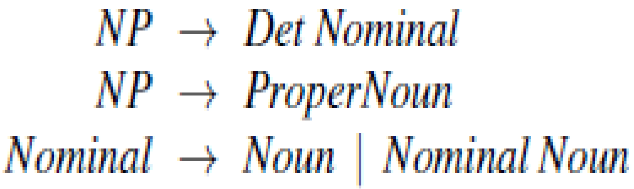
CFG

Context Free Grammars

- A widely used formal system for modeling constituent structure in natural language is the context-free grammar, or CFG
- Context-free grammars are also called phrase-structure grammars
- A context-free grammar consists of a set of rules or productions, each of which expresses the ways that symbols of the language can be grouped and ordered together, and a lexicon of words and symbols.
- For example, the following productions express that an NP (or noun phrase) can be composed of either a ProperNoun or a determiner (Det) followed by a Nominal; a Nominal in turn can consist of one or more Nouns
-



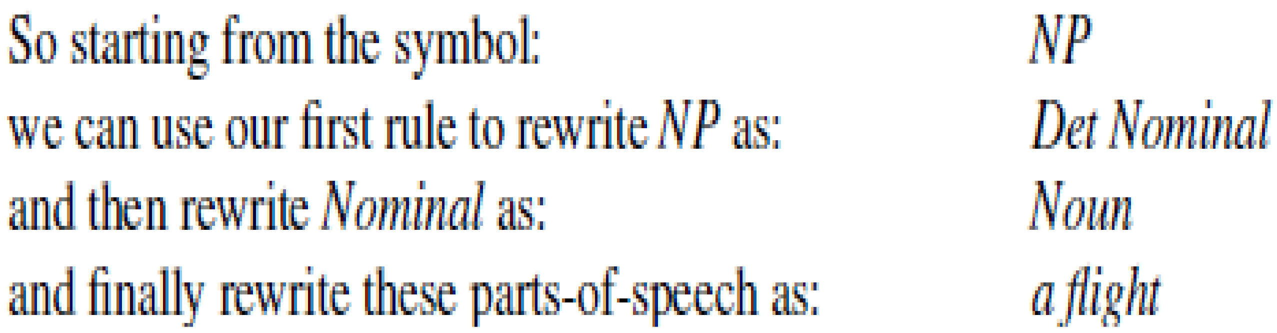
$$\begin{aligned} NP &\rightarrow \mathit{Det\ Nominal} \\ NP &\rightarrow \mathit{ProperNoun} \\ \mathit{Nominal} &\rightarrow \mathit{Noun} \mid \mathit{Nominal\ Noun} \end{aligned}$$

cntd

$S \rightarrow aSa$

$S \rightarrow a|b$

- The symbols that are used in a CFG are divided into two classes.
- The symbols that correspond to words in the language (“the”, “nightclub”) are called terminal symbols; the lexicon is the set of rules that introduce these terminal symbols.
- The symbols that express abstractions over these terminals are called non-terminals.
- In each context-free rule, the item to the right of the arrow (->) is an ordered list of one or more terminals and non-terminals; to the left of the arrow is a single non-terminal symbol expressing some cluster or generalization.
- The non-terminal associated with each word in the lexicon is its lexical category, or part of speech.
- A CFG can be thought of in two ways: as a device for generating sentences and as a device for assigning a structure to a given sentence.
- Viewing a CFG as a generator, we can read the -> arrow as “rewrite the symbol on the left with the string of symbols on the right”.



So starting from the symbol:	*NP*
we can use our first rule to rewrite *NP* as:	*Det Nominal*
and then rewrite *Nominal* as:	*Noun*
and finally rewrite these parts-of-speech as:	*a flight*

- We say the string a flight can be derived from the non-terminal NP. Thus, a CFG can be used to generate a set of strings.
- This sequence of rule expansions is called a derivation of the string of words.
- It is common to represent a derivation by a parse tree (commonly shown inverted with the root at the top). Figure 17.1 shows the tree representation of this derivation.

