Compiler Design 2020 M.   Mid Examination. For 90 Minutes.

Each question should begin in a new page. Do not interleave answers. Give page numbers to each page. Leave 1 inch margin on the left side. Unclean or illegible answers can attract negative marks.

Upload your answer script (a single PDF file) on to the google classroom of the CD course.

1. For the CFG $S \rightarrow S \vee B|B, B \rightarrow B \wedge C|C, C \rightarrow \neg C|(S)|a|b|c$, (i) remove left recursion (if you want to add new non-terminals, then call them with names $D, E, \ldots$) . Let the resulting grammar be G_1 . (ii) Apply left factoring for G_1. Let the resulting grammar is called G_2. (iii) Find FIRST and FOLLOW for each of the non-terminal for the grammar G_2. (iv) Find LL(1) parse table for G_2 and thus find whether the grammar is LL(1) or not. (v) Step by step parse the (show the steps in the form of a table) the string: $a \wedge (\neg b \vee c)$ and thus find whether the given string is in the language of the grammar or not.

1. For the CFG $S \rightarrow S \wedge B|B, B \rightarrow B \vee C|C, C \rightarrow \neg C|(S)|a|b|c$, (i) remove left recursion (if you want to add new non-terminals, then call them with names $D, E, \ldots$) . Let the resulting grammar be G_1 . (ii) Apply left factoring for G_1. Let the resulting grammar is called G_2. (iii) Find FIRST and FOLLOW for each of the non-terminal for the grammar G_2. (iv) Find LL(1) parse table for G_2 and thus find whether the grammar is LL(1) or not. (v) Step by step parse the (show the steps in the form of a table) the string: $a \wedge (\neg b \vee c)$ and thus find whether the given string is in the language of the grammar or not.

2. Consider the CFG $S \rightarrow S + A \mid A, A \rightarrow AB|B, B \rightarrow B\&|a|b$. (i) Find whether the given grammar is LR(0) or not. (ii) Find whether the given grammar is SLR or not. (iii) If the grammar is SLR, then show the parsing steps of the SLR (in the form of a table) for the string $a + ba\&$ (iv) If the grammar is not SLR then give reason why it is not.

2. Consider the CFG $S \rightarrow AS|acS|c, A \rightarrow ab$. (i) Find whether the given grammar is LR(0) or not. (ii) Find whether the given grammar is SLR or not. (iii) If the grammar is SLR, then show the parsing steps of the SLR (in the form of a table) for the string $abaccb$ (iv) If the grammar is not SLR then give reason why it is not.

2. Consider the CFG $S \rightarrow AS|abS|c, A \rightarrow ac$. (i) Find whether the given grammar is LR(0) or not. (ii) Find whether the given grammar is SLR or not. (iii) If the grammar is SLR, then show the parsing steps of the SLR (in the form of a table) for the string $acabac$ (iv) If the grammar is not SLR then give reason why it is not.

3. Consider the CFG $S \rightarrow SaS|SbS|dSd|c$. (i) Find whether this grammar is CLR(1) or not (You need to build LR(1) automaton and CLR(1) parse table). (ii) If the grammar is CLR(1), then show step by step parsing (in the form of a table) of the string: cbcac. (iii) if the grammar is not CLR(1) can you give a reason why it is not.

3. Consider the CFG $S \rightarrow SaA|A, A \rightarrow AbB|B, B \rightarrow c$. (i) Find whether this grammar is CLR(1) or not (You need to build LR(1) automaton and CLR(1) parse table). (ii) If the grammar is CLR(1), then show step by step parsing (in the form of a table) of the string: cbcac. (iii) if the grammar is not CLR(1) can you give a reason why it is not.