Non deterministic finite automata with ε transitions

Languages

- Recall.
  - What is a language?
  - What is a class of languages?

Finite Automata

- Consists of
  - A set of states (Q)
  - A start state (q_o)
  - A set of accepting states (*F*)
  - Read symbols (Σ)
  - Transition function (δ)
- Let's recap

First there was the DFA

- Deterministic Finite Automata
  - For every state and every alphabet symbol there is exactly one move that the machine can make.
  - $\delta : Q \times \Sigma \rightarrow Q$
  - δ is a total function: completely defined. I.e. it is defined for all $q \in Q$ and $a \in \Sigma$

Then, the NFA

- Non-determinism
  - When machine is in a given state and reads a symbol, the machine will have a choice of where to move to next.
  - There may be states where, after reading a given symbol, the machine has nowhere to go.
  - Applying the transition function will give, not 1 state, but 0 or more states.

Non-Deterministic Finite Automata (NFA)

- Transition function
  - δ is a function from $Q \times \Sigma$ to 2^Q
  - δ (q, a) = subset of Q (possibly empty)

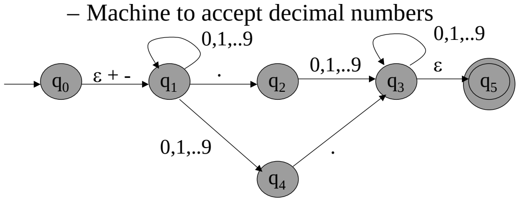
- And now…
- Introducing…
- The newest in the FA family…
- The Non deterministic finite automata with ε transitions (ε-NFA)

Nondeterministic Finite Automata with ε transitions (ε-NFA)

- For both DFAs and NFAs, you must read a symbol in order for the machine to make a move.
- In Nondeterministic Finite Automata with ε transitions (ε-NFA)
  - Can make move without reading a symbol off the read tape
  - Such a move is called a ε transition

Nondeterministic Finite Automata with ε transitions (ε-NFA)

- Example:
  - Machine to accept decimal numbers

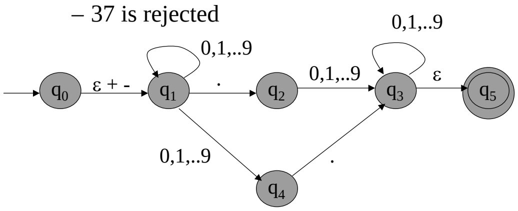


Nondeterministic Finite Automata with ε transitions (ε-NFA)

- How does such a machine accept?
  - A string will be accepted if there is <u>at least one</u> sequence of state transitions on an input (including ε transitions) that leaves the machine in an accepting state.

Nondeterministic Finite Automata with ε transitions (ε-NFA)

- Example:
  - -3.45 is accepted
  - .5678
  - 37 is rejected



Nondeterministic Finite Automata with ε transitions (ε-NFA)

- A Non-Deterministic Finite Automata with ε transitions is a 5-tuple $(Q, \Sigma, q_o, \delta, F)$ where
  - Q is a finite set (of states)
  - Σ is a finite alphabet of symbols
  - $q_o \in Q$ is the start state
  - $F \subseteq Q$ is the set of accepting states
  - δ is a function from $Q \times (\Sigma \cup \{\varepsilon\})$ to 2^Q (transition function)

Nondeterministic Finite Automata with ε transitions (ε-NFA)

- Transition function
  - δ is a function from Q x (Σ ∪ {ε}) to 2^Q
  - δ (q, a) = subset of Q (possibly empty)
  - In our example
    - $\delta(q_1, 0) = \{q_1, q_4\}$
    - $\delta(q_1, .) = \{q_1\}$
    - $\delta(q_1, +) = \varnothing$
    - $\delta(q_0, \varepsilon) = \{q_1\}$

Nondeterministic Finite Automata with ε transitions (ε-NFA)

- Transition function on a string
  - $\hat{\delta}$ is a function from Q x Σ^* to 2^Q
  - $\hat{\delta}$ (q, x) = subset of Q (possibly empty)
  - Set of all states that the machine can be in, upon following all possible paths on input x.
  - We'll need to consider all paths that include the use of ε transitions

ε-Closure

- ε closure
  - Before defining the transition function on a string ($\hat{\delta}$ (q,x)), it is useful to first define what is known as the ε closure.
  - Given a set of states S, the ε closure will give the set of states reachable from each state in S using only ε transitions.

