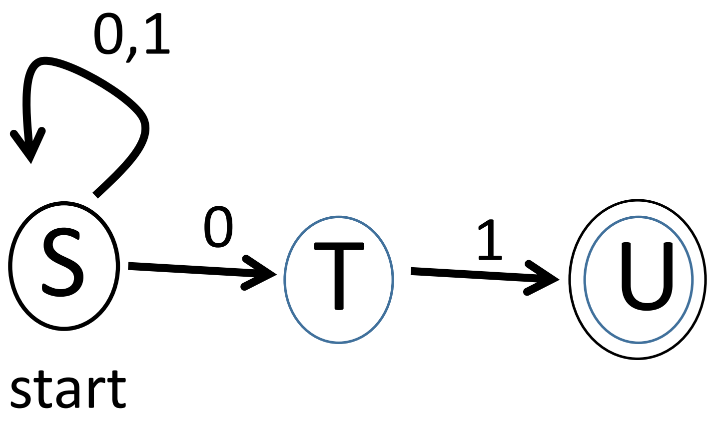
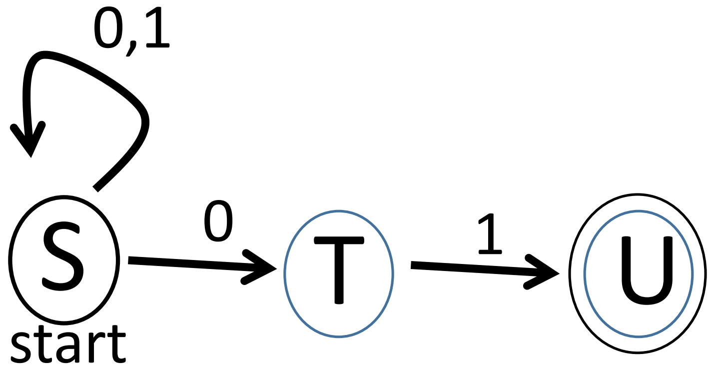
Nondeterministic Finite Automata

See Section 2.3 of the text

Consider the following automaton:



This is called a "Nondeterministic Finite Automaton", or NFA because in state S there are two options on input 0: we can stay in state S or transition to state T.



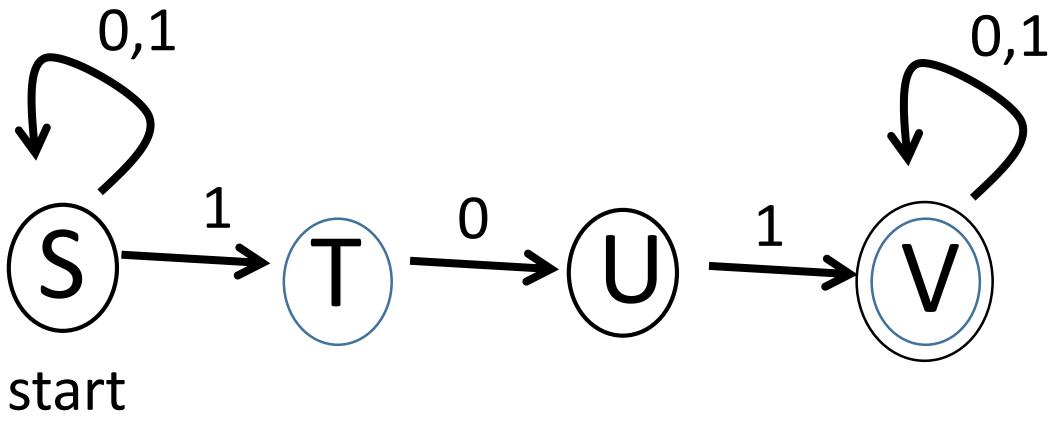
In general, an NFA is a quintuple $(Q, \Sigma, \delta, s, F)$ where Q, Σ, s, and F have the same meanings as in a DFA, and for each state t and letter a in Σ, $\delta(t,a)$ is a set of states.