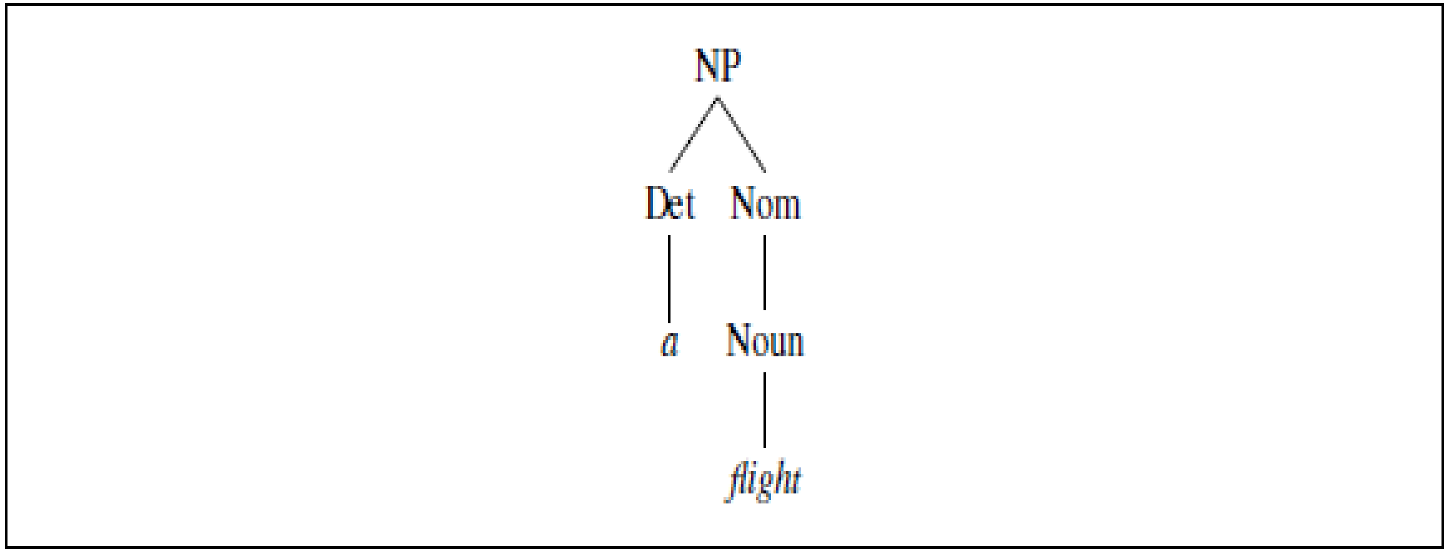
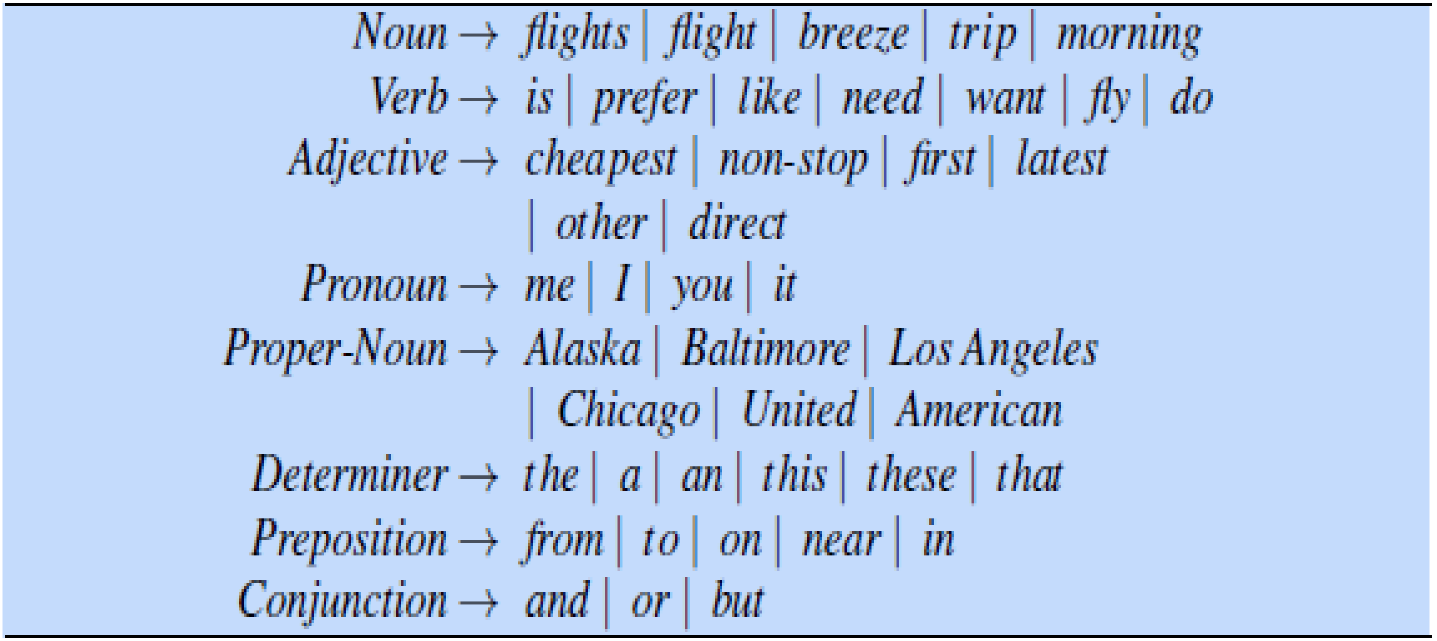


