

Deliverable 3: YACC grammar

Problem description:

After translating to tokens, the now the job of the parser is to take those tokens and write into the *.asm* file the valid syntax for the CPU to use. The grammar provided allows the user to define instructions for the robot to follow; it allows the user to control the robot's movements, rotation and position on the grid.

Context Free Grammar

```
<STATEMENT_LIST> → <STATEMENT> | <STATEMENT_LIST> <STATEMENT>
<STATEMENT> → <NOUN_PHRASE> <ROBOT_COMMAND>
<ROBOT_COMMAND> → <ACTION> | <ACTION> <CONJUNCTION> <ACTION> | <ACTION> <CONJUNCTION>
<ADVERB> <ACTION>
<NOUN_PHRASE> → <NOUN> <KIND_WORD>
<ACTION> → <MOVEMENT> | <ROTATION> | <ACTION> <ADVERB> <ACTION> | <ACTION> <CONJUNCTION>
<ACTION>
<MOVEMENT> → <POSITION> <NUMBER> <BLOCKS> <DIRECTION> | <POSITION> <NUMBER> <BLOCKS> |
<POSITION> <BLOCKS> <NUMBER> <DIRECTION>
<ROTATION> → <ORIENTATION> <ANGLE> <DEGREES> | <ORIENTATION> <DIRECTION>
```

Valid Sentences

With the tokens already defined, examples of valid sentences are as follows:

```
* Robot please move 2 blocks ahead
* Robot please move 3 blocks ahead and then turn 90 degrees, then move 2
blocks
```

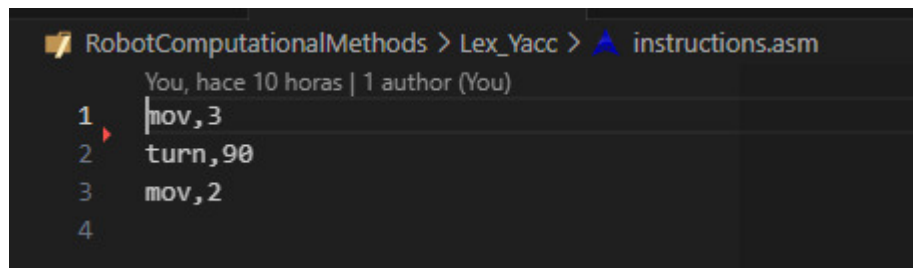
Examples of invalid sentences:

```
* Robot moves 2 blocks
* Robot moves 2 blocks quickly
* Move 2 blocks right now
* Robot 2 blocks moves
* Moves Robot 2 blocks and turns 89 degrees
```

Run example:

Input:

```
robot please move 3 blocks ahead and then turn 90 degrees, then move 2 blocks
```

OUTPUT instructions.asm:

```
RobotComputationalMethods > Lex_Yacc > instructions.asm
You, hace 10 horas | 1 author (You)
1 | mov,3
2 | turn,90
3 | mov,2
4 |
```