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

- ε-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.
- Delta(q, c) where q is a set of NFA states and c is a symbol from Σ, returns the set of NFA states reachable from an NFA state in q on the symbol c
  - $\cup_{s \in q} \delta_N(s,c)$

Subset Construction Algorithm Transitions

- T[q,c] where q is a set of NFA states and c is a symbol in Σ, is given the value of the ε-closure of the set of NFA states that states in q can reach on c

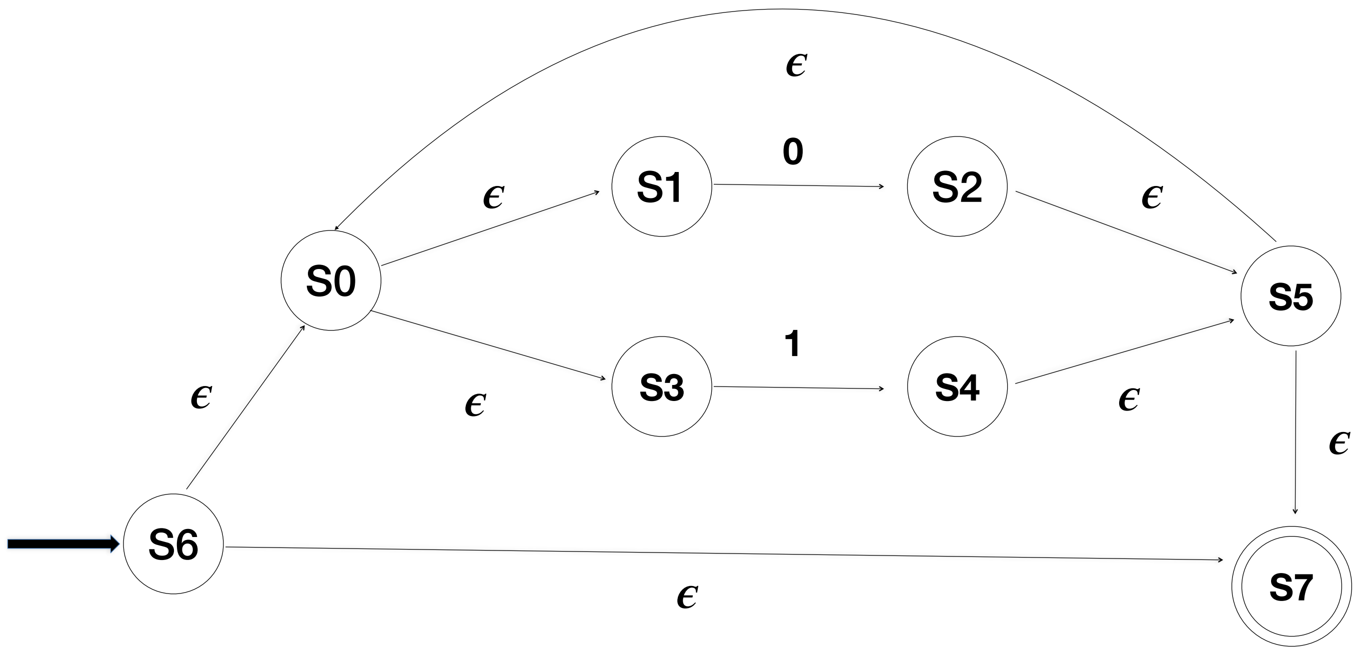
Subset Construction Algorithm

```
q0 <-- ε-closure(n0);
Q <-- q0;
Worklist <-- {q0};
while (Worklist ≠ ∅ ) do
    remove q from Worklist;
    for each c ∈ Σ do
        t <-- ε-closure(Delta(q,c));
        T[q,c] <-- t;
        if t ∉ Q then
            add t to Q and to Worklist;
    end;
end;
```