We say that such an automaton accepts string $w=w_0w_1..w_{n-1}$ if there is a sequence of states $s= t_0t_1..t_n$ where each t_{i+1} is in $\delta(t_i,w_i)$ and t_n is final.

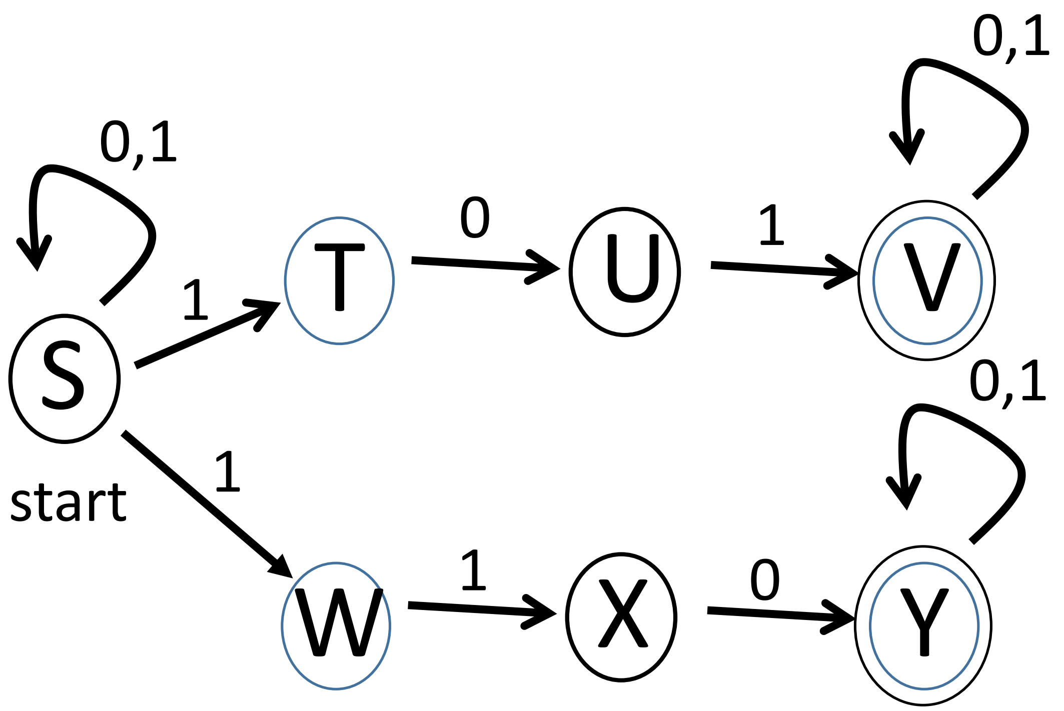
The automaton above accepts $(0+1)^*01$, which is the set of all strings of 0's and 1's that end in 01.

NFAs are often easier to design than DFAs.

Example: Construct an NFA that accepts strings containing 101.



Example. Find an NFA that accepts strings containing either 101 or 110.



First Theorem of the Course: For any NFA there is a DFA accepting the same language. So the language accepted by any NFA is regular.

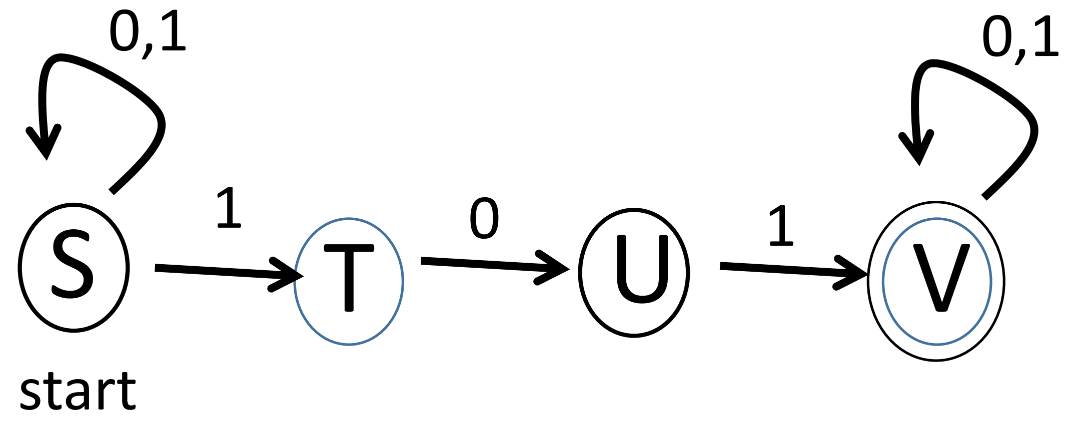
Proof: Start with NFA (Σ, Q, δ, s, F) . Construct DFA (Σ, Q', δ', s', F') :

