Homework 02

Aaron Wang

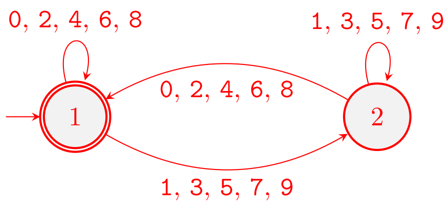
January 31 2025

1. **Divisibility tests.** Define, for all $k > 0$,

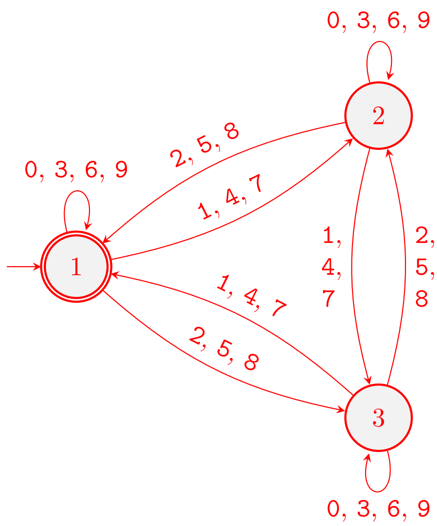
$$D_k = \Big\{w \in \{\texttt{0}, ..., \texttt{9}\}^* | w \text{ is the decimal representation of a multiple of } k\Big\}$$

where ε is considered to represent the number `0`. For example, the strings ε, `0`, `88`, and `088` all belong to D_2, but `99` and `099` do not.

(a) Prove that D_2 is regular by writing a DFA for D_2.



(b) Prove that D_3 is regular by writing a DFA for D_3.



(c) Prove that D_6 is regular. An explicit DFA is not necessary.

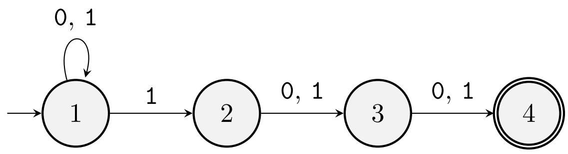
From the DFA of D_2 and D_3, we know that they are regular. Notice that for any number w, w is a multiple of 6 if and only if w is a multiple of 2 and 3. Thus, we can represent D_6 as:

$$D_6 = \Big\{w \in \{\texttt{0}, ..., \texttt{9}\}^* | w \text{ is the decimal representation of a multiple of 2 and 3}\Big\} = D_2 \cap D_3$$

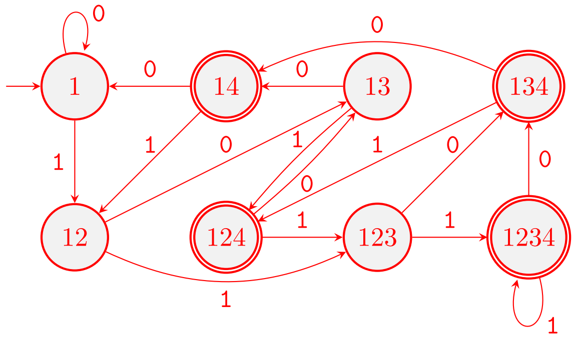
Thus D_6 is regular as it is the intersection of D_2 and D_3.[1]

[1] Theorem from class: If A and B are regular languages, then $A \cap B$ is also regular.

2. **Nondeterminism.** Consider the following NFA N_2 (same as in Figure 1.31), which accepts a string iff the third-to-last symbol is a 1:



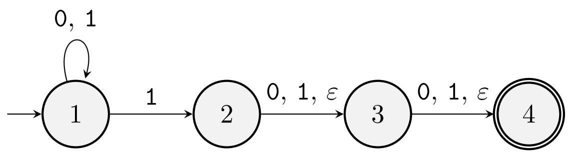
(a) Use the subset construction (Theorem 1.39) to convert N_2 to a DFA M. You may omit curly braces and commas when naming states; for example, instead of $\{1, 2, 3, 4\}$ you may write 1234. (Hint: the DFA should be equivalent to the one in Figure 1.32.)



