DFA Minimization

- Minimize the DFA \( \langle D, \Sigma, \delta, d_0, D_A \rangle \)
- The algorithm builds a new machine from subsets of states of the original machine
- The algorithm first builds two subsets: the set of final states and the set of non-final states
- A subset is split if the subset has a conflict on a symbol
- A subset has a conflict on a symbol, \( c \), when the transitions on \( c \) of two (or more) states in the subset do not go to states in the same subset.
- The algorithm halts when no subsets have conflicts (i.e. no more splits need to be done)
Split (S is a set of states from the original DFA)

Split(S) {
    for each \( c \in \Sigma \) do
        if \( c \) splits \( S \) into \( s_1 \) and \( s_2 \) then return \( \{s_1, s_2\} \);
    end;
    return \( S \);
}

DFA Minimization

\[ T = \{ D_A, (D - D_A) \}; \]
\[ P = \emptyset \]
while ( \( P \neq T \) ) do
\[ P \leftarrow T; \]
\[ T \leftarrow \emptyset; \]
\[ \text{for each } p \in P \text{ do} \]
\[ T = T \cup \text{Split}(p); \]
end;
end;
Practice Problem

- (a) Build the NFA that recognizes the language specified by the following regular expression: \((0 \mid 1) 11 (1)^*\)
- (b) Build the DFA from the NFA created in part a
- (c) Build the minimized DFA for the DFA created in part b