ε-Closure

- ε closure: Recursive definition
  - Let M = (Q, Σ, q_o, δ, F) be a ε-NFA
  - Let S be a subset of Q
  - The ε closure, denotes ECLOSE(S) is defined:
    - For each state p ∈S, p ∈ ECLOSE(S)
    - For any q ∈ ECLOSE(S), every element of δ(q, ε) ∈ ECLOSE(S)
    - No other elements of Q are in ECLOSE(S)

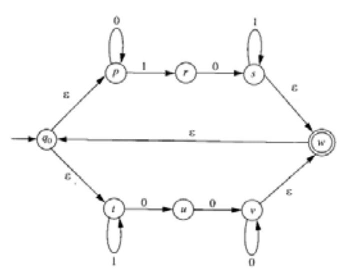
ε-Closure

- ε-Closure : Algorithm
  - Since we know that ECLOSE(S) is finite, we can convert the recursive definition to an algorithm.

  - To find ECLOSE(S) where S is a subset of Q
  - Let T = S
  - While (T does not change) do
    - Add all elements of δ(q, ε) where q ∈ T
  - ECLOSE(S) = T

ε-Closure

- Example



ε-Closure

- ε closure: Example
  - Find ECLOSE({s}) in our example
  - T = {s} initial step
  - T = {s, w} add δ(s, ε)
  - T = {s, w, q_0} add δ(w, ε)
  - T = {s, w, q_0, p,t} add δ(q_0, ε)
  - δ(w, ε) = δ(w, ε) =∅
  - We are done,
    - ECLOSE({s}) = T = {s, w, q_0, p,t}

Nondeterministic Finite Automata with ε transitions (ε-NFA)

- Now lets define $\hat{\delta}$
  1. For any q ∈Q, $\hat{\delta}$ (q, ε) = ECLOSE ({q})
  2. For any y ∈ Σ^*, a ∈ Σ, q ∈Q

$$\hat{\delta}(q, ya) = ECLOSE\left(\bigcup_{p \in \hat{\delta}(q,y)} \delta(p,a) \right)$$

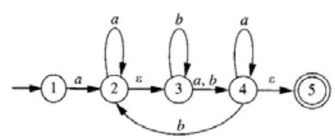
Set of all states obtained by applying δ to all states in δ^* (q,y) and input a and taking the ε closure of the result

Nondeterministic Finite Automata with ε transitions (ε-NFA)

- Accepting a string
  - A string x is accepted if running the machine on input x, considering all paths, including the use of ε transitions, puts the machine into one of the accepting states
  - Formally:
    - x ∈ Σ^* is accepted by M if
    - $\hat{\delta}$ (q_0, x) ∩ F ≠ ∅

Nondeterministic Finite Automata with ε transitions (ε-NFA)

- Are the following strings accepted by the ε-NFA below:
  - aba
  - ababa
  - aaabbb



Nondeterministic Finite Automata with ε transitions (ε-NFA)

- I bet that you're asking…
  - Can JFLAP handle ε-NFAs?
  - Well, let's check and see!

Nondeterministic Finite Automata with ε transitions (ε-NFA)

- Language accepted by M
  - The language accepted by M
    - L(M) = { x ∈ Σ^* | x is accepted by M }
- If L is a language over Σ, L is accepted by M iff L = L(M).
  - For all x ∈ L, x is accepted by M.
  - For all x ∉ L, x is rejected by M.

Nondeterministic Finite Automata with ε transitions (ε-NFA)

- Why they're a good idea
  - Given a regular expression, it is far easier to create an ε-NFA for the language described by the expression than it is to create a plain old DFA.
  - It will also be essential when showing the Fas accept the class of Regular Languages.
  - Questions?

