CSC488S/CSC2107S Compilers & Interpreters Midterm Test [15% of final mark]

Winter 2001 March 1, 2001

Instructions: This midterm is open book, open notes.

Total marks 100 - Total time is 50 minutes. Two pages, 5 questions. Answer all questions. Non-programmable calculators allowed. No electronic communication devices allowed.

The line numbers in example programs are for reference only and are not a part of the programs.

1. [20 marks] The C programming language standard has two conventions that complicate the design of a lexical analyzer for C.

- 1. *backslash continuation* If the backslash character \ occurs at the *end* of an input line, then the next input line is taken as an immediate continuation of the first line. The backslash can occur almost anywhere including in the middle of an identifer, reserved word or constant.
- 2. *trigraphs* ANSI C contains a brain-damaged mechanism for supporting keyboards that do not contain all of the characters required to express C programs. A trigraph is a 3-character sequence beginning with ?? that stands for some other single character.

The 9 ANSI trigraphs are shown below

```
??( is [ , ??) is ] , ??< is { , ??> is } , ??/ is \
??= is # , ??- is ~ , ??' is ^ , ??! is |
```

These trigraphs may be used anywhere that the corresponding character could be used.

Discuss for these lexical features might be implemented in a lexical analyzer for C.

2. [25 marks] Describe the static semantic checks that a competent C compiler would perform on the statements shown below

	•••
10	if ((ch = getchar()) != EOF)
11	fprintf(stderr, "next char is %c", ch);
12	else
13	return ;
14	buffer[bp++] = ch ;
15	for(K = 0 ; K
16	if(ch == buffer[K]){
17	bp — ;
18	break ;
19	};

3. [20 marks] For the declaration given below

1	union bigU {
2	unsigned char uchar ;
3	struct {
4	int ordinal;
5	double dNumb ;
6	} stra ;
7	struct {
8	unsigned char cursor ;
9	int xCoord, yCoord ;
10	char * sptr ;
11	double value ;
12	} strb ;
13	<pre>short clink[4] ;</pre>
14	};

Show how this structure would be laid out in memory using the space conserving *Algorithm 2* that was discussed in lecture. Assume the size and alignment constraints in the table below.

type	size	align	type	size	align
char *	32	32	unsigned char	8	8
double	64	64	short	16	16
int	32	32			

4. [15 marks] Show the symbol and type table entries that a typical compiler would create for the declarations in Question 4.

5.[20 marks] Which of the following four grammars are LL(1)? Justify your answers. Answers without justification will get partial credit. (λ is the empty string).

1	$\begin{array}{rrrr} S & \rightarrow & A \ B \ c' \\ A & \rightarrow & \ a' \\ A & \rightarrow & \lambda \\ B & \rightarrow & \ b' \\ B & \rightarrow & \lambda \end{array}$	2	$\begin{array}{rrrr} S & \to & A B B A \\ A & \to & \mathbf{\hat{a}} \\ A & \to & \lambda \\ B & \to & \mathbf{\hat{b}} \\ B & \to & \lambda \end{array}$
3	$\begin{array}{ccccc} S & \rightarrow & A \ {}^{\prime}b' \\ A & \rightarrow & {}^{\prime}a' \\ A & \rightarrow & B \\ A & \rightarrow & \lambda \\ B & \rightarrow & {}^{\prime}b' \\ B & \rightarrow & \lambda \end{array}$	4	$\begin{array}{rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr$