30 March 2023

BLG311E – FORMAL LANGUAGES AND AUTOMATA
2022-2023 Spring
Selected Sample Solutions HW1
Araş. Gör. Elif Ak

Questions from Recitation-1

9) Consider the following grammar

$$< S >::= a < S > b$$
$$< S >::= aa < S > b$$
$$< S >::= ab$$

a) Show that the grammar is ambiguous by providing an example over the shortest possible word.
b) Provide an equivalent unambiguous grammar heuristically. (Hint: From scratch)

Solution:

a) For $aaaabbb$

$$< S > \to a < S > b \to aaa < S > bb \to aaaabbb$$
$$< S > \to aa < S > b \to aaa < S > bb \to aaaabbb$$

b) This grammar represents the language $a^m b^n$ where $2n > m \geq n > 0$. An unambiguous grammar for this language may be as follows:

$$< S >::= aa < S > b \mid < A >$$

$$< A >::= a < A > b \mid ab$$

<u>Questions from HW-1</u>

1) Consider the following requirement set for a dumb turnstile machine:

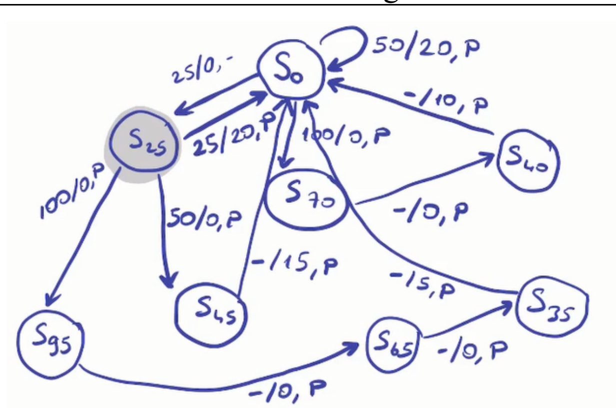
- Turnstile machine is "locked" by default
- User can push one of the following coins at once to the machine: 25, 50, 100
- Machine checks the coins presented to the machine so far and
  - either waits for additional coins if the amount of credits is not enough,
  - or unlocks to let **the maximum amount of people possible** to pass one by one.
  - Once total credits exceed the amount necessary to let one-person pass, machine lets the maximum amount of people pass, returns change and goes back to its initial state
- The amount of credits required for a single person to pass is 30
- Machine will not accept any coins while it is unlocked.

Examples:

- If the user initially inputs 25, machine will stay locked and wait for additional credits
- If the user initially inputs 100, machine will let 3 people pass, return 10 with the last person (warning: it's not possible to pay 100 and let a single person pass in this machine)

Define and explain each element in the following sets and draw the diagram for this machine using Mealy Model.

Solution:

<table>
<tr><th>Input alphabet of your design</th><th>Output alphabet of your design</th></tr>
<tr><td>$I = \{25,50,100\}$</td><td>$O_1 = \{P, -\}$
$O_2 = \{0,5,10,15,20,-\}$
$O \subset O_1 \times O_2$</td></tr>
<tr><td colspan="2">State machine diagram</td></tr>
<tr><td colspan="2">
</td></tr>
</table>

4) Consider the inductive definition of the reverse operation on a string.

$|w| = 0 \Rightarrow w^R = w = \Lambda$
$|w| = n + 1 \wedge n \in \mathbb{N} \Rightarrow |u| = n \wedge\ a \in \Sigma \wedge w = ua \Rightarrow w^R = au^R$
Using the definition above, show that $\left(w^i\right)^R = (w^R)^i$ where i is a natural number.

Solution:

This definition can be generalized for concatenation of two strings x and y:

$w = xy$

$|y| = m \Rightarrow y = y_1 y_2 \dots y_m, y_{1:m} \in \Sigma$

$w^R = (xy)^R = (xy_1y_2 \dots y_m)^R = y_m(xy_1y_2 \dots y_{m-1})^R = y_m y_{m-1}(xy_1y_2 \dots y_{m-2})^R = \cdots$

$= y_m y_{m-1} \dots y_1 x^R = y^R x^R$

Proof by induction

True for $i = 0$ as $(w^0)^R = (\Lambda)^R = \Lambda = (w^R)^0$

Assuming to be true for $i = n$ as $(w^n)^R = (w^R)^n$

For $i = n + 1$:

$(w^{n+1})^R = (w^n w)^R$

Using the generalization above: $(w^n w)^R = w^R (w^n)^R$

Using the assumption for $i = n$: $w^R (w^n)^R = w^R (w^R)^n = (w^R)^{n+1}$

8) Design context-free grammars for the following languages:

a) The $\{0^n 1^n \mid n \geq 1\}$, that is, the set of all strings of one or more 0's followed by an equal number of 1's.

b) The $\{a^i b^j c^k \mid i \neq j \text{ or } j \neq k\}$, that is, the set of strings of a's followed by b's followed by c's, such that there are either a different number of a's and b's or a different number of b's and c's, or both.

Solution:

a) S→0S1 | 01

b)

S→AB | BC | AC | DC | AE

A→aA|a

B→bB|b

C→cC|c

D→aDb|A|B

E→bEc|B|C

To understand how this grammar works, observe the following:

(i) A generates one or more a's.

(ii) B generates one or more b's.

(iii) C generates one or more c's.

(iv) D first generates an equal number of a's and b's, then produces either one or more a's (via A) or one or more b's (via B).

(v) Similarly,E generates unequal numbers of b's then c's.