Conversion of NFA's to DFAs. [Mogensen, Ch. 2]

$\varepsilon$-Closure as a special case of solving set equations

$\varepsilon$-closure $(M) = M \cup \{ t \, | \, s \in \varepsilon$-closure $(M)$ and $s \varepsilon t \in T \}$

$F_{\text{in}}(X) = M \cup \{ t \, | \, s \in X$ and $s \varepsilon t \in T \}$. With this definition to solve the $\varepsilon$-closure set equation, we just
need to solve

\[ X = F^*_m(X) \]

The \( F^*_m \) function is monotonic.

\[ F^*_m(X) \leq F^*_m(Y) \text{ if } X \leq Y. \]

The least solution \( S \) to the equation \( X = F(X) \), when \( F \) is monotonic, satisfies \( S = F(S) \).

(We say that the solution \( S \) is a fixed point of the...
set equation)

Since $\phi \leq S$ (where $S$ is the solution),

$$F(\phi) \leq F(S) = S$$

We therefore can start by guessing that $\phi$ is a solution of $x = F(x)$. If it is, we are done; otherwise, we try $F(\phi)$ as a new guess; and then we continue,
Building the chain

\[ \phi \leq f(\phi) \leq f(f(\phi)) \leq \ldots \text{ until two successive elements of the chain are equal,} \]

or until a fixed point is reached.

Example (Fig. 2.5 in McPheerson's)

NFA that recognizes \((aba)^* a c\)
We want ε-closure C(13), so \( M = \{13\} \), \( F_M = \{13\} \).

We start by guessing \( F_M(\emptyset) = F_{\emptyset 13}(\emptyset) \)

\( F_{\emptyset 13}(\emptyset) = \{13\} \cup \{ s \in \emptyset \text{ and } s \varepsilon 13 = 13\} \)

So, \( \emptyset \) is not a solution, and we continue
\[ F_{\{1\}} (\{1, 3\}) = \{1\} \cup \left\{ t \mid s \in \{1, 3\} \text{ and } s \leq t \in T \right\} = \{1\} \cup \{2, 5\} = \{1, 2, 5\} \]

So, \( \{1, 3\} \) is not a solution, and we continue.

\[ F_{\{1\}} (\{1, 2, 5\}) = \{1\} \cup \left\{ t \mid s \in \{1, 2, 5\} \text{ and } s \leq t \in T \right\} = \{1\} \cup \{2, 5, 6, 7\} = \{1, 2, 5, 6, 7\} \]

So, \( \{1, 2, 5\} \) is not a solution, and we continue.
\[ F_{1,2} \left( \{1,2,5,6,7\} \right) = \left\{ 1 \right\} \cup \left\{ 1, 2, 5, 6, 7 \right\} \text{ for } s \in \{1, 2, 5, 6, 7\} \text{ and } s^2 + 6 \leq 7 \]

\[ = \left\{ 1 \right\} \cup \left\{ 2, 5, 6, 7 \right\} = \{1, 2, 5, 6, 7\} \]

So, \( \{1, 2, 5, 6, 7\} \) is the solution, i.e., the \( \varepsilon \)-closure (\( F_{1,2} \)).

We can make this algorithm more efficient by noticing that the \( \varepsilon \)-closure function is distributive, i.e., it has the property \( F(X \cup Y) = F(X) \cup F(Y) \).
\[ f_{\{1,13\}}(\phi) = \{1,13\} \cup \{5 \ldots \} \in \{1,13\} \]

\[ F_{\{1,13\}}(\{1,13\}) = \{1,13\} \cup \{5 \ldots \} \in \{1,13\} \cup \{2,5\} = \{1,2,5\} \]

\[ F_{\{1,13\}}(\{1,2,5\}) = F_{\{1,13\}}(\{1,13\}) \cup F(\{2,5\}) = \{1,2,5\} \cup \{13\} \cup \{1,2,5\} \]

\[ \frac{1}{2} + 1 \in \{1,2,5\} \text{ and } s \in T \in \{1,2,5\} \cup \{13\} \cup \{1,2,5\} \]

\[ F_{\{1,13\}}(\{1,2,5,6,7\}) = F_{\{1,13\}}(\{1,2,5\}) \cup F_{\{1,13\}}(\{6,7\}) = \]

\[ = \{1,2,5,6,7\} \cup \{1,13\} \cup \{1,2,5\} \cup \{1,5 \in \{6,7\}\} \text{ and } s \in T \in \{1,2,5,6,7\} \cup \{1,13\} \cup \{1,2,5\} \cup \{1,5 \in \{6,7\}\} \]
\[= \{1, 2, 3, 6, 7\} \cup \{1\} \cup \{3\} = \{1, 2, 5, 6, 7\}\]

Distributivity allows to refine the algorithm by using a \underline{work-list}.

