1 NFA vs. DFA

Expressive Power of NFAs and DFAs

- Is there a language that is recognized by a DFA but not by any NFAs? No!
- Is there a language that is recognized by an NFA but not by any DFAs? No!

Main Theorem

Theorem 1. *A language L is recognized by a DFA if and only if there is an NFA N such that $L(N) = L$.*

In other words:

- *For any DFA D, there is an NFA N such that $L(N) = L(D)$, and*
- *For any NFA N, there is a DFA D such that $L(D) = L(N)$.*

---

2 NFAs for Regular Languages

Converting DFAs to NFAs

Proposition 2. *For any DFA D, there is an NFA N such that $L(N) = L(D)$.*

Proof. Is a DFA an NFA? Essentially yes! Syntactically, not quite. The formal definition of DFA has $\delta_{\text{DFA}} : Q \times \Sigma \to Q$ whereas $\delta_{\text{NFA}} : Q \times (\Sigma \cup \{\epsilon\}) \to \mathcal{P}(Q)$.

For DFA $D = (Q, \Sigma, \delta_D, q_0, F)$, define an "equivalent" NFA $N = (Q, \Sigma, \delta_N, q_0, F)$ that has the exact same set of states, initial state and final states. Only difference is in the transition function.

$$\delta_N(q, a) = \{\delta_D(q, a)\}$$

for $a \in \Sigma$ and $\delta_N(q, \epsilon) = \emptyset$ for all $q \in Q$. □

---

3 NFAs recognize Regular Languages

3.1 Simulating an NFA

Simulating an NFA on Your Computer

NFA Acceptance Problem
Given an NFA N and an input string w, does N accept w?

How do we write a computer program to solve the NFA Acceptance problem?

Two Views of Nondeterminsm

Guessing View
At each step, the NFA "guesses" one of the choices available; the NFA will guess an "accepting sequence of choices" if such a one exists.

Very useful in reasoning about NFAs and in designing NFAs.

Parallel View
At each step the machine "forks" threads corresponding to each of the possible next states.

Very useful in simulating/running NFA on inputs.

Algorithm for Simulating an NFA

Algorithm
Keep track of the current state of each of the active threads.

Example 3.

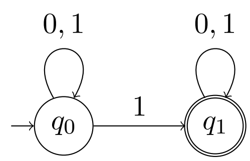


Figure 1: Example NFA N

Consider the input $w = 111$. The execution (listing only the states of currently active threads) is

$$\langle q_0 \rangle \xrightarrow{1} \langle q_0, q_1 \rangle \xrightarrow{1} \langle q_0, q_1, q_1 \rangle \xrightarrow{1} \langle q_0, q_1, q_1, q_1 \rangle$$

Algorithm
With optimizations

Observations

- Exponentially growing memory: more threads for longer inputs. Can we do better?
- Exact order of threads is not important
  - It is unimportant whether the 5^{th} thread or the 1^{st} thread is in state q.
- If two threads are in the same state, then we can ignore one of the threads
  - Threads in the same state will "behave" identically; either one of the descendent threads of both will reach a final state, or none of the descendent threads of both will reach a final state

Parsimonious Algorithm in Action

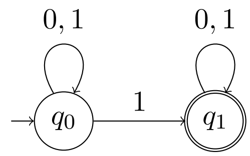
Example 4. 

Figure 2: Example NFA N

Consider the input $w = 111$. The execution (listing only the states of currently active threads) is

$$\{q_0\} \xrightarrow{1} \{q_0, q_1\} \xrightarrow{1} \{q_0, q_1\}$$
$$\xrightarrow{1} \{q_0, q_1\}$$

3.2 DFAs equivalent to NFAs

Revisiting NFA Simulation Algorithm

- Need to keep track of the states of the active threads
  - *Unordered:* Without worrying about exactly which thread is in what state
  - *No Duplicates:* Keeping only one copy if there are multiple threads in same state
- How much memory is needed?
  - If Q is the set of states of the NFA N, then we need to keep a subset of Q!
  - Can be done in $|Q|$ bits of memory (i.e., $2^{|Q|}$ states), which is finite!!

