1) Find a string for every state
2) Check where those strings send you in machine 2
3) Verify that corresponding states in $M_1, M_2$ are the same: finally and equal on every input, you go to equal states

1) BFS or DFS $|K| \cdot |\Sigma|$
2) traverse BFS tree in other DFA to set up tentative equivalence of states \([k1, k2]\)

creates an array \(s\) we think \(equiv[i]\) is the index in \(M1\) of what state \(i\) in \(M2\)

is equivalent to

\[
\begin{array}{c|ccc}
   & 0 & 1 & 2 \\
\hline
0  & 0 & 1 & 2 \\
1  & 2 & 3 & 0 \\
2  & 3 & 1 & 2 \\
3  & 1 & 2 & 3 \\
\end{array}
\]

3) for each state \(q_i\) in \(M2\)

\[
\text{if } (M2 \cdot q_i \cdot \text{isFinal()} \neq M1 \cdot equiv[q_i] \cdot \text{isFinal()} )
\]

DFAs not \(equiv\)!
for each $\sigma \in \Sigma$

find $j$ s.t. $M_2.\delta(q_i, \sigma) = q_j$

find $k$ s.t. $M_1.\delta(q_{\text{equiv}(i)}) = q_k$

if $\text{equiv}(i) \neq k$

$\text{DFAs not equiv}$

$\text{DFAs are equiv}$

$|k| \cdot |\Sigma|$

total is $O(|k| \cdot |\Sigma|)$
Given NFA $M$, string $w$, does $M$ accept $w$?

backtracking accepts $(q, w)$

for each $(q, e, q') \in \Delta$

if accepts $(q', w)$ return T

return F

make sure not stuck in loop of e-transitions

for each $(q, w.chrAt(0), q')$

if accepts $(q', w.substr(1))$

return T

idea: keep track of set of states we might be in

Set<State> possible; ← precompute e-closures

possible = e-closure (q) (O(k^2*|w|)?)

for each $a \in w$
Set \((\text{State})\) \(\text{next}\); \(\text{linked list of } S\)

for \((q, \text{possible})\in \text{dest from } q\) on \(\sigma\)

\(\text{next. addAll } (S(q, \sigma))\)

\(\text{possible} = e\text{-closure}(\text{next})\)

check if \(F \cap \text{possible} \neq \emptyset\)

if not \(\text{ACCEPTS}\) else \(\text{REJECTS}\)

\(O(1k^2|w|)\)