1. Q' consists of sets of states from Q.
2. s'={s}
3. For each state P={q_0...q_k} in Q' and each a in Σ, make a new state P'= $\bigcup_{i=0}^{k} \delta(q_i, a)$. Then δ'(P,a)=P'.
4. F' consists of all of the states in Q' that contain a state in F.

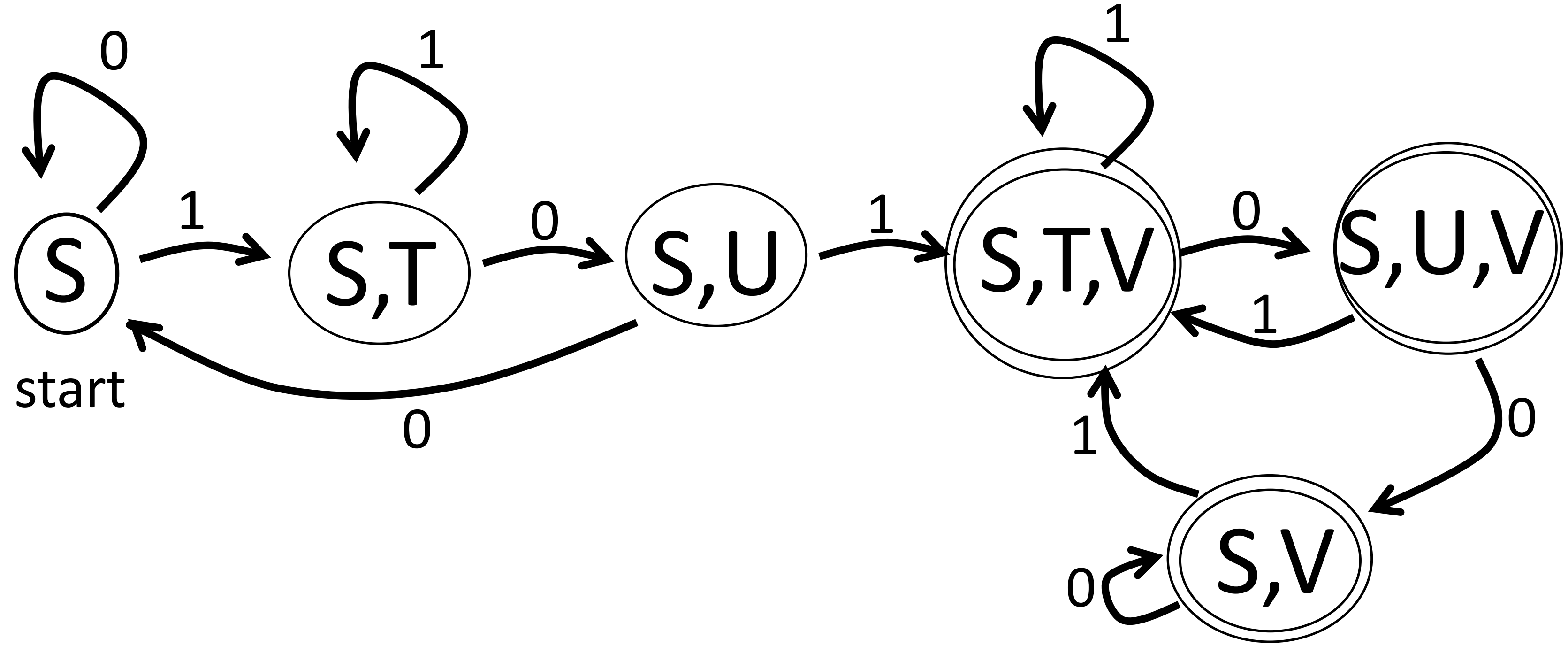
In English, the DFA models all of the states where we could be in the NFA.

