Regular Expressions (RE)

Asai Asaithambi

Spring 2018

Topic Overview

What does this topic cover?

- Basics
  - Precedence of Operations
- Examples of REs
- Things to Remember
- REs $\equiv$ Regular Languages
- Generalized Nondeterministic
  Finite Automata (GNFA)
- Converting DFAs to REs

Let $\Sigma = \{0, 1\}$. We wish to combine

elements of Σ using $\cup$, $\circ$ and *.

- Consider $(0 \cup 1)0^*$.
- This represents the language consisting of all strings starting with a 0 or a 1 followed by any number of 0s.
- How do we know this?
  - 0 and 1 are shorthand for $\{0\}$ and $\{1\}$.
  - Thus, $0 \cup 1$ represents $\{0\} \cup \{1\} = \{0, 1\}$.
  - Also, 0^* represents $\{0\}^* = \{\varepsilon, 0, 00, 000, \cdots\}$
  - Finally, $(0 \cup 1)0^*$ is shorthand for $(0 \cup 1) \circ 0^*$
- As another example, consider $(0 \cup 1)^*$
  - This represents the language consisting of all possible strings of 0s and 1s
- With $\Sigma = \{0, 1\}$, we may write Σ in place of $0 \cup 1$.

Regular Expressions (cont.)

If Σ is any alphabet, then

- Σ describes the language consisting of
  all strings of length 1 over that alphabet
- Σ^* describes the language consisting of
  all strings over that alphabet
- Σ^*1 describes the language consisting of
  all strings that end in a 1
- The language $(0\Sigma^*) \cup (\Sigma^*1)$ consists of
  all strings that start with a 0 or end in a 1

Operator Precedence:

- Star is done first;
- then concatenation; and
- then the union.

Regular Expressions: Formal Definition

Let Σ be the alphabet.

R is a *Regular Expression* (RE), if R is:

- ε
- a, for some a $\in \Sigma$
- $\emptyset$
- $R_1 \cup R_2$, where R_1 and R_2 are REs
- $R_1 \circ R_2$, where R_1 and R_2 are REs
- (R_1^*), where R_1 is a RE

Note:

- ε represents the language $\{\varepsilon\}$.
- a $\in \Sigma$ represents the language $\{\mathrm{a}\}$.
- $\emptyset$ represents the empty language.

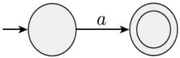
Things to Remember

- ε is the language containing one string, the empty string.
- $\emptyset$ is the language that contains no strings.
- R^k represents the concatenation of k R's.
- $R^+ = RR^* = R^k$ for $k \geq 1$. Also, $R^* = \varepsilon \cup R^+$.
- R is the RE; $L(R)$ is the language represented by R.
- $R \circ \varepsilon = R$; $R \cup \varepsilon$ may not equal R.
- $R \cup \emptyset = R$; $R \circ \emptyset \neq R$

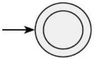
RE Examples with $\Sigma = \{0, 1\}$

- $0^*10^* = \{w \mid w \text{ contains a single } 1\}$.
- $\Sigma^*1\Sigma^* = \{w \mid w \text{ has at least one } 1\}$.
- $\Sigma^*001\Sigma^* = \{w \mid w \text{ contains } 001 \text{ as a substring}\}$.
- $1^*(01^+)^* = \{w \mid \text{every } 0 \text{ is followed by at least one } 1\}$.
- The length of a string is the number of symbols in it.
- $(\Sigma\Sigma)^* = \{w \mid w \text{ is a string of even length}\}$.
- $(\Sigma\Sigma\Sigma)^* = \{w \mid w \text{ the length of } w \text{ is a multiple of } 3\}$.
- $01 \cup 10 = \{01, 10\}$.
- $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1 =$
  $\{w \mid w \text{ begins and ends with the same symbol}\}$.
- $(0 \cup \varepsilon)1^* = 01^* \cup 1^*$
  $0 \cup \varepsilon = \{0, \varepsilon\}$ - puts 0 or ε preceding 1^*.
- $1^*\emptyset = \emptyset$ and $\emptyset^* = \varepsilon$

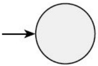
- A language is regular if and only if some RE describes it.
- Converting RE R to NFA:
- $R = a$ for $a \in \Sigma$



- $R = \varepsilon$



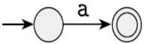
- $R = \emptyset$



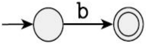
- The NFAs for $R_1 \cup R_2$, $R_1 \circ R_2$ and R_1^* are constructed
  as was done previously using ε edges.

