Subiecte 1

June 13, 2023

Exercise 1 (25p)**.** *Imagine a DFA with an attached list (LA). You can think of a PDA but with a list (the order of elements does not matter, you can see it as a set) instead of a stack. An LA should function in the following manner:*

- *If the current state is q and the input symbol is a and the list contain symbol b then LA will transition to state r will remove symbol c from list and add a symbol d to the list.*
- *If the current state is q and the input symbol is a and the list does not contain symbol b then LA will transition to state r will remove symbol c from list and add a symbol d to the list.*
- *ϵ transition are not allowed however operation with the attached lists allows ϵ (e.g it is not necessary to remove or add a symbol to the list)*

Below are few examples of possible transitions:

- *$(q_1, a_1, s_1) \to (q_2, a_2, a_3)$ means that if our LA is in state q_1 and reads a_1 from the input string and attached list contains s_1 then it will move to state q_2, removes a_2 from the list (if exists) and adds a_3 to the list.*
- *$(q_1, a_1, \epsilon) \to (q_2, a_2, a_3)$ means that if our LA is in state q_1 and reads a_1 from the input string it will move to state q_2, removes a_2 from the list (if exists) and adds a_3 to the list.*
- *$(q_1, a_1, \epsilon) \to (q_2, \epsilon, \epsilon)$ means that if our LA is in state q_1 and reads a_1 from the input string it will move to state q_2 and it does not remove or add any symbol to LA list.*

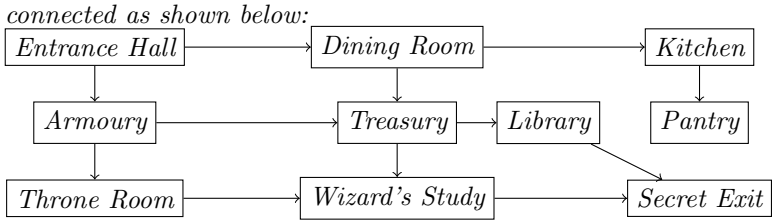
You need to:

1. *Give a formal definition of the LA.*
2. *Give a clear explanation of computation for the LA. Include a formal description of computation.*

Exercise 2 (25p)**.** *Implement a library/program (in a programming language of your choice) to load and validate a configuration file of a LA (see Ex 1). Also implement a simulator/emulator for this LA.*

Exercise 3 (50p). *Build the following Text-Based Adventure Game:*
Game Description: *In this text-based adventure game, you play the role of an adventurer trapped in the mystifying "Castle of Illusions". Each room within the castle contains unique challenges and rewards. Your ultimate goal is to discover the secret exit that leads to the outside world. However, navigating through the castle isn't as easy as it may seem - certain rooms are accessible only if you possess specific items.*

Game Map: *Your adventure unfolds across the following rooms, each interconnected as shown below:*



Rooms Description:

- *Entrance Hall: The grand foyer of the Castle of Illusions.*
- *Dining Room: A room with a large table filled with an everlasting feast.*
- *Kitchen: A room packed with peculiar ingredients.*
- *Armoury: A chamber filled with antiquated weapons and armour.*
- *Treasury: A glittering room overflowing with gold and gemstones.*
- *Library: A vast repository of ancient and enchanted texts.*
- *Pantry: A storage area for the Kitchen.*
- *Throne Room: The command center of the castle.*
- *Wizard's Study: A room teeming with mystical artifacts.*
- *Secret Exit: The hidden passage that leads out of the Castle of Illusions.*

Game Commands: *Here are the commands that the player can use to interact with the game world:*

- *go [room name]: Moves the player to the specified room if it is adjacent and the necessary conditions (e.g. having a certain item) are met.*
- *look: Provides a description of the current room and the adjacent rooms.*
- *inventory: Shows the items currently in the player's possession.*
- *take [item]: Allows the player to pick up an item found in a room.*

- *drop [item]: Allows the player to drop an item from their inventory into the current room.*

Command Examples:

- *go Kitchen: This command would move the player from the current room to the Kitchen, given that the Kitchen is adjacent and the player meets the conditions to enter (e.g., possessing the 'chef's hat' item).*
- *take key: This command would add the 'key' item to the player's inventory, if the item is present in the current room.*

Initial Configuration: *In the initial setup, the following items are located in the corresponding rooms:*

Room	**Item**	**Used for Accessing**
Entrance Hall	key	Armoury
Dining Room	invitation, chef's hat	Entrance Hall, Kitchen
Kitchen	spoon	Pantry
Armoury	sword, crown	Treasury, Throne Room
Treasury	ancient coin	Library
Library	spell book	Wizard's Study
Pantry	-	-
Throne Room	-	-
Wizard's Study	magic wand	Secret Exit
Secret Exit	-	-

Table 1: Items and Their Uses in Each Room

Remember, each room requires a specific item to enter. For example, to enter the Armoury from the Entrance Hall, the player needs the 'key'. These conditions create a logical progression and challenge in the game, driving the player to explore, strategize, and interact with their environment.

Requirements:

- *LA automata (see Ex 2) must be used for game mechanics*
- *Commands must be checked for correctness by constructing the CFG and using the equivalent PDA to check if a command is accepted or not*