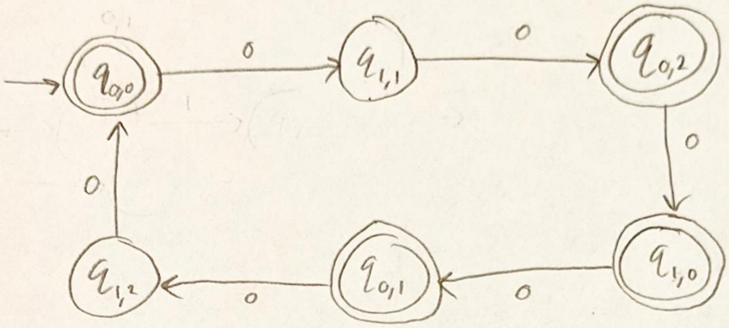
Nondeterminism is the introduction of "chance" or the "multiverse" into our machines. This is mostly syntactic sugar right now (see DFA/NFA Equivalence) but foreshadows important topics later.

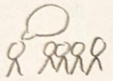
Recall that DFAs are deterministic in the sense that how it processes input at any given time is fully determined at "compilation time", whereas NFA behavior is determined at "runtime".

Ex



M

$\Sigma = \mathbb{Z}_1$

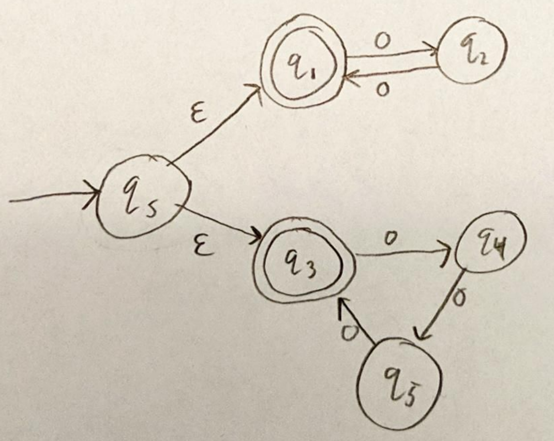


what does M do?

$L(M) = \{ w \in \Sigma^* \mid 2 \mid |w| \vee 3 \mid |w| \}$

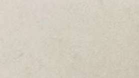
DFAs accept if the final state of the unique path through it as specified by its input is an accepting state. NFAs accept if the final state of a path through it as specified by its input is an accepting state.

Ex



N

$\Sigma = \mathbb{Z}_1$



what does N do? $L(N) = \{ 0^{2n} \mid n \in \mathbb{N} \} \cup \{ 0^{3n} \mid n \in \mathbb{N} \}$

Notice two key features of NFAs in N. We allow empty ε transitions which occur spontaneously without consuming an input. We also allow multipule identical transitions out of a state. In N, we test if either 2 divides the input length or if 3 does seperately. But since N is an NFA, if <u>either</u> path ends in an accepting state, we accept.

Ex) Prove $L(M) = L(N)$.

We do so in the usual way by showing $L(M) \subseteq L(N)$ and $L(N) \subseteq L(M)$ (double [set] inclusion).

Let $w \in L(M)$. Then $2 \mid |w|$ or $3 \mid |w|$. Thus $w = 0^{2n}$ or $w = 0^{3n}$ for some $n \in \mathbb{N}$. Hence $w \in L(N)$.

Now let $w \in L(N)$. Then $w = 0^{2n}$ or $w = 0^{3n}$ for some $n \in \mathbb{N}$,

Case $w = 0^{2n}$: Then $|w| = 2n$ and $2 \mid 2n$.

Case $w = 0^{3n}$; Then $|w| = 3n$ and $3 \mid 2n$.

In either case $w \in L(M)$.

$\square$

Formal NFA

A <u>nondeterministic finite automaton</u> M is a 5-tuple

$$M = (\underset{\text{states}}{Q}, \underset{\text{alphabet}}{\Sigma}, \underset{\substack{\text{transition}\\\text{function}\\ \delta: Q \times \Sigma_\varepsilon \to P(Q)}}{\delta}, \underset{\substack{\text{start}\\\text{state}}}{q_0}, \underset{\substack{\text{accepting}\\\text{states}}}{F}),$$

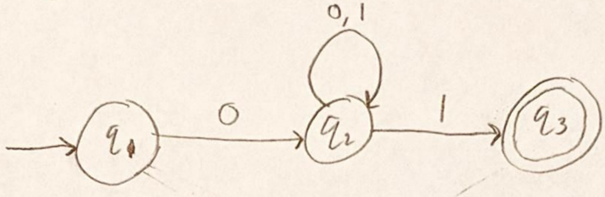
where $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$.

M <u>accepts</u> a string w if there exists a path $p = s_1, s_2 s_3 \cdots s_K$ where $k \in \mathbb{N}$, $s_1 = q_0$, $s_K \in F$, and for each i, $s_{i+1} \in \delta(s_i, a)$ where a is the next uprocessed input or $s_{i+1} \in \delta(s_i, \varepsilon)$. M <u>rejects</u> w otherwise.

Draw an NFA for each of the following languages.

