Context-Free Grammars

Prof. Shrisha Rao
IIIT Bangalore

srao@iiitb.ac.in

2025-08-28

Parts of a Language

A *language* is just a set of strings in an alphabet. *Or is it?*

- Syntax: grammar rules for proper statements in the language.
- Semantics: the meaning or real-life consequence of syntactically valid statements in a language.
- Pragmatics: what statements are appropriate to use, and in what circumstances.

Syntax, Semantics, and Pragmatics

- Syntax is often given in terms of a *grammar* for a language. (Think about grammars for the real-life languages you know.) It can also be given by syntax rules (e.g., HTML syntax).
- Semantics is the mapping of meaning to proper statements in a language; it can also refer, e.g., to the way a web browser displays something based on specific HTML commands.
- Pragmatics deals with knowing when some statement is proper to make; e.g., not to make false statements, not to use impolite language, avoiding GOTO statements in imperative programming.

Natural vs. Programming Languages

- Natural languages are considerably more rich than programming languages, and feature allegory, idioms, sarcasm, humor, etc.
- Not every syntactically correct statement in a natural language has a proper meaning; e.g., Chomsky's "Colorless green ideas sleep furiously." There can also be disagreements about whether a particular statement is syntactically correct or not.
- Some natural-language statements may also be ambiguous, e.g., "Visiting relatives can be boring." Their meaning can also change by punctuation and emphasis.
- Programming, markup, and such languages are carefully designed so that every syntactically valid statement has a meaning, and is unambiguous.

A WOMAN: WITHOUT HER, MAN IS NOTHING.

I LIKE COOKING,
MY FRIENDS & PETS

I WOULD LIKE TO

THANK MY PARENTS,

AYN RAND,

AND GOD.

Grammars

A *grammar* is a four-tuple $\langle \Sigma, N, P, S \rangle$ where the parts are:

- A finite alphabet Σ, also called *terminal symbols* in the language.
- A finite set N of *non-terminal* symbols or *syntactic categories*.
- A finite set P of *production rules*.
- A distinguished nonterminal S, called the *start symbol*.

A Grammar for English

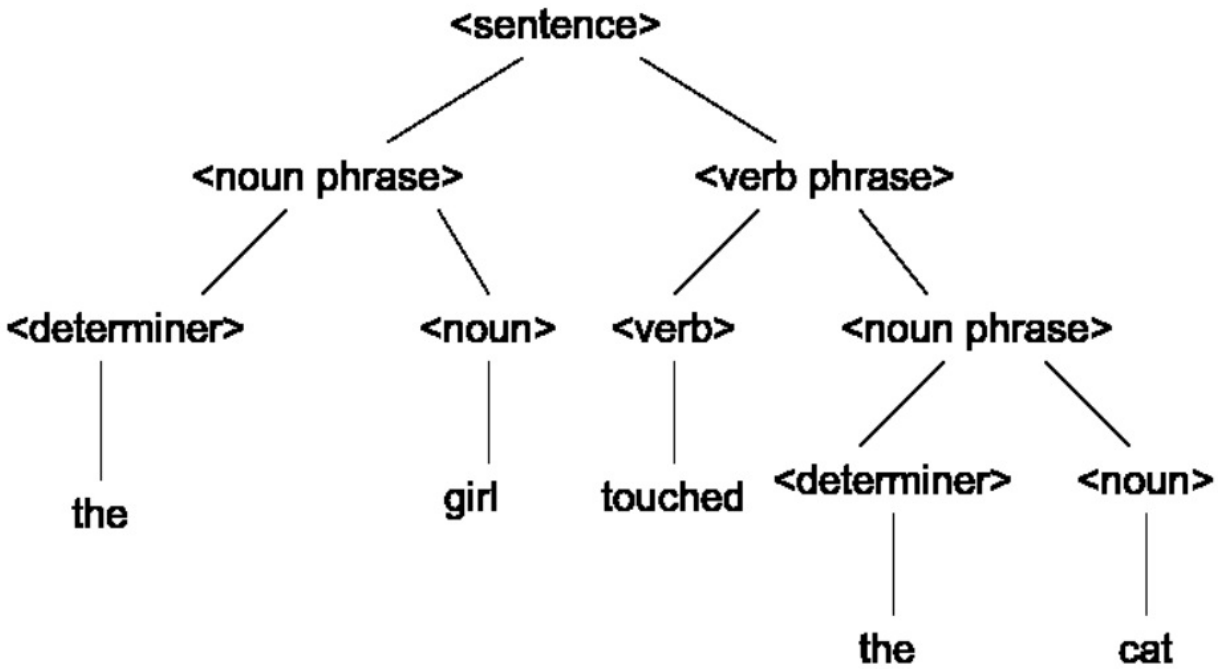
```
<sentence> := <noun phrase><verb phrase>
<noun phrase> := <determiner> <noun> |
                 <determiner> <noun> <prepositional phrase>
<verb phrase> := <verb> | <verb> <noun phrase> |
                 <verb> <noun phrase> <prepositional phrase>
<prepositional phrase> := <preposition> <noun phrase>
<noun> := boy | girl | cat | telescope | song | feather
<determiner> := a | the
<verb> := saw | touched | surprised | sang
<preposition> := by | with | of
```