(b) Why are the states in Figure 1.32 named q_{abc} where $a, b, c \in \{\texttt{0}, \texttt{1}\}$?

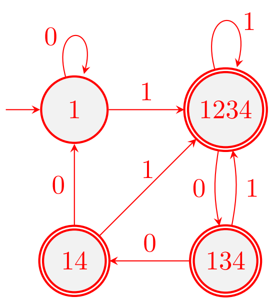
This is because abc represents the last 3 characters that have been read such that it is `1` if `1` was read and `0` otherwise. where a is the third-to-last symbol, b is the second-to-last symbol, and c is the last symbol.[2]

(c) In Example 1.30, Sipser asks what happens if you modify N_2 into the following NFA - let's call it N_2':



This now accepts the string if `1` is the last, second-to-last, or third-to-last character.

(d) Use the subset construction (Theorem 1.39) to convert N_2' to a DFA M'.



[2]If nothing has been read yet, aka no "symbols" yet, we assume symbols to be `0`

3. **Procrustean closure properties.** Let Σ be an alphabet, and let $L_3 = \{\texttt{theory}, \texttt{of}, \texttt{computing}\}$ be an example language.

   (a) For any $w = w_1 w_2 \cdots w_{n-1} w_n$, define

$$\text{STRETCH}(w_1 w_2 \cdots w_n) = w_1 w_1 w_2 w_2 \cdots w_{n-1} w_{n-1} w_n w_n.$$

This induces an operation on languages,

$$\text{STRETCH}(L) = \{\text{STRETCH}(w) | w \in L\}.$$

For example,

$$\text{STRETCH}(L_3) = \{\texttt{tthheeoorryy}, \texttt{ooff}, \texttt{ccoommppuuttiinngg}\}.$$

Prove that if L is a regular language, then STRETCH(L) is also regular.

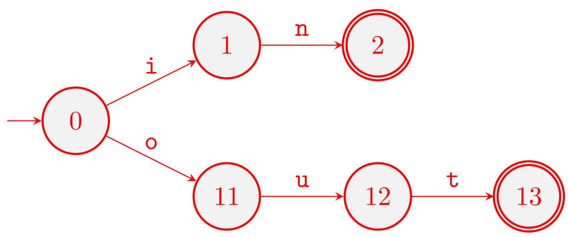
If L is regular, we have an NFA without ε transitions N for L s.t. $N = (Q, \Sigma, \delta, s, F)$. Construct $N' = (Q', \Sigma, \delta', s, F)$ to recognize STRETCH(L)

1. $Q' = Q \cup (\Sigma_\varepsilon \times Q)$. This way we have the regular states and a state for every transition.
2. Σ does not change.
3. Define δ' so if q is an original state ($q \in Q$), it will give transition state(s) and vice versa.

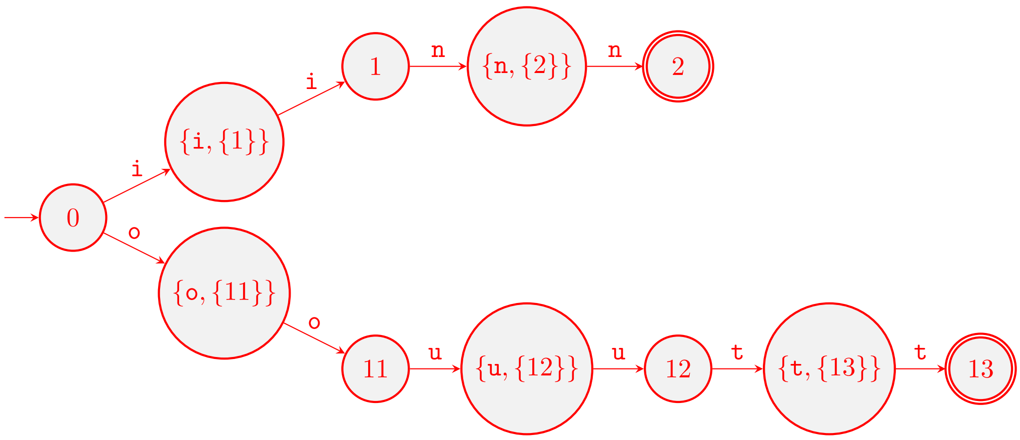
$$\delta'\Big(q, a\Big) = \begin{cases} (a, \delta(q, a)) & q \in Q \\ q_{\text{next(s)}} & \text{s.t. } \big(a, q_{\text{next(s)}}\big) = q \\ \emptyset & \text{otherwise} \end{cases}$$

