1 Context Free Grammars

1.1 Substitution rules

- A context free grammar consists of a collection of substitution rules or productions of the form:

\[ A \rightarrow w \]

\( A \) is a symbol called a variable and \( w \) is a string that consists of variables (sometimes also called non-terminals) and other symbols called terminals.

- Variables are said to be rewritten as the string that appears on the right-hand side of the substitution rule.

- Terminals appear only on the right-hand side of substitution rules and never on the left-hand side. Since terminals only appear on the right-hand side of a rule they can never be rewritten as anything else.

- One variable is designated as the start variable. It usually occurs on the left-hand side of the topmost rule.

- Example: Grammar G1

  Rules of G1:
  \( A \rightarrow 0A1 \)
  \( A \rightarrow \epsilon \)

  \( A \) is the start symbol.

  Question:
  Terminals of G1:
  Variables of G1:

1.2 Derivation of strings using a CFG

- Use the following procedure to generate a string in a context free grammar:

  1. Write down the start variable.

  2. Find a variable that is written down and a rule that starts with that variable. Replace the written down variable with the right-hand side of that rule.

  3. Repeat step 2 until no variables remain.
• The sequence of substitutions to obtain a string is called a DERIVATION. For example, a derivation of string 00001111 in grammar G1 looks like this (the last substitution is $A \to \epsilon$):

$$A \Rightarrow 0A1 \Rightarrow 00A11 \Rightarrow 000A111 \Rightarrow 0000A1111 \Rightarrow 00001111$$

• The same information can be represented in a PARSE TREE:

```
        A
       /\  \
      0  A  1
     /\  /\  /
   0  A  1  0 1
```

1.3 Context free languages

• All strings generated by applying the rules of the grammar constitute the LANGUAGE OF THE GRAMMAR.

• The language of grammar G1:

$$L(G1) = \{\epsilon, 01, 0011, 000111, 00001111, 0000011111, \ldots\} = \{0^n1^n| n \geq 0\}$$

2 Formal Definition of a Context Free Grammar

• A CONTEXT FREE GRAMMAR is a 4-tuple $< V, \Sigma, R, S >$, where:

  1. $V$ is a finite set called the variables;
  2. $\Sigma$ is a finite set of terminals.
  3. $R$ is a finite set of rules, with each rule being a variable that is rewritten as a string of variables and terminals.
  4. $S \in V$ is the start variable.

• If $u, v$ and $w$ are strings of variables and terminals, and $A \to w$ is a rule of the grammar, we say that $uAv$ YIELDS $uvw$. This relationship is written as:

$$uAv \to uvw$$

• We write $u \xrightarrow{*} v$ if $u = v$ or if there is a sequence of strings $u_1, u_2, \ldots, u_k$ with $k \geq 0$ such that $u$ can (eventually) be rewritten as $v$. In other words:

$$u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \ldots \Rightarrow u_k \Rightarrow v$$

• The language of the grammar is $\{w \in \Sigma^*|S \xrightarrow{*} w\}$. 
2.1 Example: Grammar G2

- Let the rules of the context free grammar G2 be:
  \[ S \rightarrow NP \ VP \]
  \[ NP \rightarrow \text{Det} \ N \ (PP) \]
  \[ VP \rightarrow V \ (NP) \ (PP) \]
  \[ PP \rightarrow P \ NP \]
  \[ \text{Det} \rightarrow \text{a} \mid \text{the} \]
  \[ N \rightarrow \text{boy} \mid \text{girl} \mid \text{flower} \]
  \[ V \rightarrow \text{touches} \mid \text{likes} \mid \text{sees} \]
  \[ P \rightarrow \text{with} \]

(Note: ‘A|B’ means ‘A or B’.)

- Question:
  Variables of G2 =

Terminals of G2 =

Start Variable of G2 =

- Some strings in L(G2) are:
  a boy sees
  the boy sees a flower
  a girl with a flower likes the boy

Question: What are some other strings in L(G2)?
3 Ambiguity

3.1 Definitions

- A derivation of a string $w$ in a grammar $G$ is a **leftmost derivation** if, at every step, the leftmost remaining variable is the one replaced.

