CS 442/542

NFA -> DFA
Subset Construction
Building an DFA from an NFA

- Subset construction algorithm
  - Constructs a DFA from an NFA by building a DFA whose states represent sets of states of the NFA
  - NFA: \((N, \Sigma, \delta_N, n_0, N_A)\)
  - DFA: \((D, \Sigma, \delta_D, d_0, D_A)\)
    - Note the alphabets are the same
Subset Construction Algorithm

Functions

• \( \varepsilon \text{-closure}(q) \) returns the set of states that can be reached from state \( q \) in the NFA on an epsilon transition. \( q \) is included in the result.

• \( \text{Delta}(q, c) \) where \( q \) is a set of NFA states and \( c \) is a symbol from \( \Sigma \), returns the set of NFA states reachable from an NFA state in \( q \) on the symbol \( c \)
  
  \[ \bigcup_{s \in q} \delta_N(s, c) \]
Subset Construction Algorithm

Transitions

- $T[q,c]$ where $q$ is a set of NFA states and $c$ is a symbol in $\Sigma$, is given the value of the $\varepsilon$-closure of the set of NFA states that states in $q$ can reach on $c$
Subset Construction Algorithm

$q_0 <-- \epsilon$-closure($q_0$);
$Q <-- q_0$;
$\text{Worklist} <-- \{q_0\}$;

while (Worklist $\neq \emptyset$) do
    remove $q$ from Worklist;
    for each $c \in \Sigma$ do
        $t <-- \epsilon$-closure(Delta($q, c$));
        $T[q, c] <-- t$;
        if $t \not\in Q$ then
            add $t$ to $Q$ and to Worklist;
            end;
        end;
end;
Subset Construction Algorithm

- How to create the DFA \((D, \Sigma, \delta_D, d_0, D_A)\) from \(Q\) and \(T\)
  - Each \(q_i\) in \(Q\) is named \(d_i\) (in particular \(q_0\) is named \(d_0\))
  - For each \(q_i\) in \(Q\) and each \(c\) in \(\Sigma\) where \(T[q_i, c] = q_j\), \(\delta_D(d_i, c) = d_j\)
  - \(D\) is the set of all \(d_i\)
  - \(D_A\) is the set of all \(d_i\) where \(q_i\) contained an accept state from \(N_A\)
Practice Problem

• (a) Create an NFA from the RE $(0 \mid 1) \ (0 \mid 1)$
• (b) Create a DFA from the answer to part a