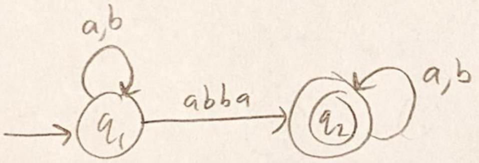
- $\{0w1 \mid w \in \Sigma^*\}$ $(\Sigma = \mathbb{Z}_2)$
- $\{x\,abba\,y \mid x,y \in \Sigma^*\}$ $(\Sigma = \{a,b\})$
- $\{xy \mid x,y \in \mathbb{Z}_5^+$ are increasing$\}$ $(\Sigma = \mathbb{Z}_5)$
- $\{xy \mid x,y \in \mathbb{Z}_5^+$ are decreasing$\}$

1)



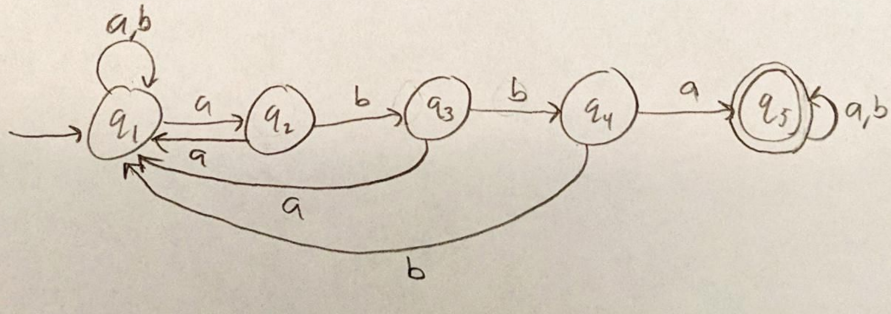
Note: It is not unusual for the absence of a transition to mean either immediate halt and reject or go to an implied state that consumes all input and then rejects.

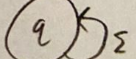
2)



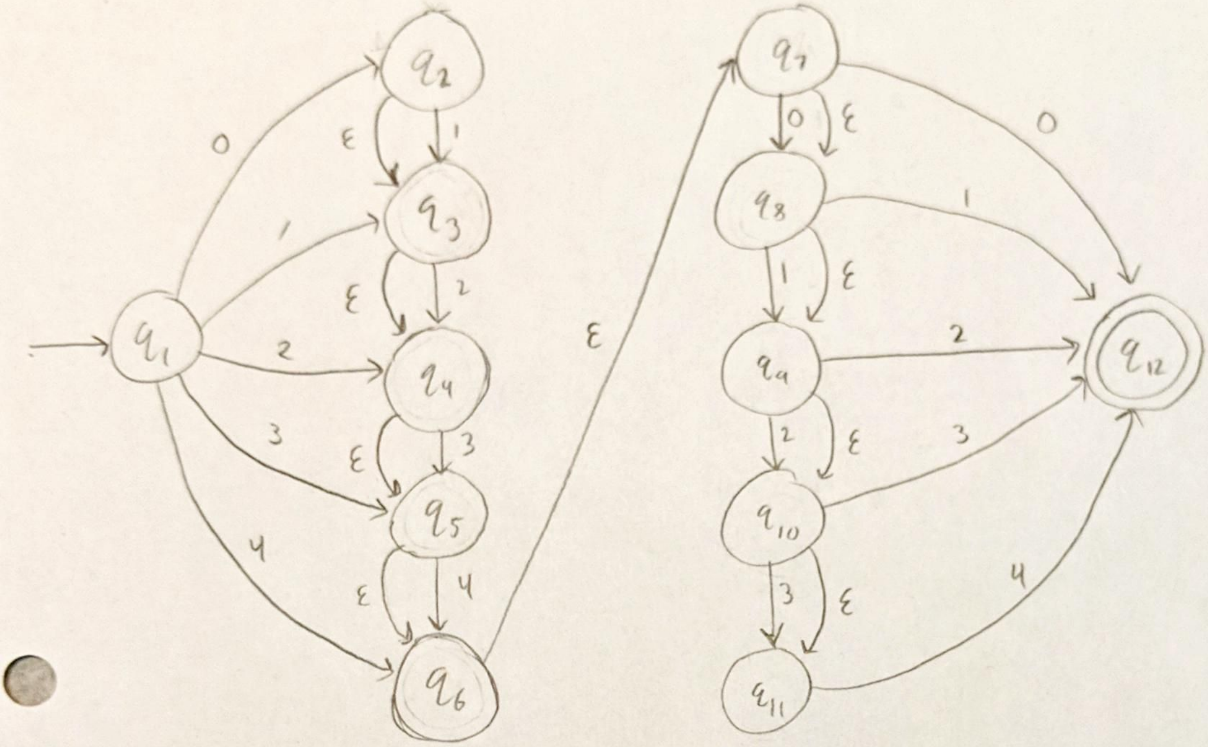
Note: It is common to label a long transition chain that goes back to the initial state on failure with the full string we're looking for.

or



Notice that in each example we can implement a $w \in \Sigma^*$ requirement simply with  and let the nondeterminism figure it out from there.

3)



4) Reverse the transitions of (3) and swap the start and accepting states.

(4) above highlights an interesting property. The regular languages are closed under reversal.