NFA for $(ab \cup a)$

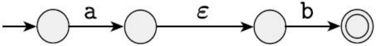
a:



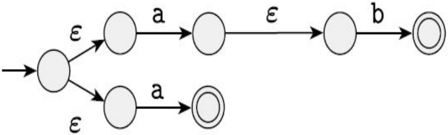
b:



ab:

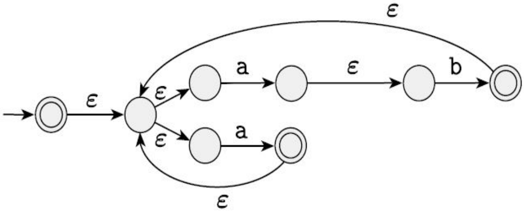


ab ∪ a:



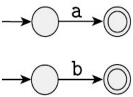
NFA for $(ab \cup a)^*$

$(\mathrm{ab} \cup \mathrm{a})^*$:

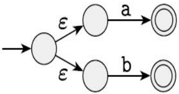


NFA for $(a \cup b)^*$

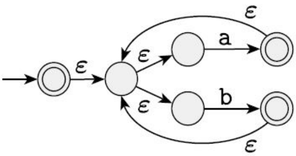
a, b:



$a \cup b$:

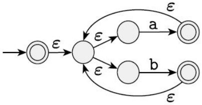


$(a \cup b)^*$:

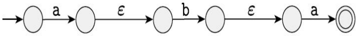


NFA for $(a \cup b)^*$ aba

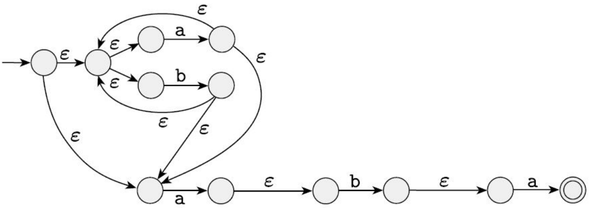
$(a \cup b)^*$:



aba:



$(a \cup b)^*$aba:



DFA to RE

- A regular language can be described by a RE.
- To see/show this, we need the idea of a GNFA.
- A GNFA is a NFA with the following properties:
  - Has only one start state and only one accept state.
  - Transitions (arcs) are labeled with REs.
  - The single start state has:

    arcs going to every other state.

    no arcs coming into the start state.
  - The single accept state has:

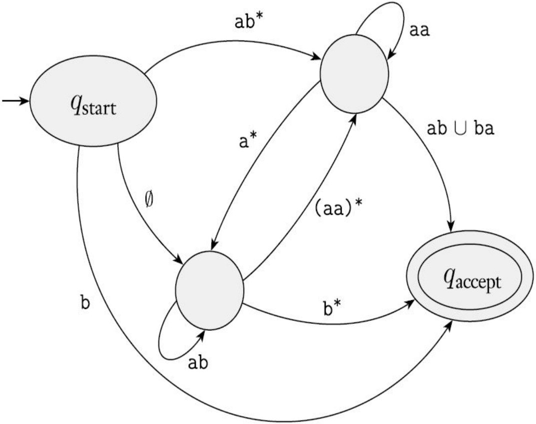
    arcs coming from every state.

    no arcs going out from the accept state.
  - For all other states there is:

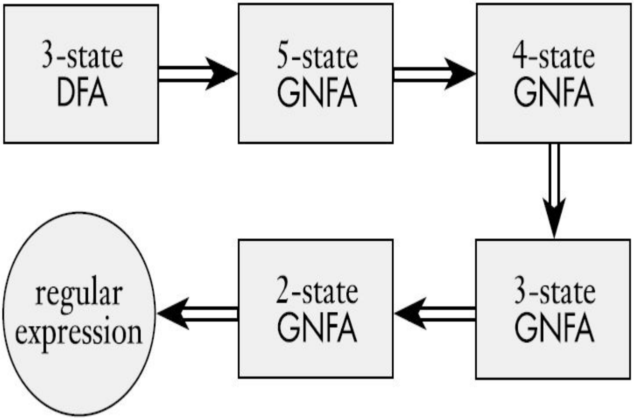
    an arc from every state to every other state.
  - There is a loop at every state,

    except the start and accept states.

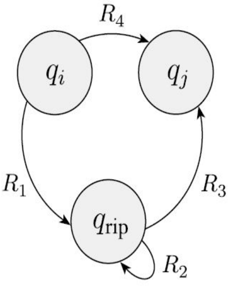
Generalized Nondeterministic FA (GNFA)



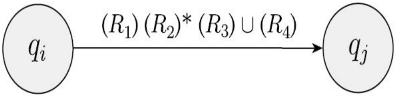
Stages in the Conversion of DFA to RE



Reducing/Ripping One State from a GNFA

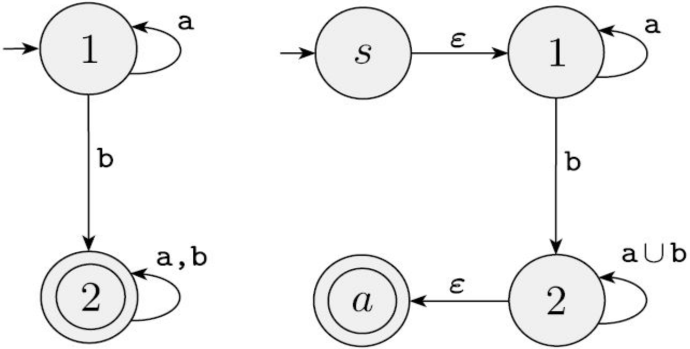


before

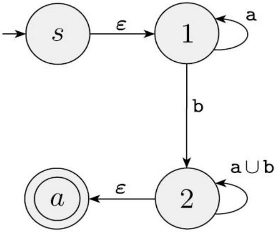


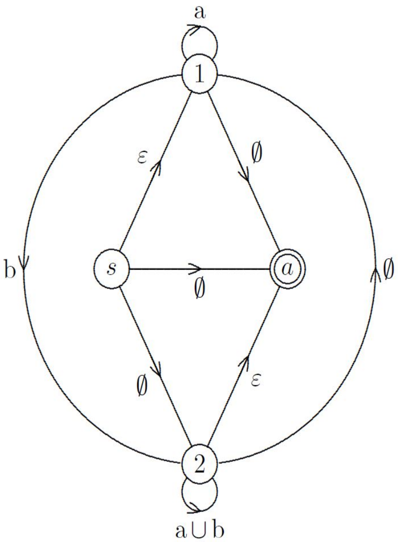
after

Converting DFA to RE: An Example

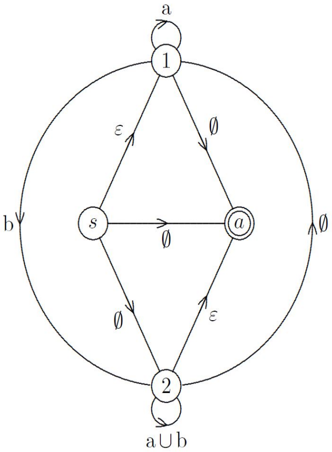


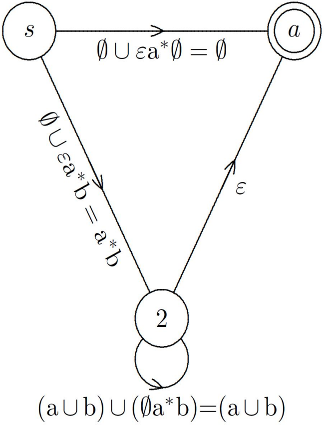
DFA to RE Example (cont.)



