Compare the following two loops, given \( G = (V, \Sigma, R, S) \):

\[
\text{set}(V) \quad N = \emptyset \quad \{ \text{or init to } \emptyset \} \quad \{ \text{A is a rule?} \} \quad \text{set}(V, \Sigma) \quad N = \Sigma
\]

bool changed = true;
while (changed) {
    changed = false;
    for (each \( A \in V \setminus N \)) {
        if (\( A \) has a rule \( A \to \alpha \) with \( \alpha \in N^* \)) {
            \( N = N \cup \{ A \} \);
            changed = true;
        }
    }
}

Output \( N \);

Claim: \( G \) Routine is a decider - same reason.
\( \emptyset \) At the end, \( S \in N \Leftrightarrow \Sigma \in L(G) \).
Inductively, \( N \) becomes \( \{ A = A \Rightarrow \alpha \} \).
\( \emptyset \) also \( \text{CFG} = \{ G : \varepsilon \in L(G) \} \) is available in \( O(|V| \cdot |<R>|) \) time.

The while loop must halt within \( \text{IVL} \) iterations, so this algm is total, ie a decider. Time \( \approx O(|IVL| \cdot |KR|) \).

Claim: At the end, \( N \) equals the set of variables that are "live" in the sense of deriving terminal strings, and \( S \in N \Leftrightarrow \L(G) \neq \emptyset \).

Since the set \( N \) of nullable variables is identified, we can "bypass" -rules by taking every rule \( B \to \alpha \) with some number \( k \) of nullable variables and making \( 2^{k-1} \) more rules by deleting any subset of their ancestry.

The text does this on the fly, while building \( N \).

Other wise, this step toward \( CNF \) can take expo time. But there is a list-based algorithm to do it in polynomial \( (|V| \cdot |<R>|) \) time.

The text does this on the fly, while building \( N \).

We might have to add \( \emptyset \) to \( S \to \varepsilon \) (5) and \( N = \{ S \} \).

\[
S \to \varepsilon \quad S \to (S) \quad S \to (S,S)
\]

Then remove \( S \rightarrow \varepsilon \) to get \( G' \). Then \( L(G') \) is \( \{ \} \).
ACFG: \( \text{INST}: \langle G, x \rangle \), i.e., a CFG \( G \) and an input \( x \in \Sigma^* \).

Ques: Is \( x \in L(G) \)?

Decider: If \( x = \varepsilon \), use the decider for \( S \in N \). Else, convert \( G \) to strict Chomsky, NF \( \overline{G} \), so that \( x \in L(\overline{G}) \iff x \in L(G) \). Key fact about \( \overline{G} \) in CHNF is that if \( x \) is derivable at all, it is derivable using \((1 \times 1)\) \( A \to BC \) kind of rule and \((1 \times 1) \ A \to \varepsilon \) kind of rules. Hence with \( n = |x| \) we can try all derivations of \( 2n - 1 \) steps, and reject if none gives \( x \).

FYI: There is an algorithm called "CKY" or "CYK" that does this in \( \Theta(n^3) \) time via dynamic programming. Combined with the better conversion to GNf, this in fact classifies ACFG into polynomial time.

We've seen that ECFG, NFCFG, ECFG, and ACFG are all decidable. How about \( \text{ALLCFG} = \{ \langle G \rangle : L(G) = \Sigma^* \} \)? All\( \text{CFG} \) is not decided, nor even recognizable.

Can we show that there are undecidable, indeed unrecognizable, languages?

Diagonalization:

\( \text{D}_TM \): \( \text{INST}: \) The code \( w = \langle M \rangle \) of a dec Tm \( M \).

Ques: Does \( M \) not accept its own code, i.e., \( w \notin L(M) \)?

As a language, \( D_{TM} = \{ \langle M \rangle : M \text{ does not accept } \langle M \rangle \} \).

Theorem: There does not exist a TM \( \mathcal{Q} \) s.t. \( L(\mathcal{Q}) = D_{TM} \). So \( D_{TM} \) is not c.e. so undecidable.

Proof: If so, then the "quixotic" \( \mathcal{Q} \) \( \mathcal{Q} \) would have a code \( q = \langle \mathcal{Q} \rangle \). Then

\[
q \in D_{TM} \iff q \notin L(\mathcal{Q}) \quad \text{by defn of } q \in D_{TM} \\
\iff q \in L(\mathcal{Q}) \quad \text{by } L(\mathcal{Q}) = D_{TM}.
\]

A logical stmt can never be equivalent to its negation. This is a contradiction, so \( \mathcal{Q} \) cannot exist. \( \Box \)
Conventional "Landscape" is shown in the mirror image to $L$. Hence also $\overline{\text{co-RE}} = \{ \overline{L} : L \text{ is recognizable} \}$.

**What is the complement of $\text{Dm}$?** Essentially answering the issue of invalid codes, or lumping them with $\text{Dm}$

\[
\overline{\text{Km}} = \{ \langle M \rangle : M \text{ does not accept } \langle M \rangle \}.
\]

Recall that I did define $\overline{\text{Am}} = \{ \langle M, w \rangle : M \text{ accepts } w \}$, and showed $\text{Am}$ is r.e. since it is the language of a universal TM. So $\overline{\text{Km}}$ too is r.e.

**Theorem:** $L$ is decidable $\iff \overline{L}$ is decidable

\[
\overline{\text{Km}} \text{ is (r.e.-but) not decidable. This is because } \text{Dm} \text{ is undecidable and not even r.e.}
\]

**Proof:** If $L \equiv \text{Lm}$ with $M$ total, we can interchange 9acc and 9rei.

This finishes Ch 4 coverage — know as facts for the exam.

**Added:** The diagram also conveys that $\text{RE} \cap \overline{\text{co-RE}} = \text{DfC}$, which says that a language is decidable if (and only if) it is c.e. and co-c.e. If $L$ is decidable then $L$ is automatically c.e. and since $\overline{L}$ is decidable too, $\overline{L}$ is c.e., so $L$ is co-c.e. as well. Conversely, if $L$ and $\overline{L}$ are both c.e., then one can use non-total TMs $M$ and $M'$ for them to build a total TM deciding $L$. Next week we will prove something more general involving reductions to c.e. and co-c.e. languages.