Decidability and CFL I

- Acceptance problem of CFG

\[ A_{\text{CFG}} = \{ \langle G, w \rangle \mid G : CFG, \text{ generates } w \} \]

- We prove that \( A_{\text{CFG}} \) is decidable
- But an issue is the \( \infty \) possible derivations of a CFG
- For example,

\[ A \rightarrow B, B \rightarrow A \]

- Chomsky normal form

\[ A \rightarrow BC \]
\[ A \rightarrow a \]
Decidability and CFL II

- Any $w$, $|w| = n$, derivation in exactly $2n - 1$ steps
- If $q$ is the number of rules, check all $q^{2n-1}$ possibilities
- Proof
  1. Convert $G$ to Chomsky
  2. Check all $q^{2n-1}$ possibilities
- Results apply to PDA as well: for PDA we have a finite procedure to generate a CFG.
$$E_{\text{CFG}} = \{ \langle G \rangle \mid G : \text{CFG}, L(G) = \emptyset \}$$

- idea: bottom up setting to see if any string can be generated from the start variable. From

$$A \rightarrow a$$

We search if there is a rule

$$B \rightarrow A$$
Proof:
1. Mark all terminals
2. Repeat until no new variables are marked
   if
   \[ A \rightarrow U_1 \cdots U_k \]
   and
   all \( U_1, \ldots, U_k \) marked
   \[ \Rightarrow \text{mark } A \]
3. If start variable is not marked, accept
   Otherwise, reject
Number of iterations is finite: bounded by the number of variables

Each iteration is a finite procedure: we check all rules
\[ EQ_{CFG} = \{ \langle G, H \rangle \mid G, H : CFG, L(G) = L(H) \} \]

- Remember that \( EQ_{DFA} \) is decidable.
- However, we cannot apply the same proof as CFL is not closed for \( \cap \) and complementation.
- It’s proved in Chapter 5 that this language is not decidable.
- We do not discuss details.
Let $A$ be a CFL. The goal is to show that $A$ is decidable.

How about converting PDA to a TM and use the TM to run any $w \in A$?

But a difficulty is that our simulation of a PDA on $w$ may not be a finite procedure.

Specifically, some branches of the PDA’s computation may go on forever, reading and writing the stack without ever halting.

For example, consider the following PDA.
CFL decidable II

By our way mentioned before for constructing a tree, at the first layer we have

\[ q_0 \emptyset \quad q_0 \{1\} \quad q_0 \{1, 1\} \quad \ldots \]

Then we may have troubles to go to the next layer for processing the first character.
So converting PDA to TM does not really work.
We need a different way.
Because we know $A$ is a CFL, there is a corresponding grammar $G$.

Then we run TM for $\langle G, w \rangle$ by using $A_{CFG}$. 
Classes of languages I

- Regular
- Context-free
- Decidable
- Turing-recognizable