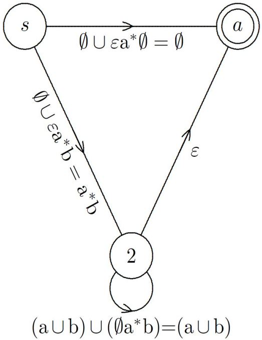


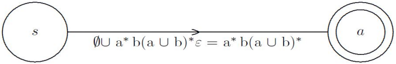
DFA to RE: Rip State 1 First



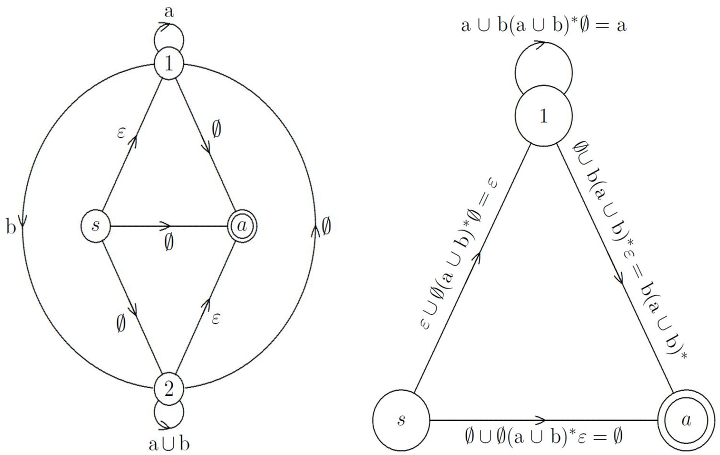


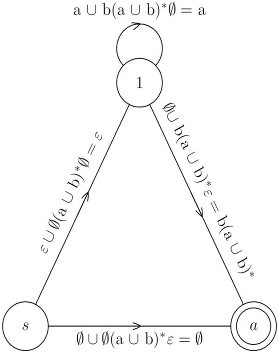
DFA to RE: Rip State 2 Second

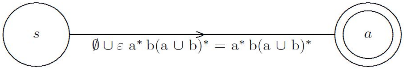




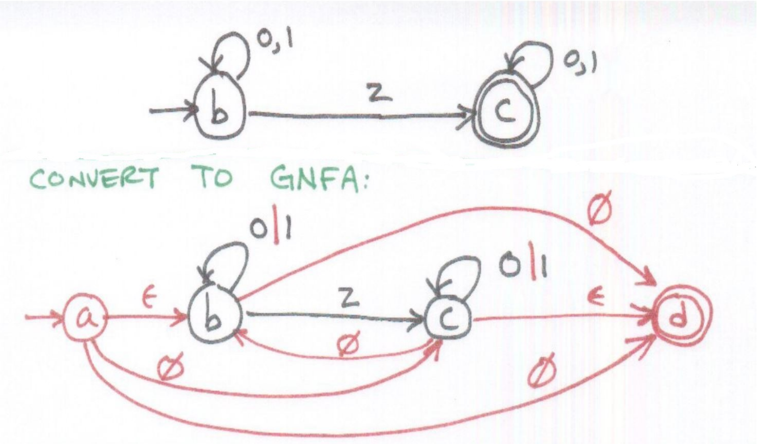
DFA to RE: Rip State 2 First



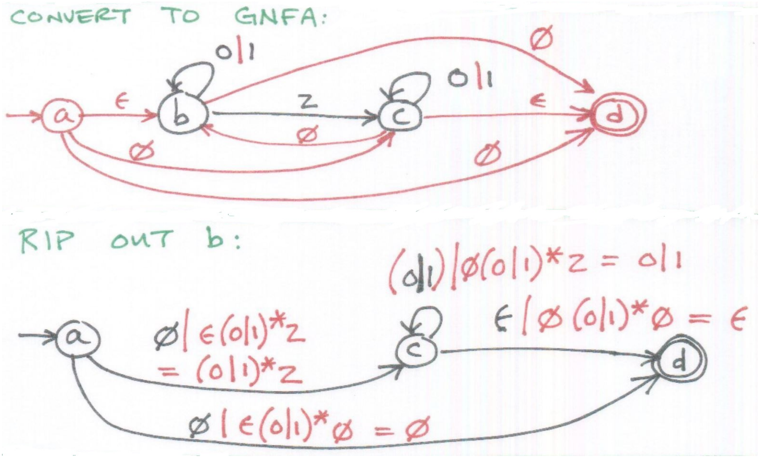

a ∪ b(a ∪ b)*∅ = a
1
ε ∪ ∅(a ∪ b)*∅ = ε
