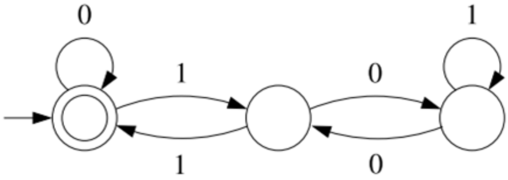
1.3 Examples

Example 6

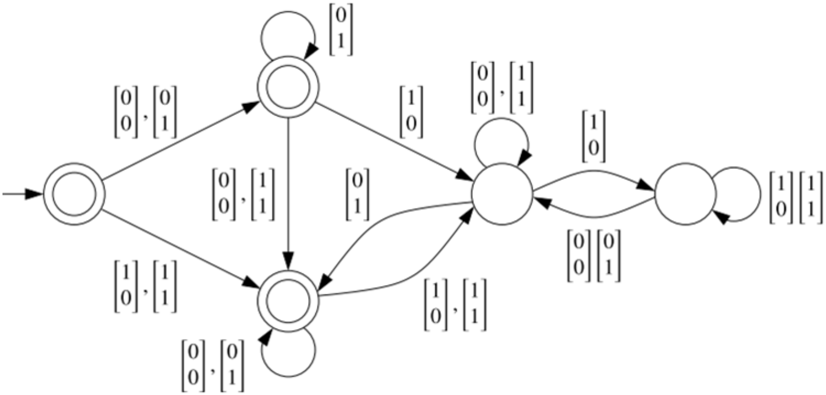
This is a DFA recognizing the multiples of 3, in binary notation:



The states, from left to right, correspond to the residue $\bmod 3$ of the binary number read so far. If this residue is r and the next digit being read is b, then the new residue is $2r + b \bmod 3$, as reflected by the arrows in the above diagram.

Example 7

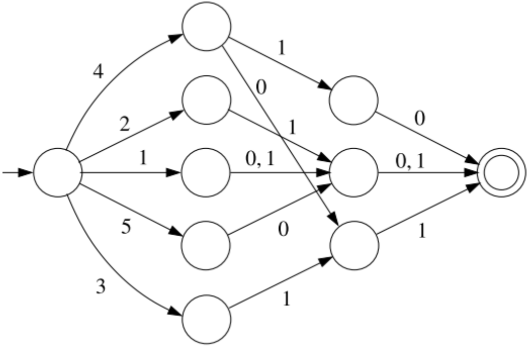
This is a DFA recognizing the nonnegative solutions of $2x - y \leq 2$ in binary (with least significant digit first):



Example 8

This is a DFA recognizing the (initial or intermediate) states of the program leading to termination. The inputs to the DFA are (in order) the number of the current line in the program, the value of the (binary) variable x, and the value of the (binary) variable y:

```
1   while x = 1 do
2      if y = 1 then
3         x ← 0
4      y ← 1 − x
5   end
```



Definition 9

Let $A = (Q, \Sigma, \delta, q_0, F$ be an automaton. A state $q \in Q$ is reachable from $q' \in Q$ if $q = q'$ or if there exists a run $q' \xrightarrow{a_1} \dots \xrightarrow{a_n} q$ on some input $a_1 \dots a_n \in \Sigma^*$. A is in normal form if every state is reachable from the initial state.

Unless we say otherwise, we always assume that automata are in normal form!

2. Conversion algorithms

2.1 NFA to DFA, power set construction

Theorem 10

Let L be the language accepted by some nondeterministic finite automaton. Then we can effectively construct a DFA M with

$$L = L(M) \ .$$

Proof.

Let $N = (Q, \Sigma, \delta, S, F)$ be an NFA.

Define

1. $M' := (Q', \Sigma, \delta', q'_0, F')$
2. $Q' := \mathcal{P}(Q)$ $\quad$ ($\mathcal{P}(Q) = 2^Q$ power set of Q)
3. $\delta'(Q'', a) := \bigcup_{q' \in Q''} \delta(q', a)$ for all $Q'' \in Q'$, $a \in \Sigma$
4. $q'_0 := S$
5. $F' := \{Q'' \subseteq Q;\ Q'' \cap F \neq \emptyset\}$

Thus

NFA N:	Q	Σ	δ	S	F
DFA M':	2^Q	Σ	δ'	S	F'

Proof (cont'd):

We have:

$$\begin{aligned} w \in L(N) \quad &\Leftrightarrow \quad \hat{\delta}(S, w) \cap F \neq \emptyset \\ &\Leftrightarrow \quad \widehat{\delta'}(q'_0, w) \in F' \\ &\Leftrightarrow \quad w \in L(M'). \end{aligned}$$

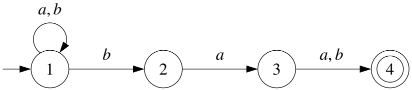
Here, $\hat{\delta}$ denotes the canonical extension of δ to words $w \in \Sigma^*$, and analogously $\widehat{\delta'}$. $\square$

The corresponding algorithm for converting an NFA into a DFA is called subset construction, power set construction, or Myhill construction.

