CSC236 *Intro. to* the Theory of Computation

Lecture 10: fsa and regular languages

Amir H. Chinaei, Fall 2016

Office Hours: W 2-4 BA4222

ahchinaei@cs.toronto.edu
http://www.cs.toronto.edu/~ahchinaei/

Course page:
http://www.cdf.toronto.edu/~csc236h/fall/index.html

Section page:
http://www.cdf.toronto.edu/~csc236h/fall/amir_lectures.html

review

- last week
  - intro to FSA:
    - useful to recognize a language
      - e.g. used in the lexical analyzer
    - and in many other problems that can be encoded to language recognition
- this week:
  - what languages FSAs can recognize?

FSA formal definition

- is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$
  - Q is the set of states, which is finite & non-empty
  - Σ is the alphabet, which is finite & non-empty
  - $\delta : Q \times \Sigma \rightarrow Q$ is the transition function
  - $q_0 \in Q$ is the start state
  - $F \subseteq Q$ is the set of accept states
- $L(M)$ is a language that machine M accepts,
  - i.e., set of all strings that machine M accepts

Example 86 revisited

- FSA that only accepts strings with an odd number of a's, and any number of b's.

$\Sigma = \{a, b\}$

$L(M)$ for Example 86?

- set of all strings that M accepts:

regular languages

languages that can be recognized by an FSA.

- e.g.,
  - The language recognized by FSA in **Example 85**: *float numbers*. $\Sigma = \{0..9, +, -, .\}$
  - The language recognized by FSA in **Example 83**: *simple identifiers*. $\Sigma = \{a..z, 0..9\}$

formal definition

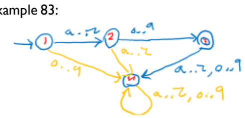
- let Σ be the alphabet:
  - the empty set, $\varnothing$, is a RL
  - the set $\{\varepsilon\}$ is a RL
  - for each a $\in \Sigma$, the set $\{a\}$ is a RL
  - If L_1 and L_2 are regular languages, then
    - union: $L_1 \cup L_2$ is RL
    - concatenation: $L_1 . L_2$ is a RL
    - Kleene star: $L_1{}^*$ is a RL
- no other RL over Σ exists.

example

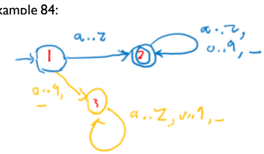
let $\Sigma = \{a, b\}$

notes

Example 83:

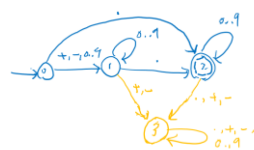


Example 84:



notes

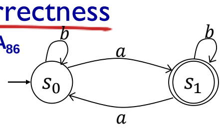
Example 85:



Example 87: FSA correctness

- Revisit Example 86:

FSA$_{86}$

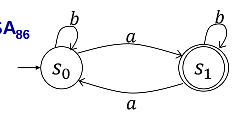


- **FSA$_{86}$** accepts $\boldsymbol{L_{86}}$ = $\{\omega \in \Sigma^* | \omega$ has odd # of a's$\}$
  - **nothing less, nothing more**

Example 87:

- **FSA$_{86}$** accepts $\boldsymbol{L_{86}}$

FSA$_{86}$

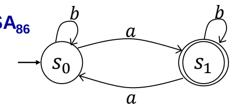


- **FSA$_{86}$** accepts $\boldsymbol{L_{86}}$ = $\{\omega \in \Sigma^* | \omega$ has odd # of a's$\}$
  - **nothing less, nothing more**
  -
  -

Example 87:

❖ **FSA$_{86}$** accepts L_{86}

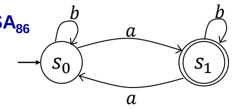
FSA$_{86}$



- to prove it, show that a string ω takes (from the beginning) to s_1 **IFF** ω has an odd # of a's.
- cannot reach s_1, without transitions from previous states
- Hence, we must define & **prove a SI for every state**
-

Example 87:

FSA$_{86}$



❖ **FSA$_{86}$** accepts L_{86}

- $P(\omega)$: $\delta^*(s_0, \omega) = \begin{cases} s_0 & only\ if\ \omega\ has\ even\ \#\ of\ a's \\ s_1 & only\ if\ \omega\ has\ odd\ \#\ of\ a's \end{cases}$

❖ prove SI for all states holds.

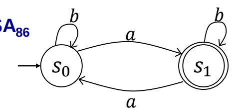
❖ proof by structural induction.

❖ **basis step:**

-

Example 87:

FSA$_{86}$



❖ **FSA$_{86}$** accepts L_{86}

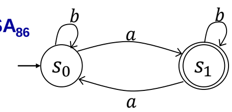
- $P(\omega)$: $\delta^*(s_0, \omega) = \begin{cases} s_0 & only\ if\ \omega\ has\ even\ \#\ of\ a's \\ s_1 & only\ if\ \omega\ has\ odd\ \#\ of\ a's \end{cases}$

❖ **inductive step:**

-

Example 87:

FSA$_{86}$



- case 2: $y = b$
  - left as a practice for you.

formal definition of RL (revisit)

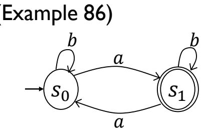
❖ let Σ be the alphabet:

- the empty set, $\varnothing$, is a RL
- the set $\{\varepsilon\}$ is a RL
- for each a $\in \Sigma$, the set $\{a\}$ is a RL
- If L_1 and L_2 are regular languages, then
  - union: $L_1 \cup L_2$ is RL
  - concatenation: $L_1.L_2$ is a RL
  - Kleene star: $L_1{}^*$ is a RL

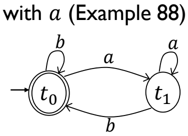
❖ no other RL over Σ exists.

union (closer look)

Let L_{86} be set of all strings with an odd number of a's, (Example 86)



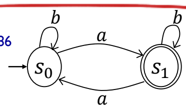
Let L_{88} be set of all strings that do not end with a (Example 88)



❖ Draw the FSA that accepts $L_{86} \cup L_{88}$ (Example 89)

union (closer look)

FSA$_{86}$

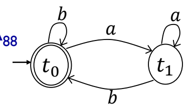


$\Sigma = \{a, b\}$ $\quad q_0 = s_0$

$Q = \{s_0, s_1\}$ $\quad F = \{s_1\}$

$\delta = \cdots$

FSA$_{88}$



$\Sigma = \{a, b\}$ $\quad q_0 = t_0$

$Q = \{t_0, t_1\}$ $\quad F = \{t_0\}$

$\delta = \cdots$

- Devise FSA$_{89}$ that accepts $L_{86} \cup L_{88}$ (Example 89)

notes