construction:

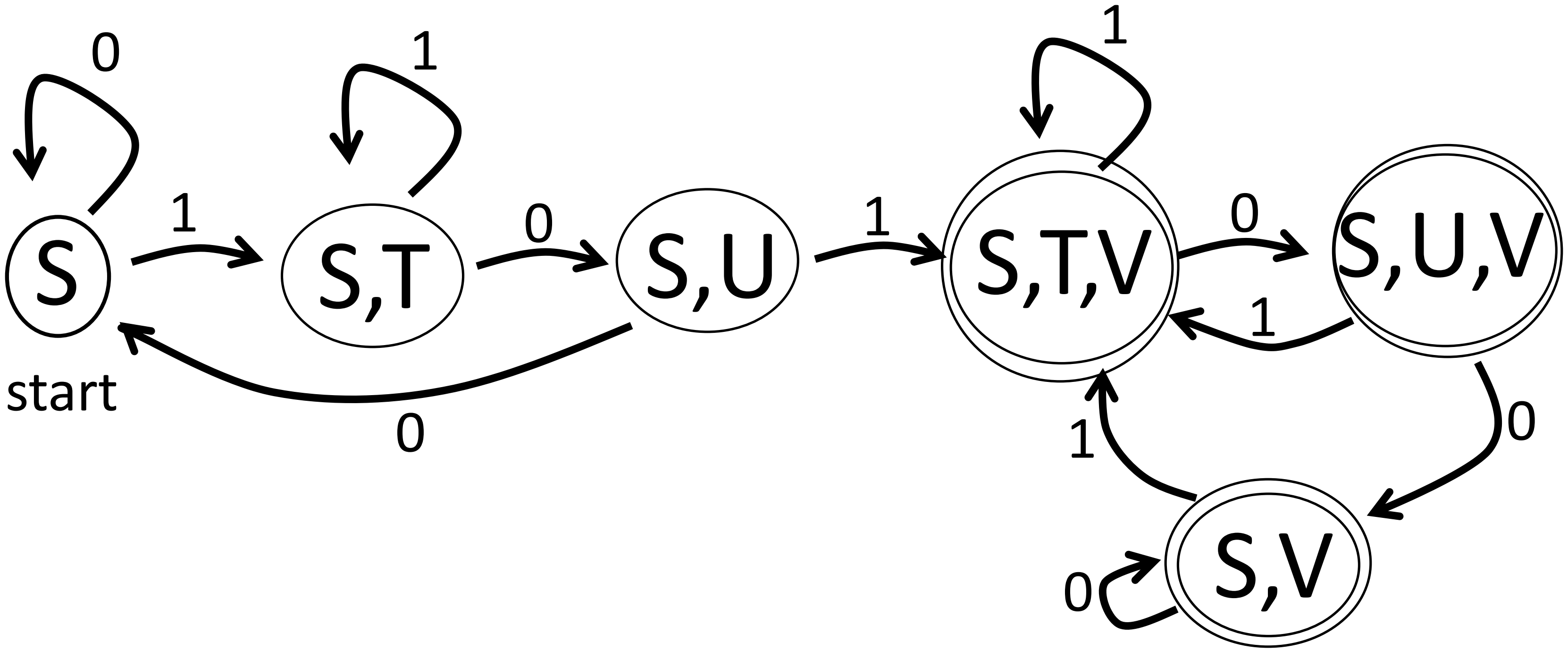
NFA:



