Nondeterministic Finite Acceptors

Why use **nondeterministic** algorithms or automata?

They are sometimes **simpler** than their deterministic counterparts

Famous open problem: $P \stackrel{?}{=} NP$

Nondeterministic finite acceptors (**nfa's**) are also often **simpler** than the corresponding dfa.

An **nfa** can have >1 possible next state (or none!) from a given state with a given input symbol.

An nfa, A, **accepts** a word, w, if there is **at least** 1 path through A with input w from the start state to some final state.

Definition: An **nfa** is a quintuple [cf. dfa's, p.2-2]

$$M = (Q,\ \Sigma,\ \delta,\ q_0,\ F)$$

where Q, Σ, q_0, F are as for **dfa's**, but

$$\delta\ :\ Q \times (\Sigma \cup \{\lambda\}) \longrightarrow 2^Q$$

Note:

There are 3 major differences with dfa's:

(1) Range of δ in **powerset** of $\boldsymbol{Q}$.

(2) Edge of transition graph can be labeled

$$q_1 \xrightarrow{\lambda} q_2, \quad (q_1, q_2 \in Q)$$

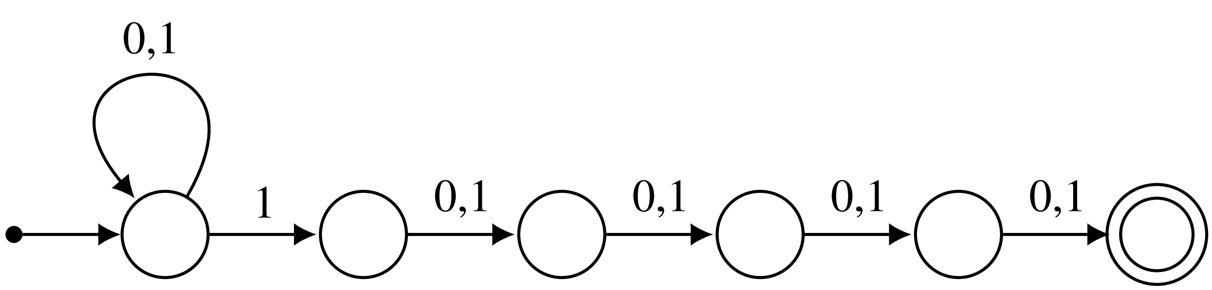
i.e. $\boldsymbol{M}$ can change state without reading a symbol.

(3) $\boldsymbol{\delta(q, a) = \varnothing}$ is possible.

Example 1: Let $\Sigma = \{0, 1\}$,

$\boldsymbol{L = \{u \in \Sigma^* \mid \text{the } 5^{\text{th}} \text{ symbol from the right is } 1\}}$.

Here is an nfa, $\boldsymbol{N}$ for $\boldsymbol{A}$:



When $\boldsymbol{N}$ sees a 1 at the start state, it must "guess" whether to loop around again or go right.

If $u \in L$, then there is **some** path for x from q_0 to F.

If $u \notin L$, there is no such path.

<u>Exercises:</u> [Linz § 2.1: Exercise 4.(a, b, c)]

Construct **dfa's** that accept the following languages:

4.a) All strings with exactly one 'a'

4.b) All strings with at least two 'a''s

4.c) All strings with no more than two 'a''s

The Subset Construction

We will see: for every **nfa**, N, which accepts a language, L, there is a **dfa**, M, (usually more complicated) which accepts the same language, i.e. $L(m) = L(n)$.

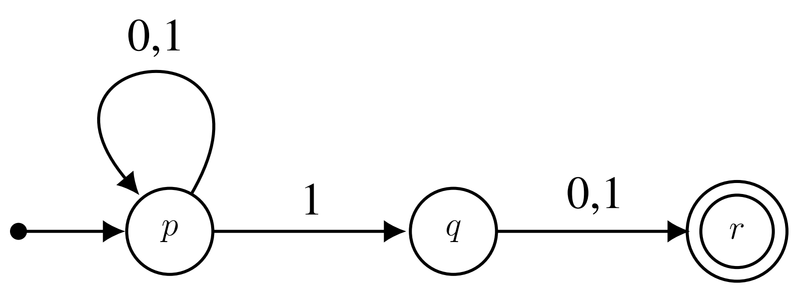
The General Idea:

Each state of M is a **set of states** of N: the set of **all states** of N which can be reached by a particular word.

