

9. Grundbegriffe des Parsens

9.1 Deklarative und prozedurale Aspekte des Parsens

9.1.1 Deklarative und prozedurale Aspekte der Sprachanalyse

- Der *deklarative* Aspekt einer computerlinguistischen Analyse wird durch die generative Grammatik repräsentiert, die für die zu analysierende Sprache im Rahmen eines mathematisch wohldefinierten Formalismus geschrieben wurde.
- Der *prozedurale* Aspekt einer computerlinguistischen Analyse umfaßt alle Teilbereiche der Programmierung, die den generativen Formalismus bei der automatischen Analyse sprachlicher Eingaben umsetzen und verwenden.

9.1.2 Unterspezifikation der deklarative Definition bzgl. Ableitungsordnung

Regel 1: $A \rightarrow B C$

Regel 2: $B \rightarrow c d$

Regel 3: $C \rightarrow e f$

9.2 Anpassen von Grammatik auf Sprache

9.2.1 Eine kontextfreie Struktur im Deutschen

Der Mann, schläft.
 der die Frau, liebt,
 die das Kind, sieht,
 das die Katze füttert,

9.2.2 Kontextsensitive Struktur im Schweizerdeutsch

mer em Hans es huus hälfed aastriiche
wir dem Hans das Haus helfen anstreichen

9.2.3 Mögliche Schlüsse aus der Annahme, daß die natürlichen Sprachen nicht kontextfrei sind

1. Die PS-Grammatik ist der einzige Elementarformalismus der generativen Grammatik, weshalb man sich damit abfinden muß, daß die natürlichen Sprachen von einer hohen mathematischen Komplexität (und daher *computationally intractible*) sind.
2. Die PS-Grammatik ist nicht der einzige Elementarformalismus. Vielmehr gibt es andere generative Grammatiken, die alternative Sprachhierarchien definieren, die quer (orthogonal) zur PS-Hierarchie liegen.

9.2.4 Mögliche Beziehungen zwischen zwei Formalismen

- *Keine Äquivalenz*

Zwei Grammatikformalisten sind nicht äquivalent, wenn sie unterschiedliche Sprachklassen erzeugen und erkennen; das heißt, die beiden Formalismen sind von unterschiedlicher generativer Kapazität.

- *Schwache Äquivalenz*

Zwei Grammatikformalisten sind schwach äquivalent, wenn sie dieselben Sprachen erzeugen und erkennen; das heißt, die beiden Formalismen haben dieselbe generative Kapazität.

- *Starke Äquivalenz*

Zwei Grammatikformalisten sind stark äquivalent, wenn sie (i) schwach äquivalent sind und (ii) darüber hinaus dieselben Strukturbeschreibungen liefern; das heißt, die beiden Formalismen sind lediglich *notationelle Varianten*.

9.2.5 Weak equivalence between C-grammar and PS-grammar

The problem arose of determining the exact relationships between these types of [PS-]grammars and the categorial grammars. I surmised in 1958 that the BCGs [Bidirectional Categorical Grammar *à la* 7.4.1] were of approximately the same strength as [context-free phrase structure grammars]. A proof of their equivalence was found in June of 1959 by Gaifman. ... The equivalence of these different types of grammars should not be too surprising. Each of them was meant to be a precise explicatum of the notion *immediate constituent grammars* which has served for many years as the favorite type of American descriptive linguistics as exhibited, for instance, in the well-known books by Harris [1951] and Hockett [1958].

[Es stellte sich das Problem, die genaue Beziehung zwischen diesen Typen der [PS-]Grammatik und der C-Grammatik zu bestimmen. Ich vermutete 1958, daß die bidirektionale C-Grammatik [siehe Definition 7.4.1] ungefähr dieselbe generative Kapazität hat [wie die kontextfreie PS-Grammatik]. Der Beweis ihrer Äquivalenz wurde im Juni 1959 von Gaifman gefunden. ... Die Äquivalenz dieser beiden Grammatiktypen sollte nicht zu sehr überraschen. Jeder von ihnen war mit der Absicht entwickelt worden, den Begriff *immediate constituent grammars* präzise zu explizieren. Dieser Begriff hat viele Jahre lang als die favorisierte Form der deskriptiven Linguistik in Amerika gedient, wie die bekannten Bücher von Harris 1951 und Hockett 1958 zeigen.]

