Lab Activity 4(Understanding Predictive Parsing)
Nathan Njonge
ICS4A
094230

1 Using an example or illustration, explain what is predictive parsing?
Predictive parsing is a special form of recursive descent parsing, where no backtracking is required, so this can predict which products to use to replace the input string.
Non-recursive predictive parsing or table-driven is also known as LL(1) parser. This parser follows the leftmost derivation (LMD).

Example: Consider the subsequent LL(1) grammar:
S -> A
S -> (S * A)
A -> id

Parsing table
row-> for each and every non-terminal symbol,
column-> for each and every terminal (including the special terminal)

LL(1) table

	(	)	id	*	$
S	s->(s*A)	-	S->A	-	-
A	-	-	A-id	-	-

2. As an extension to question 1,
a. What is LL (1) parsing?
A top-down parser that uses a one-token lookahead is called an LL(1) parser.The first L indicates that the input is read from left to right.The second L says that it produces a left-to-right derivation.And the 1 says that it uses one lookahead token.

b. Why do we get the First () and Follow () i.e. what do these two functions Really Represent?
We need to find FIRST and FOLLOW sets for a given grammar so that the parser can properly apply the needed rule at the correct position.

FIRST(X) gives you the set of terminals that can begin the strings derived from X. For eg. if there is a derivations S->X->aX, then a is in First(S).
If your grammar is
S-> X | Y
X->aX | a
Y->bB | b

Now if you want to have a derivation for bbb starting from S, since you know that b is not in FIRST(X) but is in FIRST(Y), the first production to apply is S->Y.

FOLLOW(X) gives you the set of terminals that can follow X in a string derived in this grammar. For eg. if there is a derivation S->Y->XaY, then a is in FOLLOW(X).

3. Are there any unique attributes associated with Top-Down parsing?

Recursive descent parsing : It is a common form of top-down parsing. It is called recursive as it uses recursive procedures to process the input. Recursive descent parsing suffers from backtracking.

Backtracking : It means, if one derivation of a production fails, the syntax analyzer restarts the process using different rules of same production. This technique may process the input string more than once to determine the right production.