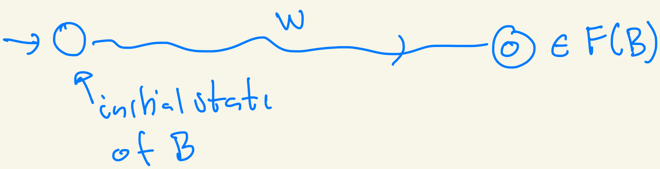
Sipser Section 4.1 Decidable Languages

$A_{DFA} = \{\langle B\rangle\langle w\rangle \mid B \text{ is a DFA and } w \in L(B)\}$

$\langle B\rangle$ and $\langle w\rangle$ are coded via the universal alphabet and the universal state set.

Notation $\langle B, w\rangle = \langle B\rangle\langle w\rangle$

$\langle B, w\rangle \in A_{DFA} \iff B$ is a DFA and



Theorem 4.1 A_{DFA} is decidable:

Let M_1 be a DTM which works as follows:

M_1: on input $\langle B, w\rangle = \langle B\rangle\langle w\rangle$

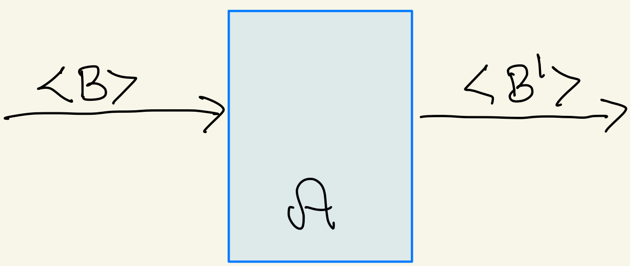
1. check whether B is a DFA and reject $\langle B, w\rangle$ if it is not
2. Simulate B on w
3. If B is in an accept state after reading w $\to$ accept $\langle B, w\rangle$
   else reject $\langle B, w\rangle$

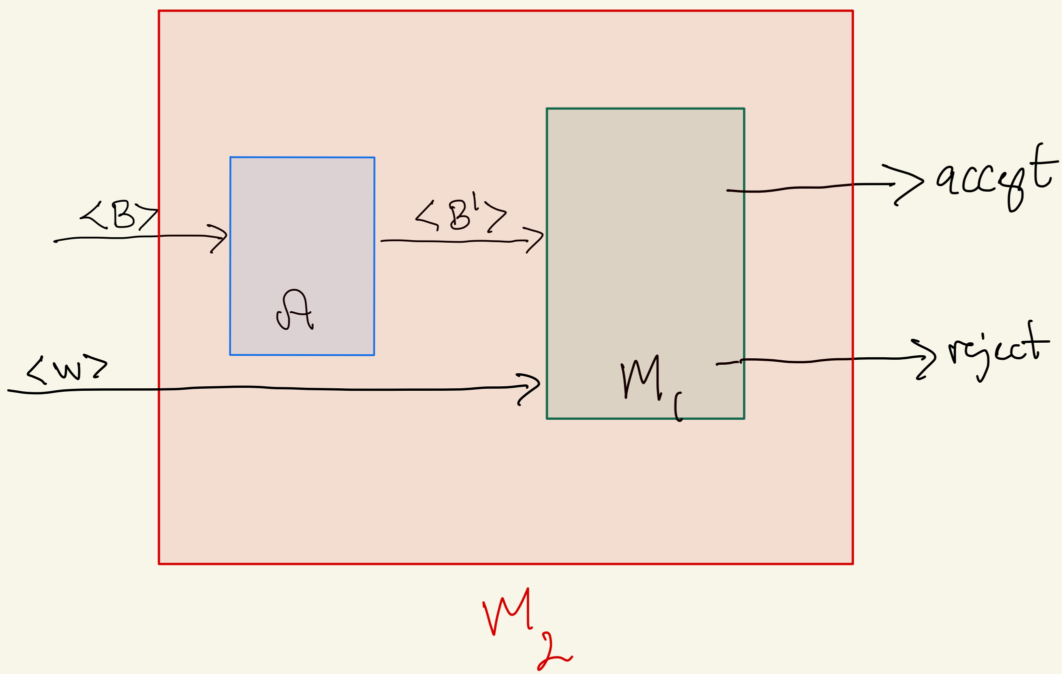
$A_{NFA} = \{ \langle B, w \rangle \mid B \text{ is an NFA and } w \in L(B) \}$

