Chomsky normal form EXAMPLE

\[ G : S \rightarrow \gamma \varepsilon \mid z z z \]
\[ X \rightarrow \varepsilon \]
\[ Y \rightarrow X Y \]
\[ Z \rightarrow \varepsilon \mid a Z \]

1. Eliminate useless symbols
   2. Eliminate symbols that cannot generate a terminal string
      \[ G' : S \rightarrow z z z \]
      \[ X \rightarrow \varepsilon \]
      \[ Z \rightarrow \varepsilon \mid a Z \]

3. Eliminate symbols that are not reachable (e.g., \( X \))
   \[ G'' : S \rightarrow z z z \]
   \[ Z \rightarrow \varepsilon \mid a Z \]

4. Eliminate \( \varepsilon \) productions (\( Z \rightarrow \varepsilon \), \( S \rightarrow \varepsilon \))
   \[ G''' : S \rightarrow z z z \mid z z \mid a Z \mid a \]
   \[ Z \rightarrow a Z \mid a \]

5. Eliminate unit productions (\( S \rightarrow z \))
   \[ G'''' : S \rightarrow B Z \mid z z \mid A Z \mid a \]
   \[ B \rightarrow z z \]
   \[ Z \rightarrow A Z \mid a \]
   \[ A \rightarrow a \]

This is the CNF form grammar that we obtain mechanically. Looking closer, it can be simplified to

\[ S \rightarrow a \mid s s \]

with \( L(S) = L(a^*) \), compared to \( L(G) = L(a^+e) = L(a^+ \varepsilon) \).
Decision problems.

1. **Emptiness:**
   - Given: CFL $L$ in CNF form of a CFG $G$.
   - Question: Is $L(G) = \emptyset$?
   - Idea: If $L(G) = \emptyset$ then $S$ is not generating.
   - $\Rightarrow$ $S$ is useless.
   - Easy to check.

2. **Membership:**
   - Given: CFL $L$, string $w$.
   - Question: Is $w \in L$?
   - Idea: (Much harder than for regular languages.) Assume $L = L(G)$ for CFG $G$.
     1. If $w = \epsilon$, check if $S$ is nullable.
     2. Construct a Chomsky normal form grammar $G'$ for $L \cup \{\epsilon\}$.
        - If $\text{length}(w) = n$ then $S \Rightarrow w$ in exactly $2n-1$ steps.
     4. Try all possible derivations of length $2n-1$.
        - $V = \{V_1\cdots V_n\}$ possibilities, $10^V = \#$ productions in $G'$.

   Better approach:
   - Dynamic Programming $V_{ij} = \begin{cases} \text{null} & i = 0 \\ \text{true} & i = n \\ \text{false} & 0 < i < n \end{cases}$

3. **Equality:**
   - Given: CFGs $G_1, G_2$.
   - Question: Is $L(G_1) = L(G_2)$?
   - Claim: There is no decision procedure? (Show later)

   Remark:
   1. Easy to test equality for regular languages $L_1 = L_2 \iff (L_1 \cup \overline{L_1}) \cap (L_2 \cup \overline{L_2}) = \emptyset$.
   2. [Sennitzescues, 2001]:
      Many other undecidable properties of CFGs, but
      algorithm for deciding $L(M) = L(M')$ for deterministic PDAs $M, M'$.
      Deciding of $L(M) \cap L(M')$ still impossible. [Gödel 1963]
      or $L = \emptyset \Rightarrow \emptyset \subseteq L \Rightarrow \emptyset \subseteq L$.
Dynamic programming alg. to decide membership in a CFL

Take a CFG in CNF form.

Goal: For each substring $x(i:j) := x_i x_{i+1} \cdots x_j$
compute $V_{i:j} := \{ X \in V \mid X \Rightarrow x(i:j) \}$

Induction in $j-i$:

Base case $j=i$: Easy,

Induction:

$X \Rightarrow x(i:j)$

must start with $X \Rightarrow BC$

where for some $k$, $i \leq k \leq j$,

$0(n)$ time

where $n = |x|$

$\Rightarrow O(n^4)$ time total, if CFG size is constant.

(then also conversion to CNF is $O(n)$)
Pumping Lemma for Context-Free Languages

Question: Of \( L_1 = \{ a^n b^n \mid n \geq 1 \} \)
\( L_2 = \{ a^n b^{n+1} c^n \mid n \geq 1 \} \)
\( L_3 = \{ a^n b^{2n} c^n \mid n \geq 1 \} \)

which are CFLs?

