1 Context-free grammar

A **context-free grammar** (CFG) can be defined as the tuple $G = (\Gamma, \Sigma, P, S)$, where:

- Γ : variables (A,B,...)
- Σ : terminals (a,b,...)
- P : productions (e.g A $\rightarrow$ aB)
- S : start symbol ($S \in \Gamma$)

Example (Palindromes):

$P \rightarrow \varepsilon$ (the empty string is a palindrome)
$P \rightarrow 0$
$P \rightarrow 1$
$P \rightarrow 0P0$ (any palindrome surrounded by two 0s is also a palindrome)
$P \rightarrow 1P1$ (any palindrome surrounded by two 1s is also a palindrome)

We could write this in a simpler way as:

$P \rightarrow \varepsilon \mid 0 \mid 1 \mid 0P0 \mid 1P1$
(the vertical line represents OR, allowing you to merge multiple productions with the same head)

Other example:

$\Sigma = \{a, b, 0, 1, (,), +, *\}$

$I \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$
$E \rightarrow I \mid E * E \mid E + E \mid (E)$

Leftmost derivation $E \Rightarrow_{lm} E * E \Rightarrow_{lm} I * E \Rightarrow_{lm} a * E \Rightarrow_{lm} a * (E) \Rightarrow_{lm} a * (E + E) \Rightarrow_{lm} ...$

Rightmost derivation $E \Rightarrow_{rm} E * E \Rightarrow_{rm} E * (E) \Rightarrow_{rm} E * (E + E) \Rightarrow_{rm} E * (E + I) \Rightarrow_{rm} ...$

Continue the above derivations, and show that $a * (a + b00)$ is in the language of E.

1.1 Recursive inference

1.2 Parse trees

Parse trees for a grammar are trees, where:

- Each interior node is labeled by a variable.
- Each leaf is labeled by a variable, terminal, or ε
- If an interior node is labeled A, and its children are labeled $B_1, B_2, \ldots, B_n$, then the grammar has a production $A \rightarrow B_1 B_2 \ldots B_n$

The *yield* of a parse tree is the concatenation of all of its leaves from leftmost to rightmost.

2 Chomsky Normal Form

Every context-free language without ε has a grammar G in which all productions have one of the following forms:

1. $A \to BC$, where A, B and C are all variables, or
2. $A \to a$, where A is variable and a is a terminal

Further, G also has no useless symbols. Such a grammar is in *Chomsky Normal Form* (CNF).

We say that X is a useful symbol if: $S \Rightarrow^* \alpha X \beta \Rightarrow^* w$
X is generating if: $X \Rightarrow^* w$
X is reachable if: $S \Rightarrow^* \alpha X \beta$
Useful symbols are both generating and reachable.

What do you think of this grammar?
$S \to AB \mid a$
$A \to b$

Answer:
B is a useless symbol. It's reachable ($S \to AB$), but not generating. Because of this, we should delete $S \to AB$. This leaves us with
$S \to a$
$A \to b$
Again, A is useless. It's generating ($A \to b$), but it's not reachable from S. Deleting $A \to b$ leaves us with
$S \to a$
which is the final form of our grammar.
Variable A is nullable if $A \Rightarrow^* \varepsilon$

To convert a grammar to CNF, we should first make sure it has no ε-productions, unit productions, or useless symbols. This is done with the following steps:

1. remove ε productions (e.g. $A \to \varepsilon$)
2. remove unit productions (e.g. $A \to B$)
3. remove useless symbols

Keep the order of these steps to have a safe transformation, as only this guarantees the desired features in every case.

If we have a grammar G that has no ε-productions, unit productions, or useless symbols, we can convert in to CNF using the following steps.

1. modify bodies of length 2 or more to contain only variables
2. break bodies of length 3 or more

Example for simplifying a grammar:

Simplify the following grammar by removing all ε-productions, unit productions, and useless symbols:

$S \to AC$
$A \to a$
$C \to B \mid Bd \mid d$

$B \to D \mid \varepsilon$
$D \to E$
$E \to b$

Step 1: The only ε-production of the grammar is $B \to \varepsilon$. This results in two nullable symbols: C and B. $S \to AC$ has a nullable symbol in it, we have to examine both cases where it's present or absent. This gives $S \to AC \mid A$. $C \to B \mid Bd$ has nullable symbol B in both productions. Again, we have to consider the present and absent cases. In $C \to B$, we cannot choose B to be absent, because we cannot get rid of the body completely. In $C \to Bd$, we can consider both cases: $C \to d \mid Bd$. By adding these new productions to our existing ones, and deleting ε-productions, we will get:

$S \to AC \mid A$
$A \to a$
$C \to B \mid Bd \mid d$
$B \to D$
$D \to E$
$E \to b$

Step 2: Remove unit productions $C \to B$, $B \to D$, $D \to E$, $S \to A$. In the case of such unit production chains as $C \to B \to D \to E$, the following steps give the best result

1. Find all X and Y pairs where the $X \Rightarrow^* Y$ derivation consists only of unit productions. In this case, these are (C,B), (C,D), (C,E), (B,D), (B,E), (D,E).
2. For each such (X,Y) pair, add all $X \to \alpha$ productions, where $Y \to \alpha$ is not a unit production. These do not exist for (C,B), (C,D) and (B,D). For (C,E), we will get $C \to b$ because of $E \to b$. We will get $B \to b$ and $D \to b$ in a similar fashion from (B,E) and (D,E).

For simple unit productions without a chain like $S \to A$, we can get $S \to a$ in a similar fashion because of $A \to a$. Again, adding the new productions to our existing ones, and deleting unit-productions, we get:

$S \to AC \mid a$
$A \to a$
$C \to b \mid Bd \mid d$
$B \to b$
$D \to b$
$E \to b$

Step 3: Now we remove useless symbols. First, we remove non-generating symbols, then we remove non-reachable ones.

- All symbols are generating.
- The only reachable symbols are S, A, B, C. This means that D and E are useless, and can be removed.

By removing the all productions that contain our useless symbols, we get our simplified grammar:

$S \to AC \mid a$
$A \to a$
$C \to b \mid Bd \mid d$
$B \to b$

Examples for CNF conversions:

Example 1:
$S \to AB$
$A \to aAA \mid \varepsilon$
$B \to bBB \mid \varepsilon$

Answer:
We have to get rid of ε-productions, unit-productions and useless symbols, in this order.
First, we get rid of ε-productions $A \to \varepsilon$ and $B \to \varepsilon$. For every nullable symbol in the production bodies, we choose all possible ways for them to be present or absent. We cannot make the body completely be absent. (For example, $S \to AB$ has both A and B as nullable symbols, which gives us $S \to AB \mid A \mid B$ as possible options, but we cannot nullify both A and B because that would get rif of the body completely. In the case of $A \to aAA$, we can choose to nullify both A-s, as there would still be a body)
The resulting productions are:
$S \to AB \mid A \mid B$
$A \to aAA \mid aA \mid a$
$B \to bBB \mid bB \mid b$
Then we remove unit productions $S \to A$ and $S \to B$
$S \to AB \mid aAA \mid aA \mid a \mid bBB \mid bB \mid b$
$A \to aAA \mid aA \mid a$
$B \to bBB \mid bB \mid b$
All symbols are useful. A and B are reachable ($S \Rightarrow^* AB$), and they are both generating ($A \Rightarrow^* a$ and $B \Rightarrow^* b$).
The grammar now has no ε-productions, unit-productions or useless symbols, so we can continue with the transformation to CNF.
To modify bodies of length 2 or more to contain only variables, we introduce $A_1 \to a$ and $B_1 \to b$
$S \to AB \mid A_1AA \mid A_1A \mid a \mid B_1BB \mid B_1B \mid b$
$A \to A_1AA \mid A_1A \mid a$
$B \to B_1BB \mid B_1B \mid b$
$A_1 \to a$
$B_1 \to b$
To break up bodies of length 3 or more, we introduce $A_2 \to A_1A$ and $B_2 \to B_1B$
$S \to AB \mid A_2A \mid A_1A \mid a \mid B_2B \mid B_1B \mid b$
$A \to A_2A \mid A_1A \mid a$
$B \to B_2B \mid B_1B \mid b$
$A_1 \to a$
$B_1 \to b$
$A_2 \to A_1A$
$B_2 \to B_1B$
The above grammar is in CNF, because it satisfies the above requirements. If we still had more productions with body length 3 or more, we would have to repeat this last step again in a similar fashion.

Example 2:
$S \to aXbX$
$X \to aY \mid bY \mid \varepsilon$
$Y \to X \mid c$

Example 3:
$S \to AbA \mid a$
$A \to Aa \mid \varepsilon$