Constructing an Equivalent DFA

- The DFA runs the simulation algorithm
- DFA remembers the current states of active threads without duplicates, i.e., maintains a subset of states of the NFA
- When a new symbol is read, it updates the states of the active threads
- Accepts whenever one of the threads is in a final state

Example of Equivalent DFA

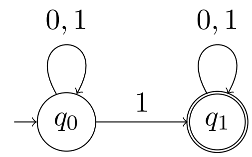


Figure 3: Example NFA N

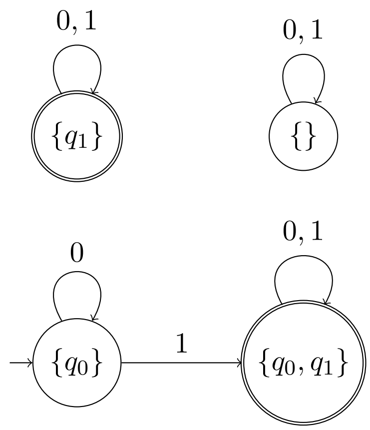


Figure 4: DFA D equivalent to N

Recall . . .

Definition 5. For an NFA $M = (Q, \Sigma, \delta, q_0, F)$, string w, and state $q_1 \in Q$, we say $\hat{\delta}_M(q_1, w)$ to denote states of all the active threads of computation on input w from q_1. Formally,

$$\hat{\delta}_M(q_1, w) = \{q \in Q \mid q_1 \xrightarrow{w}_M q\}$$

Formal Construction

Given NFA $N = (Q, \Sigma, \delta, q_0, F)$, construct DFA $\det(N) = (Q', \Sigma, \delta', q_0', F')$ as follows.

- $Q' = \mathcal{P}(Q)$
- $q_0' = \hat{\delta}_N(q_0, \epsilon)$
- $F' = \{A \subseteq Q \mid A \cap F \neq \emptyset\}$
- $\delta'(\{q_1, q_2, \ldots q_k\}, a) = \hat{\delta}_N(q_1, a) \cup \hat{\delta}_N(q_2, a) \cup \cdots \cup \hat{\delta}_N(q_k, a)$ or more concisely,

$$\delta'(A, a) = \bigcup_{q \in A} \hat{\delta}_N(q, a)$$

Correctness

Lemma 6. *For any NFA N, the DFA* $\det(N)$ *is equivalent to it, i.e.,* $L(N) = L(\det(N))$.

Proof Idea
Need to show

$$\forall w \in \Sigma^*.\ \det(N) \text{ accepts } w \text{ iff } N \text{ accepts } w$$
$$\forall w \in \Sigma^*. \hat{\delta}_{\det(N)}(q_0', w) \cap F' \neq \emptyset \text{ iff } \hat{\delta}_N(q_0, w) \cap F \neq \emptyset$$
$$\forall w \in \Sigma^*. \text{ for } \{A\} = \hat{\delta}_{\det(N)}(q_0', w),\ A \cap F \neq \emptyset \text{ iff } \hat{\delta}_N(q_0, w) \cap F \neq \emptyset$$

We will instead prove a stronger claim. There are two possible ways to strengthen this:

(a) $\forall w \in \Sigma^*.\ \hat{\delta}_{\det(N)}(q_0', w) = \{A\}$ iff $\hat{\delta}_N(q_0, w) = A$. In other words, this says that the state of the DFA after reading some string is exactly the set of states the NFA could be in after reading the same string.

(b) $\forall w \in \Sigma^*.$, for all ϵ-closed sets $A \subseteq Q$ (ϵ-closed to be defined later), $\hat{\delta}_{\det(N)}(A, w) \cap F' \neq \emptyset$ iff for some $q \in A$, $\hat{\delta}_N(q, w) \cap F \neq \emptyset$. In other words, (ignoring the technical condition of ϵ-closed sets) this says that the DFA $\det(N)$ accepts w from a state A iff the NFA N accepts w from some state in A.

Notice that both statements (a) and (b) (modulo the notion of ϵ-closure), if proved, would prove that the construction is correct. Both the strengthenings (a) and (b) can be proved by induction, and their proofs have subtle differences. We, therefore, present both proofs completely.

Correctness Proof I

Lemma 7. $\forall w \in \Sigma^*.\ \hat{\delta}_{\det(N)}(q_0', w) = \{A\}$ *iff* $\hat{\delta}_N(q_0, w) = A$.

