9. Basic notions of parsing

9.1 Declarative and procedural aspects of parsing

9.1.1 Declarative & procedural aspects in linguistics

- The *declarative* aspect of computational language analysis is represented by a generative grammar, written for the specific language to be analyzed within a general, mathematically well-defined formalism.
- The *procedural* aspect of computational language analysis comprises those parts of the computer program which interpret and apply the general formalism in the automatic analysis of language input.

9.1.2 Example

rule 1: A $\rightarrow$ B C
rule 2: B $\rightarrow$ c d
rule 3: C $\rightarrow$ e f

9.2 Fitting grammar onto language

9.2.1 Context-free structure in German

Der Mann, schläft.
(*the man*) (*sleeps*).
der die Frau, liebt,
(*who the woman*) (*loves*)
die das Kind, sieht,
(*who the child*) (*sees*)
das die Katze füttert,
(*who the cat*) (*feeds*)

9.2.2 Alternative implications of natural language not being context-free

1. PS-grammar is the only elementary formalism of generative grammar, for which reason one must accept that the natural languages are of high complexity and thus computationally intractable.
2. PS-grammar is not the only elementary formalism of generative grammar. Instead, there are other elementary formalisms which define other language hierarchies whose language classes are orthogonal to those of PS-grammar.

9.2.3 Possible relations between two grammar formalisms

- *No equivalence*
  Two grammar formalisms are not equivalent, if they generate/recognize different language classes; this means that the two formalisms are of different generative capacity.
- *Weak equivalence*
  Two grammar formalisms are weakly equivalent, if they generate/recognize the same language classes; this means that the two formalisms have the same generative capacity.
- *Strong equivalence*
  Two grammar formalisms are strongly equivalent, if they are (i) weakly equivalent, and moreover (ii) produce the same structural descriptions; this means that the two formalisms are no more than *notational variants*.

9.2.4 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 Categorial Grammar *á 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].
>
> Y. Bar-Hillel 1960 [1964, p. 103]

9.2.5 General relations between notions of generative grammar

- *Languages* exist independently of generative grammars. A given language may be described by different formal grammars of different grammar formalisms.
- *Generative grammars* is (i) a general formal framework or (ii) a specific rule system defined for describing a specific language within the general framework.
- *Subtypes of generative grammars* result from different restrictions on the formal framework.
- *Language classes*
  The subtypes of a generative grammar may be used to divide the set of possible languages into different language classes.

  Nota bene: *languages* exist independently of the formal grammars which may generate them. The *language classes*, on the other hand, do not exist independently, but result from particular restrictions on particular grammar formalisms.
- *Parsers*
  Parsers are programs of automatic language analysis which are defined for whole subtypes of generative grammars.
- *Complexity*
  The complexity of a subtype of generative grammar is determined over the number of *primitive operations* needed by an equivalent abstract automaton or parsing program for analyzing expressions in the worst case.

9.3 Type transparency between grammar and parser

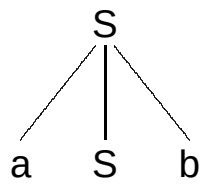
9.3.1 Natural view of a parser as the motor or driver of a grammar

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

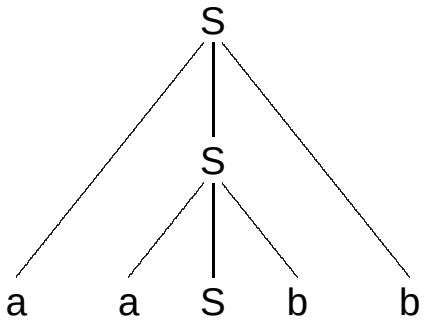
9.3.2 Definition of absolute type transparency

- For any given language, parser and generator use the *same* formal grammar,
- whereby the parser/generator applies the rules of the grammar *directly*.
- This means in particular that the parser/generator applies the rules in the *same order* as the grammatical derivation,
- that in each rule application the parser/generator takes the *same input* expressions as the grammar, and
- that in each rule application the parser/generator produces the *same output* expressions as the grammar.