Theorem 4.2 A_{NFA} is decidable

Proof: Let $\mathcal{A}$ be a DTM which given an input B, first checks whether $\langle B \rangle$ is an encoding of an NFA

- If B is not an NFA, $\mathcal{A}$ will output the encoding $\langle B' \rangle = \langle B \rangle$ (which is not a DFA)
- If B is an NFA, $\mathcal{A}$ will output $\langle B' \rangle$ where B' is a DFA with $L(B') = L(B)$





M_2: on input $\langle B \rangle \langle w \rangle$

- run A to convert $\langle B \rangle$ into $\langle B' \rangle$
- run M_1 on $\langle B' \rangle \langle w \rangle$

  accept if M_1 accepts

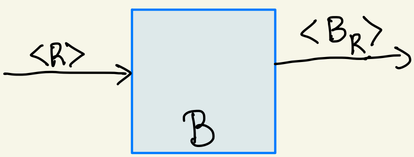
  reject if M_1 rejects

M_2 decides A_{NFA}

$A_{REX} = \{ \langle R \rangle \langle w \rangle \mid R \text{ is a regular expression and } w \in L(R) \}$

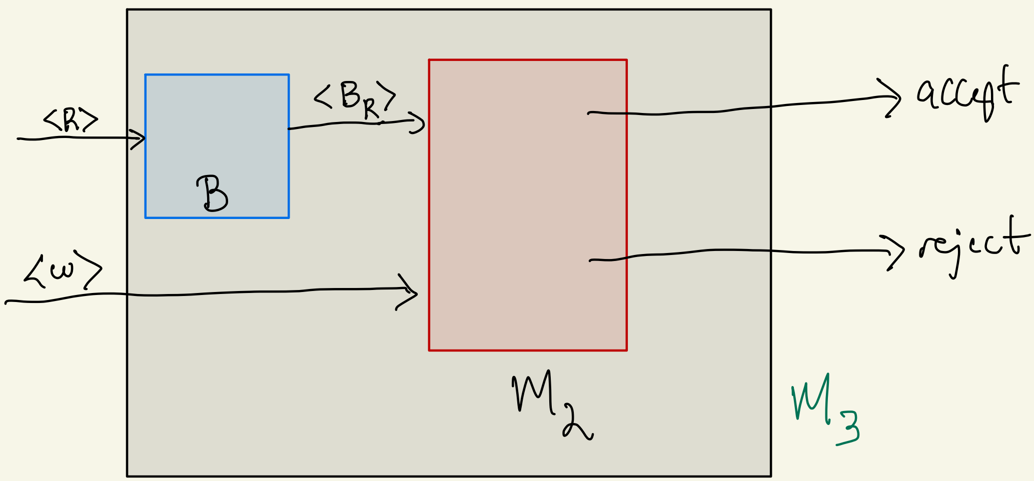
Theorem 4.3 A_{REX} is decidable

Proof:



B first checks whether R is a legal regular expression

- If it is not, then B outputs $\langle B_R \rangle$ which codes a non-NFA
- If R is a regular expr. then B generates the code of an NFA B_R with $L(B_R) = L(R)$



M_3 decides A_{REX}

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

Theorem 4.4 E_{DFA} is decidable

M_4: on input $\langle A \rangle$

1) check if $\langle A \rangle$ codes a DFA. If not, reject $\langle A \rangle$
2) let D_A be the underlying digraph of A
3) If D_A has a directed path from the vertex v_0 corresponding to the initial state q_0 of A to some vertex v_i corresponding to a state $q_i \in F(A)$

