Running from command line

The program can be run via the jarfile CYK.jar with the following synopsis :

java -jar CYK.jar [*tests* | *grammar*] [*BU* | *TD* | *naive* | *linear*] "testString"

If the first argument is set to “tests” the given example tests will be run. If the first argument is not “tests” it will try to open it as a file which the grammar will be constructed from. The second argument specifies which parser to use, can be either “BU”, “TD” or “naive”. The last argument is the string which will be parsed.

The JAR file can be found in **out/artifacts/CYK_jar/CYK.jar**.

The grammar files dyck, stupid-grammar and linearGrammar have to be present in the JAR folder for the tests to be able to run.

If the program is run with the linear parser then the grammar has to be linear for it to work.

Class descriptions

Grammar
This class represents the grammar. Its constructor takes a File that reads a grammar with a space separated syntax:

S AB
A CS
C a

The grammar stores unary and binary rules in two separate two-dimensional arrays. The number of columns in the array is 3, the first column represents the left side, second and last column is the right hand side of the rule. Each row in the array is a rule. It also translates each rule into an integer identifier starting from 0. The mappings are stored in a translation table.

Parser
The parser class contains methods for each parsing method parseNaive, parseBU, parseTD and parseLinear. It also holds a counter that is reset each time a parser is called. The counter is incremented for each iteration in each parse. It also holds a timer that stores the time for when the methods are called.

StringEnumerator
This class can take a start string and enumerate strings of different variants and increments. The string to be appended each call is set, if the string is to be appended on the start, middle or end is also specified.