The work-list for the previous example (i.e. \(\varepsilon\text{-closure}(\{1, 3\})\)) is:

\[
\begin{align*}
\{1\} \\
\{1, 2, 5\} \\
\{1, 2, 5\}
\end{align*}
\]
\{1,2,5,6,7\}
\{1,2,5,6,7\}
\{1,2,5,6,7\}
\{1,3,5,6,7\} \& \text{Incomplete}
Now the algorithm to construct a DFA from an NFA.

Algorithm 2.3 (The subset construction).

NFA  $N$  $\leadsto$  DFA  $D$

states  $S$  $\leadsto$  $S'$

starting state  $s_0$  $\leadsto$  $s_0'$

accepting states  $F \subseteq S$  $\leadsto$  $F'$

alphabet  $\Sigma$  $\leadsto$  $\Sigma$ (same)
Transition relation

\[ s'_0 = \varepsilon\text{-closure } \{s_0\} \]

move \((s', c) = \varepsilon\text{-closure } \{t \mid s \in s' \text{ and } s^c t \in T\} \)

\[ S' = \{s'_0\} \cup \{\text{move } (s', c) \mid s' \in S', c \in \Sigma\} \]

\[ S = \text{a set function, to be solved in the same way as the } \varepsilon\text{-closure function} \]
\[ F' = \{ s' \in S' | s' \cap F \neq \emptyset \} \]

Example: again, we use the NFA of Figure 2.5 (see above) that recognizes \((a \bar{b})^* a c\)

The initial state of the DFA is:

\[ s_0' = \varepsilon - \text{closure} (\{ s_0 \}) = \varepsilon - \text{closure} (\{ 1, 2 \}) = \{ 1, 2, 5, 6, 7 \} \]

To compute more, we use the work-list procedure.
Start with the worklist (i.e., the uncompleted set of states of the DFA) \( S' = \{ s_0 \} \)

\[
\text{move}(s_0', a) = \varepsilon\text{-closure}\left( \{ s \mid s \in \{1,2,5,6,7\} \text{ and } s^a \in T \} \right) = \\
= \varepsilon\text{-closure}\left( \{ 3,8 \} \right) = \\
= \{ 3,8,1,2,5,6,7 \} \\
= S_1'
\]
\text{move}(s_0',b) = \ldots = \emptyset \cup \{8,1,2,5,6,7\} = s_2'

\text{move}(s_0',c) = \mathcal{E}\text{-closure}\left(\{t\mid s \in \{1,2,5,6,7\} \text{ and } s^c + t \in \mathcal{T}\}\right) = \\
= \mathcal{E}\text{-closure}(\{1\}) = \emptyset

The empty set of NFA states is not a DFA state, so there is no transition from \(s_0'\) on \(c\).

Now, our worklist (incomplete set of DFA states), \(s_1'\) is
\{ s_0', s_1', s_2' \}

We pick $s_1'$ and calculate its transitions:

\text{move} (s_1', \varepsilon) = \cdots = s_1' \quad \text{and} \quad \text{move} (s_1', b) = \cdots = s_2' \quad \text{and} \quad \text{move} (s_1', c) = \varepsilon \text{- closure} (\{ t | s \in \{ 3, 8, 1, 2, 5, 6, 7 \} \text{ and } s^t \in \Gamma \}) = \{ 4 \} = s_3'
The work list changes to

\[
\tilde{\{ S_0, S_1, S_2, S_3 \}}
\]

We pick \( S_2' \) and compute

\[
\text{move}(S_5', a) = \cdots = s_1'
\]

\[
\text{move}(S_2', b) = \cdots = s_2'
\]

\[
\text{move}(S_7', c) = \cdots = s_3'
\]

The worklist is \( \tilde{\{ S_0, S_1, S_2, S_3 \}} \).
We pick $s_3'$ and compute

\[ \text{move} \left( s_3', a \right) = \{3\} \]

\[ \text{move} \left( s_3', b \right) = \{1\} \]

\[ \text{move} \left( s_3', c \right) = \{3\} \]

No new state to the old to the worklist.

All elements of the worklist are checked - done.
$F' = \{S_2\}$

$S_0 \text{ and } S_2'$ are equivalent
Minimality!