Undecidability of CFG Complementation
Exposition by William Gasarch

1 The Problem

Given a CFG G we want to know if $\overline{L(G)}$ is also a CFG. We will show this is undecidable. The proof we give was emailed to us by Harry Lewis. It is likely well known.

2 Needed Lemmas

Lemma 2.1 *Let G be a CFG over Σ. Let $\$ \in \Sigma$. Let Let L' be the set of strings w such that*

- *w does not contain \$, and*
- *there exists $w' \in L(G)$ such that $w' = w\$\Sigma^*$.*

Then L' is a CFL.

Proof:
We show how to transform the CFG G into a CFG for L'.
Replace every rule of the form

$$X \rightarrow \alpha\$\beta \text{ where } \alpha \in (\Sigma - \$)^*$$

with the rule

$$X \rightarrow \alpha.$$

▮

Def 2.2

1. $D(M_e) = \{x : M_e(x) \downarrow\}$.
2. A *promise problem* is a problem where you are given e and promised something about it. We give the only example of a promise we will use in the next item.
3. PROM is the following promise:
   $D(M_e)$ is either $\emptyset$ OR is NOT a CFL.

Lemma 2.3 *The following promise problem, which we denote PROMEMPTY, is undecidable: Given e which satisfies PROM, determine if $D(M_e) = \emptyset$.*

Proof: Assume, BWOC, that PROMEMPTY decidable. We show HALTONZ is undecidable.

1. Input x (so we want to know if $M_x(0)\downarrow$).
2. CREATE a machine M_e as follows:
   (a) Input y. If $y \notin \{a^n b^n c^n : n \in \mathbb{N}\}$ then go into an infinite loop.
   (b) If you got here then there exists n such that $y = a^n b^n c^n$. Run $M_x(0)$ for n steps. If it halts then HALT otherwise go into an infinite loop.
3. (This is a program comment. Note that
   1) $M_x(0)\downarrow$ implies there exists n_o (the number of steps it took to halt) such that
   $$\{y : M_e(y)\downarrow\} = \{a^n b^n c^n : n \geq n_o\}$$
   which is NOT a CFL.
   2) $M_x(0)\uparrow$ implies that $D(M_e) = \emptyset$.)
4. Note that either $D(M_e) = \emptyset$ or $D(M_e)$ is NOT a CFL. Hence e satisfies PROM. Since PROMEMPTY is decidable we can determine if $D(M_e) = \emptyset$. If $D(M_e) = \emptyset$ then $e \notin HALTONZ$, so output NO. If $D(M_e) \neq \emptyset$ then $e \in HALTONZ$, so output YES.

▮

3 Main Theorem

Def 3.1 The CFG-COMP problem is as follows. Given a CFG G, determine if $\overline{L(G)}$ is CFL.

Theorem 3.2 *CFG-COMP is undecidable.*

Proof:
Assume, by way of contradiction, that CFG-COMP is solvable. We use this to show that PROMEMPTY is decidable.

1. Input e
2. Construct a CFG G_1 that generates the COMPLEMENT of strings of the form START CONFIG of M_e $(w\$w^R)^*$ END CONFIG OF M_e.
3. Construct a CFG G_2 that generates the COMPLEMENT of strings of the form

   $$C_1\$C_1^R\$C_2\$C_2^R\$\cdots\$C_L\$C_L^R$$

   where C_{i+1} is the next config after C_i.
4. Using G_1 and G_2 (easily) construct a CFG G such that

   $$L(G) = L(G_1) \cup L(G_2)$$

5. (This is a program comment.

   Look at

   $$\overline{L(G)} = \overline{L(G_1) \cup L(G_2)} = \overline{L(G_1)} \cap \overline{L(G_2)}$$

   This is the set of all strings that represent accepting computations of M_e.

   We were promised that $D(M_e)$ was either empty or NOT a CFL.

   If $D(M_e) = \emptyset$ then $\overline{L(G)} = \emptyset$ and hence a CFL.

   If $D(M_e)$ is NOT a CFL, then, by Lemma 2.1, $\overline{L(G)}$ is not a CFL.)

   Since CFG-COMP is decidable we can determine $\overline{L(G)}$ is a CFL. If the answer is YES then $D(M_e) = \emptyset$ so we output EMPTY. If the answer is NO then $D(M_e)$ is NOT CFL so we output NOT CFL.

∎