Decidability and CFL I

- Acceptance problem of CFG

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

- We prove that \( A_{CFG} \) is decidable
- But an issue is the \( \infty \) possible derivations of a CFG
- For example,
  \[ A \rightarrow B, B \rightarrow A \]
- Chomsky normal form

\[ A \rightarrow BC \]
\[ A \rightarrow a \]
Any $w, |w| = n$, derivation in exactly $2n - 1$ steps

If $q$ is the \# 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.
$E_{CFG} = \{ \langle G \rangle \mid G : CFG, L(G) = \emptyset \}$

- 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
   \( \Rightarrow \) 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
$EQ_{CFG}$

$$EQ_{CFG} = \{ \langle G, H \rangle \mid G, H : CFG, L(G) = L(H) \}$$

- 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
This question is different from $A_{CFG}$ decidable or not
How about converting PDA to a TM?
For nondeterministic PDA we can do NTM
But nondeterministic PDA may have $\infty$-long branches
Specifically, some branches of the PDA’s computation may go on forever, reading and writing the stack without ever halting.
Then TM runs forever
So converting PDA to TM does not really work
A proof that works:
Find grammar $G$ for this CFL
Run TM for $\langle G, w \rangle$ by using $A_{CFG}$
Classes of languages I

- Fig 4.10

Diagram showing the hierarchy of language classes:
- Regular
- Context-free
- Decidable
- Turing-recognizable