Y. Bar-Hillel 1960 [1964, p. 103]

9.2.6 Allgemeinen Beziehungen zwischen den Begriffen

Sprache, generative Grammatik, Untertypen von Grammatiken, Sprachklassen, Parser und Komplexität

- *Sprachen* existieren unabhängig von generativen Grammatiken. Eine Sprache kann von verschiedenen Grammatiken aus verschiedenen Formalismen beschrieben werden.
- Eine *generative Grammatik* ist einerseits ein allgemeiner formaler Rahmen, andererseits ein spezifisches Regelsystem, das innerhalb des allgemeinen Rahmens für eine bestimmte Sprache geschrieben wurde.
- *Untertypen von generativen Grammatiken* werden über verschiedene Beschränkungen des verwendeten Grammatiktyps (insbesondere seiner Regelstruktur) definiert.
- *Sprachklassen* entstehen aus den Untertypen eines generativen Grammatikformalismus.
Nota bene: *Sprachen* existieren unabhängig von den tatsächlichen oder möglichen formalen Grammatiken, die sie generieren. *Sprachklassen* werden dagegen über spezifische Beschränkungen spezifischer Grammatikformalismen definiert.
- *Parser* sind automatische Analyseprogramme, die für einen ganzen Untertyp einer generativen Grammatik funktionieren.
- Die *Komplexität* eines Grammatiktyps wird über die Zahl der Rechenschritte (*primitive operations*) bestimmt, die schlimmstenfalls von äquivalenten abstrakten Automaten oder Parsingprogrammen bei der Analyse zugehöriger Sprachausdrücke benötigt wird.

9.3 Typentransparenz zwischen Grammatik und Parser

9.3.1 Die natürliche Auffassung eines Parsers als Motor einer Grammatik

Miller and Chomsky's original (1963) suggestion is really that grammars be realized more or less directly as parsing algorithms. We might take this as a methodological principle. In this case we impose the condition that the logical organization of rules and structures incorporated in the grammar be mirrored rather exactly in the organization of the parsing mechanism. We will call this *type transparency*.

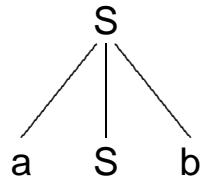
[Der ursprüngliche Vorschlag von Miller und Chomsky (1963) war eigentlich, daß Grammatiken mehr oder weniger direkt als Parsingalgorithmen zu realisieren seien. Wir können dies zu einem methodischen Prinzip machen. Das heißt, wir verlangen, daß die logische Organisation der Regeln und die Struktur der Grammatik in der Organisation des Parsingmechanismus sehr genau wiedergespiegelt wird. Dieses methodische Prinzip nennen wir *Typentransparenz*.]

R.C. Berwick & A.S. Weinberg 1984, p. 39.

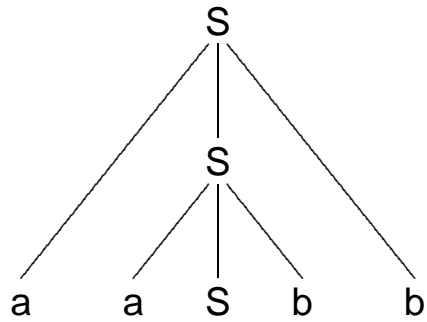
9.3.2 Definition der absoluten Typentransparenz

- Für eine gegebene Sprache verwenden Parser und Generator dieselbe formale Grammatik,
- wobei Parser und Generator die Regeln der Grammatik direkt anwenden.
- Das heißt insbesondere, daß Parser und Generator die Regeln in derselben Reihenfolge anwenden wie die grammatische Ableitung,
- daß Parser und Generator bei jeder Regelanwendung dieselben Eingaben nehmen wie die Grammatik und
- daß Parser und Generator bei jeder Regelanwendung dieselben Ausgaben ergeben wie die Grammatik.

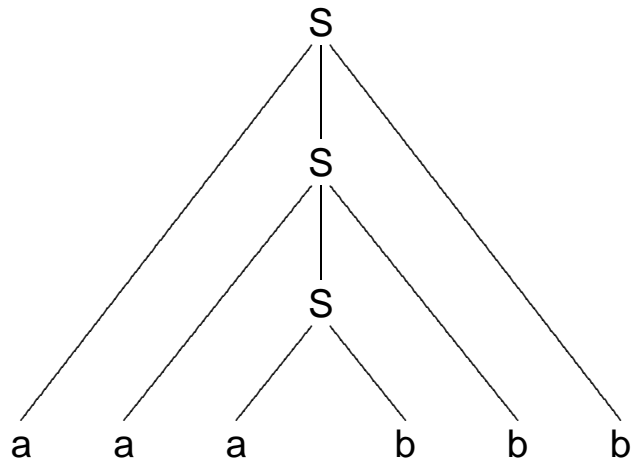
9.3.3 Top-down Ableitungsstruktur von a a a b b b