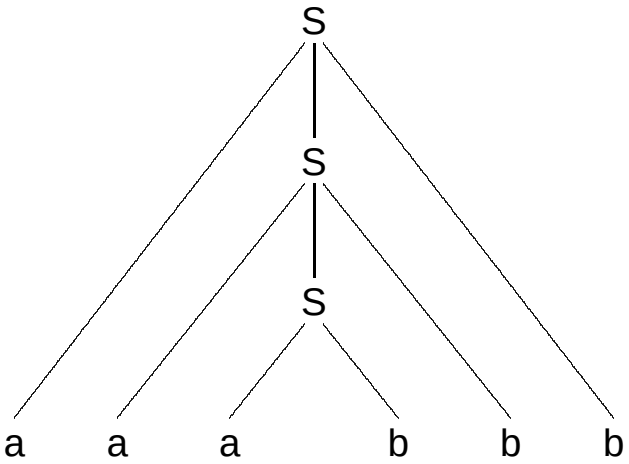
9.3.3 *Top-down* derivation of a a a b b b



S → a S b step 1

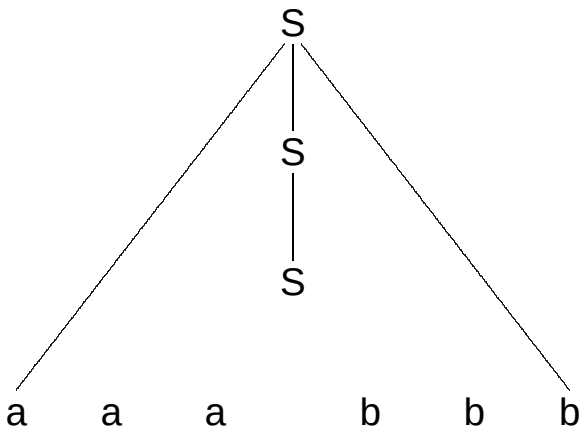


S → a S b step 2

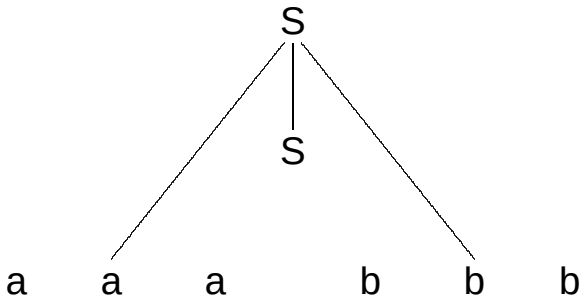


S → a b step 3

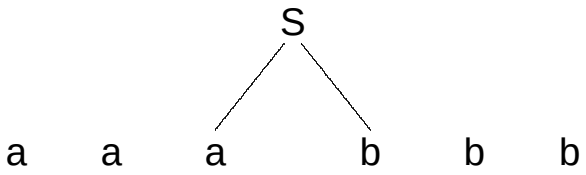
9.3.4 *Bottom-up* derivation of a a a b b b



