ECE374 Assignment 2

02/01/2023

Group & netid

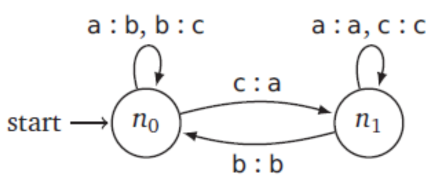
Chen Si *chensi3*

Jie Wang *jiew5*

Shitian Yang *sy39*

Problem 4

4. **Other types of automata:** A *finite-state transducer* (FST) is a type of deterministic finite automaton whose output is a string instead of just *accept* or *reject*. The following is the state diagram of finite state transducer FST_0.



Each transition of an FST is labeled at least an input symbol and an output symbol, separated by a colon (:). There can also be multiple input-output pairs for each transitions, separated by a comma (,). For instance, the transition from n_0 to itself can either take a or b as an input, and outputs b or c respectively.

When an FST computes on an input string $s := \overline{s_0 s_1 \ldots s_{n-1}}$ of length n, it takes the input symbols $s_0, s_1, \ldots, s_{n-1}$ one by one, starting from the starting state, and produces corresponding output symbols. For instance, the input string abccba produces the output string bcacbb, while cbaabc produces abbbca.

(a)

(a) Assume that FST_1 has an input alphabet Σ_1 and an output alphabet Γ_1, give a formal definition of this model and its computation. (Hint: An FST is a 5-tuple with no accepting states. Its transition function is of the form $\delta : Q \times \Sigma \to Q \times \Gamma$.)

Solution:

The finite-state transducer FST_1 could be defined as the following 5-tuple:

$FST_1 = (Q, \Sigma_1, \delta, s, \Gamma_1)$, with

Q: a set of states;

Σ_1: input alphabet, a set of input characters;

Γ_1: output alphabet, a set of output characters;

δ: transition rules that $Q\times\Sigma_1 \rightarrow Q\times\Gamma_1$:

for each rule in the set δ, we have $\delta(q_1, a) = (q_2, b)$, in which $q_1, q_2 \in Q,\ a \in \Sigma_1, b \in \Gamma_1$
s: start state, with $s \in Q$

The computation of FST that takes in a string of length n: $\overline{s_0 s_1 \dots s_{n-1}}$ and output a string of length n is described in the following part:

(1) Initial: input = $\overline{s_0 s_1 \dots s_{n-1}}$, output = “”, state=s
(2) At each step, when at state q_1, for the ith input symbol s_i, follow the transition rules $\delta(q_1, s_i) = (q_2, o_i)$ and move to a new state q_2 while concatenate the ith output symbol o_i on to the output string $\overline{o_0 o_1 \dots o_{n-1}}$.

For instance, the computation details of FST_0 that takes in the string abccba and output the string bcacbb is:

(1) Initially: input= “abccba”, output= “”, state=n_0
(2) input= “a” → $\delta(n_0, a) = (n_0, b)$ → move state to n_0, output= “b”
(2) input= “b” → $\delta(n_0, b) = (n_0, c)$ → move state to n_0, output= “bc”
(2) input= “c” → $\delta(n_0, c) = (n_1, a)$ → move state to n_1, output= “bca”
(2) input= “c” → $\delta(n_1, c) = (n_1, c)$ → move state to n_1, output= “bcac”
(2) input= “b” → $\delta(n_1, b) = (n_0, b)$ → move state to n_0, output= “bcacb”
(2) input= “a” → $\delta(n_0, a) = (n_0, b)$ → move state to n_0, output= “bcacbb”

(b)

(b) Give a formal description of FST_0.

Solution:
The finite-state transducer FST_0 could be described as the following 5-tuple:
$FST_0 = (Q, \Sigma, \delta, s, \Gamma)$, with
$Q = \{n_0, n_1\}$
$\Sigma = \{a, b, c\}$
$\Gamma = \{a, b, c\}$
$s = n_0$
δ: As shown in the following table

	a	b	c
n_0	(n_0, b)	(n_0, c)	(n_1, a)
n_1	(n_1, a)	(n_0, b)	(n_1, c)

in which each row represents the current state and each column represents the input at the current state, and each resulting state (q, k) indicates the next state q and the output k.
e.g. $\delta(n_0, a) = (n_0, b)$ as shown in the table, which indicates taking a as input at state n_0 would result in a new state n_0 and have an output of “b”.

(c)

(c) Give a state diagram of an FST with the following behavior. Its input and output alphabets are {T, F}. Its output string is inverted on the positions with indices divisible by 3 and is identical on all the other positions. For instance, on an input TFTTFTFT it should output FFTFFTTT.

Solution:

The state diagram of the FST required is