DFA / NFA / ε-NFA Equivalence

- Surprisingly enough
  - ε transitions to our NDFA does NOT give it any additional language accepting power.
  - DFAs and NFAs and ε-NFA are all equivalent
    - Every language that can be accepted by a ε-NFA can also be accepted by an DFA which can also be accepted by a NFA.
  - Let's show this

ε-NFA -> DFA

- Given ε-NFA find DFA
  - Let $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$ be a ε-NFA then
    - There exists a DFA, $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$
    - Such that $L(E) = L(D)$

ε-NFA -> DFA

- Basic idea
  - Very similar to the subset construction algorithm
    - Recall that for a ε-NFA , $\delta: Q \times \Sigma \rightarrow 2^Q$
    - Use the states of D to represent subsets of Q.

ε-NFA -> DFA

- Formal definition
  - $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$ be a ε-NFA
  - We define DFA, $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$
    - $Q_D = 2^{QE}$
    - $q_D = ECLOSE(q_0)$
    - F_D = sets containing at least one state from F_E

ε-NFA -> DFA

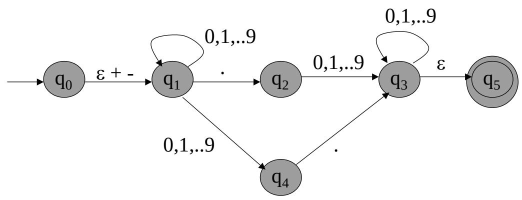
- Computing δ_D
  - $\delta_D(S, a)$ for $S \in Q_D$, $a \in \Sigma$
    - Let $S = \{ p_1, p_2, \ldots, p_n \}$
    - Compute the set of all states reachable from states in S on input a using transitions from E.

$$\{r_1, r_2, \cdots, r_m\} = \bigcup_{i=1}^{n} \delta_E(p_i, a)$$

    - $\delta_D(S, a)$ will be the union of the ε closures of the elements of $\{r_1, \ldots, r_m\}$

$$\delta_D(S, a) = \bigcup_{j=1}^{m} ECLOSE(r_j)$$

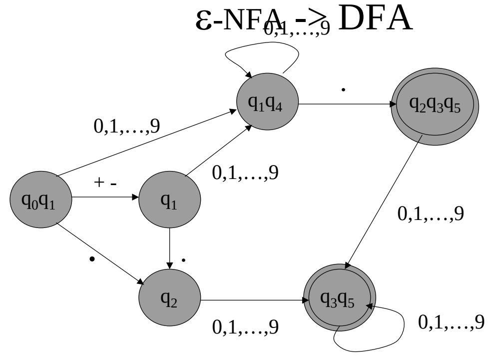
ε-NFA -> DFA



ε-NFA -> DFA

State	ε closure
q_0	$\{q_0, q_1\}$
q_1	$\{q_1\}$
q_2	$\{q_2\}$
q_3	$\{q_3, q_5\}$
q_4	$\{q_4\}$
q_5	$\{q_5\}$

ε-NFA -> DFA



ε-NFA -> DFA

- Now we must show that D accepts the same language as E
  - Can be shown (using induction) that for all $x \in \Sigma^*$
    - $\hat{\delta}_D(q, x) = \hat{\delta}_E(q, x)$
    - See Theorem 2.22

ε-NFA -> DFA

- Show that D and E recognize the same language
  - x is accepted by E iff $\hat{\delta}_E(q_0, x) \cap F_E \neq \emptyset$
  - x is accepted by D iff $\hat{\delta}_D(q_D, x) \cap F_E \neq \emptyset$
  - Thus,
    - x is accepted by D iff x is accepted by E
- Questions?

DFA -> ε-NFA

- The other direction is fairly straighforward.
  - For any DFA, there is an equivalent NFA
  - An NFA is nothing more than a ε-NFA with no ε transitions. Thus
    - $\delta(q, \varepsilon)$ for all states $q = \emptyset$

What have we shown

- For every DFA, there is an NFA that accepts the same language and visa versa
- For every DFA, there is a ε-NFA that accepts the same language, and visa versa
- Thus, for every NFA there is a ε-NFA that accepts the same language, and visa versa
- DFAs, NFAs, and ε-NFA s are equivalent!

Questions?

- Let’s take a break.