4. s does not change.
5. F does not change.

Let $L = \{\texttt{in}, \texttt{out}\}$ as an example.[3]



STRETCH(L)



By proof of construction STRETCH(L) is regular.

[3]States without transitions were ommited from drawing

(b) For any $w = w_1 w_2 \cdots w_{n-1} w_n$ with $n \geq 2$, define

$$\text{CHOP}(w_1 w_2 \cdots w_{n-1} w_n) = w_2 \cdots w_{n-1}.$$

This induces an operation on languages,

$$\text{CHOP} = \{\text{CHOP}(w) | w \in L \text{ and } |w| \geq 2\}.$$

For example,

$$\text{CHOP}(L_3) = \{\texttt{heor}, \varepsilon, \texttt{omputin}\}.$$

Prove that if L is a regular language, then CHOP(L) is also regular.

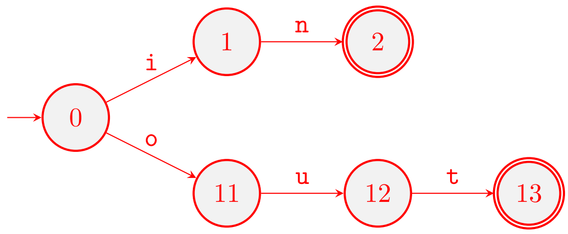
If L is regular, we have an NFA without ε transitions N for L s.t. $N = (Q, \Sigma, \delta, s, F)$.[4] Construct $N' = (Q', \Sigma, \delta', s', F')$ to recognize CHOP(L)

1. $Q' = Q \cup \{s'\}$ is Q with a new start state.
2. Σ does not change.
3. Define δ' as the same as before with the addition of ε transitions from s' to every state that s originally went to and any other transition a from s' maps to $\emptyset$

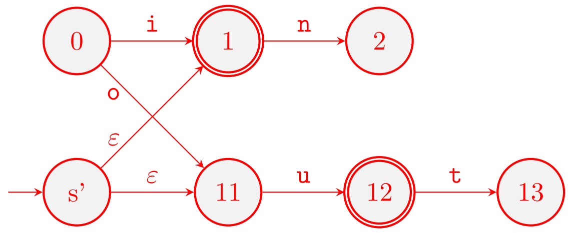
$$\delta'(q, a) = \begin{cases} \delta(q, a) & q \neq s' \\ \cup_{x \in \Sigma} \delta(s, x) & q = s' \wedge a = \varepsilon \\ \emptyset & q = s' \wedge a \neq \varepsilon \end{cases}$$

4. s' is the new start state.
5. $F' = \{q | \exists q_f \in F(\exists a \in \Sigma(q_f \in \delta(q, a)))\}$ is the set of all states that were originally one transition away from the end.

Let $L = \{\texttt{in}, \texttt{out}\}$ as an example.



CHOP(L)



By proof of construction CHOP(L) is regular.

[4]This is important for edge cases in which random ε transitions mess up construction of N'. We know such an N without ϵ always exists b/c an equivalent DFA exists for N and an equivalent DFA is an equivalent NFA without ε transitions