A sample derivation

Consider a derivation of "the girl touched the cat."

```
<sentence> := <noun phrase> <verb phrase>
           := <determiner> <noun> <verb phrase>
           := the <noun> <verb phrase>
           := the girl <verb> <noun phrase>
           := the girl touched <determiner> <noun>
           := the girl touched the cat
```

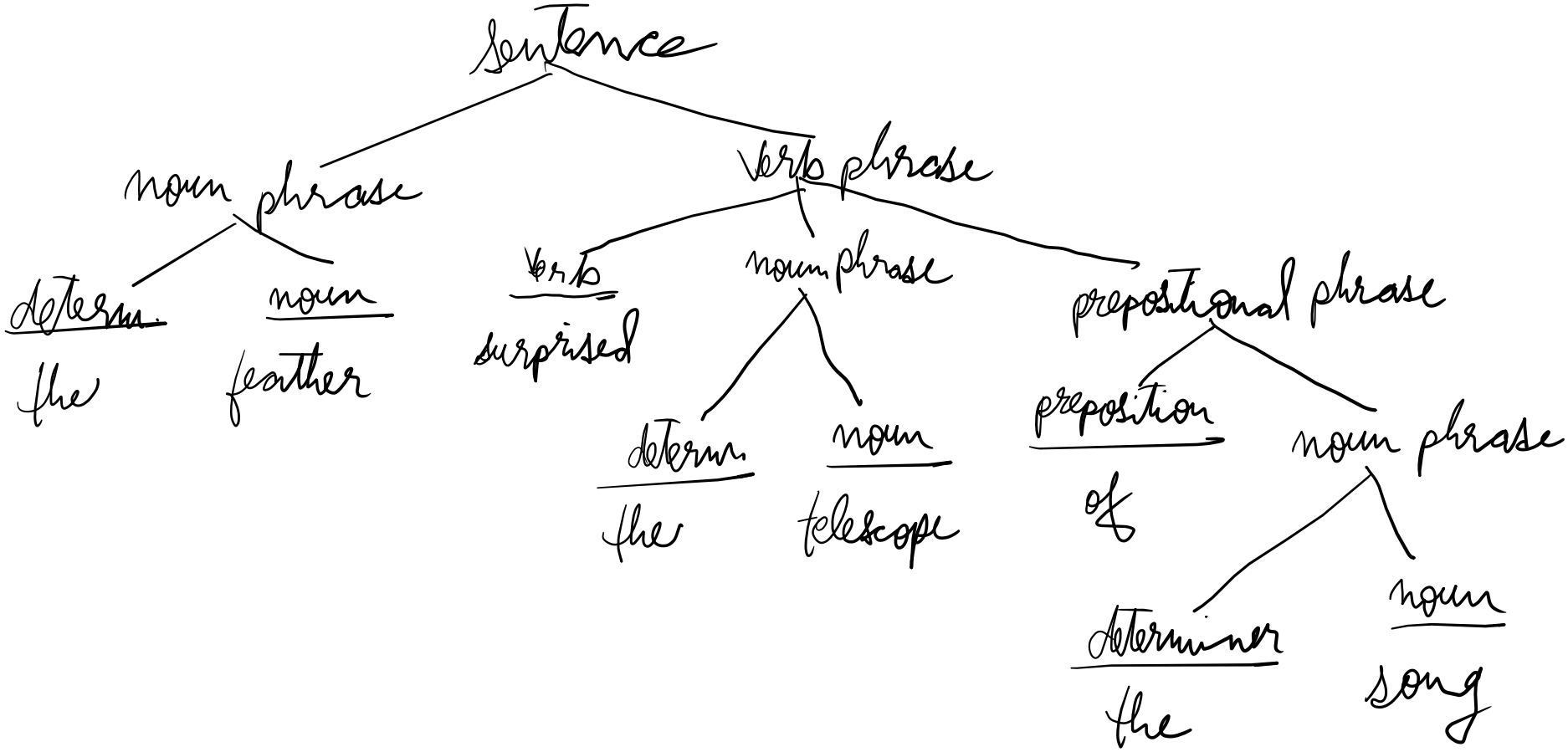
A derivation tree



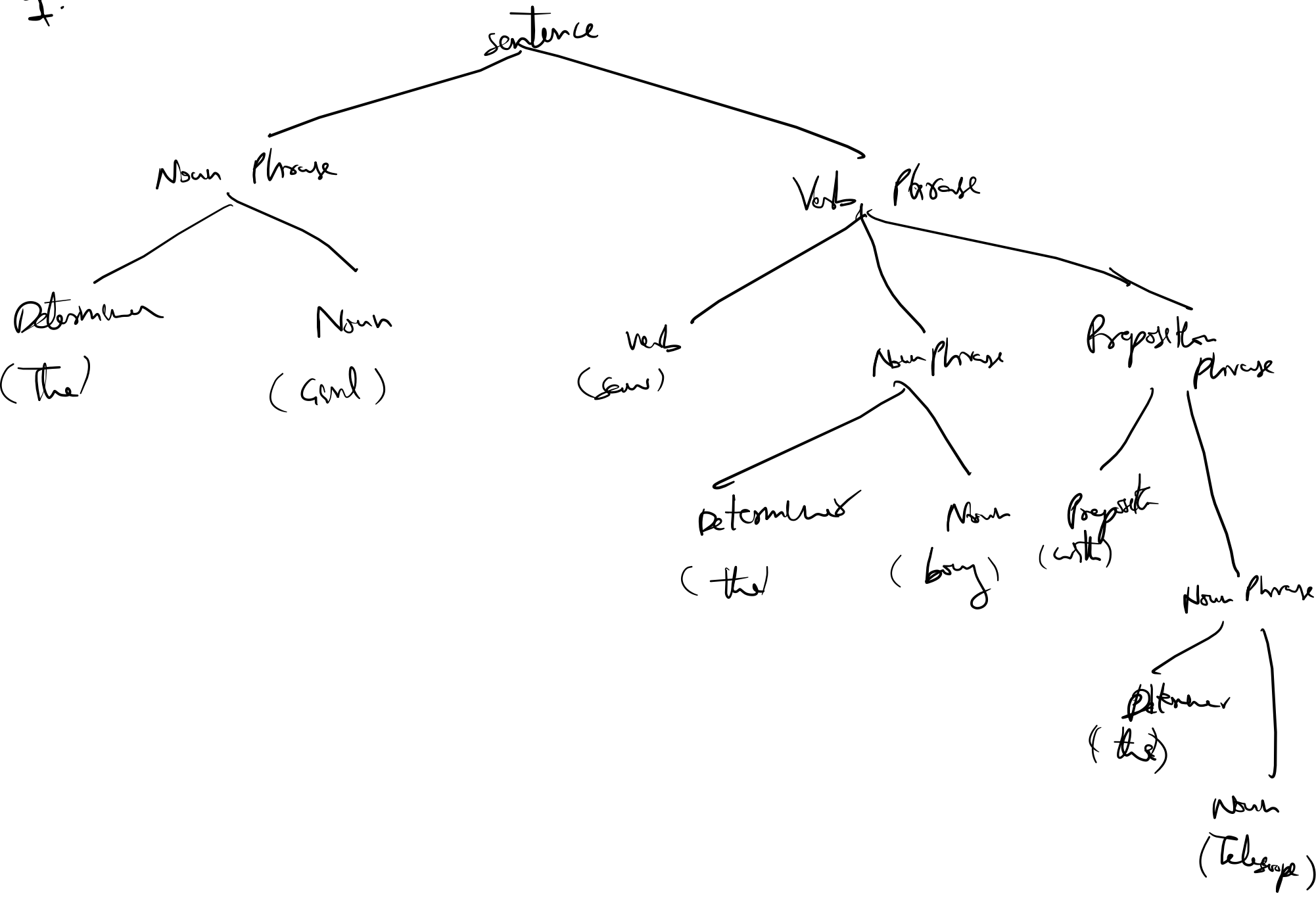
1. using the telescope
2. boy was using the telescope.

