COMPUTATION: DAY 3

BURTON ROSENBERG
UNIVERSITY OF MIAMI

Friday, January 30, 2026.

1. Power of Nondeterminism

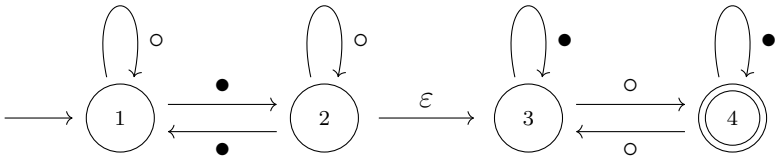
1.1 Nondeterminism in NFAs can be thought of as consulting an oracle (intuitive), exploring all paths via search (computational), or as an existential condition over sequences of states (formal). These perspectives are complementary: the first builds intuition, the second guides construction, and the third gives precise semantics for proofs.

1.2 In the oracle perspective, when an ε-move exists, or there is more than one outgoing edges, the automaton consults and oracle for which transition to make. If the string is in the language, the oracle always responds with a transition along a path to accepting the string, In the case that the string is not in the language, the oracle answers randomly. At this point we don't worry how can such an oracle exist. We leave that as a technical detail.

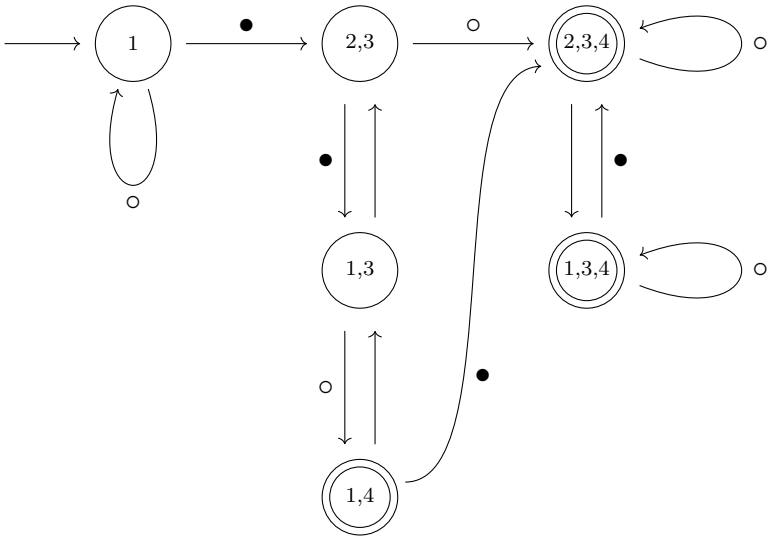
1.3 In the formal perspective, we have the mathematical definition as presented in Day 2. It says "if there exists and computation path". This is completely logical and consistent. For a deterministic machine, that path is calculated effortlessly by following the transitions demanded by each symbol. For a non-deterministic machine, we don't have a calculation of the path, at least, not just yet.

1.4 In the computational perspective, we want to be able to produce a path to accepting if one exists and show there is none else. In the case of finite automata we can do this by searching all the computation paths exhaustively. One version of this exhaustive search is the *power set construction* that, for an NFA N with state set Q, builds an DFA M with state set $\mathcal{P}(Q)$. In that very large space we can deterministically follow all computation paths simultaneously.

Theorem 1.1. Given a language S and an DFA M such that $\mathcal{L}(M) = S$, there is an NFA N such that $\mathcal{L}(N) = S$.



(A) NFA for $S \circ T$



(B) DFA for $S \circ T$

FIGURE 1. Power Set Construction

Proof: Given the DFA $M = \langle\, Q, \Sigma, \delta, q_0, F \,\rangle$ the required NFA is,

$$N = \langle\, Q, \Sigma, \delta', q_0, F \,\rangle$$

with,

$$\delta'(q, \sigma) = \{\, \delta(q, \sigma) \,\},\ \delta(q, \dot{\varepsilon}) = \emptyset.$$

□

Theorem 1.2. Given a language S and an NFA N such that $\mathcal{L}(N) = S$, there is an DFA M such that $\mathcal{L}(M) = S$.

Proof by example: We show an example of the power set construction in figure 1. The NFA in figure 1a is converted into the DFA in figure 1b. The two automata accept exactly the same language. The method of conversion works for any NFA.

1.5 Given NFA $N = \langle\, Q, \Sigma_\varepsilon, \delta, q_0, F \,\rangle$ we construct a DFA M,

$$M = \langle\, \mathcal{P}(Q), \Sigma, \bar{\delta}, \bar{q}_0, \bar{F} \,\rangle$$

with states the power set of Q.

