Theory of Computation

Lab Session 7

News

Essay

- The essay is to be submitted on April 08th, 2016.
- Live presentation will be on April 23rd, 2016.

Agenda

- Recap.
  - exercises on FSA,
  - operations on FSA,
  - the pumping lemma,
  - exercises on DPDA.
- FSA transducer.
- Acceptance of DPDAs by empty stack.

FSA: Exercises.

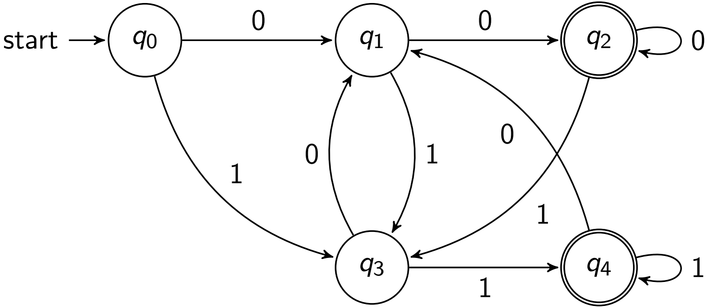
Exercise 1

Build a complete FSA accepting the following language over the alphabet $A = \{0, 1\}$

- $L_0 = \{x \in A^* \mid |x| \geq 2 \ \wedge \text{ the two final symbols of } x \text{ are the same}\}$;

Solution Ex. 1

$L_0 = \{x \in A^* \mid |x| \geq 2 \ \wedge \ \text{the two final symbols of } x \text{ are the same}\}$

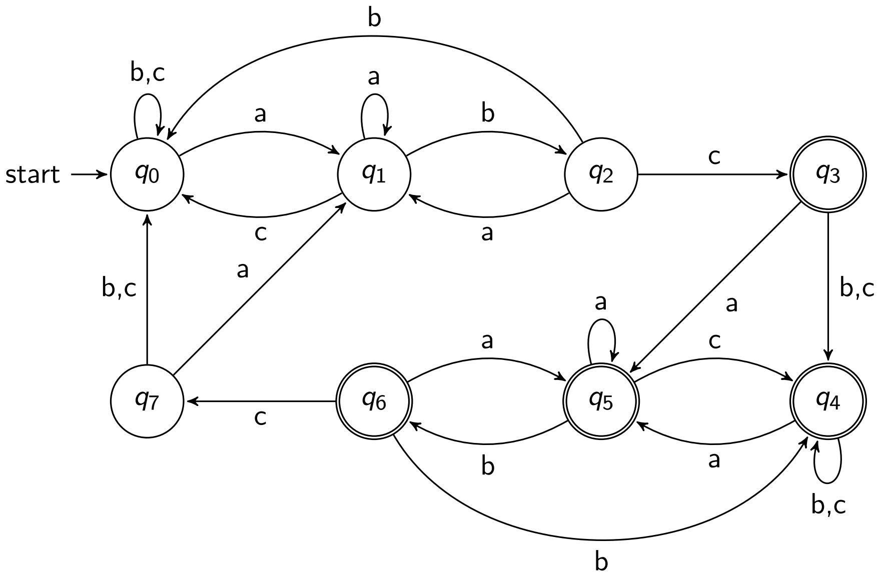


Exercise 2

Build a complete FSA accepting the following language over the alphabet $A = \{a, b, c\}$

- $L_1 = \{x \in A^* \mid \text{the substring } abc \text{ in } x \text{ occurs an odd number of times}\}$;

Solution Ex. 2



FSA: operations.

Exercise 3

Build a complete FSA accepting the following language over the alphabet $A = \{0, 1\}$

- $L_2 = \{x \in A^* \mid |x| \geq 1 \land x \text{ ends with } 10\}$;

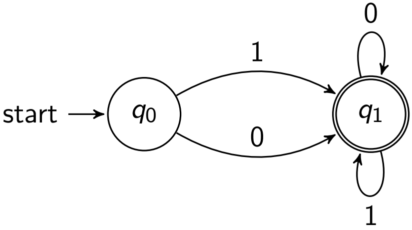
Solution Ex. 3

Let's start by building a complete FSA accepting the following language over the alphabet $A = \{0, 1\}$

- $L_a = \{x \in A^* \mid |x| \geq 1\}$;

