COMP 330 Fall 2023
Supplementary Note
Lecture 5

Cesare Spinoso-Di Piano

Last updated: September 24, 2023

This supplementary note formally[1] proves some of the facts/theorems from Lecture 5.

1 DFA and NFA

We begin this note by recalling the formal definitions of DFA and NFA.

Definition 1.1 (Deterministic Finite Automaton)**.** A **deterministic finite automaton (DFA)** M is a 5-tuple $M = (Q, \Sigma, \delta, s_0, F)$ where

- Q is the finite set of states
- Σ is the input alphabet
- δ is the transition function $\delta : Q \times \Sigma \to Q$
- $s_0 \in Q$ is the (unique) start state[2]
- $F \subseteq Q$ is the set of accept (final)[3] states

Recall that the "workhorse" of the DFA is its transition function and, in particular, its extended transition function δ^* which I defined recursively (see Lecture 3 notes).

Definition 1.2 (Non-deterministic Finite Automaton)**.** A **non-deterministic finite automaton (NFA)** N is a 5-tuple $N = (Q, \Sigma, \Delta, S_0, F)$ where

- Q is the finite set of states
- Σ is the input alphabet
- Δ is the transition function $\Delta : Q \times \Sigma \to 2^Q$

[1] A few students have been complaining about the "lack of rigour" in my proofs. Just because a proof is simple, doesn't mean it's not rigorous!

[2] Some books use q_0 as the start state. I do too sometimes. Regardless, I will be consistent in using the 4^{th} element of the tuple as the start state.

[3] Some books call F the set of accept states. Others say the set of final states. I prefer the term "accept" since it tells you exactly what F is for. Out of habit, I sometimes say the set of final states.

- $S_0 \subseteq Q$ is the set of start states
- $F \subseteq Q$ is the set of accept (final) states

Much like the DFA, the NFA's "workhorse" is its transition function and, in particular, its extended transition function Δ^*. Recall from Lecture 5 that Δ^* takes as input parameters a *set of states* and a string. This first input type is necessary to accommodate for the fact that an NFA has a *set of* start states.

After introducing Δ^*, I presented the following two facts without proofs.

Fact. *Given an NFA* $N = (Q, \Sigma, \Delta, S_0, F)$*,* $A \subseteq Q, B \subseteq Q$*,* $x, y \in \Sigma^*$*, we have that*

1. $\Delta^*(A, xy) = \Delta^*(\Delta^*(A, x), y)$
2. $\Delta^*(A \cup B, x) = \Delta^*(A, x) \cup \Delta^*(B, x)$

Proof. We prove Fact 1 here by induction on the length of y.
Base case: Suppose $|y| = 0$ then $y = \varepsilon$. Then,

$$\begin{aligned}\Delta^*(A, x \cdot y) &= \Delta^*(A, x \cdot \varepsilon) \\ &= \Delta^*(A, x)\end{aligned}$$

and

$$\begin{aligned}\Delta^*(\Delta^*(A, x), y) &= \Delta^*(\Delta^*(A, x), \varepsilon) \\ &= \Delta^*(A, x)\end{aligned}$$

This proves the base case.
Inductive hypothesis: We assume the statement is true for every $y \in \Sigma^*, |y| = n$ for some $n \in \mathbb{N}$.
Inductive step: We must show the statement for $y \in \Sigma^*, |y| = n + 1$. We rewrite y as $w\sigma$ where $w \in \Sigma^*, \sigma \in \Sigma$. Then

$$\begin{aligned}\Delta^*(A, xy) &= \Delta^*(A, x \cdot (w\sigma)) \\ &= \Delta^*(A, (x \cdot w)\sigma) \\ &= \bigcup_{q \in \Delta^*(A, xw)} \Delta(q, \sigma) \\ &= \bigcup_{q \in \Delta^*(\Delta^*(A,x), w)} \Delta(q, \sigma) && \text{by IH} \\ &= \Delta^*(\Delta^*(A, x), w\sigma) && \text{by the definition of } \Delta^* \text{ - let } B = \Delta^*(A, x) \text{ to convince yourself} \\ &= \Delta^*(\Delta^*(A, x), y)\end{aligned}$$

This completes the proof. ■

2 Equivalence of DFA and NFA

We are now ready to prove the first theorem which I stated during Lecture 5.