Figure 17.1 A parse tree for “a flight”.

- In the parse tree shown in Fig. 17.1, we can say that the node NP dominates all the nodes in the tree (Det, Nom, Noun, a, flight).
- We can say further that it immediately dominates the nodes Det and Nom.
- The formal language defined by a CFG is the set of strings that are derivable from the designated start symbol.
- Each grammar must have one designated start symbol, which is often called S.
- Since context-free grammars are often used to define sentences, S is usually interpreted as the “sentence” node, and the set of strings that are derivable from S is the set of sentences in some simplified version of English.

S → NP VP .

- Figure 17.2 gives a sample lexicon, and Fig. 17.3 summarizes the grammar rules we’ve seen so far, which we’ll call L0.
- Note that we can use the or-symbol | to indicate that a non-terminal has alternate possible expansions.



Noun → *flights* | *flight* | *breeze* | *trip* | *morning*
Verb → *is* | *prefer* | *like* | *need* | *want* | *fly* | *do*
Adjective → *cheapest* | *non-stop* | *first* | *latest*
| *other* | *direct*
Pronoun → *me* | *I* | *you* | *it*
Proper-Noun → *Alaska* | *Baltimore* | *Los Angeles*
| *Chicago* | *United* | *American*
Determiner → *the* | *a* | *an* | *this* | *these* | *that*
Preposition → *from* | *to* | *on* | *near* | *in*
Conjunction → *and* | *or* | *but*

Figure 17.2 The lexicon for $\mathcal{L}_0$.

S → NP VP

- We can use this grammar to generate sentences of this “ATIS-language”.
- We start with S, expand it to NP VP, then choose a random expansion of NP (let’s say, to I), and a random expansion of VP (let’s say, to Verb NP), and so on until we generate the string I prefer a morning flight.
- Figure 17.4 shows a parse tree that represents a complete derivation of *I prefer a morning flight.*

