CFL Homework 1

John Pham

3. Read the handout of the first lecture and the handout about notation. Make sure you understand the concepts of strings and languages. In the context of the CFL course, what is meant by the term *language?*

A language is a set of all strings that match some regular expression. For instance, L(r) denotes *the language of the regular expression r.*

$L(0) = \{\}$
$L(1) = \{[]\}$
$L(c) = \{\text{“c”}\}$
$L(r1 + r2) = L(r1) \cup L(r2)$
$L(r1 \cdot r2) = L(r1)\ @\ L(r2)$
$L(r^*) = (L(r))\star$

4. Give the definition of regular expressions — this is an inductive datatype. What is the meaning of a regular expression?

$L(0) = \{\}$
$L(1) = \{[]\}$
$L(c) = \{\text{“c”}\}$
$L(r1 + r2) = L(r1) \cup L(r2)$
$L(r1 \cdot r2) = L(r1)\ @\ L(r2)$
$L(r^*) = (L(r))\star$

5. Assume the concatenation operation of two strings is written as s1@s2.
Define the operation of *concatenating* two sets of strings. This operation is also written as _@_.
According to this definition, what is A@{} equal to?
Is in general A@B equal to B@A?

$A @ B = \{ \; s1 @ s2 \; | \; s1 \in A \;\; \land \;\; s2 \in B \; \}$

A @ {} = A

No, A@B is not equal to B@A in general.
Lets say A = {“a”} and B = {“b”}
A@B = {“ab”}
B@A = {“ba”}

6. Assume a set A contains 4 strings and a set B contains 7 strings. None of the strings is the empty string. How many strings are in A@B?

4 x 7 = 28 strings.

7. How is the power of a language defined?

$L(r)^0 = \{[]\}$
$L(r)^{n+1} = L(r) @ L(r)^n$

8. Let A = {[a], [b], [c], [d]}. (1) How many strings are in A^4? (2) Consider also the case of A^4 where one of the strings in A is the empty string, for example A = {[a], [b], [c], []}

1. 4 * 4 * 4 * 4 = 256 strings
2. (require some assistance!)

9.
(1) How many basic regular expressions are there to match the string abcd?
(2) How many if they cannot include 1 and 0?
(3) How many if they are also not allowed to contain stars?
(4) How many if they are also not allowed to contain _ + _?

(1) Infinite ((abcd · 1) + (abcd · 1 · 1) …..)
(2) Infinite ((abcd) + ((abcd)*) + ((abcd)* + abcd) + ((abcd)* + (abcd)* + abcd)….)
(3) Depends on alphabet
(4) One

10.
When are two regular expressions equivalent? Can you think of instances where two regular expressions match the same strings, but it is not so obvious that they do? For example a + b and b + a do not count…they obviously match the same strings, namely [a] and [b].

Two regular expressions are equivalent when the language of both regular expressions are equivalent – meaning the set of strings both regular expressions match are equal.

(a)* = (1 + (a · a)*) a

11. What is meant by the notions evil regular expressions and by catastrophic backtracking?

Evil regular expressions are regular expressions for which it is computationally infeasible to determine if a string matches that regular expression or not.

More often than not, programming languages employ depth-first search methods to match strings to regular expressions. When searching possible paths to match the regular expression, if it finds the *wrong* choice, it has to 'backtrack' and find all potential candidates. Which may take a while.