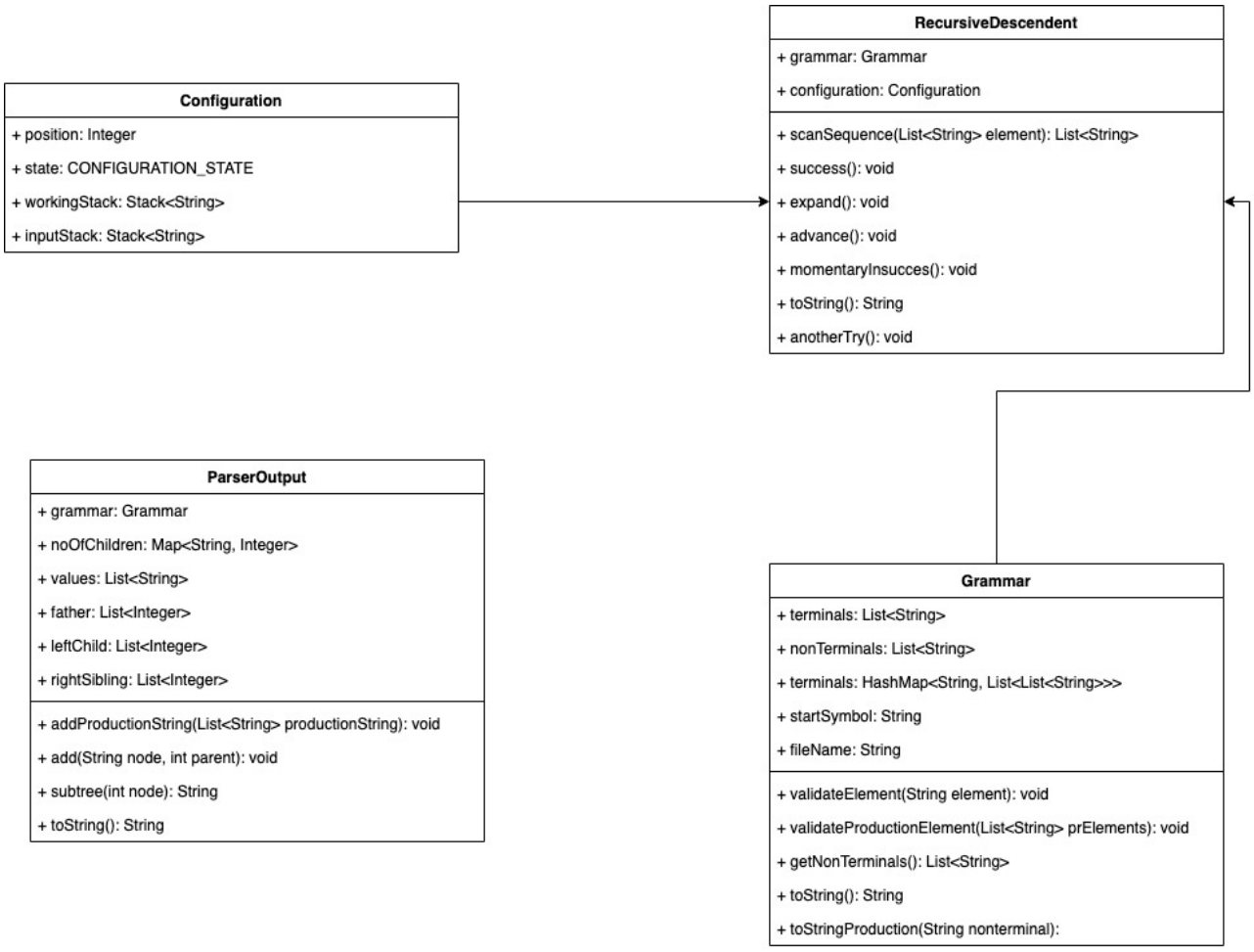
Documentation

Recursive Descendent Parser



Configuration

(s, i, α, β) :

- s - the current state - represented as an enum
- i - the current index
- α - the working stack - represented as a list of strings - contains a value and a production number in case it's a nonterminal
- β - the input stack - represented as a list of strings

RecursiveDescendent
Class Structure

The class contains the following attributes:

- Grammar: the grammar specific to the parser
- Configuration: the current configuration of the parsing

Method

- scanSequence - wraps the descendent recursive algorithm, using the correct method corresponding to the move that is required. It stops when the current configuration is equivalent to an error or a success.
- Specific methods for each move:
    - State is normal:
        - if the head of the input stack is a nonterminal: **EXPAND**
        - if the head of the input stack equals ε and the current index of the configuration equals the size of input + 1: **SUCCESS**
        - if the head of the input stack equals the current symbol in the input and the current index is less or equal than the size of the input: **ADVANCE**
        - if the head of the input stack is a terminal: **MOMENTARY INSUCCESS**
    - State is back:
        - if the head of the working stack is not a terminal: **ANOTHER TRY**
        - if the head of the working stack is a nonterminal: **BACK**

Parser Output

The output file has the following structure:

- Father: list with the index of the parent of each node (or 0 if it's the root)
- Left child: list with the index of the left child for each node (or -1 if it does not exist)
- Right siblings: list with the index of the right sibling of each node (or -1 if it does not exist)
- Tree

Grammar - input file:

G1:

S

a,b,c

S

S -> a S b S | a S | c

Output for grammar G1 with input sequence: aacbc

Values: [ε, S#1, a, S#2, a, S#3, c, b, S#3, c]

Father: [0, 1, 1, 3, 3, 5, 1, 1, 8]

Left child: [1, 2, -1, 4, -1, 6, -1, -1, 9, -1]

Right sibling: [-1, -1, 3, 7, 5, -1, -1, 8, -1, -1]

```
ε
    \-- S
        |-- a
        |-- S
        |       |-- a
        |       \-- S
        |               \-- c
        |-- b
        \-- S
                \-- c
```