprachen.

Technische Implementierung: Verwendung im CPU,

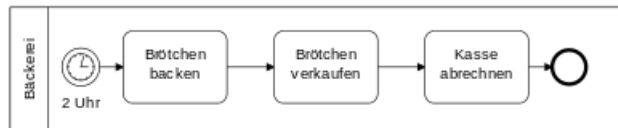
Compiler: automatische Übersetzung von einer
Programmsprache in die Maschinensprache

Maschinelle Übersetzung: Hilfsmittel, weil die Semantik die Regeln
sprengt

Die Theoretische Informatik behauptet zu recht, was sie nicht
beschreiben können, kann der PC nicht berechnen.

Was ich nicht zeigen wollte

- ▶ Assemblersprachen
- ▶ höhere Programmiersprache
- ▶ Abfragesprachen (relationale Datenbanken)
- ▶ Auszeichnungssprachen ermöglichen Texte zu formatieren
- ▶ Modellierungssprachen



Quellen

- ▶ Theoretische Informatik, SS2014
- ▶ Technische Grundlagen und Systeme, WS15
- ▶ <https://de.wikipedia.org/wiki/Lincos>
- ▶ https://de.wikipedia.org/wiki/Konstruierte_Sprache#Formale_Sprachen

Bilder

- ▶ <https://de.wikipedia.org/wiki/Turingmaschine>
- ▶ Technische Grundlagen und Systeme WS15
- ▶ Kap3_Sprachmodelle
- ▶ https://de.wikipedia.org/wiki/Business_Process_Model_and_Notation