Theorem 1. *Given some alphabet Σ, the family of languages accepted by DFA, $L_{DFA} = \{L(M) : M \text{ is a DFA}\}$, is exactly the same as the family of languages accepted by NFA, $L_{NFA} = \{L(N) : N \text{ is an NFA}\}$.*

Proof. We prove this set equality by double inclusion.
$\boldsymbol{L_{\mathrm{DFA}} \subseteq L_{\mathrm{NFA}}}$. This statement says that any language accepted by some DFA $M = (Q, \Sigma, \delta, s_0, F)$, $L(M)$, belongs to L_{NFA}. To show this is true, we must show that there is some NFA N such that $L(M) = L(N)$. We construct $N = (Q', \Sigma, \Delta, S_0, F')$ *explicitly* as follows

$Q' \coloneqq Q$

$S_0 \coloneqq \{s_0\}$

$F' \coloneqq F$

For $q \in Q, \sigma \in \Sigma$, $\Delta(q, \sigma) \coloneqq \{\delta(q, \sigma)\}$

That is, N looks *exactly like* M except that we've changed the types of some of the elements of the tuple such that they respect the definition of NFA. We must show that $L(M) = L(N)$. To do so (and this will be a common procedure throughout this note), we must first prove some relation between δ^* and Δ^*. We state this relation in the following claim.

Claim. *Given a DFA $M = (Q, \Sigma, \delta, s_0, F)$ and the NFA $N = (Q', \Sigma, \Delta, S_0, F')$ which has been constructed as a function of M, we have that $\forall w \in \Sigma^*, q \in Q$, $\Delta^*(\{q\}, w) = \{\delta^*(q, w)\}$.*

Proof. I omit this proof. It is a straightforward proof by induction. Please come to OH if you'd like to discuss it.

We are now ready to show $L(M) = L(N)$. Consider some arbitrary $w \in \Sigma^*$, then

$$\begin{aligned} w \in L(M) &\iff \delta^*(s_0, w) \in F \\ &\iff \{\delta^*(s_0, w)\} \cap F \neq \emptyset \\ &\iff \Delta^*(\{s_0\}, w) \cap F \neq \emptyset \\ &\iff \Delta^*(S_0, w) \cap F' \neq \emptyset \\ &\iff w \in L(N) \end{aligned}$$

$\boldsymbol{L_{\mathrm{DFA}} \subseteq L_{\mathrm{NFA}}}$. To prove this direction, given any NFA $N = (Q, \Sigma, \Delta, S_0, F)$, we must show that there exists an equivalent DFA $M = (Q', \Sigma, \delta, s_0, F')$ such that $L(M) = L(N)$. We reproduce the explicit construction I showed during Lecture 5 (I drop the subscripts for compactness).

$Q' \coloneqq 2^Q$

$s_0 \coloneqq S_0$

$F' \coloneqq \{B \subseteq Q : B \cap F\}$

$\delta(B,\sigma) := \bigcup_{q\in B} \Delta(q,\sigma)$, for $B \in Q'$ and $\sigma \in \Sigma$. Note that, by definition, the RHS is also equal to $\Delta^*(B,\sigma)$. This will come up in the proof of the upcoming claim.

We must now prove that $L(M) = L(N)$. To do so, we first must establish a relation between δ^* and Δ^*. We state it in the claim below.

Claim. *Given an NFA $N = (Q, \Sigma, \Delta, S_0, F)$ and the DFA $M = (Q', \Sigma, \delta, s_0, F')$ which has been constructed as a function of N, we have that $\forall w \in \Sigma^*, B \subseteq Q$, $\Delta^*(B,w) = \delta^*(B,w)$.*

Study this equality carefully. In particular, do the data types make sense?

Proof. We prove this by induction on the length of w.