Exercises

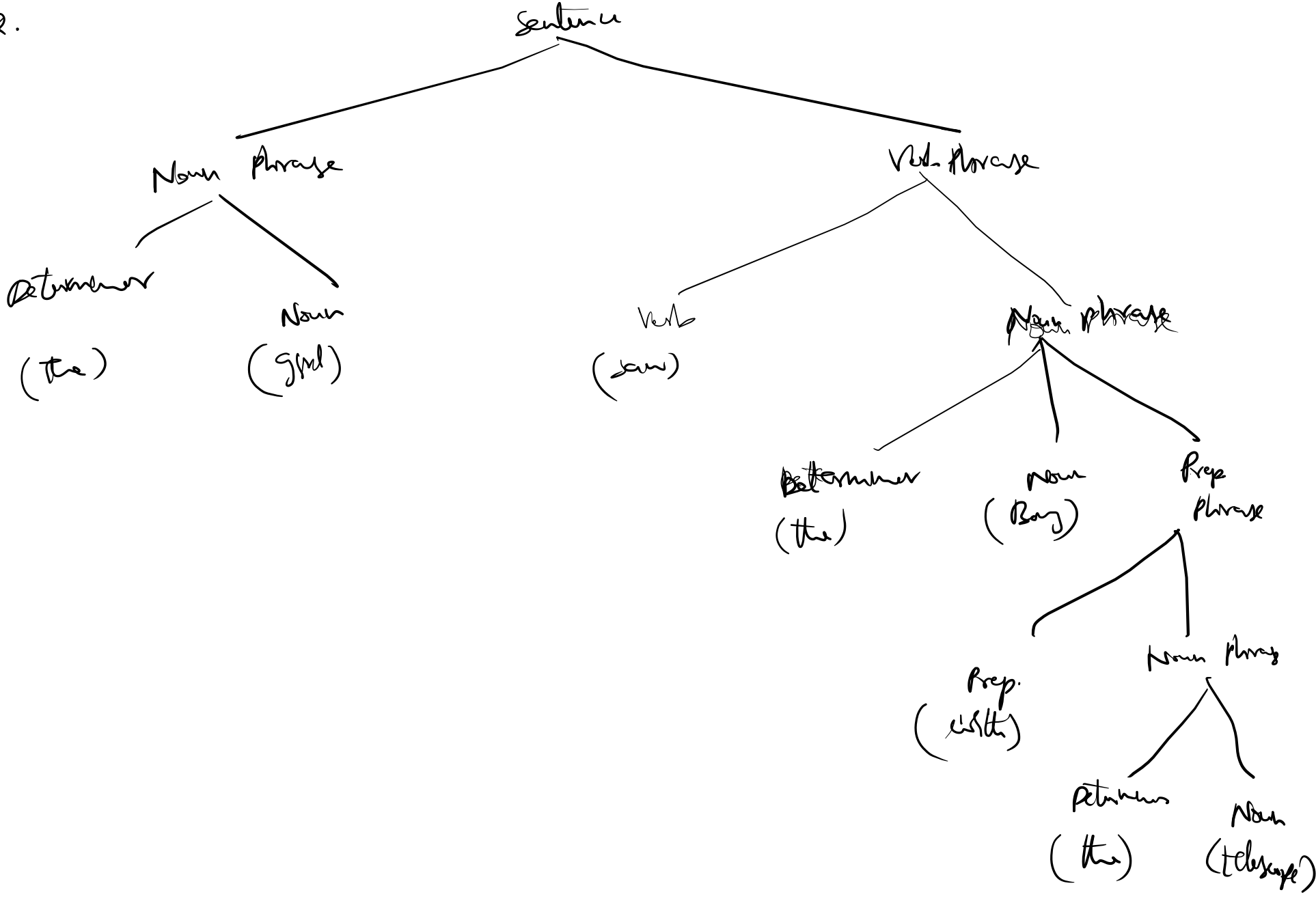
1. Give *two* derivations and draw the corresponding derivation trees, for "the girl saw the boy with the telescope."
2. Give a derivation and draw the corresponding tree for the syntactically valid but meaningless statement "the feather surprised the telescope of the song."