∅∪ b(a ∪ b)*ε = b(a ∪ b)*
s
a
∅ ∪ ∅(a ∪ b)*ε = ∅


s
∅ ∪ ε a* b(a ∪ b)* = a* b(a ∪ b)*
a

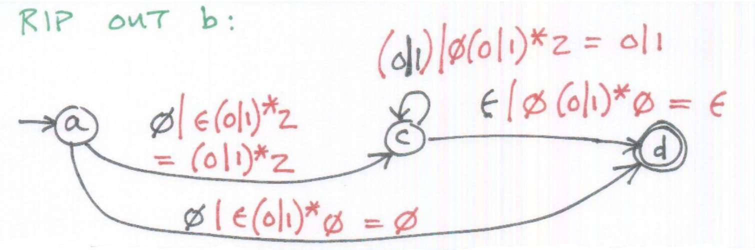
DFA to RE: A Second Example

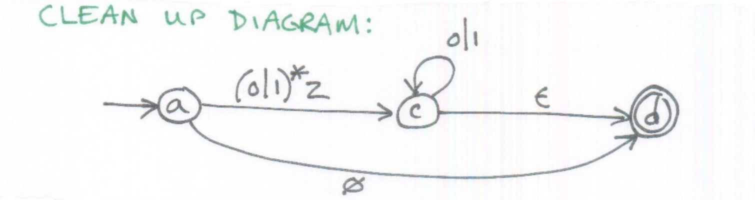


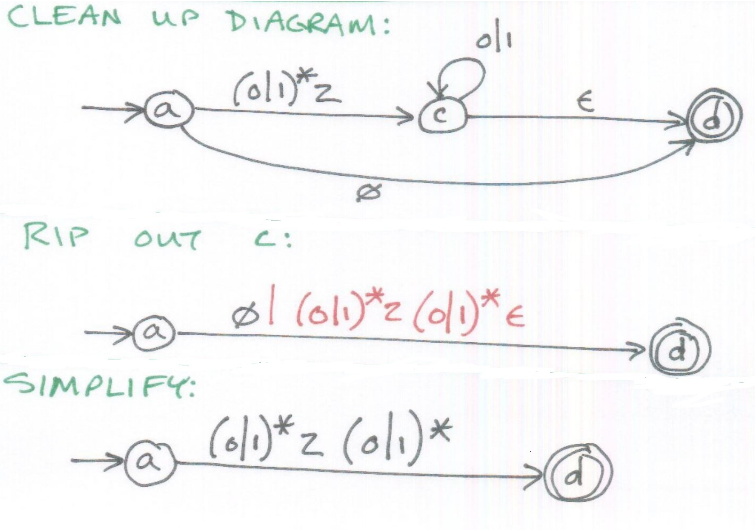
DFA to RE: Second Example (cont.)



DFA to RE: Second Example (cont.)






CLEAN UP DIAGRAM:
0|1
(0|1)*z
ϵ
a
c
d
∅
RIP OUT c:
∅ | (0|1)*z (0|1)* ϵ
a
d
SIMPLIFY:
(0|1)*z (0|1)*
a
d