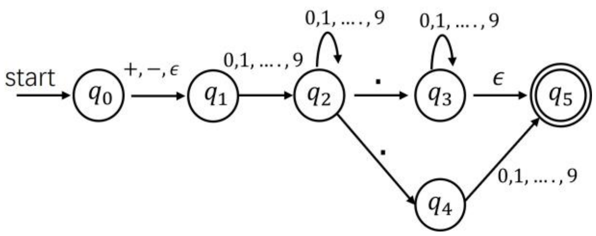
Week 3 Summary Sheet
Wanrong Yang
11/10/2024

NFA with ϵ-Transitions ($\epsilon - NFA$)

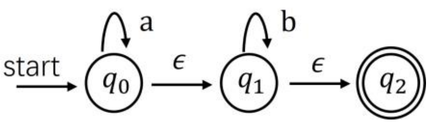
1. Introduction
   - $\epsilon - \text{NFA},\ E = \{Q, \Sigma, \delta, q_0, F\}$
   - $\delta(q, a) = \{q_1, q, q_3, \dots, q_k\}$
   - Keep it in mind that for a, it can be input symbols from Σ or ϵ
2. Examples
   - $\epsilon - \text{NFA}$ for decimal numbers (they are strings)
   - languages look like this $L = \{-467.35, 467.35, +467.35, .354, 467., \dots\dots\}$
   - 

3. $\epsilon - \text{close}$(more details are in the second part)
   - all accessible following states though ϵ transtions
   - taking the last $\epsilon - NFA$ as example
   - $\epsilon - \text{close}(q_0) = \{q_0, q_1\}$
   - $\epsilon - \text{close}(q_1) = \{q_1\}$
   - $\epsilon - \text{close}(q_2) = \{q_2\}$
   - $\epsilon - \text{close}(q_3) = \{q_3, q_5\}$
   - $\epsilon - \text{close}(q_4) = \{q_4\}$
   - $\epsilon - \text{close}(q_5) = \{q_5\}$

Epsilon closure

1. $\epsilon - closure(q)$
   it is a set of states reachable from q on $\epsilon - transition$ alone.
   - it always starts with q itself. $\epsilon - closure(q) = \{q\}$
   - if there is an $\epsilon - transition$ from state q to state p, then update $\epsilon - closure(q) = \{q, p\}$
2. easy concepts, but please do exercise as much as you can to get familiar with

Convert the given ϵ-NFA to DFA

- we will start with an simple example to explain
- 

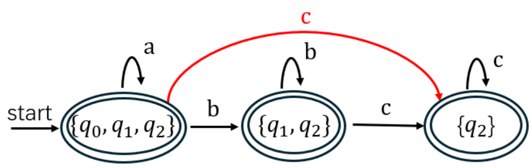
1. Firstly, we will need to have the transition table for this ϵ-NFA
- the transition table is :

	a	b	c	ϵ
q_0	$\{q_0\}$	$\emptyset$	$\emptyset$	$\{q_1\}$
q_1	$\emptyset$	$\{q_1\}$	$\emptyset$	$\{q_2\}$
q_2	$\emptyset$	$\emptyset$	$\{q_2\}$	$\emptyset$

2. then get epsilon closure for each state
- $\epsilon - closure(q_0) = \{q_0, q_1, q_2\}$
- $\epsilon - closure(q_1) = \{q_1, q_2\}$
- $\epsilon - closure(q_2) = \{q_2\}$
3. take epsilon closure result of each state as states in your targeted DFA
- then targeted DFA may have 3 states, $\{q_0, q_1, q_2\}$, $\{q_1, q_2\}$ and $\{q_2\}$ respectively
4. get transitions for targeted DFA transition table
- let δ_D as transition function of targeted DFA, and δ_E as ϵ-NFA transition func.
- we have following principles
  $\delta_D(\{q_0, q_1, q_2\}, a) = \varepsilon - closure(\delta_E(\{q_0, q_1, q_2\}, a))$
- $\delta_E(\{q_0, q_1, q_2\}, a) = \delta_E(q_0, \mathrm{a}) \cup \delta_E(q_1, \mathrm{a}) \cup \delta_E(q_2, \mathrm{a}) = \{q_0\}$
- so first equation changes to $\delta_D(\{q_0, q_1, q_2\}, a) = \varepsilon - closure(q_0) = \{q_0, q_1, q_2\}$
5. apply the principle on each possible DFA states and then you can have the following transition table for targeted DFA
- arrow means start state (because it includes the state q_0, which is a start state of given epsilon NFA)
- * start means all the final states, as q_2 is included in, which is a final state of given epsilon NFA)

