Tue, February 12 Name ______________________

6.035 Spring 2013 Miniquiz #4 5 minutes

Suppose that our language calls a procedure by giving its name, with parameters surrounded by parentheses, and that arrays are referenced by the same syntax. Since the translation of parameters to procedure calls and indices in array references are different, we want to use different productions to generate lists of parameters and lists of indices. Our grammar might therefore have (among others) productions as follows:

(1)	stmt	$\rightarrow$	ID ‘(’ parameter_list ‘)’
(2)	stmt	$\rightarrow$	expr “:=” expr
(3)	parameter_list	$\rightarrow$	parameter_list ‘,’ parameter
(4)	parameter_list	$\rightarrow$	parameter
(5)	parameter	$\rightarrow$	ID
(6)	expr	$\rightarrow$	ID ‘(’ expr_list ‘)’
(7)	expr	$\rightarrow$	ID
(8)	expr_list	$\rightarrow$	expr_list ‘,’ expr
(9)	expr_list	$\rightarrow$	expr

The ID ‘(’ ID ‘,’ ID ‘)’ token stream will be given to a shift-reduce parser for the grammar. Assume that the parser never takes a step that produces a configuration from which it is not possible to produce a parse tree.

1. What configuration (stack and input) will the parser be in after 3 steps?

2. What conflict will occur at the moment? Why?

3. What are the pros and cons of having nondeterminism in the behavior of a program?