Decidable

Does a DFA, B, accept a string w?

\[ A_{\text{DFA}} = \{ <B, w> \mid \text{B is a DFA that accepts string w} \} \]

\[ A_{\text{DFA}} \] is decidable.

Proof: construct TM, M that decides \[ A_{\text{DFA}} \]

M: on input \(<B, w>\)

1. Simulate B on input w.
2. If the simulation ends w/ B accepting w, accept
   otherwise reject.
\[ A_{NFA} = \{ \langle B, w \rangle \mid B \text{ is an NFA that accepts str. } w \} \]

\[ A_{NFA} \text{ is decidable} \]

\[ N : \text{on input } \langle B, w \rangle \]

1. Convert \( B \) to a DFA, \( C \)
2. Run \( M \) on input \( \langle C, w \rangle \) (\( M \) from \( A_{DFA} \))
3. If \( M \) accepts, accept; else reject.

\[
\text{boolean } \text{NFA-is-decidable}(\text{NFA } n) \}
\]
\[
\text{DFA } d = \text{nfa2dfa}(n);
\]
\[
\text{return } \text{DFA-is-decidable}(d);
\]

```
N:
\langle B, w \rangle
```
\[ E_{DFA} = \{ \langle A \rangle \mid A \text{ is a DFA and } L(A) = \emptyset \} \]

**T:** on input \( \langle A \rangle \) where

1. Mark the start state.
2. Repeat 3 until no new states are marked.
3. Mark any state that has a transition coming from a marked state
4. If no final state is marked, accept.
   Otherwise, reject.
\[ \text{EQ}_{\text{DFA}} = \{\langle A, B \rangle \mid A \text{ and } B \text{ are DFA's s.t. } L(A) = L(B) \} \]

\text{EQ}_{\text{DFA}} \text{ is decidable.}

Construct DFA \( C \) s.t. \( C \) accepts strings accepted by \( A \) or \( B \), but not both.

\[
L(C) = \left( L(A) \cap \overline{L(B)} \right) \cup \left( \overline{L(A)} \cap L(B) \right)
\]

\[ L(C) = \emptyset \text{ iff } L(A) = L(B) \]

\text{TM } F = \text{ on input } \langle A, B \rangle \text{ where } A, B \text{ are DFA's}

1. Construct \( C \) (as above).
2. Run TM \( T \) on input \( \langle C \rangle \) (\( T \) accepts \( E_{\text{DFA}} \)).
3. If \( T \) accepts, accept; Else reject.
\[ A_{CFG} = \{\langle G, w \rangle | G \text{ is a CFG that generates } w \} \]

\[ A_{CFG} \text{ is decidable if } G \text{ is in Chomsky Normal Form, any derivation has } 2n-1 \text{ steps (} n=|w|). \]

Every rule is of the form

- \[ A \to BC \]
- \[ A \to a \]
- \[ S \to \epsilon \]

1. Convert \( G \) to C.N.F.
2. List all derivations up to length \( 2n-1 \)
3. If \( w \) is derived, accept
   Else, reject.