Example 2: $\Sigma = \{0, 1\}$. [Kozen, p. 29]

$L = \{u \in \Sigma^* \mid$ **the 2nd symbol from the right is 1**$\}$

L is accepted by **nfa**, $N =$



Here is the (**nondeterministic**) **transition table** for $\boldsymbol{N}$: [See p. 2-3]

	0	1
$\longrightarrow p$		
q		
F : r		

From this we can make a **deterministic transition table** for an **equivalent** dfa, $\boldsymbol{M}$.

Its **states** are all possible **sets of states** of $\boldsymbol{N}$:

	0	1
$\varnothing$		
$\longrightarrow$ $\{p\}$		
$\{q\}$		
F : $\{r\}$		
$\{p, q\}$		
F : $\{p, r\}$		
F : $\{q, r\}$		
F : $\{p, q, r\}$		

Note:

- The **start state** of M is the set containing the **start state** of N, i.e. $\{p\}$.
- The **final states** of M are those states that **contain any of the final states** of N
- Some states of M are **inaccessible** from the start state, $\{p\}$: $\{q, r\}$, $\{q\}$, $\{r\}$, $\varnothing$.

  These can be left out of the table.

Note:

The dfa, M, formed by the subset construction from N in this way is more **complex** than N:

If N has k states, then M has 2^k states.

Example: Let $\Sigma = \{a\}$, [Kozen, p. 30]

$L = \{x \in \{a\}^* \mid |x| \text{ is divisible by 3 or 5}\}$

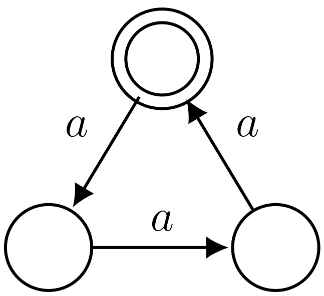
Find a **dfa** and **nfa** for L.

Let $L_1 = \{u \in \Sigma^* \mid |u| \text{ is divisible by 3}\}$,

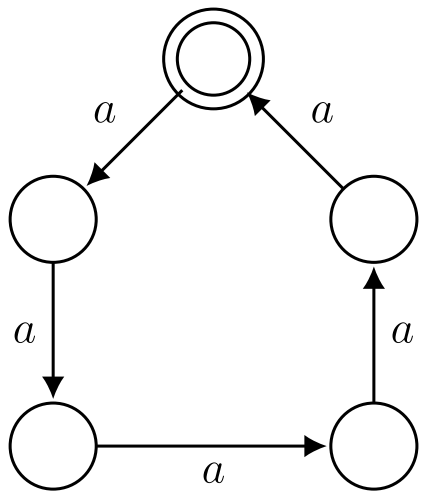
and $L_2 = \{u \in \Sigma^* \mid |u| \text{ is divisible by 5}\}$.

Then have **dfa**'s for $\boldsymbol{L_1}$ and $\boldsymbol{L_2}$:

$\boldsymbol{M_1}$:

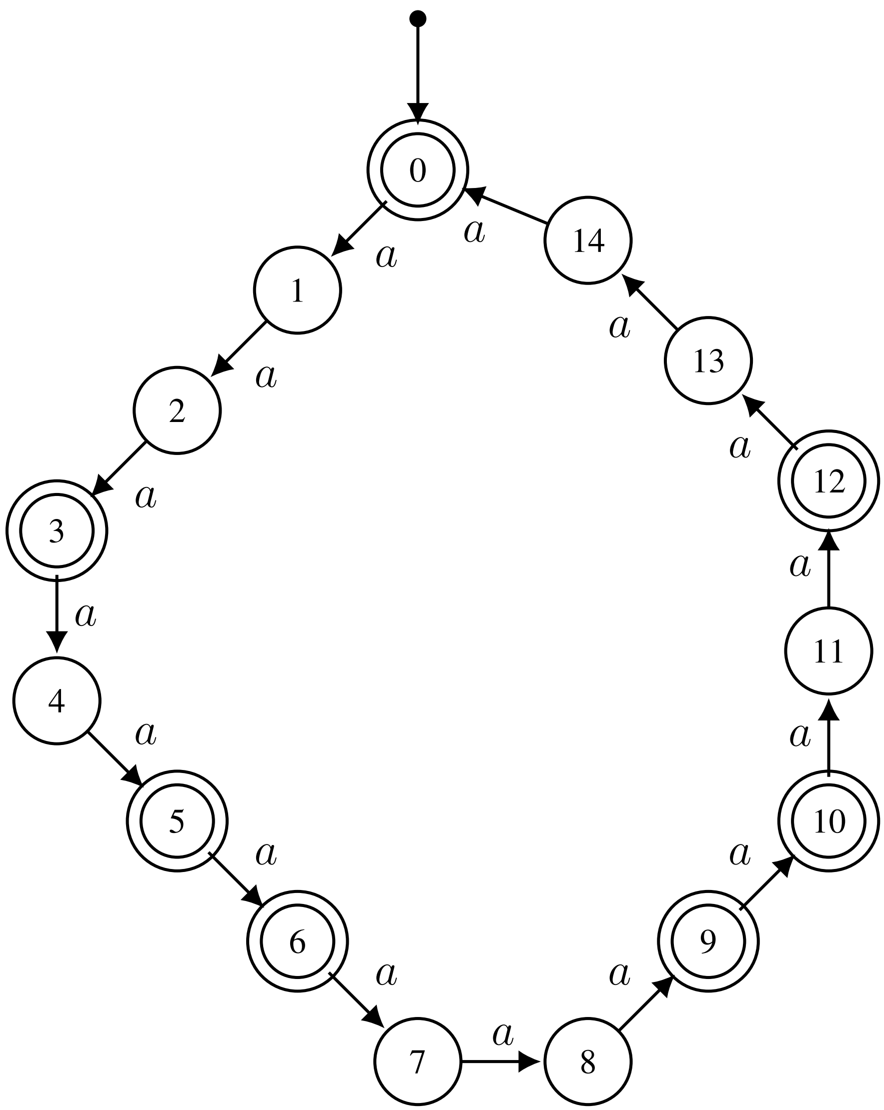


$\boldsymbol{M_2}$:



Then an **nfa** for $L = L_1 \cup L_2$ is...

Here is a **dfa** for ***L***:



15 states!

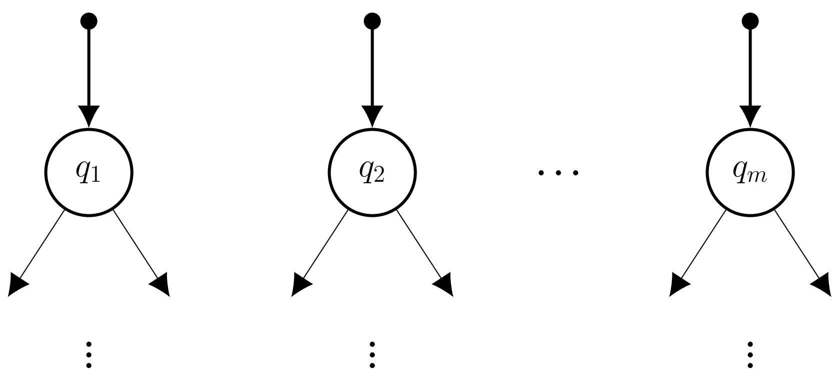
Which are the **final states**?

Note:

Suppose we **change** the **definition** of **nfa** to allow > 1 start state.

Q. Would that make a difference to the power of nfa's?

A. No! Consider an **nfa** with $\boldsymbol{m > 1}$ start states:



We can **transform** this to an **equivalent** nfa with 1 new start state $\boldsymbol{q_0}$, and $\boldsymbol{\lambda}$-edges from $\boldsymbol{q_0}$ to $\boldsymbol{q_1, \ldots, q_m}$, which are no longer start states.

[See Linz Sec. 2.2, Exercises: 19 (6ed), or 18 (5ed).]