QUIZ 2

DATE: 13th FEBRUARY 2002 **Maximum Marks: 40**

---

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.

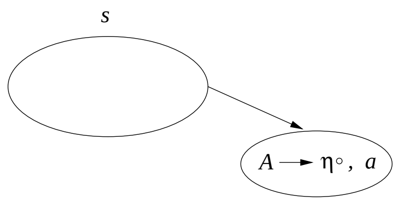
   (a) **not SLR(1)**
   $S \rightarrow X$ `@`
   $X \rightarrow M\ a \mid b\ M\ c \mid d\ c \mid b\ d\ a$
   $M \rightarrow d$

   (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 $\mid$ `id := ` be
   $be \rightarrow be$ `and id` $\mid$ be `or id` $\mid$ ae `== id` $\mid$ `id`
   $ae \rightarrow ae$ `+ id` $\mid$ `id` (3 ∗ 5 = 15 marks)

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 marks)

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. (5 marks)



4. Consider the grammar shown below:

   $P \rightarrow D;\ S$
   $D \rightarrow$ `label` id | `proc` $id;\ D;\ S$ | $D1;\ D2$
   $S \rightarrow$ `a` | `go to` id | id: `a` | id: `go to` id | $S1;\ S2$

   The scope rules regarding labels and gotos is as follows:

   (a) A label must be both declared and defined in the same procedure.

   (b) A use of a label (through a goto) is bound to a definition in the same way as that of an ordinary variable. Every use of a label should be bound to some definition.

We want to write syntax directed definitions to achieve two things:

(a) A violation of the above rules should be flagged as an error at soon as possible.

(b) A goto statement S should have an attribute called *where*. This should evaluate to the name of the procedure containing the definition to which the goto will be targeted.

Here is an example program produced by the grammar along with the output we want to achieve.:

```
label x1;

proc P;
  label x2;  label x3;

  proc Q;
    label x1; label x4

    x3: a        /* error, x3 not declared in this procedure */
    x4: goto x1  S.where = Q, S.visible ={(x1,Q), (x2,P), (x3,P),
                 (x4, Q)}, S.dnd = {x1, x4}
    x1: goto x5  /* error, x5 not in scope*/
  end proc /*Q*/

  x2: goto x4    /* error, x4 not in scope*/
  goto x1        S.where = main, S.visible ={(x1,main), (x2,P),
                 (x3,P)}, S.dnd = {x3}
end proc /*P*/   /* error, x3 declared but not defined in P*/
x1: a
```

You must use the following attributes:

S.visible - the labels visible just before S and the procedure names containing them.

S.dnd - the labels which have been declared in the procedure containing S but not yet defined at a point just prior to S.

You may use any other attribute and any function that you wish. However, clearly specify what they mean. *Use attributes only, do not use any global variables.* (17 marks)