$S \longrightarrow a S b$ Schritt 1

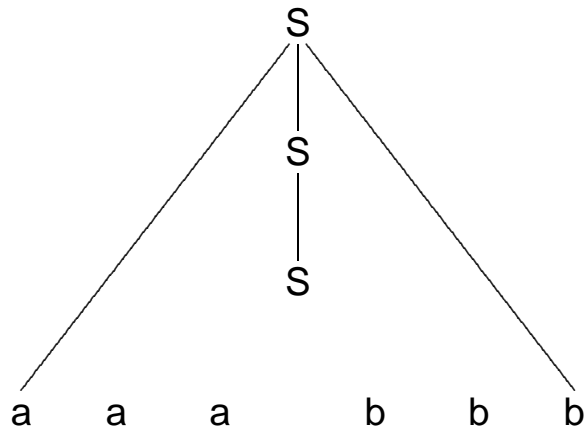


$S \longrightarrow a S b$ Schritt 2

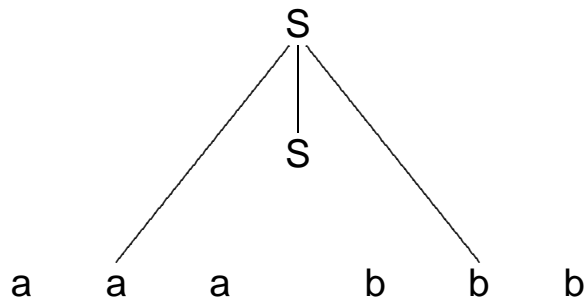


$S \longrightarrow a b$ Schritt 3

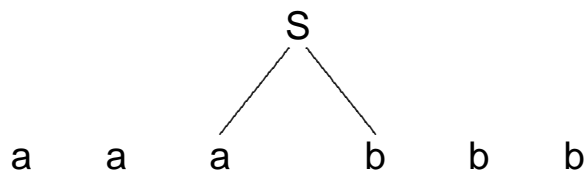
9.3.4 Bottom-up Ableitungsstruktur von a a a b b b



$S \leftarrow a S b$ Schritt 3



$S \leftarrow a S b$ Schritt 2



$S \leftarrow a b$ Schritt 1

9.3.5 Der Earley-Algorithmus am Beispiel von $a^k b^k$

.aaabbb

.S

| a.aabbb

.ab -> a.b

.aSb -> a.Sb

| aa.aabbb

a.abb -> aa.bb

a.aSbb -> aa.Sbb

| aaa.bbb aaab.bb

aa.aabbb -> aaa.bbb -> aaab.bb-> ...

