CS2104 Programming Language Concepts
Tutorial 9 : Grammars and Parser Combinators
(week of 26th Oct 2015)

Please attempt this tutorial in OCaml with help of code given.

Q1 For each of the following EBNF, write two valid and two invalid examples.

(i) <S> ::= a{b|c|d}{aa}+

(ii) <S> ::= a{[bc]d}+

Q2 Implement scanners for two regular expressions in Q1 by using:
(i) tail-recursive OCaml functions.
(ii) regular expression regexp type supported by the OCaml Str module.
http://caml.inria.fr/pub/docs/manual-ocaml/libref/Str.html

Q3. The parser combinator (repeat ph) would repeatedly apply a parser, ph, zero or more times. Write a parser combinator that would apply a given parser ph one or more times. That is the parser must succeed at least once.

Q4. Consider the BNF grammar rules below:

<S> ::= '(' <A> ')'
<A> ::= '[' <A> ']'
| '{' <A> '}' <S>
| a | … | z

(i) Write two valid and invalid instances of the above grammar form.
(ii) Use parser combinators given to implement a parser for this language.
(iii) Can this parser be implemented using regular expression form of the OCaml Str module? If not, briefly explain why.