Universidad de los Andes

ISIS-1106 Lenguajes y máquinas
Project 2
Date: March 27, 2020

As we have seen in projects 0 and 1, the project work for the course will consist of managing and operating a language to program a robot in a two-dimensional world. The robot is able to move in the world (by a matrix); the robot moves from cell to cell. Cells are indexed by rows and columns. The top left cell is indexed as (1,1). North is top; West is left. Additionally, the robot interacts (picks and puts down) with two different types of objects (chips and balloons).

Robot Routine Description

Let us recall the description of a routine for the robot:

- A routine is the keyword ROBOT_R possibly followed by a declaration of variables, followed by a block of instructions.
- A declaration of variables is the keyword VARS followed by a list of names separated by commas. A name is a string of alphanumeric characters that begins with a letter.
- A block of instructions is a sequence of instructions separated by semicolons preceded by the keyword BEGIN and followed by the keyword END.
- An instruction can be a command or a control structure.
  - A command can be any one of the following:
    * **assign**: `n` **to**: `name` where `name` is a variable's name and `n` is a number. The result of this instruction is to assign the value of the number to the variable.
    * **move**: `n` – where `n` is a number or a variable. The robot should move n steps forward.
    * **turn**: `D` – where `D` can be left, right, or around. The robot should turn 90 degrees in the direction of the parameter.
    * **face**: `O` – where `O` can be north, south, east or west. The robot should turn so that it ends up facing direction `O`.
    * **put**: `n` **of**: `X` – where `X` corresponds to either Balloons or Chips, and `n` is a number or a variable. The Robot should put `n` `X`'s.
    * **pick**: `n` **of**: `X` – where `X` is Balloons or Chips and `n` is a number or a variable. The robot should pick `n` `X`'s.
    * **move**: `n` **toThe**:`D` – where `n` is a number or a variable. `D` is a direction, either front, right, left, back. The robot should move `n` positions to the front, to the left, the right or back and end up facing the same direction as it started.
    * **move**: `n` **inDir**: `O` – here `n` is a number or a variable. `O` is north, south, west, or east. The robot should face `O` and then move `n` steps.

    * `Skip`
  - A control structure can be:

    Conditional: **if**: `condtion` **then**: `Block1` **else**: `Block2` – Executes `Block1` if `condition` is true and `Block2` if `condition` is false.

    Loop: **while**: `condition` **do**: `Block` – Executes `Block` while `condition` is true.

    RepeatTimes: **repeat**: `Block` **times**:`n` – Executes `Block` `n` times, where `n` is a variable or a number.
  - A condition can be:
    * **facing**: `O` – where `O` is one of: north, south, east, or west
    * **canPut**: `n` **of**:`X` – where `X` can be chips or balloons, and `n` is a number or a variable
    * **canPick**: `n` **of**: `X` – where `X` can be chips or balloons, and `n` is a number or a variable
    * **canMove**:`D` – where `D` is one of: north, south, east, or west
    * **not**: `cond` – where `cond` is a condition

Spaces, and tabulators are separators.

Task 1. For this project we will use GOLD to perform a lexical analysis of routines for the Robot.

In particular we will use finite automata (Mealy and Moore machines, a.k.a finite transducers) for the lexical analysis of the Robot language.

For this project, you will only have to verify lexical correctness. You don't have to determine if a variable is undefined.

A finite transducer, reads the text of the routine and tokenizes it producing a string of tokens: each token must be represented by a single character.

As an example, consider the following program.

```
ROBOT_R
VARS a, b
BEGIN
assign: 3 to: a
assign: 1 to: b
move: a
turn: left
move: 1
END
```

would be translated to (for example, you need to figure out your own encoding):

```
RZVCVBA(ICV)A(ICV)M(I)T(L)M(I)E
```

We include a Gold project that performs lexical analysis for a very simple language. A ppt file is found in the Doc folder of the Gold project. This presentation provides a guide to implementing lexers with GOLD.

You should hand in the Lexer.gold file for the Robot Routines and a document in which you explain the encoding that you used.