Solution Ex. 3

$L_a = \{x \in A^* \mid |x| \geq 1\}$



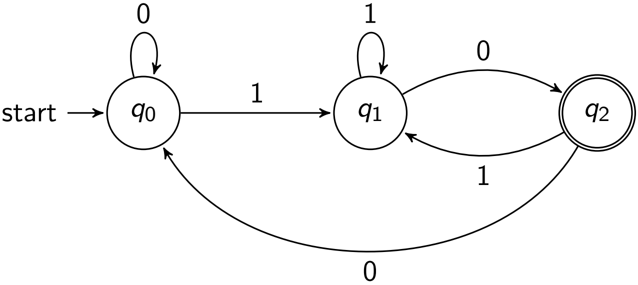
Solution Ex. 3

Now, let's build a complete FSA accepting the following language over the alphabet $A = \{0, 1\}$

- $L_b = \{x \in A^* \mid x \text{ ends with } 10\}$;

Solution Ex. 3

$L_b = \{x \in A^* \mid x \text{ ends with } 10\}$



Solution Ex. 3

Now, let's build a complete FSA that accepts L_a and L_b.

Solution Ex. 3

Now, let's build a complete FSA that accepts L_a and L_b.
To solve it we need to build the intersection of the FSAs.

Reminder: Intersection of FSAs

Suppose $M^1 = (Q^1, A, \delta^1, q_0^1, F^1)$ and $M^2 = (Q^2, A, \delta^2, q_0^2, F^2)$ are finite automata accepting L_1 and L_2, respectively. Let M be the complete FSA $M = (Q, A, \delta, q_0, F)$, where

$$Q = Q^1 \times Q^2$$
$$q_0 = (q_0^1, q_0^2)$$

the transition function δ is defined by the formula

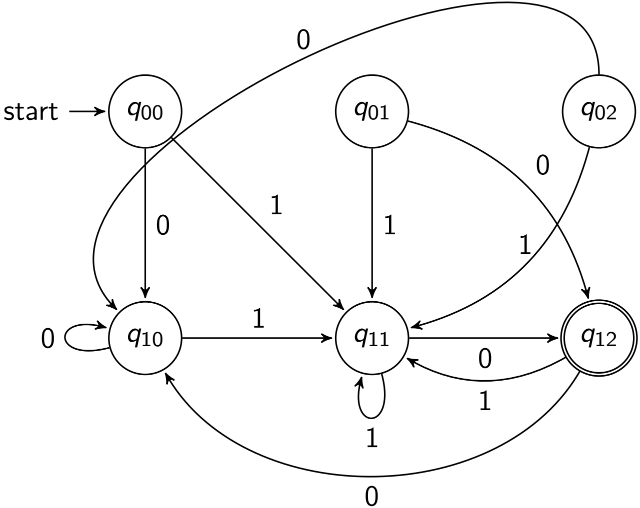
$$\delta((q, p), a) = (\delta^1(q, a), \delta^2(p, a))$$

for every $q \in Q^1$, every $p \in Q^2$, and every $a \in A$. And the set of final states is defined as

$$F = \{(q, p) \mid q \in F^1 \wedge p \in F^2\}$$

M accepts the language $L_1 \cap L_2$.

Solution Ex. 3



FSA: transducers.

Finite State Transducer

Build a complete FSA over the language $A = \{a, b\}$ that accepts only strings ending with the letter b. The FSA will translate the input string where every second symbol a in the input is erased.

Finite State Transducer

Build a complete FSA over the language $A = \{a, b\}$ that accepts only strings ending with the letter b. The FSA will translate the input string where every second symbol a in the input is erased.

- So far we have seen (in the lab. sessions) acceptors FSAs.

Finite State Transducer

Build a complete FSA over the language $A = \{a, b\}$ that accepts only strings ending with the letter b. The FSA will translate the input string where every second symbol a in the input is erased.

- So far we have seen (in the lab. sessions) acceptors FSAs.
- Now, we will see, not just acceptance, but also translation of input string: Finite State Transducer

Finite State Transducer

Finite State Transducer

A Finite State Transducer (FST) is a tuple $\langle Q, I, \delta, q_0, F, O, \eta\rangle$ where

- Q, I, δ, q_0, F: just like acceptors;
- O is the output alphabet;
- $\eta : Q \times I \to O^*$.

Remark:

- the condition for acceptance remains the same as in acceptors;
- the translation is performed only on accepted strings.