Base case: $|w| = 0 \Rightarrow w = \varepsilon$. Then,

$$\delta^*(B,w) = \delta^*(B,\varepsilon) = B = \Delta^*(B,\varepsilon) = \Delta^*(B,w)$$

Inductive hypothesis: We assume that the statement is true for every $w \in \Sigma^*$ where $|w| = n$ for some $n \in \mathbb{N}$.

Inductive step: Suppose $w \in \Sigma^*$ and $|w| = n+1$. Then $w = x\sigma$ for $x \in \Sigma^*, \sigma \in \Sigma$.

$$\begin{aligned}
\Delta^*(B,x\sigma) &= \Delta^*(\Delta^*(B,x),\sigma) && \text{using Fact 1}\\
&= \Delta^*(\delta^*(B,x),\sigma) && \text{by IH}\\
&= \delta(\delta^*(B,x),\sigma) && \text{by definition of } \delta\\
&= \delta^*(B,x\sigma)\\
&= \delta^*(B,w)
\end{aligned}$$

We are now ready to show $L(N) = L(M)$. Consider some arbitrary string $w \in \Sigma^*$, then

$$\begin{aligned}
w \in L(N) &\iff \Delta^*(S_0,w) \cap F \neq \emptyset && \text{by definition of acceptance for NFA}\\
&\iff \delta^*(S_0,w) \cap F \neq \emptyset && \text{using the previous claim}\\
&\iff \delta^*(s_0,w) \in F' && \text{by construction of } s_0 \text{ and } F'\\
&\iff w \in L(M) && \text{by definition of acceptance for DFA}
\end{aligned}$$

This completes the proof. ■

3 Equivalence of NFA and NFA+ϵ

3.1 A brief recall and a comment about notation

Recall from Lecture 5 that NFA+ϵ behave exactly like NFA except that they *allow* ϵ-transitions (pronounced "epsilon"-transitions). ϵ-transitions are transitions that allow an automaton to transition from one state to another without reading any letter from the input tape.

Note that, in order to be extremely explicit, I have used the symbol ϵ (in tex, `\epsilon`) to talk about "epsilon"-transitions rather than the symbol ε (in tex, `\varepsilon`) which I've reserved for the empty string. This difference in notation is because, strictly speaking, ε, the empty string, and ϵ, the symbol representing "epsilon"-transitions, are not the same. The former *is a string* and thus has data type "string". The latter is a special symbol used to label "epsilon"-transitions in automaton. I will be precise in this section and use ϵ and ε diligently, but, in general, I will abuse notation and use ε in all cases (e.g., Lecture 5 notes).

3.2 A formal definition of NFA+ϵ

Definition 3.1 (NFA with ϵ-transitions)**.** A **non-deterministic finite automaton with ϵ-transitions (NFA+ϵ)** N is a 6-tuple $N = (Q, \Sigma, \epsilon, \Delta, S_0, F)$ where

- Q is the finite set of states
- Σ is the input alphabet and $\epsilon \notin \Sigma$
- ϵ is the special symbol representing "epsilon"-transitions
- Δ is the transition function $\Delta : Q \times (\Sigma \cup \{\epsilon\}) \to 2^Q$
- $S_0 \subseteq Q$ is the set of start states
- $F \subseteq Q$ is the set of accept (final) states

To talk about string acceptance for NFA+ϵ, we must create a way to formally talk about the states the NFA+ϵ can reach "for free" using ϵ-transitions. Note that we cannot use the Δ^* extended transition function for vanilla NFA because we do not want to consider strings of the form $a\epsilon b$.

Definition 3.2 (ϵ-`closure`)**.** Given an NFA+ϵ, $N = (Q, \Sigma, \epsilon, \Delta, S_0, F)$, a state $q \in Q$ and a set of states $A \subseteq Q$, we define the ϵ-`closure`[4] for q and A as

$$\epsilon\text{-}\mathtt{closure}(q) = \{p \in Q : \exists \text{ a walk of 0 or more } \epsilon\text{-transitions from } q \text{ to } p\}$$

and

$$\epsilon\text{-}\mathtt{closure}(A) = \bigcup_{q \in A} \epsilon\text{-}\mathtt{closure}(q)$$

Note that, by definition $q \in \epsilon$-`closure`(q) and $A \subseteq \epsilon$-`closure`(A). Next, to talk about string and language acceptance, we need to define NFA+ϵ's extended transition function.