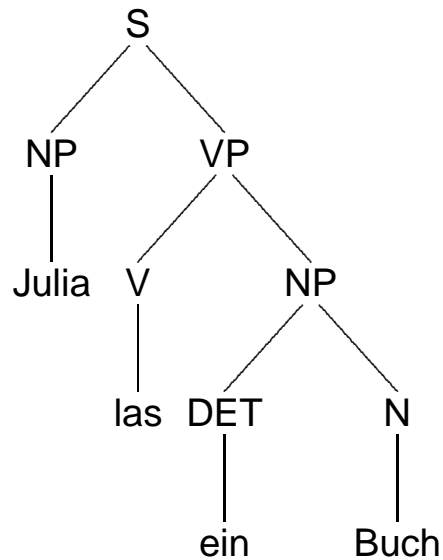
aa.aSbbb -> aaa.Sbbb

9.4 Ein-Ausgabeäquivalenz mit dem Sprecher-Hörer

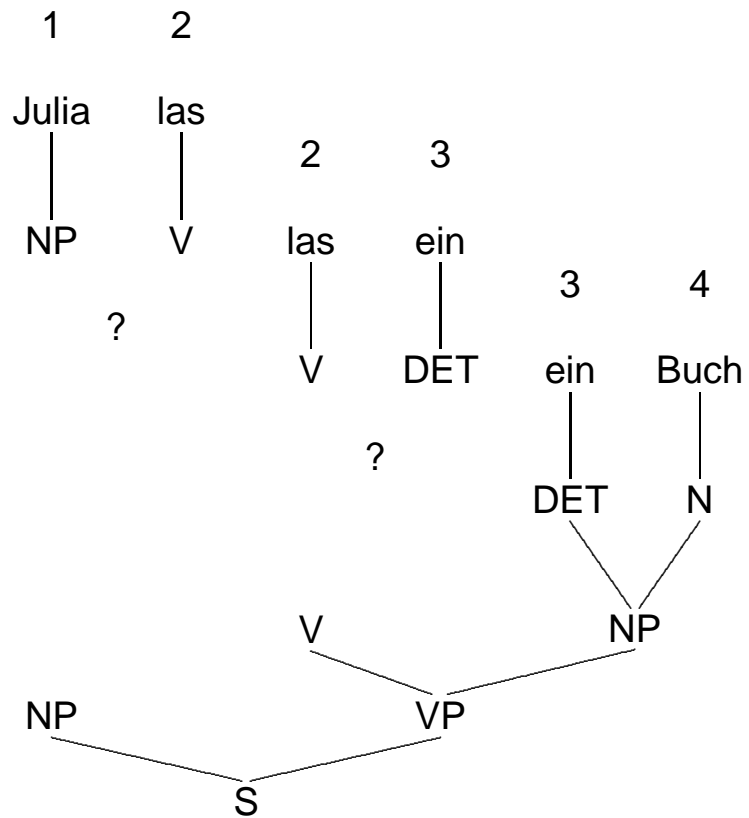
9.4.1 Eine kontextfreie PS-Grammatik

- | | | | |
|-------|---------|--------|--------|
| 1. S | → NP VP | 5. V | → las |
| 2. NP | → DET N | 6. DET | → ein |
| 3. VP | → V NP | 7. N | → Buch |
| 4. NP | → Julia | | |

9.4.2 PS-grammatische Analyse (*top-down-Ableitung*)



9.4.3 Zeitlinearer Analyse-Versuch in der PS-Grammatik



9.5 Konvergenz-Desiderata an einen Grammatikformalismus

9.5.1 Symptome fehlender Konvergenz im Nativismus

- Statt einer Konsolidierung des ursprünglichen Elementarsystems werden ständig neue abgeleitete Systeme entwickelt.
- Die als notwendig erachtete Einführung zusätzlicher Mechanismen (Transformationen, Metaregeln etc.) führt ausnahmslos zu einer Verschlechterung der mathematischen und programmiertechnischen Eigenschaften des ursprünglichen Formalismus kontextfreier PS-Grammatik.
- Die empirische Beschreibung natürlicher Sprache führt ständig zu Problemen vom Typ ‘deskriptive Aporie’ und ‘Qual der Wahl’.
- Die theoretischen Konstrukte des Nativismus werden von praktischen Systemen der Sprachverarbeitung höchstens mit Lippenbekenntnissen beachtet, meist aber ganz ignoriert.

9.5.2 Ursachen der fehlenden Konvergenz im Nativismus

- Der Nativismus ist empirisch unterspezifiziert weil er keine funktionale Sprachtheorie enthält.
- Der vom Nativismus verwendete Formalismus der PS-Grammatik ist mit den Ein-Ausgabebedingungen des Sprecher-Hörers nicht kompatibel.

9.5.3 Eigenschaften der PS-Grammatik

- *Mathematisch:*

Praktisch verwendbare Parsingalgorithmen existieren nur für die kontextfreie PS-Grammatik. Diese ist zwar von ausreichend niedriger Komplexität (n^3), aber nicht von ausreichend hoher generativer Kapazität für die natürlichen Sprachen. Erweiterungen der generativen Kapazität sind dagegen mathematisch so komplex (unentscheidbar oder exponentiell), daß es für sie keine praktisch verwendbaren Parsingalgorithmen geben kann.

- *Programmiertechnisch:*

Die PS-Grammatik ist nicht typentransparent. Dies erschwert die Fehlersuche und Erweiterung von Grammatiken mit automatisch erstellten Parsingprotokollen. Außerdem erfordert die indirekte Beziehung zwischen Grammatik und Parsingalgorithmus den Einsatz aufwendiger Routinen und umfangreicher Zwischenstrukturen.

- *Empirisch:*

Die substitutionsbasierte Ableitungsordnung der PS-Grammatik widerspricht der zeitlinearen Grundstruktur natürlicher Sprachen und ist mit den Bedingungen der natürlichen Kommunikation nicht in Einklang zu bringen.

9.5.4 Desiderata an generative Grammatikformalismen

1. Der Formalismus sollte mathematisch einwandfrei definiert sein und somit
2. eine *deklarative* Spezifikation der Strukturen in natürlichen und künstlichen Sprachen erlauben.
3. Dabei sollte der Formalismus *rekursiv* (und damit entscheidbar)
4. sowie *typentransparent* in bezug auf seine Parser und Generatoren sein.
5. Der Formalismus sollte über strukturell offensichtliche Einschränkungen des Regelapparats eine *Hierarchie verschiedener Sprachklassen* definieren, analog – aber orthogonal – zur PS-grammatischen Hierarchie,
6. wobei diese Hierarchie eine Sprachklasse von niedriger, möglichst linearer Komplexität enthält, deren *generative Kapazität* für die Analyse der natürlichen Sprachen ausreicht.
7. Der Formalismus sollte *ein-ausgabeäquivalent* mit dem Sprecher-Hörer sein und somit eine zeitlineare Ableitungsordnung verwenden.
8. Der Formalismus sollte für *Produktion* (im Sinne einer Abbildung von Bedeutungen auf Oberflächen) und *Interpretation* (im Sinne einer Abbildung von Oberflächen auf Bedeutungen) gleichermaßen geeignet sein.