2048 Game Language

CS F363 Compiler Construction, Sem II 2020-21

Dependencies

```
pip install sly==0.4
```

Introduction

The game elements, its lexicon, and grammatical demands of its programming language are given. It is a 2048-game family.

Here, variations on the original 2048 game are to be also provided for. The variations are:

- Allowing subtraction, multiplication and division in addition to the plain doubling operation at tile mergers.
- The operations:
    - **Moves**: Add/Subtract/Multiply/Divide Left/Right/Up/Down
    - **Assignment**: Assign ≪ value ≫ to ≪ x ≫,≪ y ≫
    - **Var** ≪ varname ≫ is ≪ x ≫,≪ y ≫
    - **Query**: Value in ≪ x ≫,≪ y ≫
- Remaining token types will be identifiers, numbers, and punctuation symbols (,.?). Com- mands must end with a full-stop. Co-ordinates must be separated by a comma, and optional whitespace.

Interpretation

This is a fun assignment to learn about compilers and their working through a game, the objective is to make a parser-translator (that mean a complete syntax-directed translation scheme) for a game programming language.

1. Tiles with 0 values are considered to be empty and not printed.
2. Each command will be ended by a full stop, there will be only one line per command and lexer will ignore the whitespaces and tabs.
3. The user is allowed to change the size, shape of the game grid prior to starting the engine.
4. Compression of tiles merging is upto the user.

Assumptions

1. No Foreign characters should be allowed, even though '?' is unused we keep it as a part of the lexer but throw wrong reference error on encountering it, additionally '-' has been allowed for negative numbers (Although they are cosidered illegal for index / values)
2. The user may wish to check the output of a number, and just pass a number say command is

```
2048 >>> 7 .
7
```

3. Queries may be recursive

```
ASSIGN VALUE IN VALUE IN 1 , 1 , 2 TO VALUE IN 1 , 1 .
#Assigns value in (M(1,1), 2) to M(1, 1) also prints M(1,1), (M(1,1), 2)
```

4. On moving if intermediate empty tiles are to be compressed or not, user selects the option before-hand.
5. Game runs even after being over, negative tiles are invalid, and ends only on encountering EOF -> ctrl + d

Handled Basic Errors

1. Foreign Characters, Misplaced ? , will throw errors
2. If fullstop is encountered in the middle its considered to be an error (no trimming/splitting to multiple commands is done here)
    - If wished it can be done by updating the preprocessor
3. Two tiles cannot have same variable names.
4. Negative Values cannot be assigned to tiles.
5. Empty tiles cannot be named or queried.
6. Syntax Errors

Advanced Error Handling

1. On encountering each of the above errors a proper message conveying the addressed syntax/runti,e error is displayed in the interactive shell.
2. Recursive Queries will be handled. Non-Numeric Expressions as Indexes, Assigning Values will be thrown proper messages with the printed issue.
3. Every Syntax Error that has one keyword token wrong will have a guided error message explaining the error and suggesting the correct alternative as well.
    - Example:

```
 2048 >>> ASSIGN 3 IN 2, 2 .
STDOUT: Syntax Error: Assign Value cannot be followed by IN, try TO
STDERR: -1

2048 >>> UP UP .
STDOUT: Syntax Error: DIRECTION cannot be followed by UP, try ADD/MULTIPLY/SUBTRACT/DIVIDE
STDERR: -1
```

Instructions To Run

Running in debug mode

```
 make errfile
#once the output is checked
make clean
```

For redirecting stderr to a file.

```
 make console
#once the output is checked
make clean
```

Running with specifications

```
 import sys
from game import Board
from lexer import Lexer2048
from parser import Parser2048

board = Board(size=(4, 4))
lexer, parser = Lexer2048(), Parser2048(fmap=board.fmap)
print("Welcome to the 2048 Gaming Language, Below is the Board. Happy Coding!")

while True:
    try:
        inp = input("2048 >>>")
        command = lexer.err(inp)
        out = parser.parse(lexer.tokenize(command))
        board.eout()

    except EOFError:
        exit()

    except Exception as E:
        print(str(E))
        print("-1", file=sys.stderr)
```