Definition 3.3 (Δ^*_ϵ)**.** Let $N = (Q, \Sigma, \epsilon, \Delta, S_0, F)$ be an NFA+ϵ, and let $A \subseteq Q, x \in \Sigma^*, \sigma \in \Sigma$. The extended transition function for NFA+ϵ $\Delta^*_\epsilon : 2^Q \times \Sigma^* \to 2^Q$ (note that the string input does not allow ϵ) is defined as follows. For the base case, we have that

$$\Delta^*_\epsilon(A, \varepsilon) = \epsilon\text{-}\mathtt{closure}(A) \qquad \text{note the difference between } \epsilon \text{ and } \varepsilon$$

[4]This definition is tacitly assuming you are somewhat familiar with graph theory.

And, in the recursive (inductive) case, we have that[5]

$$\Delta^*_\epsilon(A, x\sigma) = \bigcup_{q \in \Delta^*_\epsilon(A,x)} \epsilon\text{-}\texttt{closure}(\Delta(q, \sigma))$$

Note how similar this recursive definition is to the one for vanilla NFA. The only difference now is that *before* and *after* reading a letter, we expand the set of destination sets by checking which states we can reach *for free*. To get a feel for this definition, let's apply it (recursively) to the string $w = ab$ for some subset of states $A \subseteq Q$ and see if it matches the way I was presenting computations of NFA+ϵ during the lecture

$$\begin{aligned}\Delta^*_\epsilon(A, ab) &= \bigcup_{q \in \Delta^*_\epsilon(A,a)} \epsilon\text{-}\texttt{closure}(\Delta(q, b)) \\ &= \bigcup_{q \in A'} \epsilon\text{-}\texttt{closure}(\Delta(q, b))\end{aligned}$$

Where $A' = \Delta^*_\epsilon(A, a)$ is

$$\begin{aligned}\Delta^*_\epsilon(A, a) &= \bigcup_{p \in \Delta^*(A,\varepsilon)} \epsilon\text{-}\texttt{closure}(\Delta(p, a)) \\ &= \bigcup_{p \in \epsilon\text{-}\texttt{closure}(A)} \epsilon\text{-}\texttt{closure}(\Delta(p, a))\end{aligned}$$

This exactly matches how we would run through *all* of the computations for an NFA+ϵ given the string ab. Let's work through this sequence of recursive calls *bottom-up*, i.e., starting from the base case and working our way up to the original function call. This would look like the following

1. Check if there are any states we can reach for free from the states in A. Call this set of states A_1 ($A \subseteq A_1$).
2. Read a from each of the states in A_1. Call the set of destination states A_2.
3. For each of the states in A_2, check if there are any states that we can reach for free. Call this set of states A_3 ($A_2 \subseteq A_3$).
4. Read b from each of the states in A_3. Call the set of destination states A_4.
5. For each of the states in A_4, check if there are any states that we can reach for free. Call this set of states A_5 ($A_4 \subseteq A_5$).

[5] There was a mistake in a previous version of this note where I defined the recursive case as $\Delta^*_\epsilon(A, x\sigma) = \epsilon\text{-}\texttt{closure}(\Delta(\Delta^*_\epsilon(A, x), \sigma))$. What is the problem with this recursive case? Hint: Take a look at the data types. Thanks to the student who caught this!

6. $\Delta^*_\epsilon(A, ab) = A_5$

We note the following facts about Δ^*_ϵ which are analogous to the facts about Δ^*.