FST: an example

Build a complete FST accepting the following language over the alphabet $A = \{0, 1\}$

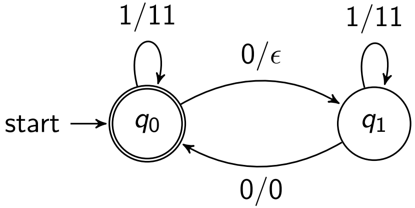
$$L = \{x \in A^* \mid \text{ the number of 0's is even}\}$$

The FST outputs the string obtained by removing every odd occurrence of 0 and doubling every occurrence of 1. Examples of inputs recognised by L and their respective outputs:

- input: 010010, output: 110110
- input: 00, output: 0
- input: 000100011, output: 011001111

FST: an example

$$L = \{x \in A^* \mid \text{ the number of 0's is even}\}$$

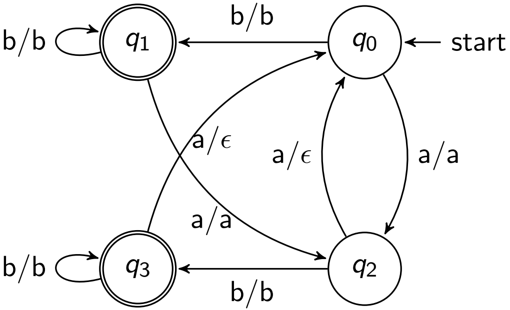


Finite State Transducer

Coming back to the exercise:
Build a complete FSA over the language $A = \{a, b\}$ that accepts only strings ending with the letter b. The FSA will translate the input string where every second symbol a in the input is erased.

Finite State Transducer

Coming back to the exercise:
Build a complete FSA over the language $A = \{a, b\}$ that accepts only strings ending with the letter b. The FSA will translate the input string where every second symbol a in the input is erased.



Pumping lemma.

Pumping Lemma – Exercise

Prove that $L = \{a^{n^2} \mid n \geq 0\}$ is NOT regular.

Solution

Consider $L = \{a^{n^2} \mid n \geq 0\}$.

Solution

Consider $L = \{a^{n^2} \mid n \geq 0\}$.
Suppose there were an FA for L with k states.

Solution

Consider $L = \{a^{n^2} \mid n \geq 0\}$.
Suppose there were an FA for L with k states.
Let $x = a^{k^2-k}$, $y = a^k$, $z = \epsilon$, so $xyz = a^{k^2} \in L$.

Solution

Consider $L = \{a^{n^2} \mid n \geq 0\}$.
Suppose there were an FA for L with k states.
Let $x = a^{k^2-k}$, $y = a^k$, $z = \epsilon$, so $xyz = a^{k^2} \in L$.
Given any splitting of y as uvw with $v \neq \epsilon$, we have $1 \leq |v| \leq k$.

Solution

Consider $L = \{a^{n^2} \mid n \geq 0\}$.
Suppose there were an FA for L with k states.
Let $x = a^{k^2-k}$, $y = a^k$, $z = \epsilon$, so $xyz = a^{k^2} \in L$.
Given any splitting of y as uvw with $v \neq \epsilon$, we have $1 \leq |v| \leq k$.
So taking $i = 2$, we have $xuv^2wz = a^n$ where $k^2 + 1 \leq n \leq k^2 + k$.

Solution

Consider $L = \{a^{n^2} \mid n \geq 0\}$.
Suppose there were an FA for L with k states.
Let $x = a^{k^2-k}$, $y = a^k$, $z = \epsilon$, so $xyz = a^{k^2} \in L$.
Given any splitting of y as uvw with $v \neq \epsilon$, we have $1 \leq |v| \leq k$.
So taking $i = 2$, we have $xuv^2wz = a^n$ where $k^2 + 1 \leq n \leq k^2 + k$.
But there are no perfect squares between k^2 and $k^2 + 2k + 1$, so n is not a perfect square. Thus $xuv^2wz \notin L$.

Solution

Consider $L = \{a^{n^2} \mid n \geq 0\}$.
Suppose there were an FA for L with k states.
Let $x = a^{k^2-k}$, $y = a^k$, $z = \epsilon$, so $xyz = a^{k^2} \in L$.
Given any splitting of y as uvw with $v \neq \epsilon$, we have $1 \leq |v| \leq k$.
So taking $i = 2$, we have $xuv^2wz = a^n$ where $k^2 + 1 \leq n \leq k^2 + k$.
But there are no perfect squares between k^2 and $k^2 + 2k + 1$, so n is not a perfect square. Thus $xuv^2wz \notin L$.
By the pumping lemma, we conclude that L is not regular.

DPDA: exercises.

