Lab 002: Defining Languages and Parsing

Dr. Matthew Pike

9th October 2018

Please submit a PDF document containing your answers via the associated Moodle assignment by 4PM today (day of lab). This work is not assessed, but does contribute towards your attendance mark for this lab (you must also sign the register).

Backus-Naur Form (BNF)

Q1) Given the following BNF definition:

```
list    =       "{" items   "}"
items   =   item    |   item    "," items
item    =   text    |   text    list
```

Assume that `text` is an arbitrary (possibly empty) sequence of characters not containing { } ;

Q1A) Which of the following lists are valid:

1. {Languages {French,English,Swiss German}}
2. {Food {Health Junk}}
3. {Darth {Vader} {Maul}}
4. Table of Contents {Section 1, Section 2 {Subsection 2.1}}

Q1B) What is the shortest possible valid list?

Q2. Write a simple BNF for the following alphabet:

```
{A,B}
```

Write a BNF that enforces the following rules:

```
A can only appear in pairs like AA.
```

An example of valid words include:

```
BBBAABAA
AA
B
BAAB
```

Parse Tree

Q3. Draw the parse (or derivation) tree for the following MiniTriangle fragment. The relevant grammar is given in Appendix A in the Task Sheet for CW1. Start from the production for `Command`.

```
 if x[i] > 10 then
        putint(k)
    else
        i := (i) - 1
```

Note Hand-drawn solutions are perfectly acceptable. Simply take a photo of your diagram and include the photo in your submission. Do note invest significant time in this question.

MiniTriangle

Compile the MiniTriangle source code provided with CW1 using the included makefile. This process should generate an executable named **HMTC**.

Q4. Write a simple, valid MiniTriangle application (write your own, do not copy one from the lecture notes).

Investigate the command line flags that are available with HMTC. **Hint** try running - `./hmtc --help`

Q5. Using your simple program from Q4, provide the Tokenised output of the scanning process, as represented in HMTC.

Q6. Using your simple program from Q4, provide the concrete abstract representation output of the parsing process, as represented in HMTC.

Bonus...

Investigate how HMTC reacts or behaves to different types of inputs. Specifically look at how it detects, tracks and reports invalid input.