TUTORIAL 2 DATE: 5th FEBRUARY 2003

1. Prove that the following grammars do not belong to the class mentioned alongside by showing any parsing table entry with conflicts in each case.

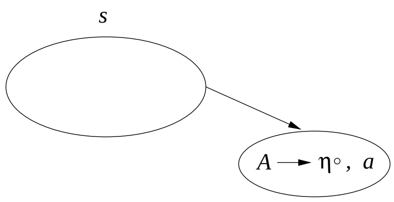
   (a) **not SLR(1)**
   $S \rightarrow X$ `@`
   $X \rightarrow M\ a \mid b\ M\ c \mid d\ c \mid b\ d\ a$
   $M \rightarrow d$

   (b) **not LALR(1)**
   $S \rightarrow A\ a \mid C\ A\ c \mid B\ c \mid C\ B\ a$
   $A \rightarrow d$
   $B \rightarrow d$
   $C \rightarrow c\ c$

   (c) **not LR(1)**
   $S \rightarrow$ `id := ` ae | `id := ` be
   $be \rightarrow be$ `and id` | be `or id` | ae `== id` | `id`
   $ae \rightarrow ae$ `+ id` | `id`

2. Explain with an example why the following statement is incorrect:
   If there is a production $A \rightarrow \alpha\ B$, then $FOLLOW(A) = FOLLOW(B)$.

3. Using the figure shown below as a hint, answer the following question. Is it possible for a LR parser to be in a situation where it has to consult $GOTO\ [s,A]$ for some state s and non-terminal A, and $GOTO\ [s,A]$ does not contain any entry? Clearly explain your answer.



4. Consider the following grammar:
   $S \rightarrow A_1\ b_1 \mid A_2\ b_2 \mid A_3\ b_3$
   $A_1 \rightarrow a_2\ A_1 \mid a_3\ A_1 \mid a_2 \mid a_3$
   $A_2 \rightarrow a_1\ A_2 \mid a_3\ A_2 \mid a_1 \mid a_3$
   $A_3 \rightarrow a_1\ A_3 \mid a_2\ A_3 \mid a_1 \mid a_2$

   Now answer the following questions with explanations.

   (a) Is the grammar ambiguous?
   (b) Is the grammar $LL(n)$ for some value of n?
   (c) Is the grammar $SLR(n)$ for some value of n?
   (d) Is the grammar $LALR(n)$ for some value of n?

5. For the grammar shown below, answer the following questions with explanations:
   $S \rightarrow$ `id` $[\ E\ ]\ := E$
   $E \rightarrow E + T \mid T$
   $T \rightarrow T * F \mid F$
   $F \rightarrow (\ E\ ) \mid$ `id`

   (a) In the automaton of the SLR(1) parser for this grammar, there is a state with a self loop? Identify the state through its shortest viable prefix.
   (b) Is the behavior of the SLR(1) parser and the LR(1) parser identical for *all strings*?
   (c) Is the behavior of the LALR(1) parser and the LR(1) parser identical for *all strings*?