Exercise

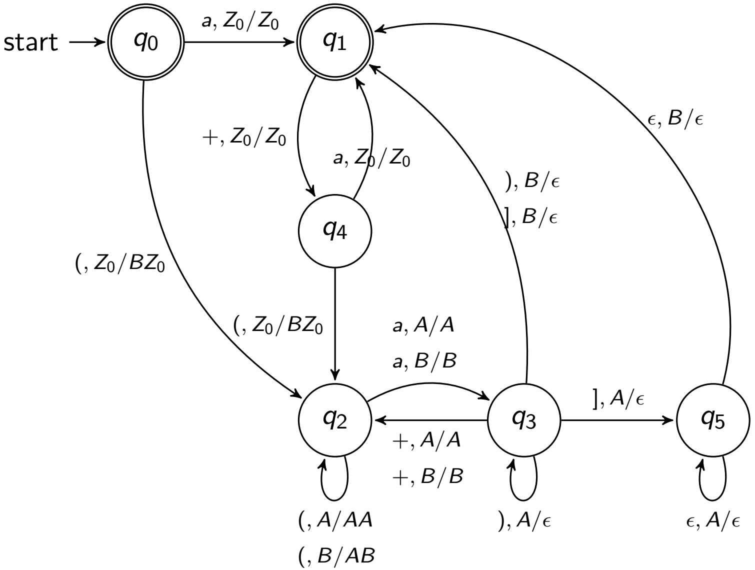
Consider the language described before. Now, suppose that in addition to regular parentheses “(” and “)”, there is also available bracket “]”, which has the effect of closing all open parentheses up to that point.

Examples of strings (not) belonging to the language are:

belongs to the language	**does not belong to the language**
(a + b]	(a + b)]
((a) + (b * c]	((a) + (b * (c)])
(((((((((a + b]	(a]]
(a + b)	a + b]
((a) + (b * (c)]	(a + b

Define a DPDA that recognises this language. For simplicity, consider the following alphabet $I = \{a, (,),], +\}$.

Exercise


start
q_0
q_1
q_2
q_3
q_4
q_5
$a, Z_0/Z_0$
$+, Z_0/Z_0$
$a, Z_0/Z_0$
$(, Z_0/BZ_0$
$(, Z_0/BZ_0$
$\epsilon, B/\epsilon$
$), B/\epsilon$
$], B/\epsilon$
$a, A/A$
$a, B/B$
$+, A/A$
$+, B/B$
$], A/\epsilon$
$(, A/AA$
$(, B/AB$
$), A/\epsilon$
$\epsilon, A/\epsilon$

DPDA: acceptance by empty stack.

Acceptance by a PDA

In a previous lab. session we defined a type of acceptance by a PDA:

Acceptance by final state

Let M be the PDA $\langle Q, I, \Gamma, \delta, q_0, Z_0, F \rangle$, and $x \in I^*$. The string x is accepted by M if

$$(q_0, x, Z_0) \vdash^* (q, \epsilon, \gamma)$$

for some $\gamma \in \Gamma^*$ and some $q \in F$.

Acceptance by a PDA

In a previous lab. session we defined a type of acceptance by a PDA:

Acceptance by final state

Let M be the PDA $\langle Q, I, \Gamma, \delta, q_0, Z_0, F \rangle$, and $x \in I^*$. The string x is accepted by M if

$$(q_0, x, Z_0) \vdash^* (q, \epsilon, \gamma)$$

for some $\gamma \in \Gamma^*$ and some $q \in F$.

Another way of acceptance is by empty stack:

Acceptance by a PDA

Acceptance by empty stack

Let M be the PDA $\langle Q, I, \Gamma, \delta, q_0, Z_0, F \rangle$, and $x \in I^*$. The string x is accepted by M if

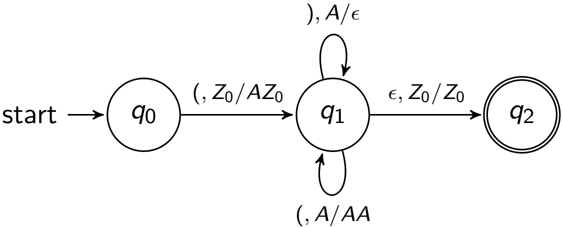
$$(q_0, x, Z_0) \vdash^* (q, \epsilon, \epsilon)$$

for any $q \in Q$.

Exercise

Build a PDA that accepts well-parenthesised input strings (acceptance by final state).

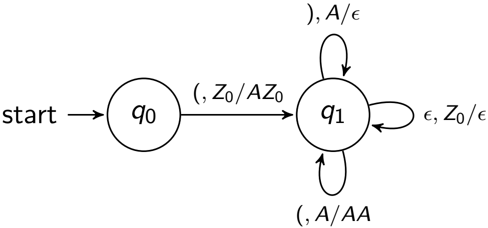
Solution



Exercise

Build a PDA that accepts well-parenthesised input strings (acceptance by empty stack).

Solution



Homework.

Homework

Define a TM that recognises this language:

$$L = \{a^n b^n | n \geq 0\} \cup \{a^n b^{2n} | n \geq 0\}$$