Proof. By induction on $|w|$

- *Base Case $|w| = 0$:* Then $w = \epsilon$. Now

$$\hat{\delta}_{\det(N)}(q_0', \epsilon) = \{q_0'\} = \{\hat{\delta}_N(q_0, \epsilon)\} \text{ defn. of } \hat{\delta}_{\det(N)} \text{ and defn. of } q_0'$$

- *Induction Hypothesis:* Assume inductively that the statement holds $\forall w.\ |w| < n$

- *Induction Step:* If $|w| = n$ then $w = ua$ with $|u| = n - 1$ and $a \in \Sigma$.

$$\begin{aligned}
\hat{\delta}_{\det(N)}(q_0', ua) = \{A\} \quad & \text{iff} \quad q_0' \xrightarrow{ua}_{\det(N)} A \\
& \qquad \text{defn. of } \hat{\delta} \text{ and } \det(N) \text{ is deterministic} \\
& \text{iff} \quad \exists B.\ q_0' \xrightarrow{u}_{\det(N)} B \text{ and } B \xrightarrow{a}_{\det(N)} A \\
& \qquad \text{property of } \longrightarrow \text{ proved in lecture 3} \\
& \text{iff} \quad \exists B.\ \hat{\delta}_{\det(N)}(q_0', u) = \{B\} \text{ and } B \xrightarrow{a}_{\det(N)} A \\
& \qquad \det(N) \text{ is deterministic and defn. of } \hat{\delta} \\
& \text{iff} \quad \exists B.\ \hat{\delta}_N(q_0, u) = B \text{ and } A = \cup_{q \in B} \hat{\delta}_N(q, a) \\
& \qquad \text{ind. hyp. and defn. of transition in } \det(N) \\
& \text{iff} \quad \hat{\delta}_N(q_0, ua) = A \\
& \qquad \text{see Lemma below}
\end{aligned}$$

□

To complete the proof, we need to prove the following lemma

Lemma 8. $\exists B.\ \hat{\delta}_N(q_0, u) = B$ *and* $A = \cup_{q \in B} \hat{\delta}_N(q, a)$ *iff* $\hat{\delta}_N(q_0, ua) = A$.

Proof. Observe that $A = \cup_{q \in B} \hat{\delta}_N(q, a)$ iff ($q \in A$ iff $\exists q' \in B$ s.t. $q' \xrightarrow{a}_N q$). Thus we have,

$$\begin{gathered} \hat{\delta}_N(q_0, u) = B \text{ and } A = \cup_{q \in B} \hat{\delta}_N(q, a) \\ \text{iff} \\ (q \in A \text{ iff } \exists q'.\ q_0 \xrightarrow{u}_N q' \text{ and } q' \xrightarrow{a}_N q) \end{gathered}$$

Now since we know that $q_1 \xrightarrow{uv}_N q_2$ iff there is q' s.t. $q_1 \xrightarrow{u}_N q'$ and $q' \xrightarrow{v}_N q_2$, we can conclude

$$\begin{gathered} \hat{\delta}_N(q_0, u) = B \text{ and } A = \cup_{q \in B} \hat{\delta}_N(q, a) \\ \text{iff} \\ (q \in A \text{ iff } \exists q'.\ q_0 \xrightarrow{u}_N q' \text{ and } q' \xrightarrow{a}_N q) \\ \text{iff} \\ (q \in A \text{ iff } q_0 \xrightarrow{ua}_N q) \\ \text{iff} \\ \hat{\delta}_N(q_0, ua) = A \end{gathered}$$

where the last step is a consequence of the definition of $\hat{\delta}_N$. □

Correctness Proof II

Definition 9. Let Q be the states of NFA N and let $A \subseteq Q$. A is said to be *ϵ-closed* iff for every $q \in A$, $\hat{\delta}_N(q, \epsilon) \subseteq A$. In other words, any state that can be reached from a state in A by following only ϵ-transitions, again belongs to A.

Example 10. Trivial examples of ϵ-closed sets include $\emptyset$ and Q. An important example (that will be critical to this proof of correctness) is the initial state of DFA $\det(N)$ $q_0' = \hat{\delta}_N(q_0, \epsilon)$.

Lemma 11. *For every string $w \in \Sigma^*$, for every $A \subseteq Q$ that is ϵ-closed, $\hat{\delta}_{\det(N)}(A, w) \cap F' \neq \emptyset$ iff for some $q \in A$, $\hat{\delta}_N(q, w) \cap F \neq \emptyset$.*

Discussion
Before proving the lemma, let us highlight one point about the lemma. Ideally, we would have liked to prove the above lemma without the condition on ϵ-closure, i.e., we would have liked to prove that for every string w, for every $A \subseteq Q$, $\hat{\delta}_{\det(N)}(A, w) \cap F' \neq \emptyset$ iff for some $q \in A$, $\hat{\delta}_N(q, w) \cap F \neq \emptyset$. Unfortunately, this stronger condition does not hold, as you will see in the proof of the base case below. However, inspite of this, when A is taken to be q_0', which we pointed out is an ϵ-closed set, the above lemma specializes to the statement establishing the correctness of the DFA construction which is good enough for our purposes.

