March 12

Exercise 1: CFL's are closed under the union operations

Solution: Suppose we have CFG's G and H, which generates $L(G)$ and $L(H)$ respectively.

Let G: $S \to X_1 Y_1$, H: $S \to X_2 Y_2$

$X_1 \to A_1 B_1$ $\quad$ $X_2 \to A_2 C_2$

$Y_1 \to B_1 C_1$ $\quad$ $Y_2 \to B_2 C_2$

$C_1 \to \cdots \mid c$ $\quad$ $A_2 \to \cdots \mid a$

$A_1 \to \cdots \mid a$ $\quad$ $B_2 \to \cdots \mid b$

$B_1 \to \cdots \mid b$ $\quad$ $C_2 \to \cdots \mid c$

Add a $'$ on every variable in G, and a $''$ on every variable in H. Also, add a new variable S and rule $S \to S' \mid S''$, the other rules in G and H are included as

$$\boxed{\begin{array}{l} S' \to X_1' Y_1' \\ X_1' \to A_1' B_1' \\ Y_1 \to B_1' C_1' \\ C_1' \to \cdots \mid c \\ A_1' \to \cdots \mid a \\ B_1' \to \cdots \mid b \end{array}} \qquad \boxed{\begin{array}{l} S'' \to X_2'' Y_2'' \\ X_2'' \to A_2'' C_2'' \\ Y_2'' \to B_2'' C_2'' \\ A_2'' \to \cdots \mid a \\ B_2'' \to \cdots \mid b \\ C_2'' \to \cdots \mid c \end{array}}$$

Obviously, the new grammar generates $L(G) \cup L(H)$.

Exercise 2: CFL's are NOT closed under the intersection operations.

Note that to disprove a statement, we only need a counterexample. (Suppose the opposite.)

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

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

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

$D = \{a^n b^n c^n \mid n \geq 0\}$

Claim: B, C are context-free.

CFG G for B:

$S \to XY$

$X \to aX \mid \varepsilon$

$Y \to bYc \mid \varepsilon$

CFG H for C:

$S \to XY$

$X \to aXb \mid \varepsilon$

$Y \to cY \mid \varepsilon$

Claim: $D = B \cap C$ is not context-free, following the pumping lemma example.

Exercise 3: CFL's are NOT closed under the complementation operations.

Suppose the opposite, i.e., they are closed under the complementation operations.

Then, $B \cap C = \overline{\overline{B} \cup \overline{C}}$ would be context-free, a contradiction.

Undecidability.

Thm. Let S be an infinite countable set.
Then 2^S (the powerset of S) is not countable.

Example: S — set of all Turing machines
2^S — groups of Turing machines (each group can solve a set of problems).

---

proof: We'll use the diagonalization method.

Let $S = \{s_1, s_2, s_3, \cdots\}$.

We encode each element t of 2^S as a sequence of 0's and 1's, with a 1 at position i iff $s_i \in t$.

Suppose 2^S were countable, then its elements could be ordered as $t_1, t_2, t_3, \cdots$, with

$t_1 = \underline{0}\ 1\ 0\ 1\ 0 \cdots$ // $\{s_2, s_4, s_6 \cdots\}$

$t_2 = 1\ \underline{0}\ 1\ 0\ 1 \cdots$ // $\{s_1, s_3, s_5, \cdots\}$

$t_3 = 1\ 0\ \underline{1}\ 1\ 1 \cdots$

$\vdots$

Take the values along the main diagonal, and complement each entry, i.e., $t' = 110\cdots$. t' is an element of 2^S, say $t' = t_i$ for some i. But by the construction t' differs from t_i at position i. $\therefore$ We reach a contradiction.

$\therefore$ 2^S is uncountable. $\square$

$A_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM which accepts } w\}$

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

$EQ_{TM} = \{\langle M_1, M_2\rangle \mid M_1, M_2 \text{ are TM's and } L(M_1) = L(M_2)\}$

We'll deal with A_{TM} in this topic.

A TM U which recognizes A_{TM}:

1. Simulate M on input w.
2. If M enters the accept state, <u>accept</u>;
   if M enters the reject state, <u>reject</u>;
   otherwise, loop. // difficult to distinguish from taking a long time to run.

A_{TM} is also called the <u>halting</u> problem, U is called a <u>universal</u> TM.

Note that A_{TM} is not the problem asking you to write a program (TM) which processes (accepts) w.

It is more like somebody wrote a huge program, possibly with no comments at all, and ask you to decide whether it fulfills some goal.