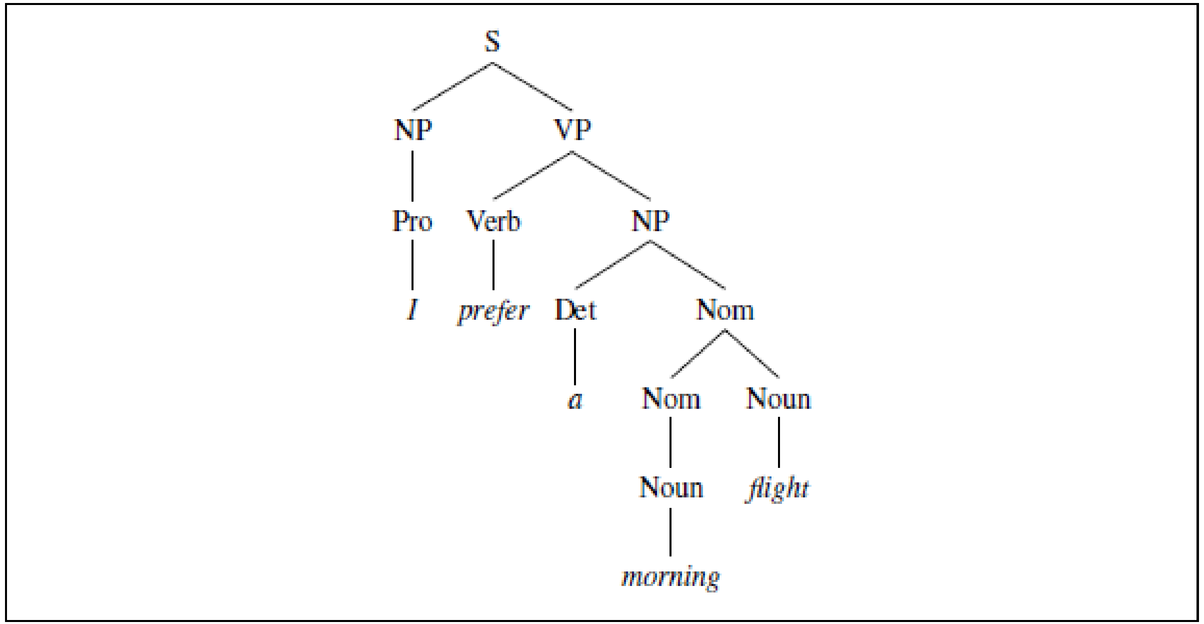


Figure 17.4 The parse tree for “I prefer a morning flight” according to grammar $\mathcal{L}_0$.

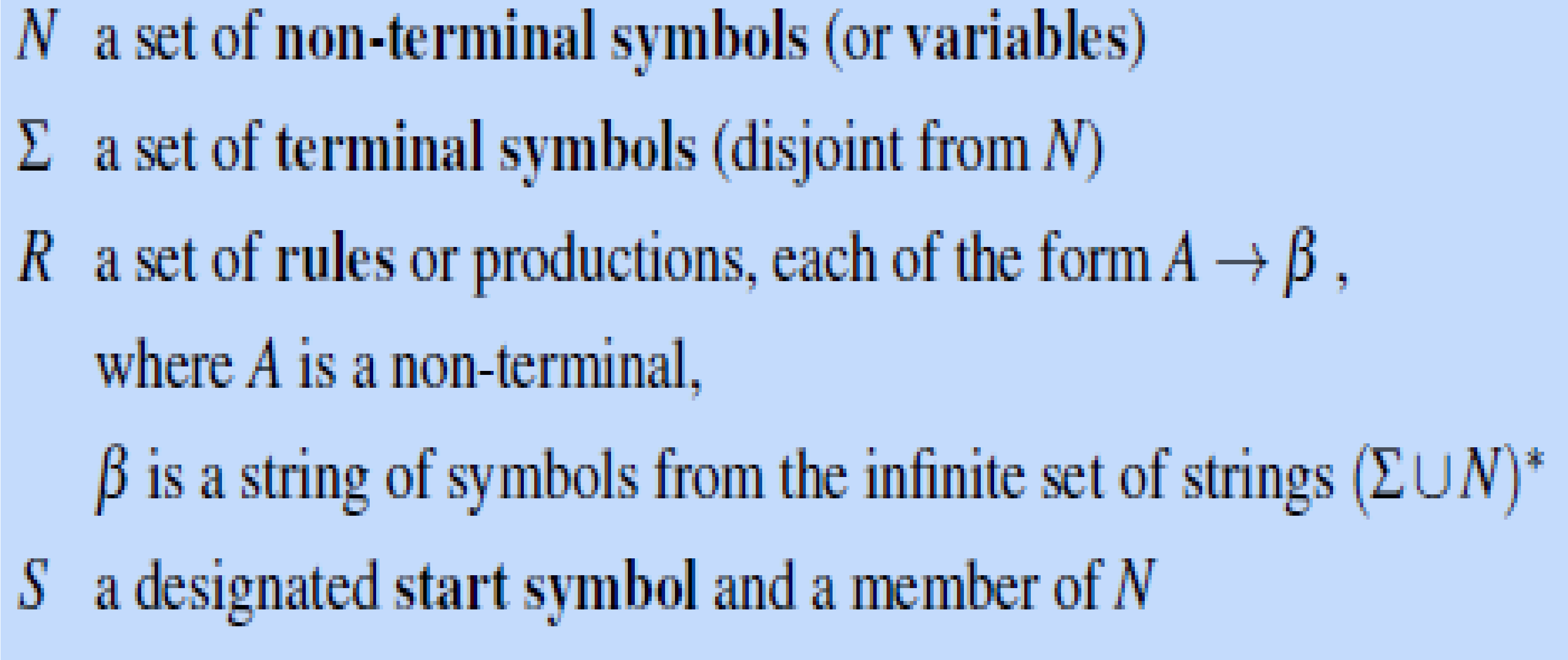
Bracketed Notation

- We can also represent a parse tree in a more compact format called bracketed notation; here is the bracketed representation of the parse tree of Fig. 17.4:

[S [NP [Pro I]] [VP [V prefer] [NP [Det a] [Nom [N morning] [Nom [N flight]]]]]]

Formal Definition of CFG

- A context-free grammar G is defined by four parameters: N; ***Σ***; R; S (technically it is a “4-tuple”).