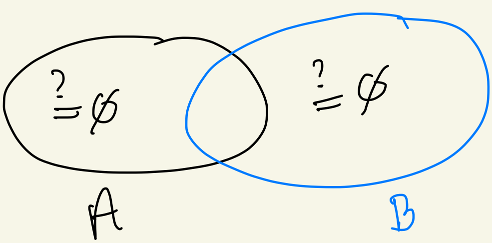
If such a path exists, ~~accept~~ reject $\langle A \rangle$

else ~~reject~~ accept $\langle A \rangle$

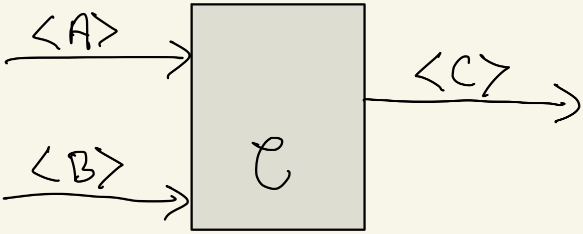
$EQ_{DFA} = \{ \langle A \rangle \langle B \rangle \mid A \text{ and } B \text{ are DFAs and } L(A) = L(B) \}$

Note $L(A) = L(B) \iff (L(A) \setminus L(B)) \cup (L(B) \setminus L(A)) = \emptyset$

$\Updownarrow$



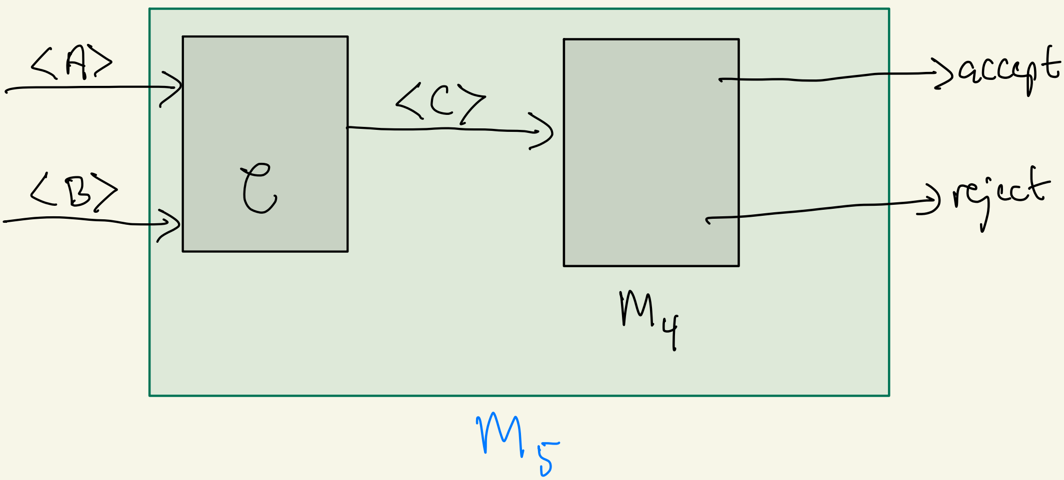
$L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B)) = \emptyset$



If ⟨A⟩ and ⟨B⟩ are DFAs then output DFA ⟨C⟩ with $L(C) = (L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B))$

Else output a DFA ⟨C⟩ with $L(C) = \Sigma^*$

$$L(C) = \emptyset \iff \langle A \rangle \langle B \rangle \in EQ_{DFA}$$



M_5 decides EQ_{DFA}

$$A_{CFG} = \{ \langle G \rangle \langle w \rangle \mid G \text{ is a CFG and } w \in L(G) \}$$

