

Midterm Test Solutions (Day 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:

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

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

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

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. ___Yes___ | x op3 x op4 x | iii. ___No_____ |
| ii. ___Yes___ | x op3 x op3 x | iv. ___No_____ |
| | | x op4 x op3 x |
| | | x op3 x op2 x |

Part (c) [3 MARKS]

Write a CFG in BNF for the following language: All strings of 0s and 1s that are palindromes (*i.e.*, are the same whether read forward or backward) **and** contain an even number of 1s.

$$\langle S \rangle \rightarrow 0 \langle S \rangle 0 \mid 1 \langle S \rangle 1 \mid 0 \mid \epsilon$$

Question 2. List Representation [4 MARKS]

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

(1 (3 ()) (4 . 5) ())

Question 3. Scheme Procedure [7 MARKS]

Write a Scheme procedure (`rem-dup lst`) which returns a copy of `lst` with consecutive duplicate elements removed. Use `eq?` to test for equality.

For example,

(`rem-dup '(a b c c d a a a a)`) returns (a b c d a).

```
(define rem-dup
  (lambda (lst)
    (cond ((null? lst) ())
          ((null? (cdr lst)) lst)
          ((eq? (car lst) (cadr lst)) (rem-dup (cdr lst)))
          (else (cons (car lst) (rem-dup (cdr lst))))))
  )
)
```

Question 4. High-Order Procedures [5 MARKS]

Write a procedure (`sums lst`) which takes a list of lists of numbers and returns the sum of each element of `lst`. Use `map` and `apply` to write your procedure. You may not use recursion or `eval`. For example,

`(sums '((1 2 3) () (3 5 7 9 11)))` returns `(6 0 35)`.

```
; Pre: lst is a list of lists of numbers.
(define sums
  (lambda (lst)
    (map (lambda (x) (apply + x)) 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 := 1;
j := 2;
k := 3;

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 value-result**?

```
print 1: i: 6 j: 4 k: 2
```

```
print 2: i: 5 j: 6
```

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

Give one advantage of compilation over interpretation.

Two potential answers: Faster since you can compile once and execute many times; allows optimizations.

Part (b) [1 MARK]

Give one advantage of interpretation over compilation.

Three potential answers: Easier to provide meaningful error messages to the programmer; more flexible; allows rapid prototyping.

Part (c) [1 MARK]

Give one advantage of pseudo-compilation.

One potential answer: Platform independent.

Part (d) [1 MARK]

Name a key operation of the imperative programming paradigm.

Loop; assignment.

Part (e) [2 MARKS]

Suppose you have a grammar that gives two different derivations for the same sentence. Is that grammar ambiguous? Circle the best answer:

Definitely yes Definitely no **There is not enough information to tell**

Part (f) [2 MARKS]

With an ambiguous 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 (g) [2 MARKS]

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

`((1) (4))`

Total Marks = 40