Decidable languages related to CFG's

We will consider following languages today.

$A_{CFG} = \{\langle G, w\rangle \mid G \text{ is a CFG and } G \text{ generates } w\}$

$E_{CFG} = \{\langle G\rangle \mid G \text{ is a CFG and } L(G) = \phi\}$

$EQ_{CFG} = \{\langle G, H\rangle \mid G, H \text{ are CFGs and } L(G) = L(H)\}$

$A_{CFG} = \{ \langle G, w \rangle \mid G$ is a CFG an w is a String and G generates $w \}$

- go through every possible derivations of the grammer G and check whether w is generated
- we do not have a bound on the # of derivations.
- what if we convert G into CNF?

Theorem 4.7

A_{CFG} is decidable

- If G were in chomsky Normal form, any derivation of w where $|w| = n$, has $2n-1$ steps.

Proof: construct TM S as follows:

S = "on $\langle G, w \rangle$ where G is CFG & w is a string:

1. Convert G into CNF.
2. List all derivations with $2n-1$ steps, where $n = |w|$
3. If any of the derivations generates w, accept; otherwise, reject."

Is this an efficient algorithm?

No, we may have to look at exponential # of derivations.

$E_{CFG} = \{ \langle G \rangle \mid G \text{ is a CFG and } L(G) = \emptyset \}$

Use A_{CFG} and try all possible $w \in \Sigma^*$

we try to check whether start variable can generate a String of terminals

would this be enough to create a decider?

The idea is simple. we check whether start variable can generate a string of terminals.

Theorem 4.8 E_{CFG} is decidable.

construct TM R for E_{CFG}:

1. Mark all terminals in G.
2. Repeat until no new variables get marked
   - Mark A, if $A \rightarrow u_1 u_2 \ldots u_k$ is a rule in G and all u_i's are marked.
3. If the start variables is not marked, accept; otherwise, reject.

Ex: $S \rightarrow ABX \mid BAX$

$A \rightarrow AX$

$B \rightarrow BX$

$X \rightarrow a$

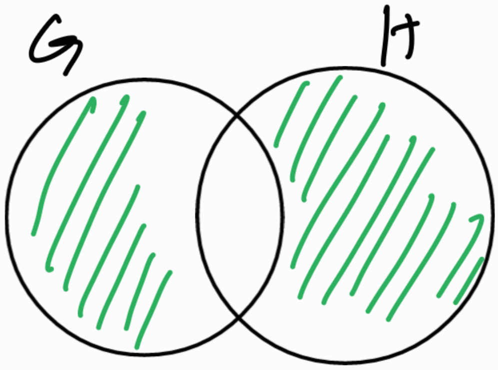
$S \rightarrow ABX$ $\quad$ $S \rightarrow BAX$ $\quad$ $A \rightarrow AX$ $\quad$ $B \rightarrow BX$

$X \rightarrow a$

$$EQ_{CFG} = \{ \langle G, H \rangle \mid G, H \text{ are CFG's and } L(G) = L(H) \}$$

Can we follow the idea used in EQ_{DFA}?

EQ_{CFG} : Given two CFGs, are they equal?



$C = (G \cap \overline{H}) \cup (\overline{G} \cap H)$

if $G = H$, then $C = \emptyset$

if $C = \emptyset$, then $G = H$

Q: Are CFG's closed under $\cup$, $\cap$, $\overline{}$ operations?

Claim 1: CFL's are closed under union operation.

Proof Idea:

G: $S_1 \rightarrow \ldots$

H: $S_2 \rightarrow \ldots$

Construct G' s.t

$$S \rightarrow S_1 \mid S_2$$

$S_1 \rightarrow \ldots$

$S_2 \rightarrow \ldots$

claim 2:

CFL's are <u>not</u> closed under

$\cap$, $-$ operation.

we only need one counter example.

$A = \{a^i b^j c^k \mid i,k,j \geq 0, i=j \text{ or } j=k\}$

* this is a CFL

$B = \{a^m b^m c^n \mid m,n \geq 0\}$ * this is a CFL

$C = \{a^n b^m c^m \mid m,n \geq 0\}$ * this is a CFL

$D = \{a^n b^n c^n \mid n \geq 0\}$ * this is not a CFL

$D = B \cap C$

Claim 02: CFL's are not closed under complement ($\overline{}$) operation.

consider CFGs B and C

Let $L = B \cap C$

$$B \cap C = \overline{\overline{B} \cup \overline{C}} \quad \text{(De morgan's law)}$$

This is a counter example.

$\overline{}$ operator should not be closed for CFL's.

- Later we show that EQ_{CFG} is not decidable.