Proof. We will prove this by induction on $|w|$.

- *Base Case* $|w| = 0$*:* Then $w = \epsilon$. Now for any set A, $\hat{\delta}_{\det(N)}(A, \epsilon) = \{A\}$. Thus, $\hat{\delta}_{\det(N)}(A, \epsilon) \cap F' \neq \emptyset$ iff $A \cap F \neq \emptyset$. If A is ϵ-closed then $\hat{\delta}_N(A, \epsilon) = A$ and so we have

  $$\hat{\delta}_{\det(N)}(A, \epsilon) \cap F' \neq \emptyset \text{ iff } A(= \hat{\delta}_N(A, \epsilon)) \cap F \neq \emptyset$$

  Thus, we have established the base case.

  Notice that the crucial step where ϵ-closedness is used — it is in arguing that $\hat{\delta}_N(A, \epsilon) = A$. Without this the base case cannot be proved, and in fact does not hold.

- *Induction Hypothesis:* Assume inductively that the statement holds $\forall w.\ |w| < n$

- *Induction Step:* If $|w| = n$ then $w = au$ with $|u| = n - 1$ and $a \in \Sigma$ [1].

  Now $\hat{\delta}_{\det(N)}(A, w = au) = \{C\}$ iff there is a B s.t. $\delta'(A, a) = B$ and $\hat{\delta}_{\det(N)}(B, u) = \{C\}$. The first important observation we make is that if A is ϵ-closed then so is B; we will prove this later after completing this proof. Thus, we can use the induction hypothesis on the computation on string u from state B.

$$\hat{\delta}_{\det(N)}(A, w = au) = \{C\} \text{ and } C \in F'$$
iff
$$\exists B.\ \delta'(A, a) = B,\ \hat{\delta}_{\det(N)}(B, u) = \{C\} \text{ and } C \in F' \qquad \text{defn. of } \hat{\delta}_{\det(N)}$$
iff
$$\exists B.\ \delta'(A, a) = B,\ \exists q \in B.\ \hat{\delta}_N(q, u) \cap F \neq \emptyset \qquad B \text{ is } \epsilon\text{-closed and ind. hyp. on } u$$
iff
$$\exists B.\ \delta'(A, a) = B,\ \exists q \in B.\ \exists q_2 \in F.\ q \xrightarrow{u}_N q_2 \qquad \text{defn. of } \hat{\delta}_N$$
iff
$$\exists q_1 \in A. \exists q. \exists q_2 \in F.\ q_1 \xrightarrow{a}_N q \text{ and } q \xrightarrow{u}_N q_2 \qquad \text{defn. of } \delta' \text{ transition of } \det(N)$$
iff
$$\exists q_1 \in A. \exists q_2 \in F.\ q_1 \xrightarrow{w=au}_N q_2$$
iff
$$\exists q_1 \in A.\ \hat{\delta}_N(q, w) \cap F \neq \emptyset$$

To complete this proof we need to show

Lemma 12. *If A is ϵ-closed and $\delta'(A, a) = B$ then B is ϵ-closed.*

Proof. Let $q \in B$ be an arbitrary element of B. Need to show that if $q \xrightarrow{\epsilon}_N q'$ then $q' \in B$. Observe that from the definition of δ', we have $q \in B$ implies there is $q_1 \in A$ such that $q_1 \xrightarrow{a}_N q$. Now if $q \xrightarrow{\epsilon}_N q'$ then we have $q_1 \xrightarrow{a=a\epsilon}_N q'$. Again by the definition of δ' this means that $q' \in B$, which completes the proof. □

□

Another Example

[1] Notice the difference in the form of w when compared to proof I. In proof I we took $w = ua$ in the induction step.

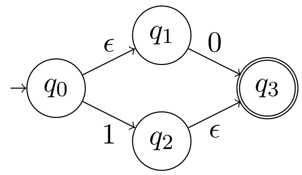


Figure 5: Example NFA N_ϵ

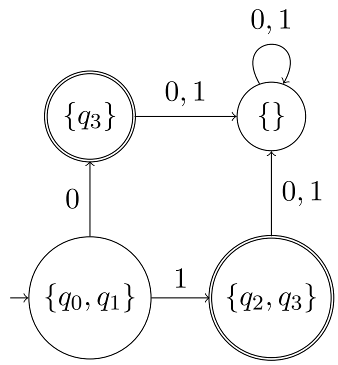


Figure 6: DFA D'_ϵ for N_ϵ (only relevant states)
