Tutorial 2

Gidon Rosalki

2025-04-02

Notice: If you find any mistakes, please open an issue at `https://github.com/robomarvin1501/notes_computability_compl`

1 Non-deterministic Finite Automaton (NFA)

Definition 1.1 (NFA). *An NFA A is defined as*

$$A = (Q, \Sigma, \delta, Q_0, F)$$

where

- Σ *is the alphabet*
- Q *is a set of the states*
- $F \subseteq Q$
- $Q_0 \subseteq Q$ *the starting states*
- $\delta : Q \times \Sigma \to 2^Q$

Definition 1.2 (Run). *A run of A on $w = w_1 \dots w_n$ is a sequence of states $r_0, \dots, r_n \in Q$, such that*

- $r_0 \in Q$
- $\forall 0 \leq i < n \; r_{i+1} \in \delta(r_i, w_{i+1})$

Definition 1.3 (Accepting run). *A run is called accepting if $r_n \in F$*

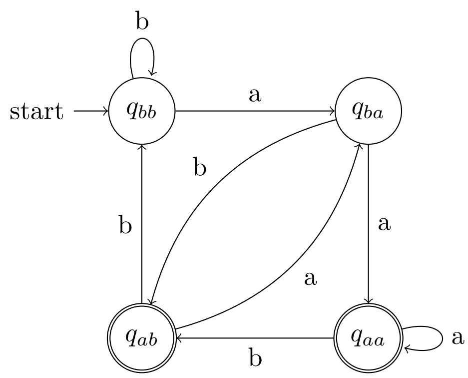
Definition 1.4. *A accepts w if there* ***exists*** *at least 1 accepting run.*

1.1 A small NFA can replace a DFA

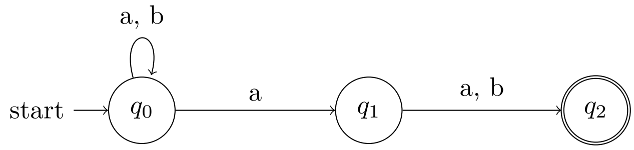
For every $k \in \mathbb{N}$ we will define L_k to be the language over $\Sigma = \{a, b\}$ that contains the words for whom the kth letter from the end is a.

1. Design a DFA that decides L_2
2. Design a NFA that decides L_2
3. For a general k, show that there exists an NFA with $k + 1$ states that decides L_k
4. For a general k, show that any DFA that decides L_k has at least 2^k states

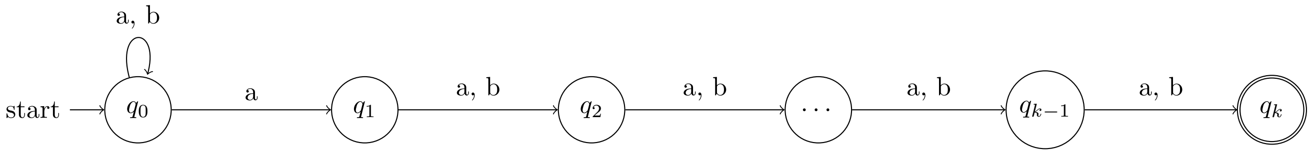
1. The following DFA decides L_2:



2. The following NFA decides L_2:



3. The following NFA with $k+1$ states decides L_k:



   We want to find for all $Q' \subseteq Q$ which words w can finish running in Q'. For the word w we will write $i_1, \ldots, i_l$ the locations of a from the end (only in the last k letters), and we will claim that these words whose a positions as described above can reach $\{q_0, q_{i_1}, \ldots, q_{i_l}\}$. For the inductive step we will decompose the word w as $w'\alpha$, where $\alpha \in \{a, b\}$. By the induction hypothesis, we determine the possible sets of states that u can reach. Then for each $\alpha \in \{a, b\}$, we verify the states that δ may lead w to are exactly those in the inductive claim.