N a set of **non-terminal symbols** (or **variables**)

Σ a set of **terminal symbols** (disjoint from N)

R a set of **rules** or **productions**, each of the form $A \rightarrow \beta$,

where A is a non-terminal,

β is a string of symbols from the infinite set of strings $(\Sigma \cup N)^*$

S a designated **start symbol** and a member of N

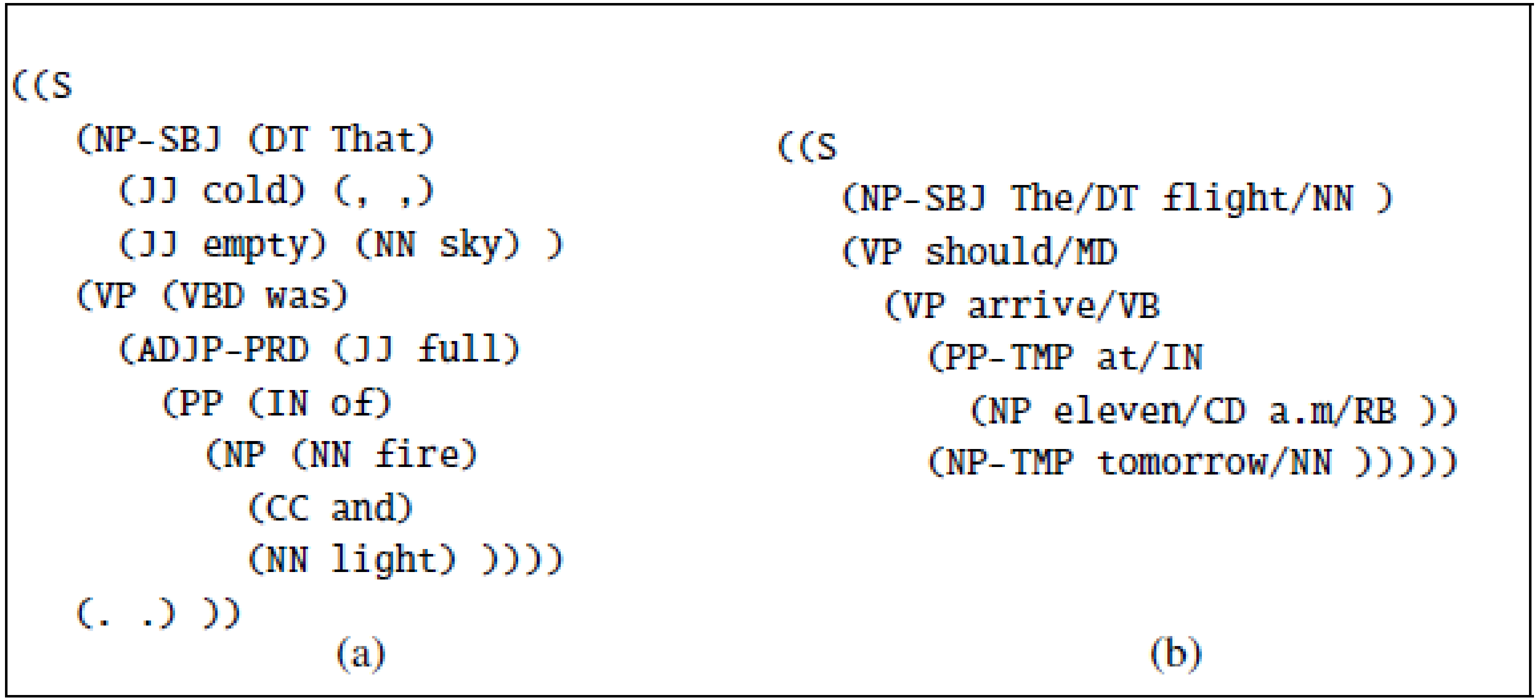
Treebanks

Treebanks

- A corpus in which every sentence is annotated with a parse tree is called a treebank
- Treebanks play an important role in parsing as well as in linguistic investigations of syntactic phenomena
- Treebanks are generally made by parsing each sentence with a parse that is then hand-corrected by human linguists.