Theorem 4.7 A_{CFG} is decidable

Proof let M_{CFG} be a deterministic TM which

1. Checks whether $\langle G \rangle$ codes a CFG. If not, reject $\langle G \rangle \langle w \rangle$
2. Convert G to G' which is a Chomsky CFG with $L(G') = L(G)$
3. Check all possible derivations of length $2|w|-1$ and **accept** $\langle G \rangle \langle w \rangle$ if one of them is the string w
4. If no derivation gives w **reject** $\langle G \rangle \langle w \rangle$

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

Note: we cannot aford to check all possible strings w via M_{CFG}, but

Theorem 4.8 E_{CFG} is decidable

P: We want to check whether $\exists \beta \in \Sigma^*$ s.t

$S \overset{*}{\Rightarrow} \beta$

1. Mark all terminal symbols
2. Repeat

   If $A \to u_1 u_2 \cdots u_n$ is a rule when all u_i are marked

   then mark A

   If S becomes marked reject $\langle G \rangle$

   Until no change
3. accept $\langle G \rangle$

example:

$S \to AB \mid CD$

$A \to AA$

$B \to BC$

$C \to c$

$D \to AB \mid d$

Next relevant question:

$$EQ_{CFG} = \{ \langle G, H \rangle \mid G \text{ and } H \text{ are CFGs and } L(G) = L(H) \}$$

Problem: We cannot use the same approach as we did for DFAs as the set of context-free languages is not closed under complementation and intersection so

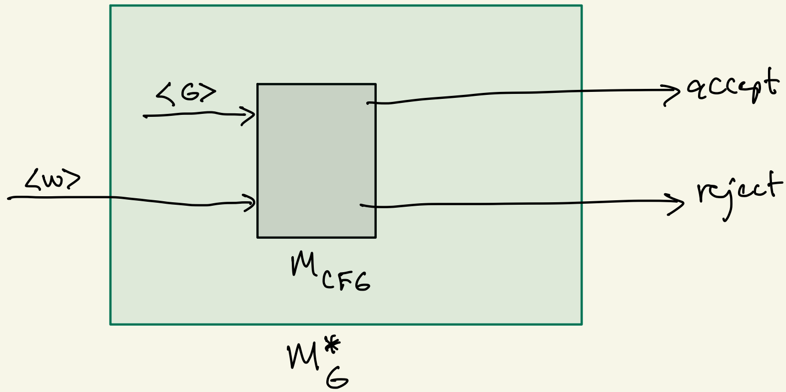
$$\left(L(G) \cap \overline{L(H)} \right) \cup \left(\overline{L(G)} \cap L(H) \right)$$ may not

be a context-free language.

