

Class 6 - Worksheet

Project 1

[Project 1](#) is due at the beginning of class on **Friday** (if you are not able to make it to class on Friday, you can drop it off at my office anytime before then). Please follow the submission directions carefully (advice on how to print Python files has been added to the [Project 1 page](#)).

Replacement Grammars and Rules of Evaluation

For each of the following grammars: (a) describe the language that can be produced starting from S ; (b) if possible, write a simpler BNF grammar (fewer rules) that describes the same language; if not, explain why; and (c) determine rules of evaluation so that the value is interpreted as a binary number (with the rightmost bit being the least significant one). So, **1** should have value 1 , **110** should have value 6 , and **10100** should have value 20 (note that not all of these strings can be produced by all of the grammars below, but your evaluation rules should work for all strings that can be produced).

1.

$S ::= 0 S$

$S ::= S 1$

$S ::= \epsilon$

2.

$S ::= S 0$

$S ::= S 1$

$S ::= \epsilon$

3.

$S ::= S S S$

$S ::= T T$

$T ::= 0$

$T ::= 1$

Here is a fragment of the Python language described by the grammar for test conditions (like in an **if** statement):

```

Test ::= OrTest
OrTest ::= AndTest OptOrTests
OptOrTests ::= or AndTest OptOrTests
OptOrTests ::=  $\epsilon$ 
AndTest ::= NotTest OptAndTests
OptAndTests ::= and NotTest OptAndTests
OptAndTests ::=  $\epsilon$ 
NotTest ::= not NotTest
NotTest ::= Comparison
Comparison ::= Expression OptComparisons
Expression ::= NumberLiteral
Expression ::= True | False
OptComparisons ::= ComparisonOperator Expression OptComparisons
OptComparisons ::=  $\epsilon$ 
ComparisonOperator ::= < | > | =

```

Assume that *NumberLiteral* is defined as in the previous class (that is, numbers like **7** and **1120**).

4. From the grammar, determine which of the following are valid *Test* expressions in Python:

```

3 < 4
3 and 4 or 5
3 < 4 > 5
3 < 4 and 4 > 5 or 4 < 5
not not not not not not False
not not not
not 3 and 4

```

You can try evaluating them in the PYthon interpreter to check your answers and see how they evaluate.

5. (You're not expected to finish this, but see how far you can get.) Develop rules of evaluation for the grammar above that matches how things are interpreted by the Python interpreter.