- We can say formally that a string $w$ is derived ambiguously in a context free grammar $G$ if it has two or more different leftmost derivations.

- Grammar $G$ is said to be ambiguous if it generates some string ambiguously.
3.2 Structure and meaning

Sometimes, ambiguity in structure is directly reflected in the meaning assigned to the terminal string.

the girl touches the boy with the flower

The above sentence has two different parse trees associated with it. And each parse tree corresponds to a different meaning.

Question: What are the two parse trees? (Use Grammar G2.)
4 Chomsky Normal Form

4.1 Definition

- Any context free grammar can be converted to a standard simplified format called the ‘Chomsky normal form’.

- A context free grammar is in CHOMSKY NORMAL FORM if every rule is of the form:

\[
\begin{align*}
A & \rightarrow BC \\
A & \rightarrow a
\end{align*}
\]

where \(a\) is any terminal, and \(A, B,\) and \(C\) are any variables, except that neither \(B\) nor \(C\) may be the start variable.

- In addition, we permit the rule \(S \rightarrow \epsilon\), where \(S\) is a start variable.

4.2 Theorem (Sipser’s Theorem 2.6)

Any context free language is generated by a context free grammar in Chomsky normal form.

We give a proof by construction by building a procedure that takes an arbitrary context free grammar as input and returns a grammar in Chomsky normal form.

1. Add a new start symbol \(S_o\) and the rule \(S_o \rightarrow S\), where \(S\) was the original start symbol. This change guarantees that the start symbol does not occur on the right-hand side of a rule.

2. Remove \(\epsilon\)-rules.

We remove any rules of the form \(A \rightarrow \epsilon\), where \(A\) is not the start variable. Then for each occurrence of an \(A\) on the right-hand side of a rule, we add a new rule with that occurrence deleted (i.e., we essentially add \(\epsilon\) to the right-hand side of that rule). In other words:

(a) If \(R \rightarrow uAv\) is a rule in which \(u\) and \(v\) are strings of variables and terminals, we add rule \(R \rightarrow uv\) (\(uv\) is equivalent to \(u\epsilon v\)).

(b) For \(R \rightarrow uAvA\), we add \(R \rightarrow uvAw, R \rightarrow uAvw,\) and \(R \rightarrow uvw\).

(c) If we have the rule \(R \rightarrow A\), we add \(R \rightarrow \epsilon\), unless we had previously removed the rule \(R \rightarrow \epsilon\).

We repeat these steps until we eliminate all \(\epsilon\) rules not involving the start variable.
3. Eliminate unit rules.
   
   (a) We remove unit rules of the form $A \rightarrow B$.
   
   (b) Then, whenever a rule $B \rightarrow u$ appears, we add the rule $A \rightarrow u$, unless this was a unit rule previously removed. Note that $u$ is a string of variables and terminals.
   
   (c) Repeat this step until all unit rules have been eliminated.

4. Convert all remaining rules to the proper form.
   
   (a) We replace each rule of the form $A \rightarrow u_{1}u_{2}...u_{k}$ where $k \geq 3$ and each $u_{i}$ is either a terminal or a variable, with the following set of rules, where $A_{j}$ is a new variable:

   \[
   \begin{align*}
   A & \rightarrow u_{1}A_{1} \\
   A_{1} & \rightarrow u_{2}A_{2} \\
   A_{2} & \rightarrow u_{3}A_{3} \\
   & \vdots \\
   A_{k-2} & \rightarrow u_{k-1}u_{k}
   \end{align*}
   \]

   (b) If there is a rule of the form $A \rightarrow u_{k}B$, where $A$ and $B$ are variables and $u_{k}$ is a terminal, make up a new variable $U_{k}$ and replace this rule with the rules:

   \[
   \begin{align*}
   A & \rightarrow U_{k}B \\
   U_{k} & \rightarrow u_{k}
   \end{align*}
   \]

Question: Convert the following CFG to Chomsky normal form.

\[
\begin{align*}
S & \rightarrow ASA | aB \\
A & \rightarrow B | S \\
B & \rightarrow b | \epsilon
\end{align*}
\]