Cntd

- Figure shows sentences from the Penn Treebank project, which includes various treebanks in English, Arabic, and Chinese.



```
((S
   (NP-SBJ (DT That)
     (JJ cold) (, ,)
     (JJ empty) (NN sky) )
   (VP (VBD was)
     (ADJP-PRD (JJ full)
       (PP (IN of)
         (NP (NN fire)
           (CC and)
           (NN light) ))))
   (. .) ))
```

(a)

```
((S
   (NP-SBJ The/DT flight/NN )
   (VP should/MD
     (VP arrive/VB
       (PP-TMP at/IN
         (NP eleven/CD a.m/RB ))
       (NP-TMP tomorrow/NN )))))
```

(b)

Figure 17.5 Parses from the LDC Treebank3 for (a) Brown and (b) ATIS sentences.

- We show a standard node-and-line tree representation in following Figure

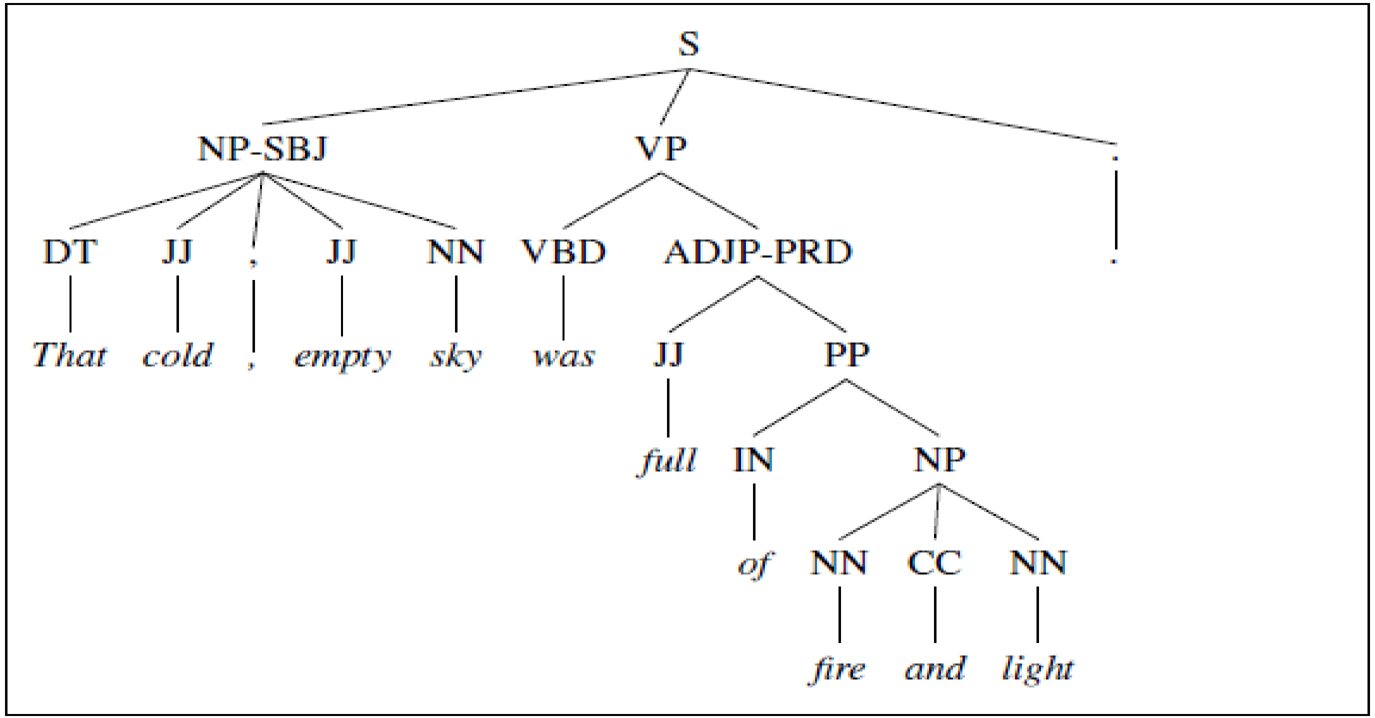
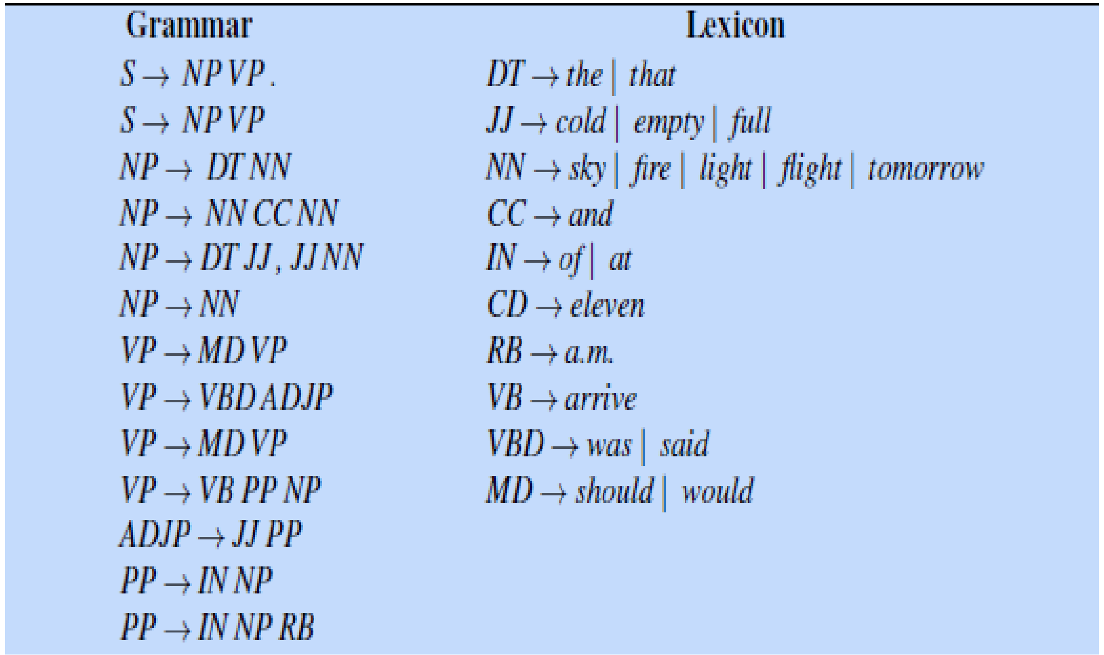


