UNIVERSITY OF TORONTO
Faculty of Arts and Science

APRIL/MAY EXAMINATIONS 2005

CSC 488/2107S

Duration - 2 hours (120 minutes)

Examination Aids: Books, Notes, Calculators.

115 marks total, 6 questions on 4 pages. ANSWER ALL QUESTIONS.
There is also a bonus 7th question.

Write answers in the exam book. If you need to make any assumptions in order to answer a question, state the assumption clearly in your answer book.

1. [25 marks] LL and LR parsing.

a. [12 marks] Convert the grammar below into LL(1) grammar. Show the director sets for your revised grammar.

1:	S	$\longrightarrow$	S a A
2:		$\longrightarrow$	B
3:		$\longrightarrow$	a B
4:	A	$\longrightarrow$	C A
5:		$\longrightarrow$	c A
6:		$\longrightarrow$	b
7:	B	$\longrightarrow$	B b A
8:		$\longrightarrow$	A
9:		$\longrightarrow$	A c A

b. [10 marks] Show that the following grammar is LL(1) but not SLR(1):

1:	S	$\longrightarrow$	A a A b
2:		$\longrightarrow$	B b B a
3:	A	$\longrightarrow$	λ
4:	B	$\longrightarrow$	λ

c. [3 marks] Assume we have a working compiler that uses an LL(1) parser. As a rule of thumb, almost all LL(1) grammars are also LALR(1). Hence, it should be feasible to replace the LL(1) parser with an LALR(1) parser. What must we do to guarantee that the replacement is transparent to the rest of the compiler?

2. [10 marks] Project experience.

a. [5 marks] If you could get rid of one CAPAL language feature, which would it be? In a couple of sentences, explain why. And what is your second least favourite?

b. [5 marks] If you were to give advice to the project architect for next year's course, which two modifications would you suggest him or her to make to `Machine.java`, to make lives of future generations of 488/2107 students easier? Explain why, in a couple of sentences.

3. [15 marks] Let the following `switch` statement be given, where `i` is an integer from 0 to 1000.

```
1: switch (i) {          13:
2:    case 3:            14:    case 18:
3:    case 17:           15:       i = i-1;
4:        i = i+1;       16:       break;
5:        break;         17:
6:                       18:    default:
7:    case 50:           19:       i = i+4;
8:    case 62:           20: }
9:    case 77:
10:      i = i*3;
11:      j = i;
12:      break;
```

a. [5 marks] Describe an efficient translation strategy for this `switch` statement.

b. [10 marks] Show the result of translating it to IR. You can use the same IR as in the lecture notes. Explicitly describe all other assumptions you made.

4. [20 marks] Optimization of array and record handling.

a. [8 marks] In the lecture notes, we considered two algorithms for record alignment. Using Algorithm 2, show how the following record will be aligned.

```
1:  struct {                7:     char* S;
2:    struct {              8:     union {
3:      unsigned char X;    9:       int R;
4:      short Y[6];         10:      double T;
5:    } first [2];          11:    } smallU;
6:    bit  Z;               12:  } second;
```

The following is a table of alignments and sizes of various types:

type	size	align	bit	1	1
char*	32	32	unsigned char	8	8
double	64	64	short	16	16
int	32	32	short[K]	16*K	16

b. [12 marks] Outline a series of optimizations that can be performed on the following code. You do not have to show how the code looks after every individual optimization, although you have to describe what each applicable optimization accomplishes and give the final code.

```
for (i = 1, i < 9, i++)
   for (j = 2, j < 5, j++)
     a[i+1] = b[i][j+2] + c[j]
```

Assume arrays are declared as follows:

```
int a [2..10], b[1..12][4..9], c[2..5];
```

5. [22 marks] Let the following Pascal program be given.

```
1: program prog (output)
2:
3:   procedure q
4:   label L:
5:     procedure exec (procedure z)
6:     begin (* exec *)
7:       z;
8:     end;  (* exec *)
9:
10:    procedure r
11:    begin (* r *)
12:      goto L;
13:    end;  (* r *)
14:
15:  begin (* q *)
16:    exec(r);
17:  end;  (* q *)
18:
19:   procedure p (procedure a)
20:   var v : integer;
21:   begin (* p *)
22:     v := 10;
23:     a;
24:     writeln (v);
25:   end (* p *)
26:
27: begin (* prog *)
28:   p(q);
29: end. (* prog *)
```

a. [7 marks] Which of the display-handling algorithms described in class will work correctly for this program? Which will fail? Why?

b. [15 marks] Write code that needs to be executed for line 16 of the above program in the target language of the project. Comment your code and discuss your choices for prolog/epilogue in setting up and tearing down function and procedure calls.

6. [23 marks] Data-flow analysis.

We often need to determine whether it is possible for a variable to be used before it has been defined. This problem is called *use-before-def*. Note that in many languages, declaration is separate from definition, e.g.

```
var x : integer;
```

does not assign `x` a value, whereas

```
x = 0;
```

clearly does. Assume that all variables that are being assigned have been declared.

a. [2 marks] Is *use-before-def* a forward-flow or a backward-flow problem?

b. [7 marks] Give the data flow equations for computing *use-before-def*.

c. [12 marks] Divide the program fragment below into basic blocks and compute `in[B]` and `out[B]` for each basic block. Assume `E1` and `E2` are boolean expressions that could be evaluated to either true or false, and ignore variables present in these expressions.

```
1:  ...
2:  x = 5;
3:  while (E1) {
4:     int a;
5:     if (E2) {
6:        x = z;
7:        y = x * 3;
8:     }
9:     else {
10:       z = 10;
11:       a = x + z;
12:    }
13:    { /* new scope */
14:       int z = 10;
15:       int x = z + 3;
16:    } /* end of new scope */
16: }
17: x = y;
```

d. [2 marks] Based on your computation, does this program contain instances of use-before-def?

7. [bonus 4 marks] Our source language is called CAPAL. What else can CAPAL stand for? Note: only top three answers will get the bonus, as judged, with all due bias, by the instructor and TAs.

END OF EXAM