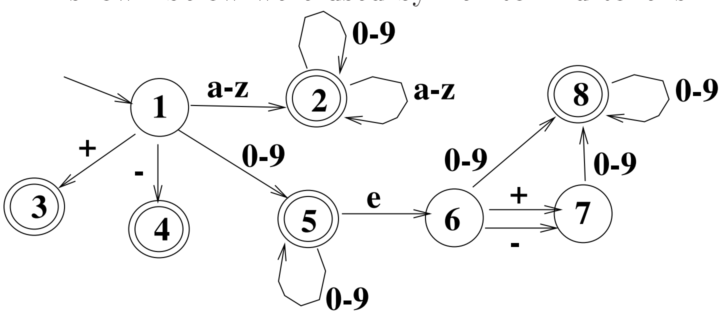
QUIZ 1

Date: 29th January 2003 Max. Marks 35

1. Suppose the DFA shown below were used by Lex to find tokens in an input file.

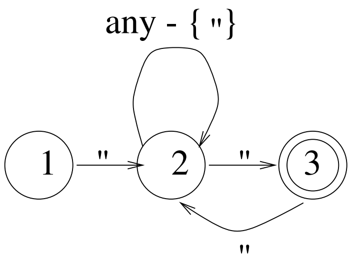


What is the maximum number of characters that Lex might have look ahead past the end of a token, to be sure of the token's arrival? Illustrate you answer with an example. (5 Marks)

Answer: To detect the end of the token represented by state 5, the lexer may have to look ahead 3 characters past the token. An example is the string `10e-f`. Only on encountering `f` will the lexer be sure of the end of the token three characters back.

2. Informally, define a comment as a sequence of characters enclosed within `"` and `"`. If the character `"` itself appears within a comment, it is to be represented as `""`.

(a) Draw a *minimised* DFA which recognizes comments.



(b) Using the DFA above, write the code for lexical analyser to detect comments. This has to be done using a language which has:

i. *assignment statements*,

ii. *if* with a *then* and an optional *else* clause,

iii. *go to* statements which can jump to a label at the beginning of a statement.

iv. *expressions* made with *simple variables*, the operator ==, and *function calls* to the three functions *nextchar*(), *error*() and *action*(). These functions have obvious meanings. (5+5=10 Marks)

```
1: if nextchar()== '"' then goto 2 else error()
2: if nextchar()== '"' then goto 3 else goto 2
3: if nextchar()== '"' then goto 2 else action()
```

3. Given the following specification of a language through regular expressions:

   $variable \rightarrow$ ($simple\_or\_array$.)* $simple\_or\_array$
   $simple\_or\_array \rightarrow$ `id` ([$elist$])?
   $elist \rightarrow$ (e ,)* e

   Consider `e` to be a terminal symbol.

   (a) Write a LL(1) grammar specifying the same language.

   (b) Construct the LL(1) parsing table for the grammar. (5+5=10 Marks)

   (a) *1.* $variable \rightarrow simple\_or\_array\ V$
   2. & 3. $V \rightarrow .\ simple\_or\_array \mid \epsilon$
   4. $simple\_or\_array \rightarrow$ `id` EL
   5. & 6. $EL \rightarrow$ [$elist$] $\mid \epsilon$
   7. $elist \rightarrow$ `e` CE
   8. & 9. $CE \rightarrow$,`e` $\mid \epsilon$

   (b) The $LL(1)$ table is shown below:

	.	id	[	]	,	e	$
variable		1					
simple_or_array		4					
V	2						3
EL	6		5				6
elist						7	
CE				9	8		

4. Consider the grammar shown below:

   $S \rightarrow X$
   $X \rightarrow M\ a \mid b\ M\ c \mid d\ c \mid b\ d\ a$
   $M \rightarrow d$

   Now complete the following sentences:

   The grammar is not SLR(1). The state which demonstrates this has the following set of items: $X \rightarrow d.c$ and $M \rightarrow d.$ Consider the two items $X \rightarrow d.c$ and $M \rightarrow d.$ in this state. The first suggest a shift on the symbol c, while the second suggests a reduce using production $M \rightarrow d$ on the same symbol. $(1 + 3 + 6 \times 1 = 10$ Marks)