University of Toronto

CSC488S/CSC2107S Compilers and Interpreters

Winter 2011/2012

Mid Term Test (15% of course mark)

March 8, 2012.

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. [20 marks] Convert the grammar below into an LL(1) grammar. Show the predict sets for your revised grammar to demonstrate that it is LL(1).

1	S	\rightarrow	S	С	А
2		\rightarrow	С	В	
3		\rightarrow	В		
4	Α	\rightarrow	b	А	
5		\rightarrow	d		
6	В	\rightarrow	В	а	А
7		\rightarrow	А	b	А
8		\rightarrow	А		

2. [25 marks] The current syntax of the project language allows functions and procedures to be declared in some unconventional places, e.g. in either alternative of an **if** statement, inside **while** and **repeat** loops. Suppose you wanted to make these unconventional placements of procedure and function declarations illegal so that procedures and functions can only be declared directly in a major scope, i.e. the main program or the body of a function or procedure and not in any contained minor scopes.

a) could this be done by changing the *syntax* of the language? Describe a possible solution.

b) could this be done during *semantic analysis*? Describe a possible solution.

3. [20 marks] A lexical analyzer for Fortran has to deal with several kinds of tokens:

identifiers	a non-en	npty seque	ence of lett	ers and dig	its starting	with a letter
integer constants	\mathcal{D}					
real constants	\mathcal{D} .	\mathcal{D} \mathcal{I}). Д	$\cdot \mathcal{D}$		
	$\mathcal{D}E\mathcal{D}$	\mathcal{D} .E \mathcal{D}	$\mathcal{D} = \mathcal{D}$	$\mathcal{D} E \mathcal{D}$	$.\mathcal{D}E\mathcal{D}$	
boolean constants	.FALSE.	.TRI	JE.			
boolean operators	.AND.	.OR.	.NOT			
comparison operators	.LT.	.LE.	.EQ.	.NE.	.GE.	.GT.

Where \mathcal{D} stands for a non-empty sequence of decimal digits and E is used to mark the exponent part of real numbers (e.g. 1.0E10). Integer and real constants may be used in the same expression with automatic conversion as required. Fortran is a *blank insensitive* language, except inside strings, the presence or absence of blanks *anywhere* doesn't change the meaning of the program.

Discuss the **problems** that this **combination of lexical tokens** will cause in the design of a lexical analyzer for Fortran. You **do not** have to give the design of a lexical analyzer.

4. [20 marks] Given the declarations in C:

1	struct dataStruct {
2	int myKey ;
3	unsigned char uTag ;
4	union {
5	char name[5] ;
б	<pre>struct { unsigned char kind ; double value ; } ;</pre>
7	int alternate1, alternative 2 ;
8	} data ;
9	short tag[3] ;
10	float rValue;
11	} ;

Assume the bit size and alignment factors listed below

Туре	Size	Alignment	Туре	Size	Alignment
char	8	8	int	32	32
unsigned char	8	8	float	32	32
short	16	16	double	64	64

Using the multi-level structure mapping algorithm discussed in lecture, show how this data structure would be laid out in memory.

5. [15 marks] Describe the symbol and type table entries that a typical compiler might create for the data structure in Question 4.