1. (5 points) Describe how a bottom-up parser works. What is put on the stack? What are the kind of actions it performs?

2. Consider the following grammar:

(1) E \rightarrow E + T
(2) E \rightarrow T
(3) T \rightarrow T F
(4) T \rightarrow F
(5) F \rightarrow F^*
(6) F \rightarrow a
(7) F \rightarrow b

with terminals $a, b, +, \ast$, nonterminals $E, T, F$, and start symbol $E$.

(a) (12 points) Construct the LR(0) items, along with the goto function (transition function).
(b) (6 points) Construct the SLR parsing table. Show how you come with your parse table entries.
(c) (6 points) Show the contents of the parse stack, input and the actions of the parse stack as well as the parse tree at each step during a parse of the input $a^*a + b$.

3. (2 points) Can SLR parsers handle left-recursive grammars? Justify your answer.