

University of Toronto

CSC488S/CSC2107S Compilers and Interpreters

Winter 2014/2015

Mid Term Test (20% of course mark)

March 5, 2015.

5 questions on 2 pages. 100 marks total. 50 minutes total

Open Book and Notes, Non-programmable calculators allowed, NO other electronic aids allowed

Answer all questions. WRITE LEGIBLY!

If you need to make any additional assumptions to answer a question, be sure to state those assumptions in your test booklet. The line numbers on the left side of programs and grammars are for reference only and not part of the program or grammar .

1. [15 marks] Given the grammar:

1	S	→	A B C
2	A	→	a A
3		→	λ
4	B	→	B b
5		→	λ
6	C	→	c C
7		→	λ

Where λ is the empty string.

Show a *leftmost derivation* and a *rightmost derivation* for the string

a b b c

2. [20 marks] Some languages have two forms of the **for** statement

for *variable* := expression **to** expression [**by** expression] **do** statements **end**

and **for** *variable* := expression , expression , ... , expression **do** statements **end**

Where the "**by** expression" part is optional in the first form.

In the second form "... " indicates that an arbitrarily long list of comma separated expressions can occur.

Write a recursive descent parser function that recognizes these two forms of the **for** statement.

You may assume

- your recognizer is called when the statement parser encounters the reserved word **for**
- there is a function **boolean function** *variable* () that recognizes one variable
- there is a function **boolean function** *expression* () that recognizes one expression
- there is a function **boolean function** *statements* () that recognizes one or more statements
- the function **boolean function** *getToken* () advances the scanner input to the next token
- all of these boolean functions return true if successful and false otherwise.
- the global variable *nextToken* contains the next input token from the scanner

3. [20 marks] In the GO programming language, semicolons are used to terminate statements. As a convenience for programmers, Go programs may omit most of these semicolons using the following two rules:

1. . When the input is broken into tokens, a semicolon is automatically inserted into the token stream at the end of a non-blank line if the line's final token is an identifier an integer, a floating point constant, a character constant , or string literal, one of the keywords **break**, **continue**, **fallthrough**, or **return** one of the operators and delimiters ++, −,),], or }
2. To allow complex statements to occupy a single line, a semicolon may be omitted before a closing ")” or ”}”.

Describe the lexical analysis and/or syntax analysis mechanisms necessary to implement these rules.

4. [25 marks]

Use the depth first structure alignment algorithm described in the lecture slides to layout the data fields in the Java classes shown below.

```

1      class outer {
2          class inner {
3              boolean U1, U2, U3, U4, U5 ;
4              short  V ;
5              long   Wa, Wb, Wc, Wd ;
6              double Y ;
7              int    X1, X2, X3 ;
8          } // end of inner
9          short A ;
10         int   B1 ,  B2 ;
11         boolean Cx, Cy, Cz ;
12         inner D, E ;
13         double F ;
14     }

```

The size and alignment factors for the basic data types:

type	align (bits)	size (bits)	type	align (bits)	size (bits)	type	align (bits)	size (bits)
boolean	1	1	long	64	64	int	32	32
short	16	16	float	32	32	double	64	64

5. [20 marks] In order to make it easier to write mutually recursive function, a proposal has been made to add a **forward** declaration to the language for procedures and functions. Examples:

```

forward integer function F
forward integer function G ( parameters )
forward procedure P
forward procedure Q ( parameters )

```

The forward declaration is a commitment that a *matching declaration* for the function or procedure will occur later in the same scope as the forward declaration. *matching declaration* means the same parameters and the same return type for functions. Describe the symbol table mechanisms and semantic analysis checking that will be required to implement this proposal.