

Programming Languages G22.2110 Summer 2007 hw01

Assigned Th 5/17/2007, due We 5/23/2007 at 1pm.

How to Submit Homework Assignments

Email your answers, in either plain text format or as pdf, to **Abhijit Guria** <guria@cs.nyu.edu>. Assignments are due on Wednesdays at 1pm. This deadline will be strictly enforced.

Reading Assignments

- For lecture on 5/17/2007: Scott 1.2, 1.3, 2.1
 - For lecture on 5/24/2007: Scott 6.0, 6.1.0-6.1.1; van Rossum 1-5 (<http://docs.python.org/tut/tut.html>)
-

Homework Assignments

1. Syntax (18 = 4 + 4 + 5 + 5 points)

Consider the following regular expression:

$(XY)^*XY^*$

- 1a. (4 points) Give two examples of words matched by the regular expressions.
 - 1b. (4 points) Give two examples of words for which the regular expression does not match.
 - 1c. (5 points) Give a context-free grammar in BNF that matches the same language as the regular expression. Do not use Kleene star or meta-level parentheses in your grammar.
 - 1d. (5 points) Give the rightmost derivation for the word XYXY using your grammar from Question 1c.
2. More syntax (20 = 5 + 5 + 5 + 5 points)
Consider the following context-free grammar:

```
start symbol <s>
<s> ::= <f> | <f> '$' <s>
<f> ::= <p> | <f> '#' <p>
<p> ::= '(' <s> ')' | '0' | '1'
```

- 2a. (5 points) Give the parse tree for the word $0\#(0\$1\$0)\#1$.

- 2b. (5 points) Assume the symbols '\$' and '#' are operators of a programming language. What is their precedence, as given by the grammar?
- 2c. (5 points) What is the associativity of operators \$ and # in the grammar?
- 2d. (5 points) Modify the grammar by adding a unary prefix operator @ that has a higher precedence than both \$ and #.
3. Python (0 points)
Start teaching yourself Python by doing the following:
- 3a. If possible, find peers (other students who want to learn Python together with you) and gurus (people who already know Python, whom you can ask questions when you get stuck).
- 3b. Make sure you have access to a Python interpreter in your preferred working environment. If you are lucky, it may already be installed. Otherwise, you can get it from the Python webpage, or through one of the common GNU package managers, such as Cygwin for Windows or Fink for Mac OS.
- 3c. Read the first 5 sections of the Python tutorial (see reading assignment above). Along the way, try things out with the Python interpreter that you installed in Step b.
- 3d. Familiarize yourself with the structure of the online Python documentation, so you can find information quickly when you need it. In particular, find the URLs for the Python library reference and the Python language reference.
4. More Python (12 = 3 + 3 + 3 + 3 points)
Continue teaching yourself Python by reading the example code from <http://www.cs.nyu.edu/courses/summer07/G22.2110-001/hw01-py-example.txt>
- 4a. (3 points) What does the script print for the example data at its top?
- 4b. (3 points) Add the line `print str(table)` to the bottom of the script. Based on the output, describe briefly what function `read_table` does.
- 4c. (3 points) Try the regular expression stand-alone in the interactive Python interpreter:
- ```
>>> import re
>>> re.findall(r'^[,]*|"[^"]+"', 'a,"b,c",d')
```
- Based on the output, describe briefly what the regular expression does.
- 4d. (3 points) Try the string interpolation stand-alone in the Python interpreter:
- ```
>>> '%-*s' % (3, 'a')
>>> '%-*s' % (4, 'aa')
```
- Based on the output, describe briefly what the format string does.

<http://www.cs.nyu.edu/courses/summer07/G22.2110-001/hw01.pdf>

Total points: 50.