Answer: \( L_1, L_2 \) are CFLs, \( L_3 \) is not a CFL.

Intuitively, because the number of \( b \)'s depends on its context, both to its left and right.

Pumping Lemma
Let \( L \) be a CFL.
Then there exists a constant \( N \) such that for all \( z \in L \) with \( |z| \geq N \)
can write \( z = uvwxy \) with
1) \( |vwx| \leq N \)
2) \( |v| > 0 \)
3) \( i \geq 0 \), \( uv^iwx^iy \in L \)

Application: Claim: \( L_3 = \{ a^n b^{2n} c^n \mid n \geq 1 \} \) is not a CFL.
Proof: 1) Assume \( L_3 \) is CFL and apply P.L.
2) Get constant \( N > 0 \)
3) Choose \( z = a^N b^{2N} c^N \in L_3 \)
4) Get \( u, v, w, x, y \) such that
\( z = uv^iwx^iy \) and \( |vw| \leq N \), \( |v| > 0 \)
5) We choose \( j = 0 \) and claim
\( z' = uv^iwx^iy = uv^0wxy \notin L_3 \) (contradiction)

Why? Observe: \( |vw| \leq N \) implies that either \( vwx \) has no \( c \)'s or it has no \( a \)'s
Case I: \( |vwx| \) has no \( a \)'s:
\( |z'| = |z| - |vwx| < |z| = 4N \)
but \( n_c(z') = N \), so more than \( \frac{3}{4} \) of its symbols are \( c \)'s.
Every string in \( L_3 \) has exactly \( \frac{3}{4} \) \( c \)'s, so \( z' \notin L_3 \).
Case II: \( |vwx| \) has no \( a \)'s: By a similar argument, \( z' \notin L_3 \).

Corollary: Unlike regular languages, CFLs are not necessarily closed under intersection.
\( L_3 \cap L_2 \neq \emptyset \)
Proof of the CFL Pumping Lemma:

**Given** CFL $L$

Let $G = (V, T, P, S)$ be a Chomsky normal form grammar for $L \setminus \epsilon$.

**Claim**: In a parse tree for $G$, all root-leaf paths have length at most $k$.

Then the yield of the tree has $\leq 2^{k+1}$ terminals.

**Proof**: By induction in $k$. Intuitively obvious since the tree is binary.

**Worst cases**:

- $l = 1$:
  - $V_i = A$
  - $V_{i+1} = \omega$
  - $2^{l-1} = 1$

- $l = 2$:
  - $V_i = A$
  - $V_{i+1} = B$
  - $V_{i+2} = C$
  - $2^{l-1} = 1$

- $l = 3$:
  - $V_i = A$
  - $V_{i+1} = B$
  - $V_{i+2} = E$
  - $V_{i+3} = F$
  - $2^{l-1} = 1$

Now define $k = \|V\| = \#variables$, $N = 2^k$.

Consider any $z \in L(G)$ with $\|z\| > N$.

⇒ by the claim, any parse tree $T$ for $z$ has a root-leaf path $Q$ of length $q \geq k+1$.

⇒ by the pigeon-hole principle, some variable repeats along the path $Q$. In fact the last $k+1$ variables on $Q$ must contain a repetition.

Let $i, j$ be such that $q-k \leq i < j \leq q$, $V_i = V_j = A$.

Therefore $z = uvwxy$, where

$S \Rightarrow uAy \Rightarrow uvAxy \Rightarrow uvywxy$

so $A \Rightarrow vAx$, $A \Rightarrow \omega$

Hence $A \Rightarrow vAx \Rightarrow vyAx^2$

$\Rightarrow \cdots \Rightarrow vy^ixx^i \Rightarrow vy^iwx^i$

and so $i \geq 0$,

$S \Rightarrow uAy \Rightarrow uvy^iwx^iy$,

implying $uv^iwx^iy \in L(G)$.

Also: $\|z\| > N$ since CNF $G$ has no $\varepsilon$ or unit productions.

Finally: $\|vwxy\| \leq N = 2^k$, since $V_i = A$ has height $\leq k+1$ and so by the claim its yield $vwxy$ has length at most $2^{(k+1)-1} = 2^k$ terminals.