Decidable Problems for CFLs

**Note:** Conversions between grammars and PDAs are effective.

**Claim:** There is an algorithm to decide whether a given CFL is empty.

**Proof:** Given a CFG, we can decide whether the start symbol is useful.

---

**Claim:** There is an algorithm to decide whether a given CFL contains a given string $w$.

1. Convert to a CFG $G$.

2. If $w = \epsilon$, decide whether the start symbol is nullable.

3. Otherwise:
   (a) Convert to CNF.
   (b) Generate all strings of length no greater than $|w|$.
Undecidable Problems for CFLs

- Is a given CFG ambiguous?
- Is a given CFL inherently ambiguous?
- Is the intersection of two given CFLs empty?
- Are two given CFLs equivalent?
- Is a given CFL equal to $\Sigma^*$?

Church-Turing Thesis

Any algorithmic procedure can be described by a \textit{Turing machine}. Such algorithmic procedures include:

- evaluation of $\lambda$-calculus expressions;
- evaluation of programs in any programming language; and
- computation of partial recursive functions.
**Defn:** A *Turing machine* (TM) is a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, where

- $Q$ is an alphabet of *states*;
- $\Sigma$ is the *input alphabet*;
- $\Gamma \supseteq \Sigma$ is the *tape alphabet*;
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ is the *transition function*;
- $q_0 \in Q$ is the *start state*;
- $B \in \Gamma - \Sigma$ is the *blank symbol*; and
- $F \subseteq Q$ is the set of *final states*.

**Defn:** Let $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ be a TM. An *instantaneous description* (ID) of $M$ is a string in $\Gamma^*Q\Gamma^*$. The set of all IDs of $M$ will be denoted by $\text{ID}(M)$. 


Defn: Let $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ be a TM. We define $\vdash_M$ to be the binary relation on $\text{ID}(M)$ such that $I_1 \vdash_M I_2$ iff for some $\alpha \beta \in \Gamma^*$, $a, b, c \in \Gamma$, and $p, q \in Q$, either

- $I_1 = \alpha qa\beta$, $I_2 = \alpha bp\beta$, and $\delta(q, a) = (p, b, R)$;
- $I_1 = \alpha aqb\beta$, $I_2 = \alpha pac\beta$, and $\delta(q, b) = (p, c, L)$; or
- $I_1' \vdash_M I_2'$ by one of the above rules, where $I_1' \in B^* I_1 B^*$ and $I_2' \in B^* I_2 B^*$.

$\vdash_M^*$ is the reflexive transitive closure of $\vdash_M$.

---

Defn: Let $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ be a TM. The language accepted by $M$ is given by

$$L(M) = \{ w \in \Sigma^* \mid \exists q \in F, \alpha \beta \in \Gamma^* : q_0 w \vdash^* \alpha q \beta \}$$

Defn: A language $L$ is recursively enumerable (RE) if there is a TM $M$ s.t. $L(M) = L$.

Defn: A language $L \subseteq \Sigma^*$ is said to be recursive if there is a TM $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F')$ such that $L(M) = L$ and there is no infinite sequence

$$q_0 w \vdash I_1 \vdash I_2 \vdash \cdots$$

for $w \in \Sigma^*$. 
Decidable Problems

**Defn:** A *decision problem* is a language \( L \subseteq \Sigma^* \), where \( \Sigma^* \) is the set of *instances* of the problem.

**Defn:** A decision problem \( L \) is said to be *decidable* if \( L \) is recursive; otherwise, \( L \) is said to be *undecidable*.