sentence
noun phrase
verb phrase
determ.
the
noun
feather
verb
surprised
noun phrase
determ.
the
noun
telescope
prepositional phrase
preposition
of
noun phrase
determiner
the
noun
song

7.


sentence
Noun Phrase
Verb Phrase
Determiner
(The)
Noun
(Girl)
Verb
(saw)
Noun Phrase
Preposition Phrase
Determiner
(the)
Noun
(boy)
Preposition
(with)
Noun Phrase
Determiner
(the)
Noun
(Telescope)

2.


Sentence
Noun Phrase
Verb Phrase
Determiner
(the)
Noun
(girl)
Verb
(saw)
Noun phrase
Determiner
(the)
Noun
(Boy)
Prep Phrase
Prep.
(with)
Noun Phrase
Determiner
(the)
Noun
(telescope)

CHOMSKY HIERARCHY

Types of Grammars

There are four types of grammars, Types 0 through 3.

- Type 3 grammars are the most restrictive grammars, and allow only a terminal, or a single terminal followed by one nonterminal, on the right side. Thus, production rules can only be of the form $\langle A \rangle \to a$ or $\langle A \rangle \to a\langle A \rangle$.
- Type 3 grammars are also known as *regular grammars*. These grammars describe the same languages that can be derived using regular expressions.

Type 2 Grammars

- Type 2 grammars are known as *context-free grammars* (CFGs for short).
- CFGs have a single non-terminal on the left side of a production rule, i.e., their rules are of the form $\langle A \rangle \rightarrow a\langle B \rangle \langle C \rangle$, etc.
- They are called "context-free" because a production rule can be invoked for a non-terminal regardless of its context, i.e., without concern for surrounding symbols and non-terminals.
- Nearly all grammars for NLP as well as programming languages are of this type.

Type 1 Grammars

- Type 1 grammars are known as *context-sensitive* grammars (CSGs); they allow production rules of the type $\langle A \rangle b \to \langle B \rangle a$ where $\langle A \rangle$ and $\langle B \rangle$ are non-terminals and a, b are symbols.
- The only restriction is that the right side contain no fewer symbols than the left side.
- These grammars are "context-sensitive" because the applicability of a production rule to a non-terminal depends on the context in which it occurs.

Type 0 Grammars

- These are *unrestricted* grammars, and only require that at least one non-terminal occur on the left side of a production rule.
- Unrestricted grammars are the most powerful possible—they are "Turing complete"(TBD)—but also the hardest to deal with.

A Grammar for $\{0^n1^n \mid n \geq 0\}$

Consider the context-free grammar

$$S \to 0S1 \mid \epsilon$$

This can derive all strings in $\{0^n1^n \mid n \geq 0\}$.

$$S \to \epsilon \tag{1}$$
$$S \to 0S1 \to 0\epsilon1 \to 01 \tag{2}$$
$$\to 0S1 \to 00S11 \to 00\epsilon11 \to 0011 \tag{3}$$
$$\to 0S1 \to 00S11 \to 000S111 \to 000\epsilon111 \to 000111 \tag{4}$$

Q3 $S \to 0S0 \mid 1S1 \mid \varepsilon$

Q4) $S \to AB \mid \varepsilon$
$A \to 0A1 \mid \varepsilon$
$B \to 1B0 \mid \varepsilon$

Exercises

3. Give a CFG for the language of all even-length palindromes on $\{0, 1\}$.
4. Give a CFG for all strings of the form $0^a1^b0^c$ where $a + c = b$.
5. Give a CFG for the language of algebraic expressions on symbols a, b, c, with operators $+, -, \times, /$, and appropriate brackets. Use the same to derive a complete syntax tree for the expression $(a + b)^2/(a^2 - c^2)$.

6. CFG for balanced brackets: $S : SS \mid (S) \mid \epsilon$