4. We will assume the contradiction that A with less than 2^k states can decide $L(A) = L_k$. We will note that there are 2^k words of length k. Therefore, from the pigeonhole principle, there exists $u \neq v$ that finish their runs at the same state. For every word $w \in \Sigma^*$, then $u \cdot w \wedge v \cdot w$ finish in the same state, and therefore are either both accepted, or both rejected. Let us denote as i the first index where the two words differ, and without loss of generality define $u_i = a$, and $v_i = b$. If we create the words $u' = u \cdot a^{i-1}, v' = v \cdot a^{i-1}$, we know that the kth letter from the end of u' is a, and thus is accepted by A, but the kth letter from the end of v' is b, and is thus rejected, in contradiction, and thus A has at least 2^k states.

2 Closure properties of NREG

Definition 2.1 (NREG).

$$NREG = \{L \subseteq \Sigma^* : \exists NFA \text{ that decides } LL\}$$

2.1 Union

Theorem 1. *NREG is closed to union:*

$$L_1, L_2 \in NREG \implies L_1 \cup L_2 \in NREG$$

Proof . $L_1, L_2 \in NREG \implies$ that there be

$$A = (Q, \Sigma, \delta, Q_0, F)$$
$$B = (P, \Sigma, \eta, P_0, G)$$

Such that $L(A) = L_1, L(B) = L_2$. We will define

$$C = (Q \cup P, \Sigma, \alpha, Q_0 \cup P_0, F \cup G)$$
$$\alpha(q, \alpha) = \begin{cases} \delta(q, \alpha), & \text{if } q \in Q \\ \eta(q, \alpha), & \text{if } q \in P \end{cases}$$

Let there be $w = w_1 \ldots w_n$, and $r_0, \ldots, r_n$ be the run of C on w. We want to show that all of the run is contained within Q or within P. For $r_0 \in Q_0$, we may see by induction that

$$r_{i+1} \in \alpha(r_i, w_{i+1}) = \delta(r_i, w_{i+1}) \in Q$$

We may similarly show for $r_0 \in P_0$.

We will prove that $L(C) = L_1 \cup L_2$ through two sided containment:

1. Let there be $w \in L_1 \cup L_2$. Without loss of generality, $w \in L_1$. There exists an accepted run of A on w which we will call r. Therefore, r is an accepted run on C.

2. Let there be $w \in L(C)$. There exists a run of C on w $r = r_0 \ldots r_n$. Without loss of generality $r_0 \in Q_0$. Therefore, the run r is contained in Q, and thus $r \in L(A)$.

Therefore, $L(C) = L_1 \cup L_2$ □

2.2 Concatenation

Theorem 2.

$$L_1 \cdot L_2 = \{w_1 \cdot w_2 : w_1 \in L_1, w_2 \in L_2\}$$

Proof . Let there be

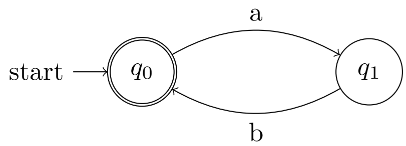
$$A = (Q, \Sigma, \delta, Q_0, F)$$
$$B = (P, \Sigma, \eta, P_0, G)$$

NFAs that recognise L_1 and L_2 respectively. We want to construct an NFA C such that $L(C) = L(A) \cdot L(B)$. We will need:

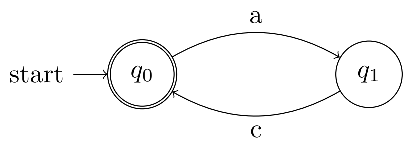
- States $= Q \cup P$
- Alphabet $= \Sigma$
- Transition function = All the transitions of A and all the transitions of B and a transition ε between all accepting states of A to all the starting states of B
- Start states $= Q_0$
- Finish states $= G$

□

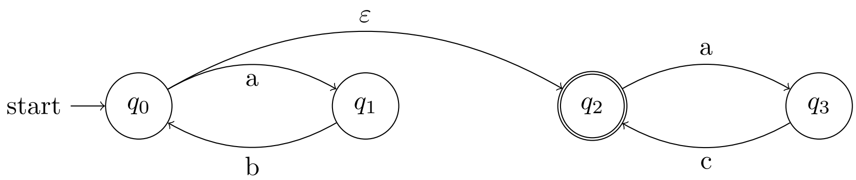
For example, let us consider the regular expressions $L_1 = (ab)^*$, $L_2 = (ac)^*$ over $\Sigma = \{a, b, c\}$. They respectively are the automota:



and

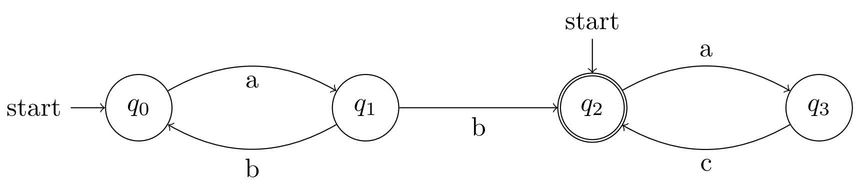


So to use the construction described above:



3 Formalising epsilon transitions

How can we perform the last NFA without ε transitions?



Let us formalise the concept. Given an NFA with ε transitions, we denote the non ε transitions by $\eta : Q \times \Sigma \to 2^Q$. Suppose that there is a transition from q to q' labelled a, and a transition from q' to q'' labelled ε. We will add in δ to be our target transition function. Therefore δ must contain a transition from q to q'' through the label a. This is almost sufficient, but what if there are more states reachable from q' through ε? We will want a transitions from q to those states as well. We will define the additional

$$E(q) = \{q' \in Q : q' \text{ is reachable from } q \text{ using only } \varepsilon \text{ transitions}\}$$

We can thus define δ as follows:

$$\delta(q,\alpha) = \bigcup_{q' \in \eta(q,\alpha)} E(q')$$

Now every transition from q to q' takes into account all ε transitions reachable from q'. We must now consider the initial states: Suppose that P_0 is the set of initial states on the graph before we take the ε transitions into account. We will define our set of initial states as follows:

$$Q_0 = \bigcup_{q \in P_0} E(q)$$

So every state reachable from an initial state via an ε transition is considered an initial state.

4 Delta star for NFAs

Definition 4.1 (δ^*).

$$\delta^* : 2^Q \times \Sigma^* \to 2^Q$$

$$\delta^*(S,w) = \{q \in Q : \text{ there is a run such that } r_0 \in S, r_n = q \wedge \forall 0 \le i < n \ r_{i+1} \in \delta(r_i, w_{i+1})\}$$

or recursively

$$\delta^*(S,w) = \begin{cases} S, & \text{if } w = \varepsilon \\ \bigcup_{q \in \delta^*(S,w')} \delta(q,\alpha), & \text{if } w = w'\alpha \end{cases}$$

Theorem 3. *Both above representations are equivalent*

Proof . We will show that the first case satisfies both the base case, and the recursive case. Since they agree on both, then they have the same definition. In this proof, δ^* will refer to the first definition, and $S \subseteq Q$.
Base case: Let $w = \varepsilon$. A run on ε starts at some state $r_0 \in S$, and immediately stops. Therefore, the reachable states are the states in S, and we have shown that $\delta^*(S,\varepsilon) = S$
Recursive case: Let $w = w'\alpha$. We want to prove that

$$\delta^*(S,w'\alpha) = \bigcup_{q \in \delta^*(S,w')} \delta(q,\alpha)$$

We shall do this through two sided containment:

1. Let $r_n \in \delta^*(S,w'\alpha)$. There is a run $r_0,\ldots,r_n$ on $w'\alpha$, that starts in S, where $r_n \in \delta(r_{n-1},\alpha)$. Since $r_0,\ldots,r_{n-1}$ is a run on w' starting in S, so $r_{n-1} \in \delta^*(S,w')$, therefore

$$r_n \in \delta(r_{n-1},\alpha) \subseteq \bigcup_{q \in \delta^*(S,w')} \delta(q,\alpha)$$

2. Let $r_n \in \bigcup_{q \in \delta^*(S,w')} \delta(q,\alpha)$. Then there is some $r_{n-1} \in \delta^*(S,w')$ for which $r_n \in \delta(r_{n-1},\alpha)$. There is a run $r_0,\ldots,r_{n-1}$ from S on w', and since $r_n \in \delta(r_{n-1},\alpha)$, we have shown that $r_0,\ldots,r_n$ is a run from S to r_n. Therefore, $r_n \in \delta^*(S,w'\cdot\alpha)$

□