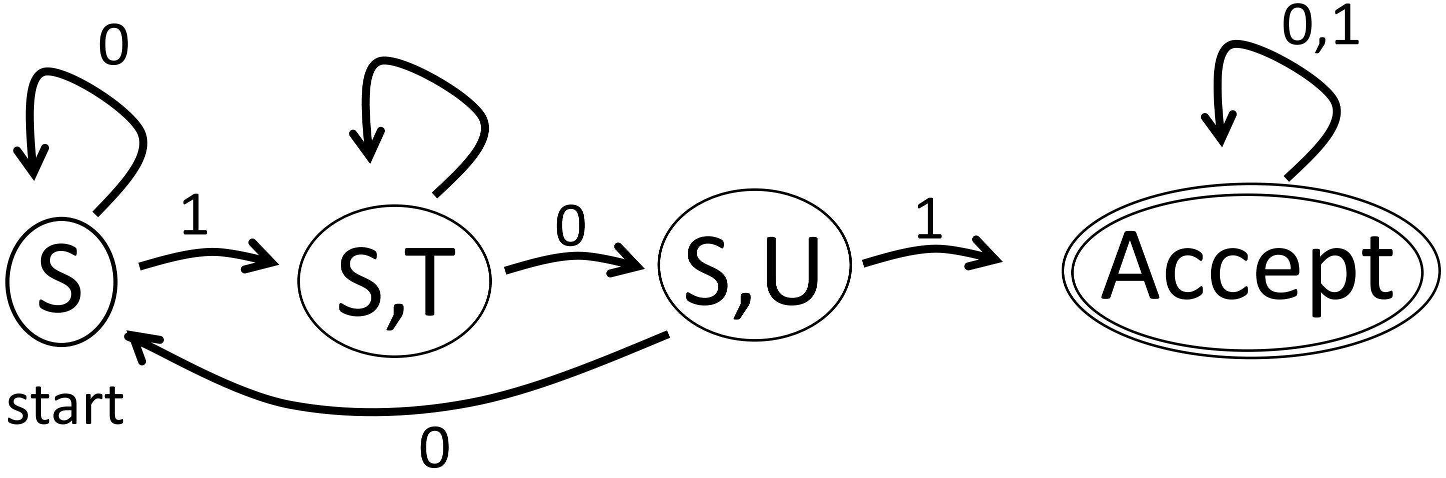
DFA:



DFA:

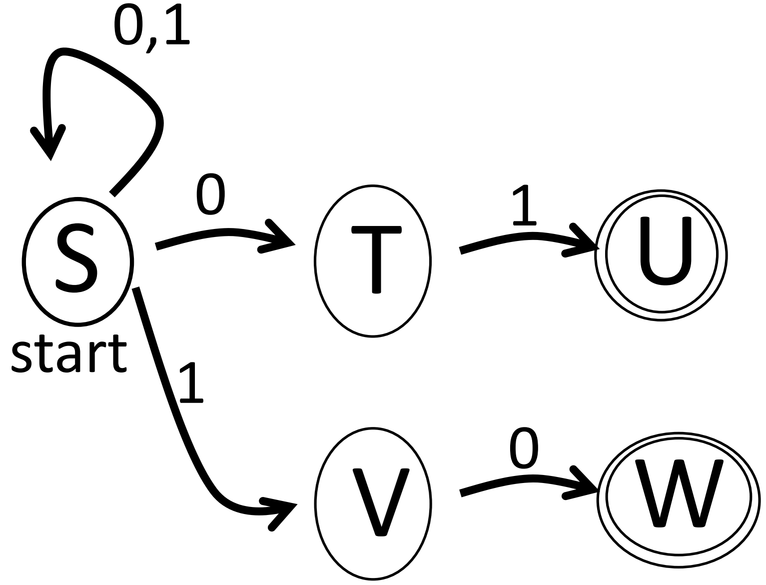


Note that this is equivalent to

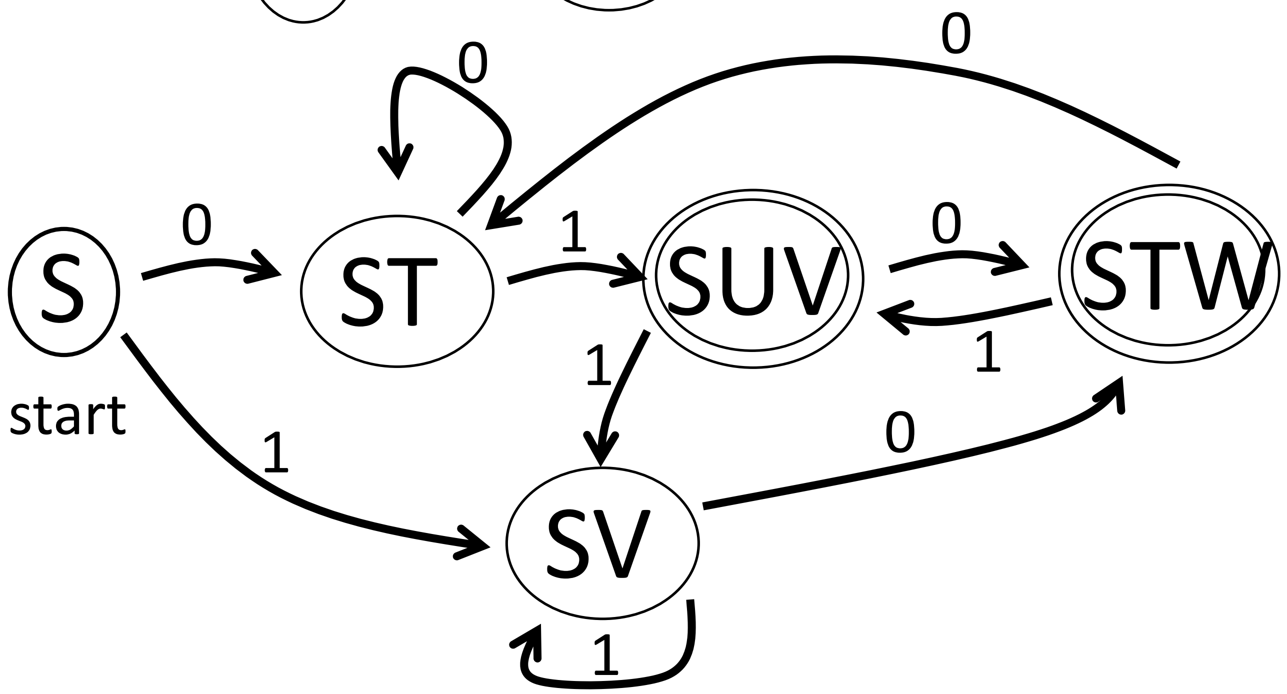


Example: Find a DFA that accepts all strings ending in 01 or 10

NFA:



DFA:



Now, we need to prove that the NFA and DFA accept the same language.

1. Suppose $w=a_0a_1....a_{n-1}$ is a string accepted by the NFA. Then there is a sequence of NFA states

   $q_0=s$
   $q_1 \in \delta(q_0,a_0)$
   $q_2 \in \delta(q_1,a_1)$
   etc. with q_n in F.

   Well, in the DFA $\delta'(\{s\},a_0)=Q_1$, where $q_1 \in Q_1$
   $\delta'(Q_1,a_1)=Q_2$, where $q_2 \in Q_2$ and so forth.

   Ultimately this produces DFA state Q_n with $q_n \in Q_n$ and $q_n \in F$, so $Q_n \in F'$.

   This means the DFA accepts w.

2. On the other hand, suppose $w=a_0a_1....a_{n-1}$ is a string accepted by the DFA. So there is a sequence of states

    $Q_0=\{s\}$

    $Q_1=\delta'(Q_0,a_0)$

    etc. where Q_n contains an element of F.

   Note that there is a path on a_0 from s to every state in Q_1. $Q_2=\delta'(Q_1,a_1)$, so every state in Q1 can be reached on a1 from a state in Q. This means there is a path on a_0a_1 from s to every state in Q_2, and so forth. In the end there is a path on input $w=a_0a_1....a_{n-1}$ from s to every state in Q_n, and one of those is an element of F, so the NFA also accepts w.

   This completes the proof.