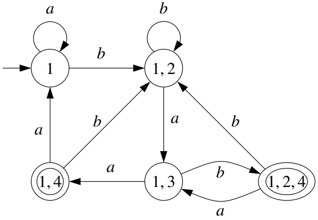
Remark: Of course, the algorithm should also put the NFA it constructs into normal form.

Example 11

NFA:

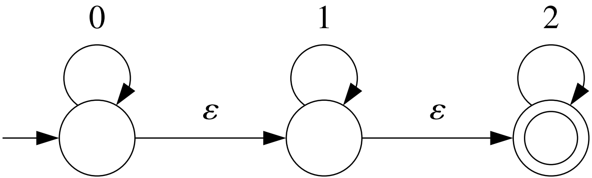


DFA:



2.2 NFA-e to DFA

Consider the NFA-ϵ

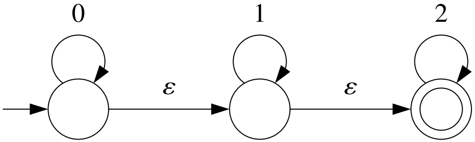


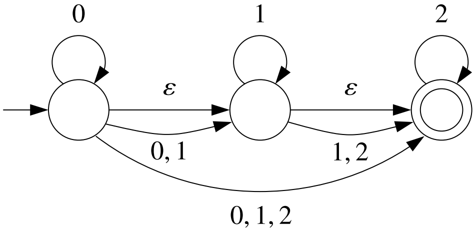
accepting $L(0^*1^*2^*)$.

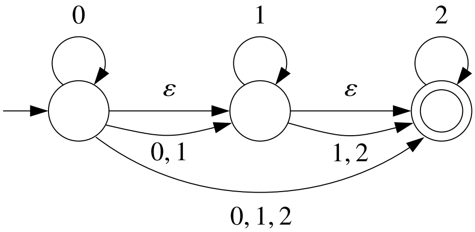
We perform the following algorithm NFA-ϵtoNFA:

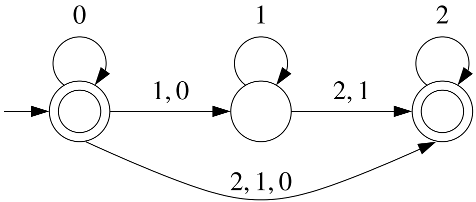
Input: NFA-ϵ $A = (Q, \Sigma, \delta, S, F)$
Output: NFA $B = (Q', \Sigma, \delta', q_0', F')$ with $L(A) = L(B)$
$Q_0' := S$; $Q' := S$; $\delta' := \emptyset$; $F' := F \cap S$
$\delta'' := \emptyset$; $W := \{(q, \alpha, q') \in \delta \mid q \in S\}$
while $W \neq \emptyset$ **do**
  pick (q_1, α, q_2) from W
  if $\alpha \neq \epsilon$ **then**
    add q_2 to Q'; add (q_1, α, q_2) to δ'; **if** $q_2 \in F$ **then** add q_2 to F' **fi**
    for all $q_3 \in \delta(q_2, \epsilon)$ **do if** $(q_1, \alpha, q_3) \notin \delta'$ **then** add (q_1, α, q_3) to W **fi**
    for all $a \in \Sigma, q_3 \in \delta(q_2, a)$ **do if** $(q_2, a, q_3) \notin \delta'$ **then** add (q_2, a, q_3) to W **fi**
  else co $\alpha = \epsilon$ **oc**
    add (q_1, α, q_2) to δ''; **if** $q_2 \in F$ **then** add q_1 to F' **fi**
    for all $\beta \in \Sigma \cup \{\epsilon\}, q_3 \in \delta(q_2, \beta)$ **do**
      if $(q_1, \beta, q_3) \notin \delta' \cup \delta''$ **then** add (q_1, β, q_3) to W **fi**
  fi
od

Example 12






0
1
2
ε
ε
0, 1
1, 2
0, 1, 2


0
1
2
1, 0
2, 1
2, 1, 0