

Homework 1

Please hand your solution to the instructor before class on the day of the deadline or email them to the instructor with CC to rh1424@nyu.edu.

The deadline for Homework 1 is September 12.

Problem 1 Regular Expressions (10 Points)

- PLP, pp. 102-103: 2.1 a,b,f (**6 Points**)

Write regular expressions to capture the following.

- (a) Strings in C. These are delimited by double quotes ("), and may not contain newline characters. They may contain double-quote or backslash characters if and only if those characters are “escaped” by a preceding backslash. You may find it helpful to introduce shorthand notation to represent any character that is *not* a member of a small specified set.
- (b) Comments in Pascal. These are delimited by (* and *) or by { and }.
- (f) Financial quantities in American notation. These have a leading dollar sign (\$), an optional string of asterisks (*—used on checks to discourage fraud), a string of decimal digits, and an optional fractional part consisting of a decimal point (.) and two decimal digits. The string of digits to the left of the decimal point may consist of a single zero (0). Otherwise it must not start with a zero. If there are more than three digits to the left of the decimal point, groups of three (counting from the right) must be separated by commas (,). Example `$$*2,345.67`. (Feel free to use “productions” to define abbreviations, so long as the language remains regular.)

- PLP, pp. 103: 2.3 (**4 Points**)

Build a regular expression that captures all nonempty sequences of letters other than `file`, `for`, and `from`. For notational convenience, you may assume the existence of a **not** operator that takes a set of letters as argument and matches any *other* letter. Comment on the practicality of constructing a regular expression for all sequences of letters other than the keywords of a large programming language.

Problem 2 Context-Free Grammars (10 Points)

- PLP, pp. 105: 2.12 a,b (**2 + 3 Points**)

Consider the following grammar with start symbol G , non-terminals $\{G, S, M, A, E, B\}$,

terminals $\{\$, a, b\}$, and productions:

$$\begin{aligned} G &::= S\$\$ \\ S &::= AM \\ M &::= S \mid \epsilon \\ A &::= aE \mid bAA \\ E &::= aB \mid bA \mid \epsilon \\ B &::= bE \mid aBB \end{aligned}$$

- (a) Describe in English the language that the grammar generates.
- (b) Show a parse tree for the string $abaa\$\$$.

• PLP, pp. 108: 2.27 (5 Points)

The dangling else problem of Pascal is not shared by Algol 60. To avoid ambiguity regarding which then is matched by an else, Algol 60 prohibits if statements immediately inside a then clause. The Pascal fragment

```
if C1 then if C2 then S1 else S2
```

must be written either

```
if C1 then begin if C2 then S1 end else S2
```

or

```
if C1 then begin if C2 then S1 else S2 end
```

in Algol 60. Show how to write a grammar for conditional statements that enforces this rule. (Hint: you will want to distinguish in your grammar between conditional statements and nonconditional statements; some contexts will accept either, some only the latter.)