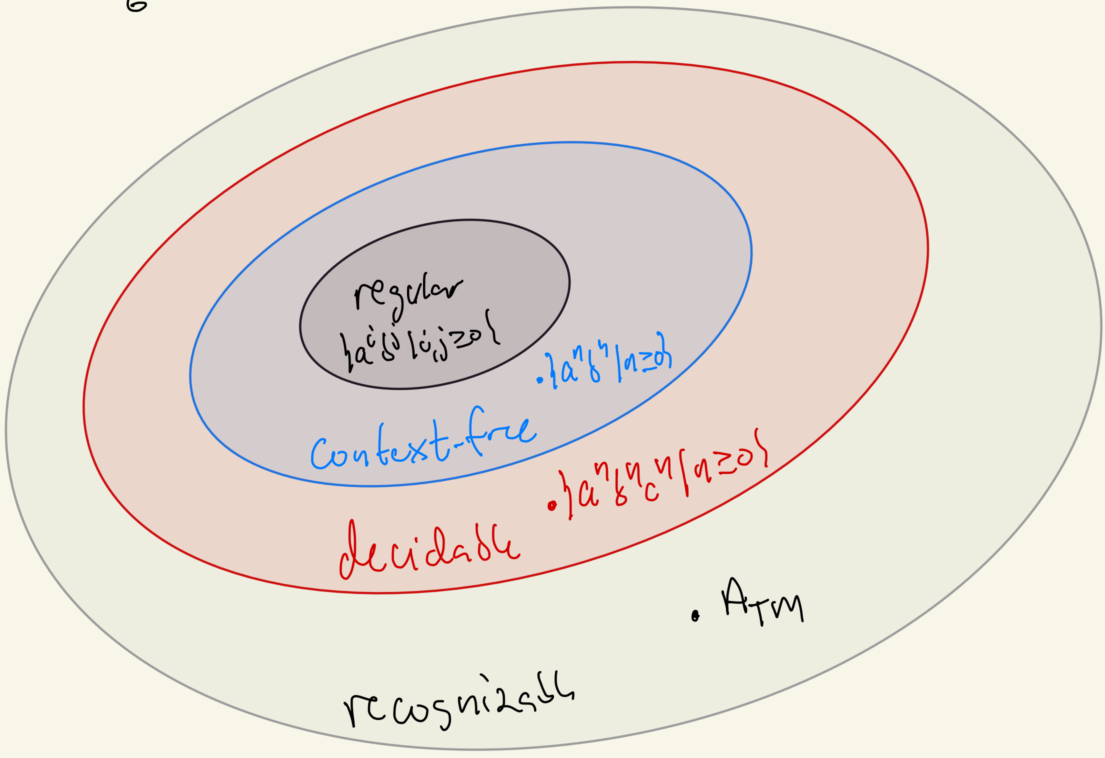
In fact no algorithm can decide EQ_{CFG}!

Theorem 4.9 Every Context-free language is decidable

Proof



M_G^* decides the context-free language $L(G)$

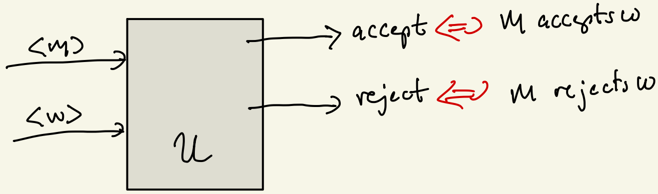


$A_{TM} = \{\langle M\rangle\langle w\rangle \mid M$ is a deterministic TM and $w \in L(M)\}$

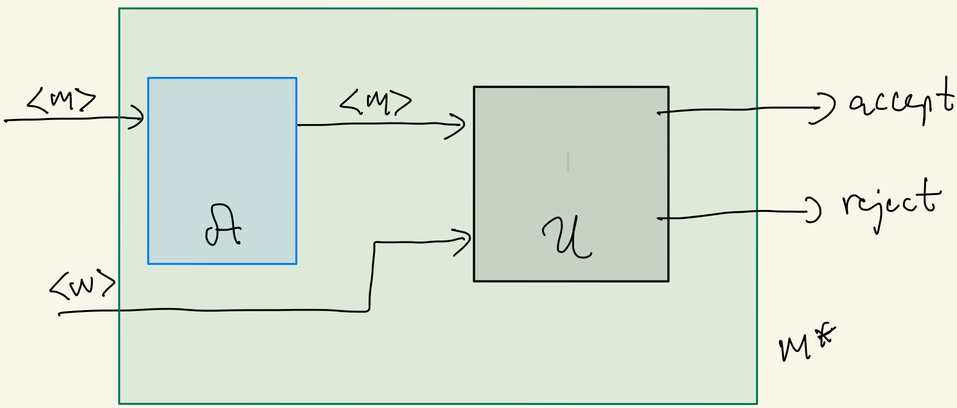
Theorem 4.11 A_{TM} is Turing-recognizable

proof: 1. check whether $\langle M\rangle$ codes a DTM
if not reject $\langle M\rangle\langle w\rangle$

2.



U is the universal TM
it will loop on $\langle M\rangle\langle w\rangle \iff M$ loops on w



A: check whether M is a DTM
if not loop
else send $\langle M\rangle$ to U

M^* recognizes A_{TM}.