Fact. *Given an NFA+ϵ, $N = (Q, \Sigma, \epsilon, \Delta, S_0, F)$, a subset $A \subseteq Q, B \subseteq Q$, strings $x, y \in \Sigma^*$, we have that*

3. $\Delta^*_\epsilon(A, xy) = \Delta^*_\epsilon(\Delta^*_\epsilon(A, x), y)$

4. $\Delta^*_\epsilon(A \cup B, x) = \Delta^*_\epsilon(A, x) \cup \Delta^*_\epsilon(B, x)$

Proof. The proofs are by induction. I omit them. The proof of the first fact uses the property that $\epsilon\text{-}\texttt{closure}(\epsilon\text{-}\texttt{closure}(A)) = \epsilon\text{-}\texttt{closure}(A)$. Do you see why?

We are now able to formally define the notion of string and language acceptance for NFA+ϵ.

Definition 3.4 (String acceptance)**.** Given an NFA+ϵ $N = (Q, \Sigma, \epsilon, \Delta, S_0, F)$ and a string $w \in \Sigma^*$, we say that N accepts w if and only if $\Delta^*_\epsilon(S_0, w) \cap F \neq \emptyset$.

Definition 3.5 (Language acceptance)**.** Given an NFA+ϵ $N = (Q, \Sigma, \epsilon, \Delta, S_0, F)$, the language accepted by N is

$$L(N) = \{w \in \Sigma^* : \Delta^*_\epsilon(S_0, w) \cap F \neq \emptyset\}$$

3.3 Equivalence between NFA and NFA+ϵ

We are (finally) ready to re-state the theorem I presented towards the end of Lecture 5[6].

Theorem 2. *Given some alphabet Σ, the family of languages accepted by NFA, $L_{NFA} = \{L(N) : N \text{ is an NFA}\}$, is exactly the same as the family of languages accepted by NFA+ϵ, $L_{NFA+\epsilon} = \{L(N) : N \text{ is an NFA+}\epsilon\}$.*

Proof. We again must prove this theorem by double inclusion.
$\boldsymbol{L_{\mathbf{NFA}} \subseteq L_{\mathbf{NFA+\epsilon}}}$. As discussed during the lecture, this direction is easy because an NFA can be thought of as an NFA+ϵ which does not have any ϵ-transitions. Thus, we could take an arbitrary NFA N and convert it to an NFA+ϵ. In this case, $\forall q \in Q, A \subseteq Q$, we will have $\epsilon\text{-}\texttt{closure}(q) = q$ and $\epsilon\text{-}\texttt{closure}(A) = A$ in which case $\Delta^*(A, w) = \Delta^*_\epsilon(A, w)$ is clearly true.
$\boldsymbol{L_{\mathbf{NFA+\epsilon}} \subseteq L_{\mathbf{NFA}}}$. Consider some arbitrary NFA+ϵ $N = (Q, \Sigma, \epsilon, \Delta, S_0, F)$. We *explicitly* construct an NFA $N' = (Q', \Sigma', \Delta', S'_0, F')$ such that $L(N') = L(N)$. We construct N' explicitly as follows

$Q' := Q$

$\Sigma' := \Sigma$. Note: This means ϵ is not part of N''s input alphabet, which is desired.

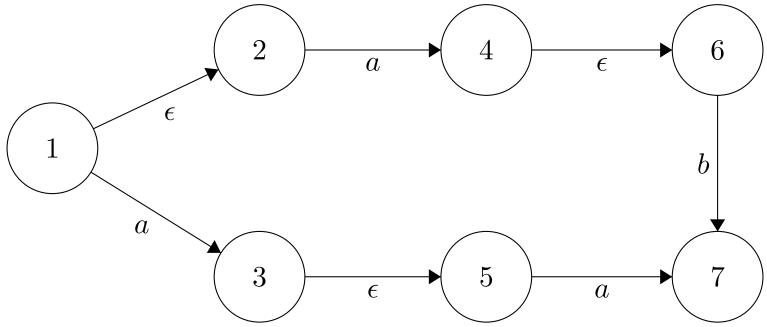
$S'_0 := S_0$

$F' := F \cup \{s \in S_0 : \epsilon\text{-}\texttt{closure}(s) \cap F \neq \emptyset\}$. We will see why this is necessary in a moment.