S ⟵ a S b step 3

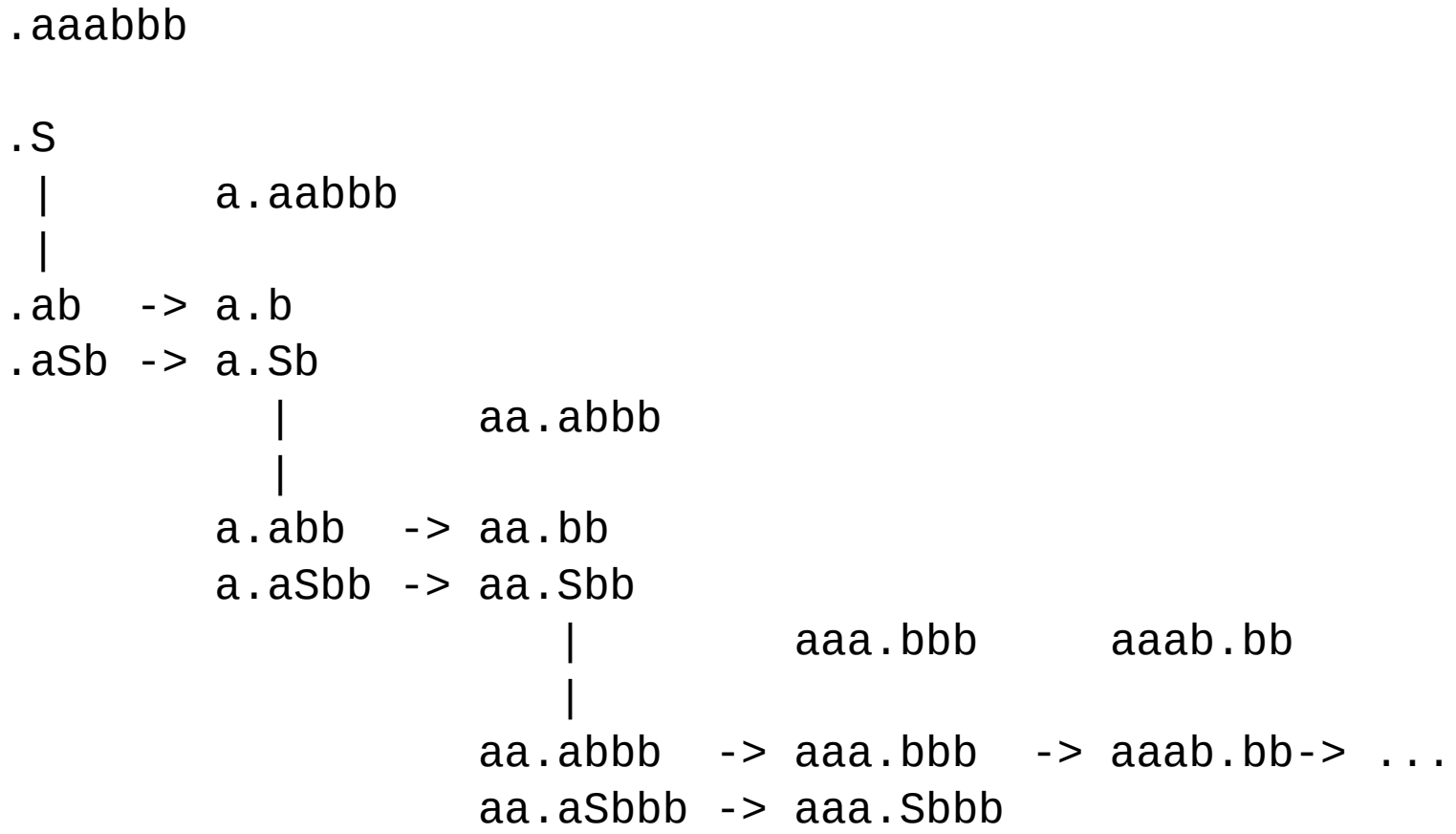


S ⟵ a S b step 2



S ⟵ a b step 1

9.3.5 The Earley algorithm analyzing $a^k b^k$

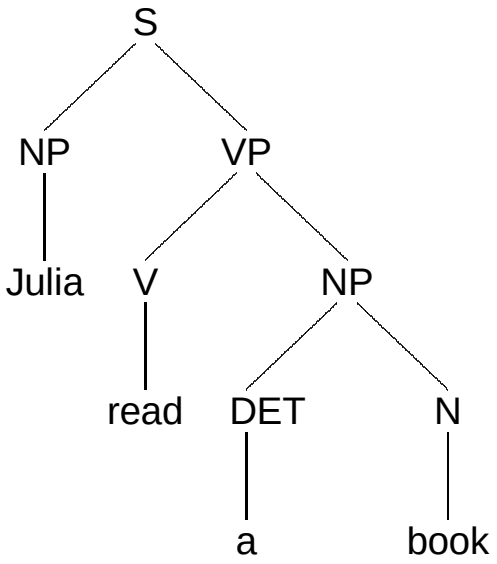


9.4 Input-output equivalence with the speaker-hearer

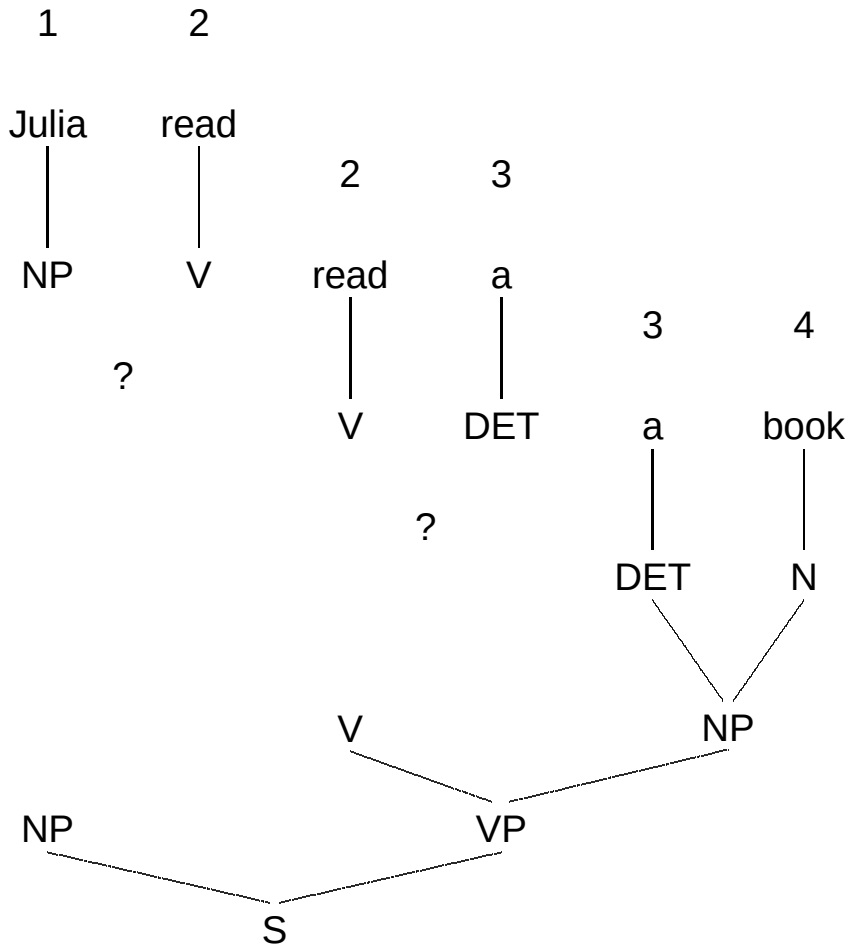
9.4.1 Context-free PS-grammar for a simple sentence of English

1. S → NP VP
2. NP → DET N
3. VP → V NP
4. NP → Julia
5. V → read
6. DET → a
7. N → book

9.4.2 PS-grammar analysis (*top-down* derivation)



9.4.3 Attempt of a time-linear analysis in PS-grammar



9.5 Desiderata of grammar for achieving convergence

9.5.1 Symptons of lacking convergence in nativism

- Development of ever new derived systems instead of consolidation.
- Additional mechanisms regarded as descriptively necessary have consistently degraded mathematical and computational properties.
- Empirical work has lead continuously to problems of the type descriptive aporia and embarrassment of riches.
- Practical systems of natural language processing pay either only lip service to the theoretical constructs of nativism or ignore them altogether.

9.5.2 Reasons for lacking convergence of nativism

- Nativism is empirically underspecified because it does not include a functional theory of communication.
- The PS-grammar formalism adopted by nativism is incompatible with the input-output conditions of the speaker-hearer.

9.5.3 Properties of PS-grammar

- *Mathematical:*
  Practical parsing algorithms exist only for context-free PS-grammar. It is of a sufficiently low complexity (n^3), but not of sufficient generative capacity for natural language. Extensions of the generative capacity for the purpose of describing natural language turned out to be of such high complexity (undecidable or exponential) that no practical parse algorithm can exist for them.
- *Computational:*
  PS-grammar is not type transparent. This prevents using the automatic traces of parsers for purposes of debugging and upscaling grammars. Furthermore, the indirect relation between the grammar and the parsing algorithm requires the use of costly routines and large intermediate structures.
- *Empirical:*
  The substitution-based derivation order of PS-grammar is incompatible with the time-linear structure of natural language.

9.5.4 Desiderata of a generative grammar formalism

1. The grammar formalism should be mathematically well-defined and thus
2. permit an explicit, declarative description of artificial and natural languages.
3. The formalism should be recursive (and thus decidable) as well as
4. type transparent with respect to its parsers and generators.
5. The formalism should define a hierarchy of different language classes in terms of structurally obvious restrictions on its rule system (analogous – but orthogonal – to the PS-grammar hierarchy),
6. whereby the hierarchy contains a language class of low, preferably linear, complexity the generative capacity of which is sufficient for a complete description of natural language.
7. The formalism should be input-output equivalent with the speaker-hearer (and thus use a time-linear derivation order).
8. The formalism should be suited equally well for production (in the sense of mapping meanings into surfaces) and interpretation (in the sense of mapping surfaces into meanings).