Figure 17.6 The tree corresponding to the Brown corpus sentence in the previous figure.

- The sentences in a treebank implicitly constitute a grammar of the language.
- For example, from the parsed sentences in Fig. 17.5 we can extract the CFG rules shown in Fig. 17.7 (with rule suffixes (-SBJ) stripped for simplicity).

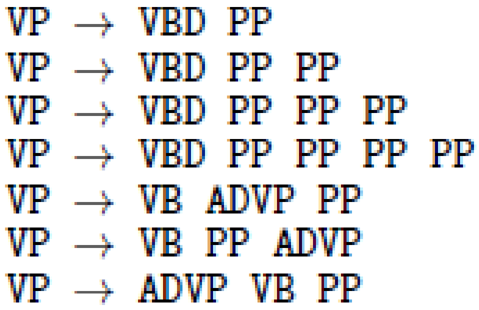


Grammar	Lexicon
S → *NP VP .*	*DT* → *the* | *that*
S → *NP VP*	*JJ* → *cold* | *empty* | *full*
NP → *DT NN*	*NN* → *sky* | *fire* | *light* | *flight* | *tomorrow*
NP → *NN CC NN*	*CC* → *and*
NP → *DT JJ , JJ NN*	*IN* → *of* | *at*
NP → *NN*	*CD* → *eleven*
VP → *MD VP*	*RB* → *a.m.*
VP → *VBD ADJP*	*VB* → *arrive*
VP → *MD VP*	*VBD* → *was* | *said*
VP → *VB PP NP*	*MD* → *should* | *would*
ADJP → *JJ PP*	
PP → *IN NP*	
PP → *IN NP RB*	

Figure 17.7 CFG grammar rules and lexicon from the treebank sentences in Fig. 17.5.

CNTD

- The grammar used to parse the Penn Treebank is very flat, resulting in very many rules
- For example, among the approximately 4,500 different rules for expanding VPs are separate rules for PP sequences of any length and every possible arrangement of verb arguments:



VP → VBD PP
VP → VBD PP PP
VP → VBD PP PP PP
VP → VBD PP PP PP PP
VP → VB ADVP PP
VP → VB PP ADVP
VP → ADVP VB PP