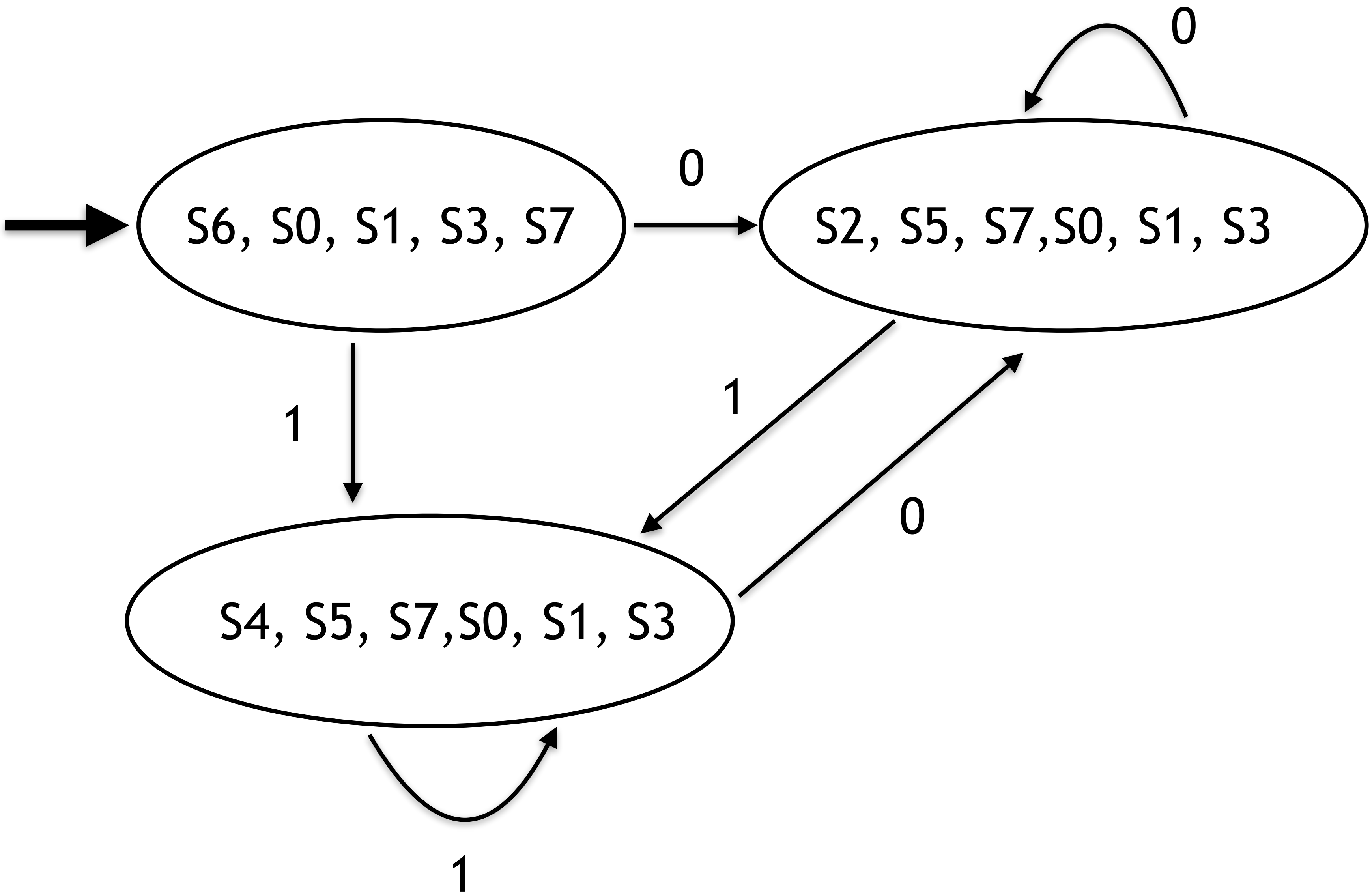
Subset Construction Algorithm

- 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 Σ 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

NFA for (0|1)*



NFA -> DFA



Practice Problem

- (a) Create an NFA from the RE (0 | 1) (0 | 1)
- (b) Create a DFA from the answer to part a