

2003 A SEMESTER EXAMINATIONS



The
University
of Waikato
Te Whare Wānanga
o Waikato

DEPARTMENT	Computer Science
PAPER TITLE	Programming Languages
TIME ALLOWED	Three Hours
NUMBER OF QUESTIONS IN PAPER	Five
NUMBER OF QUESTIONS TO BE ANSWERED	Five
VALUE OF EACH QUESTION	All questions are of equal value
GENERAL INSTRUCTIONS	Nil
SPECIAL INSTRUCTIONS	Nil
CALCULATORS PERMITTED	No

1. (a) Briefly describe the following terms with respect to the analysis phases of a compiler:
lexical analysis
top down parser
LL(1)

[6 Marks]

- (b) Given the following grammar construct its First and Follow sets. The start symbol is Bexpr. Nonterminals start with an uppercase letter. Terminals are bolded in lowercase.

Bexpr → Bterm Bexpr'
Bexpr' → **or** Bterm Bexpr' | ϵ
Bterm → Bfactor Bterm'
Bterm' → **and** Bfactor Bterm' | ϵ
Bfactor → **not** Bfactor | (Bexpr) | **true** | **false**

[9 Marks]

- (c) The following grammar has been written to give the correct precedence to the addition and multiplication operators. However there is problem with the subtraction operator. Outline the problem and rewrite the grammar so that the subtraction operator is handled appropriately. The start symbol is Exp. Nonterminals start with an uppercase letter. Terminals are bolded in lowercase.

Exp → Exp Addop Exp | Term
Addop → + | -
Term → Term * Term | Factor
Factor → (Exp) | **number** | **id**

[5 Marks]

2. (a) Study the following C++ code. Which of the assignment statements labelled 1-10 will give type errors. Why do they give type errors?

```
struct A
{ char x;
  int y;
};

struct B
{ char x;
  int y;
};

typedef struct A C;
typedef C * P;
typedef struct B * Q;
typedef struct A * R;
typedef int S[5];
typedef int T[5];
typedef int U[10];
typedef int Age;
typedef int (*F) (int);
typedef Age (*G) (Age);

main()
{
  A a;
  B b;
  C c;
  P p;
  Q q;
  R r;
  S s;
  T t;
  U u;
  int I;
  Age age;
  F f;
  G g;

  a = b; /* 1 */
  a = c; /* 2 */
  b.x = c.y; /* 3 */
  p = q; /* 4 */
  p = r; /* 5 */
  s = t; /* 6 */
  s[I] = u[I]; /* 7 */
  s = u; /* 8 */
  age = I; /* 9 */
  f = g; /* 10 */
}
```

[8 Marks]

- (b) Describe the different types of polymorphism. Illustrate your answer with examples from programming languages you have used.

[12 Marks]

3. (a) Given the following clauses outline the effect of the cut operator (!) when prolog is given *path2(1, 7)*.

```
move(1,6).
move(1,8).
move(6,1).
move(6,7).
move(8,1).
move(8,3).
move(7,6).
path2(X,Y) :- move(X,Z), !, move(Z,Y).
```

[7 Marks]

- (b) What is the closed world assumption in Prolog? Describe the effect of “negation by failure”.

[6 marks]

- (c) The following *family* predicate establishes a family structure where Charles and Margaret are parents and Tristan and Richard are their children. On 20 June, 2003 Margaret and Charles have an addition to their family, Rowan Marscelles. Write one or more selector predicates which will add Rowan to the family structure.

```
family(person(charles, marscelles, date(7, may, 1950), employed),
        person(margaret, milligan, date(20, june, 1973), employed),
        [person(tristan, marscelles, date(5, may, 1994), student),
         person(richard, dobbs, date(5, may, 1996), student),
         _]).
```

[7 Marks]

4. (a) Briefly describe the following phrases with respect to functional programming languages.

higher order functions
lazy evaluation
partial evaluation of functions
referential transparency

[9 Marks]

(b) What is the type of function *fun*?

```
fun f [] = []
```

```
fun f (x:xs) = f x : fun f xs
```

[2 Marks]

(c) What is the type of function *reduce*?

```
reduce x ys = [y | y <- ys, x \= y]
```

[3 Marks]

(d) Describe the role of the Y combinator in applying recursive functions in lambda calculus.

[6 Marks]

5. (a) Discuss the issues of polymorphism, dynamic binding and strong typing in object oriented programming languages.

[10 Marks]

(b) Discuss polymorphic type checking in Haskell. Illustrate your answer with examples.

[7 Marks]

(c) What is the difference between `let` and `let*` in Scheme?

[3 Marks]