University of Toronto

CSC 488/2107 Language Processors **Fall 2005**
Midterm Test [15% of final mark] **Oct. 20, 2005**

Total marks 65 - Total time is 50 minutes. Answer all 4 questions.
Instructions: This midterm is open book, open notes. Non-programmable calculators allowed. No electronic communication devices allowed.
The line numbers in grammars are for reference only and are not part of the grammars. Ellipses ... indicate omitted, correct text. ϵ indicates an empty string.
If you need to make any assumptions in order to answer a question, state the assumptions clearly in your answer book.

I. [7 marks] Type declarations.

Let the following type declaration in a fictional (but inspired by Pascal) language be given:

```
typedef B = *A;  /* create a type of pointers to type A */
typedef A =
  X : integer;
  Y : B;
end;

var Z : A;
```

1. **[5 marks]** Without defining new types, can you create an expression that is structure equivalent to `Z` (other than `Z` itself)? If so, do it. If not - explain why not.

2. **[2 marks]** What about name equivalent to `Z`?

II. [20 marks] Parsing and Lexical Analysis.

Suppose we are are interested in defining `const`, and are doing so with the following set of expressions:

1:	const	::=	id | number | intconst | ϵ
2:	number	::=	bin | oct | hex
3:	le	::=	a-z
4:	di	::=	0-9
5:	bin	::=	b $(0\|1)^+$
6:	oct	::=	o $(0\|1\|2\|3\|4\|5\|6\|7)^+$
7:	hex	::=	h $(\text{di}\|A\|B\|C\|D\|E\|F)^+$
8:	intconst	::=	di^+
9:	id	::=	$\text{le}(\text{le}\|\text{di})^+$

1. **[8 marks]** Create the NFA for recognizing `const`.

2. **[8 marks]** Make this automaton deterministic.

3. **[4 marks]** Is the resulting grammar LL(1)?

III. [18 marks] LL parsing.
Let the following grammar be given:

1: S ::= E
2: E ::= - E
3: | (E)
4: | E - V
5: | V
6: V ::= id
7: | id (E)

1. **[4 marks]** Is this grammar LL(1)? If not, convert it into an LL(1) grammar.
2. **[10 marks]** Create the LL(1) parse table for the (potentially modified) grammar.
3. **[4 marks]** Is this grammar LR(1)? Explain why or why not.

IV. [20 marks] LR parsing.
Show that the following grammar is LALR(1) but not SLR(1):

1: S ::= $A\ a$
2: | $b\ A\ c$
3: | $d\ c$
4: | $b\ d\ a$
5: A ::= d