Decidable

Does a DFA, B, accept a string w?

$$A_{DFA} = \{\langle B, w\rangle \mid B \text{ is a DFA that accepts string } w\}$$

A_{DFA} is decidable.

Proof: construct TM, M that decides A_{DFA}

M: on input $\langle B, w\rangle$

B is a DFA, w is a string

1. Simulate B on input w.
2. If the simulation ends w/ B accepting w, accept otherwise reject.

$$A_{NFA} = \{\langle B, w\rangle \mid B \text{ is an NFA that accepts str. } w\}$$

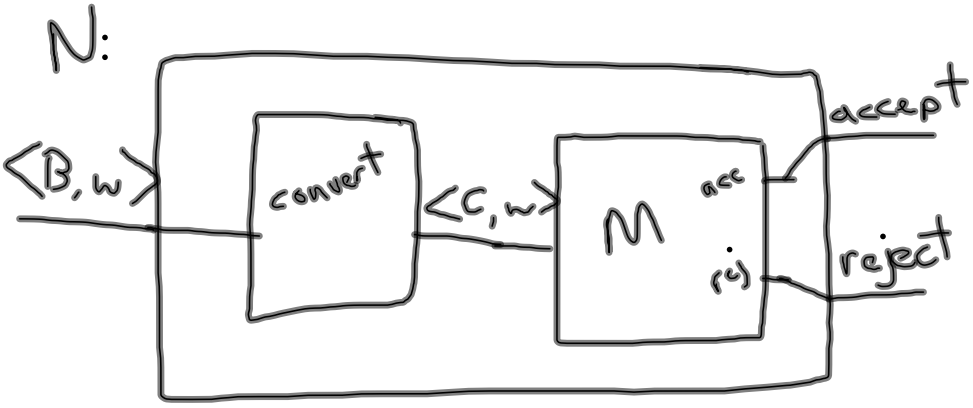
A_{NFA} is decidable

N: 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.

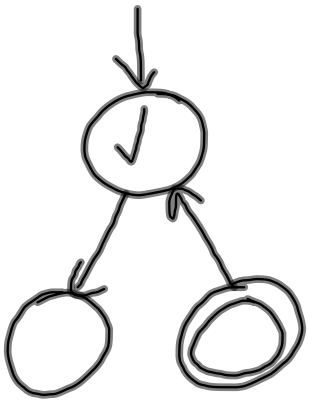
```
boolean NFA_is_decidable(NFA n){
    DFA d = nfa2dfa(n);
    return DFA_is_decidable(d);
}
```

N:



$$E_{DFA} = \{\langle A \rangle \mid A \text{ is a DFA and } L(A) = \emptyset\}$$

T: on input $\langle A \rangle$ where A is a DFA.



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.

$EQ_{DFA} = \{\langle A, B\rangle \mid A \text{ and } B \text{ are DFAs s.t. } L(A) = L(B)\}$

EQ_{DFA} is decidable.

construct DFA C s.t. C accepts strings accepted by A or B, but not both.

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

$L(C) = \emptyset$ iff $L(A) = L(B)$

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

1. construct C (as above).
2. run TM T on input $\langle C\rangle$ (T accepts E_{DFA}).
3. If T accepts, accept
   Else reject.

$A_{CFG} = \{\langle G, w\rangle \mid G \text{ is a CFG that generates } w\}$

A_{CFG} is decidable

if G is in Chomsky Normal Form, any derivation has $2n-1$ steps ($n = |w|$).

every rule is of the form

$A \to BC$

$A \to a$

$S \to \varepsilon$

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