1.6 The meaning of a state $\bar{q} \in \mathcal{P}(Q)$ is that it represents the set of states $q \in Q$ that the NFA N *could be in* after processing a string $\sigma \in \Sigma^*$. We construct the DFA M by starting with an initial state

$$\bar{q}_0 = E(\{q_0\}),$$

the ε-closure of the NFA's start state q_0 (see Definition 1.1), and then iteratively exploring all possible transitions: for each discovered DFA state $\bar{q}$ and each input symbol $a \in \Sigma$, we define the DFA transition

$$\bar{\delta}(\bar{q}, a) = E\Big(\bigcup_{q \in \bar{q}} \delta(q, a) \Big),$$

adding $\bar{\delta}(\bar{q}, a)$ as a new DFA state if it has not been seen before. Repeating this process for all newly discovered states ensures that M contains exactly all reachable subsets of Q, and the set of accepting states of M are those states $\bar{q}$ among those discovered for which

$$\bar{q} \cap F \neq \emptyset,$$

so that M accepts exactly those strings that the original NFA N accepts. □

Definition 1.1 (ε-closure)**.** Given an NFA $N = \langle\, Q, \Sigma, \delta, q_o, F \,\}$ the *ε-closure of S*,

$$E(S) : \mathcal{P}(Q) \to \mathcal{P}(Q)$$

is the smallest subset of Q such that,

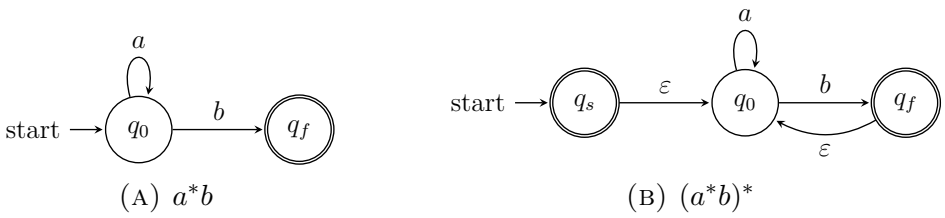
(1) $S \subseteq E(S)$,

(2) and $\forall\, q \in E(S),\ \delta(q, \varepsilon) \subseteq E(S)$.

It is called a closure operator because $E(E(S)) = E(S)$.

Theorem 1.3. The ε-closure operator commutes with set union, $E(A \cup B) = E(A) \cup E(B)$.

Corollary 1.1. Nondeterministic finite automata (NFAs) accept exactly the regular languages.



(A) a^*b (B) $(a^*b)^*$

FIGURE 2. NFA for a^*b and its Kleene star construction

1.7 Intuitively, an NFA can be thought of as a computational focus, represented by a token that journeys from state to state with the help of a beneficent oracle. The oracle always points out a valid path to acceptance if one exists, but it cannot violate the transition rules of the automaton.

1.8 The subset construction formalizes this intuition. For any NFA there exists an equivalent DFA whose states correspond to sets of NFA states. Conceptually, the token is cloned whenever the NFA has a choice of transitions, and tokens disappear when they reach dead ends. The DFA tracks, in a single deterministic computation, the entire collection of tokens simultaneously.

1.9 A string is accepted exactly when at least one token reaches an accepting state — precisely the condition under which the NFA would accept. Thus, while NFAs allow more flexible and concise descriptions of computation, they do not increase the class of languages recognized.

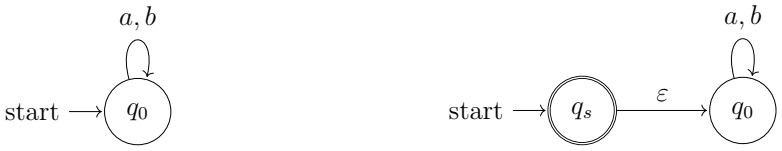
2. Kleene Algebras

2.1 We have now seen several key closure constructions for regular languages. Deterministic finite automata (DFAs) are closed under complement, and — via the product construction — under union and intersection as well. Nondeterministic finite automata (NFAs) were introduced to handle concatenation: by adding ε-transitions between component automata, we can construct an NFA that accepts exactly the concatenation of two regular languages. As any NFA can be converted to a DFA, this establishes that the concatenation of two regular languages is a regular language.

2.2 Taken together, these results show that the class of regular languages is closed under the basic operations of choice (union) and composition (concatenation). What remains is to account for iteration, captured by the Kleene star, which we address next.

2.3 The Kleene star presents iteration of a language A as the most precise solution X to the equation

$$X = \{\varepsilon\} \cup A \circ X.$$



(A) The language $\emptyset$ (B) The language $\emptyset^*$

FIGURE 3. Kleene start construction for $\emptyset^* = \{\varepsilon\}$

A computation in X either exits immediately by choosing $\{\varepsilon\}$, or continues by applying A and returning again to X. The most precise solution to this equation is $X = A^*$, satisfying

$$A^* = \{\varepsilon\} \cup A \circ A^*.$$

Theorem 2.1. If $A \subseteq \Sigma^*$ is regular, so is A^*.

Theorem 2.2 (Kleene–Kozen[1])**.** The class of regular languages forms a Kleene Algebra.

2.4 The general method for realizing this equation with an NFA is shown in Figure 2. An example NFA is shown in Figure 2a for the language

$$A = a^*b,$$

that is, a sequence of zero or more a's followed by a final b. The NFA for A^* is shown in Figure 2b. The construction proceeds as follows:

(1) Create a new initial state (here labeled q_s), which is also accepting.

(2) Add an ε-transition from the new initial state to the original initial state (here, from q_s to q_0).

(3) Add ε-transitions from each accepting state of the original NFA back to the original initial state (here, from q_f to q_0).

2.5 *A caution:* It might be thought unnecessary to create a new initial state, and instead to mark the original initial state as final. This does not work in general, and does not work in our example. In the example above, doing so yields the language $\{a, b\}^*$, by introducing too many new accepting paths.

2.6 A second example showing the need for a new start state is in the star of the empty language, see figure 3.

[1]Dexter Kozen, *A completeness theorem for Kleene algebras and the algebra of regular events,* Information and Computation, 110(2):366–390, 1994.