	a	b	c
→* $\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
* $\{q_1, q_2\}$	$\emptyset$	$\{q_1, q_2\}$	$\{q_2\}$
* $\{q_2\}$	$\emptyset$	$\emptyset$	$\{q_2\}$

6. based on the DFA transition table, get DFA diagram



Regular expressions (RE)

1. regular expression specifies pure text strings
   - language including DFA, NFA & Regular expression
   - RE also be called regular language
2. operators for language
   - 2 languages $\mathrm{L}_1 = \{01, 101, 11\}$ and $\mathrm{L}_2 = \{101, \varepsilon\}$
   - Union $\mathrm{L}_1 \cup \mathrm{L}_2 = \{01, 101, 11, \varepsilon\}$ no repeated strings
   - Concatenation $\mathrm{L}_1\mathrm{L}_2 = \{01101, 01, 101101, 101, 11101, 11\}$
   - Closure $\mathrm{L} = \{0,1\}$, closure operation means L^*

3. RE definitions

3.1 Basics

RE=ε, then the corresponding language is $\{\varepsilon\}$

RE=a, then the corresponding language is $\{\mathrm{a}\}$

3.2 if R_1 and R_2 are regular expressions

then, $R_1 + R_2$ means $\mathrm{L}(R_1) \cup \mathrm{L}(R_2)$

then, R_1R_2 means $\mathrm{L}(R_1)\mathrm{L}(R_2)$, concatenation of languages

then, $R_1{}^*$ means $(\mathrm{L}(R_1))^*$

3.3 example

construct a RE for the language accepting all strings which have bab as a substring over $\Sigma = \{a, b\}$

- then, the RE should be $\mathrm{R} = (\mathrm{a}+\mathrm{b})^*\mathrm{bab}(\mathrm{a}+\mathrm{b})^*$
- $(\mathrm{a}+\mathrm{b})^*$ means any number of combination of a and b, any number of a's and b's

  construc a RE start with ab
- $\mathrm{RE} = \mathrm{ab}(\mathrm{a}+\mathrm{b})^*$
- End with aba, regular expression of this could be $(\mathrm{a}+\mathrm{b})^*\mathrm{aba}$

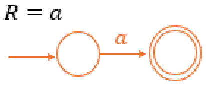
Translation of RE to NFA

1. they are equivalent when they are tying to describe a same language
2. Precedance of reguar expressions
   - ( )
   - *
   - concatenation
   - +
3. things you will need to always keep it in mind

$R = \varepsilon$

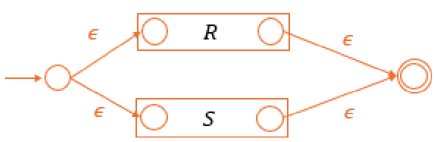
$R = \emptyset$

$R = a$



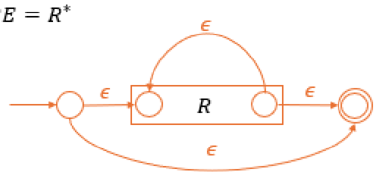
Now we have 2 regular expressions R *and* S

$RE = R + S$



$RE = RS$

$RE = R^*$



$RE = (R)$

$\varepsilon - NFA$ for R $=$ $\varepsilon - NFA$ for (R)