

## Midterm Test Solutions (Evening section)

October 31, 2002

---

**Duration:** 50 minutes

**Aids allowed:** None

**Weight:** 20% of your course grade

This exam contains a total of 5 pages (including this one). Write your answers clearly in the spaces provided. Use the back pages for your rough work.

**Surname:** \_\_\_\_\_

# 0: \_\_\_\_\_/ 1

**First name:** \_\_\_\_\_

# 1: \_\_\_\_\_/ 9

**Student #:**

# 2: \_\_\_\_\_/ 4

# 3: \_\_\_\_\_/ 7

# 4: \_\_\_\_\_/ 5

# 5: \_\_\_\_\_/ 4

# 6: \_\_\_\_\_/10

**TOTAL:** \_\_\_\_\_/40

Good Luck!

**Question 0.** [1 MARK]

Write your student number legibly in the space provided on every page of this test.

**Question 1.** Grammars [9 MARKS]**Part (a)** [2 MARKS]

Consider the following grammar:

$$\begin{aligned} \langle S \rangle &\rightarrow \langle S \rangle \text{ op1 } \langle G \rangle \mid \langle G \rangle \\ \langle G \rangle &\rightarrow \langle Z \rangle \text{ op2 } \langle G \rangle \mid \langle Z \rangle \\ \langle Z \rangle &\rightarrow \langle Z \rangle \text{ op3 } \langle M \rangle \mid \langle Z \rangle \text{ op4 } \langle Z \rangle \mid \langle M \rangle \\ \langle M \rangle &\rightarrow x \mid y \mid ( \langle S \rangle ) \end{aligned}$$

Give the associativity (**left**, **right** or **neither**) of the operators op1, op2, op3 and op4.

Operator	Associativity	Operator	Associativity
op1	___Left_____	op3	___Left_____
op2	___Right_____	op4	___Neither___

**Part (b)** [4 MARKS]

Consider the above grammar again. Which of the following strings are ambiguous? Mark the blank next to each string with either **yes** if it is ambiguous, or **no** if it is not ambiguous.

Ambiguous?		Ambiguous?	
i. ___No_____	x op3 x op4 x	iii. ___Yes_____	x op4 x op3 x
ii. ___No_____	x op4 x op2 x	iv. ___Yes_____	x op4 x op4 x

**Part (c)** [3 MARKS]

Write a CFG in BNF for the language of balanced parentheses. For example, your grammar should accept  $()()()$  and  $((())((())))$  but reject  $(($  and  $()()()$ . Make sure your grammar is unambiguous.

$$\langle S \rangle \rightarrow ( \langle S \rangle ) \langle S \rangle \mid \epsilon$$

**Question 2.** List Representation [4 MARKS]

Draw a diagram showing Scheme's internal representation for the following expression:

`((3 4 ()) 5 (7 8) . 9)`

**Question 3.** Scheme Procedure [7 MARKS]

Write a Scheme procedure (`insert-sorted x lst`) which returns a copy of the sorted list of numbers `lst` with `x` inserted in the correct location, such that the returned list is still in sorted in increasing numerical order. If `x` is already in `lst`, do not re-insert it.

For example,

`(insert-sorted 3 '(1 2 4 5))` returns `(1 2 3 4 5)`.

`(insert-sorted 3 '(1 2 3 4))` returns `(1 2 3 4)`.

```
(define insert-sorted
  (lambda (x lst)
    (cond ((null? lst) (list x))
          ((< x (car lst)) (cons x lst))
          ((= x (car lst)) lst)
          (else (cons (car lst) (insert-sorted x (cdr lst))))))
  )
)
```

**Question 4.** High-Order Procedures [5 MARKS]

Write a procedure (`count-gt x lst`) which returns the number of elements in `lst` which are greater than `x`. Use `map` and `apply` to write your procedure. You may not use recursion or `eval`. For example,

`(count-gt 5 '(4 5 6 7))` returns 2.

```
; Pre: x is a number, lst is a list of numbers.
(define count-gt
  (lambda (x lst)
    (apply + (map (lambda (y) (if (< x y) 1 0)) lst))
  )
)
```

**Question 5.** Parameter Passing [4 MARKS]

Consider the following code in some theoretical language:

```
int i, j, k;

subprogram p(int a, int b) {
  a := a + 2;
  b := b - 1;
  i := 6;
}

i := 10;
j := 11;
k := 12;

p(j, k);
print i, j, k;          --- print 1

p(j, i);
print i, j;            --- Note: p(j, i), not p(i, j)!
                      --- print 2
```

What is the output of this program if the parameter-passing mode of this language is **pass by reference**?

```
print 1: i: 6 j: 13 k: 11
```

```
print 2: i: 6 j: 15
```

**Question 6.** Short Answers [10 MARKS]**Part (a)** [1 MARK]

Give one advantage of assembly over machine language.

Easier to write **because** there are mnemonic for commands, names for variables, and you can use macros. Just writing “Easier to write” got half marks.

**Part (b)** [2 MARKS]

Give two advantages of high-level languages over assembly.

Some potential answer: Easier to port; more constructs make abstractions easier; easier to read and write; easier to debug.

**Part (c)** [1 MARK]

Name a key operation of the object oriented programming paradigm.

Inheritance; message passing.

**Part (d)** [2 MARKS]

With an ambiguous grammar, how many parse trees are there for any given sentence that is in the language? Circle the best answer:

0      exactly 1      more than 1      **1 or more**

**Part (e)** [2 MARKS]

With an unambiguous grammar, how many parse trees are there for any given sentence that is *not* in the language? Circle the best answer:

**0**      exactly 1      more than 1      1 or more

**Part (f)** [2 MARKS]

What does `(list (car '(1 2)) (cdr '(3 (4))))` return?

`((1) ((4)))`

Total Marks = 40