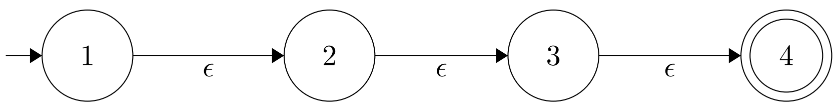
$\Delta'(q, \sigma) := \Delta^*_\epsilon(\{q\}, \sigma)$ for $q \in Q, \sigma \in \Sigma$

[6]Imagine if I had done all of this during the lecture!

Note how Δ' is defined. It is meant to account for any ϵ-transitions *before and after* reading the letter σ. For instance, suppose we have the following directed subgraph in N



Then in N', $\Delta'(1, a) = \Delta_\epsilon^*(\{1\}, a) = \{4, 6, 3, 5\}$. The only other tricky part about this conversion is the way we defined F'. The set of accept states of N' is the set of accept states of N *along with* any start states that can, for free, reach a final state. This is because, in vanilla NFA, the only way for the empty string to be accepted is for a start state to be a final state. Thus, if we have the following situation in N'



Then the state 1 will be an accept state in N.

We must now show that $L(N) = L(N')$. To do so, we need a relation between Δ'^* (which follows the definition of the extended transition function for vanilla NFA) and Δ_ϵ^*. We state it in the following claim.

Claim. *Given an NFA+ϵ $N = (Q, \Sigma, \epsilon, \Delta, S_0, F)$ and the NFA $N' = (Q', \Sigma', \Delta', S_0', F')$ which has been constructed as a function of N, we have that $\forall w \in \Sigma^*, |w| \geq 1, B \subseteq Q$, $\Delta'^*(B, w) = \Delta_\epsilon^*(B, w)$.*

Note the lower bound on the length of w - the statement is in fact false if $|w| = 0$ by definition of Δ'^* and Δ_ϵ^*.

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

Base case: $|w| = 1 \Rightarrow w = \sigma, \sigma \in \Sigma$. Then,

$$\begin{aligned} \Delta'^*(B, \sigma) &= \bigcup_{q \in B} \Delta'(q, \sigma) \\ &= \bigcup_{q \in B} \Delta_\epsilon^*(\{q\}, \sigma) \\ &= \Delta_\epsilon^*(B, \sigma) && \text{Generalization of Fact 3} \end{aligned}$$

Inductive hypothesis: We assume that the statement is true for every $w \in \Sigma^*$ where $|w| = n$ for some $n \in \mathbb{N}, n \geq 1$.

Inductive step: Suppose $w \in \Sigma^*$ and $|w| = n+1$. Then $w = x\sigma$ for $x \in \Sigma^*, \sigma \in \Sigma$.

$$\begin{aligned}
\Delta'^*(B, x\sigma) &= \Delta'^*(\Delta'^*(B, x), \sigma) && \text{using Fact 1} \\
&= \Delta_\epsilon^*(\Delta_\epsilon^*(B, x), \sigma) && \text{by IH and the same argument as in the BC} \\
&= \Delta_\epsilon^*(B, x\sigma) && \text{using Fact 4}
\end{aligned}$$

We are now able to show $L(N) = L(N')$. Pick some arbitrary $w \in \Sigma^*$. If $w = \varepsilon$ then

$$\begin{aligned}
\varepsilon \in L(N) &\iff \text{there is an } \epsilon \text{ walk of length 0 or more from some } s \in S_0 \text{ to some } f \in F \\
&\iff \exists s \in S_0 \text{ such that } \epsilon\texttt{-closure}(s) \cap F \neq \emptyset \\
&\iff S_0' \cap F' \neq \emptyset \\
&\iff \varepsilon \in L(N')
\end{aligned}$$

Otherwise, if $w \neq \varepsilon$ then it has length greater or equal to 1. Then

$$\begin{aligned}
w \in L(N) &\iff \Delta_\epsilon^*(S_0, w) \cap F \neq \emptyset \\
&\iff \Delta'^*(S_0', w) \cap F' \neq \emptyset \\
&\iff w \in L(N')
\end{aligned}$$

Done! ■

There is a much cleaner way of proving this result using *homomorphisms*. I leave that for another note.