

Duration: **50 minutes**
 Aids Allowed: **none**

Student Number: _____

Family Name(s): _____

Given Name(s): _____

*Do **not** turn this page until you have received the signal to start.*
 In the meantime, please read the instructions below *carefully*.

This term test consists of 3 questions on 10 pages (including this one), printed on both sides of the paper. *When you receive the signal to start, please make sure that your copy of the test is complete, fill in the identification section above, write your student number where indicated at the bottom of every odd-numbered page (except page 1), and write your name on the back of the last page.*

Answer each question directly on the test paper, in the space provided, and use the reverse side of the pages for rough work. If you need more space for one of your solutions, use the reverse side of a page and *indicate clearly the part of your work that should be marked.*

In your answers, you may use without proof any result or theorem covered in lectures, tutorials, homework, tests, or the textbook, as long as you give a clear statement of the result(s)/theorem(s) you are using. You must justify all other facts required for your solutions.

Write up your solutions carefully! In particular, use notation and terminology correctly and explain what you are trying to do — part marks *will* be given for showing that you know the general structure of an answer, even if your solution is incomplete.

If you are unable to answer a question (or part), you will get 20% of the marks for that question (or part) if you write “I don’t know” and nothing else — you will *not* get those marks if your answer is completely blank, or if it contains contradictory statements (such as “I don’t know” followed or preceded by parts of a solution that have not been crossed off).

MARKING GUIDE

1: _____/17

2: _____/14

3: _____/14

TOTAL: _____/45

Use this page for rough work — clearly indicate any section(s) to be marked.

Question 1. [17 MARKS]

Consider the following CFG G_0 , intended to represent polynomials in one variable.

$$\begin{aligned} \langle \text{poly} \rangle &::= \langle \text{poly} \rangle + \langle \text{poly} \rangle \mid \langle \text{poly} \rangle - \langle \text{poly} \rangle \mid \langle \text{term} \rangle \\ \langle \text{term} \rangle &::= \langle \text{number} \rangle x^{\langle \text{number} \rangle} \mid x^{\langle \text{number} \rangle} \mid \langle \text{number} \rangle x \mid x \mid \langle \text{number} \rangle \\ \langle \text{number} \rangle &::= \langle \text{digit} \rangle \mid \langle \text{digit} \rangle \langle \text{number} \rangle \\ \langle \text{digit} \rangle &::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \end{aligned}$$

Part (a) [2 MARKS]

Give a specific string over the alphabet $\{+, -, ^, x, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ that does **not** belong to $L(G_0)$, and explain briefly why your string does not belong to $L(G_0)$.

Part (b) [4 MARKS]

Give a specific string in $L(G_0)$ with two different parse trees, and draw both parse trees for your string.

Use this page for rough work — clearly indicate any section(s) to be marked.

Question 1. (CONTINUED)**Part (c)** [5 MARKS]

Give an *unambiguous* CFG G_1 equivalent to G_0 . Explain briefly why G_1 is unambiguous—in particular, discuss any decision you made regarding precedence and associativity. (Use “...” to indicate productions that remain the same as in G_0 , so you do not have to waste time copying them over.)

Part (d) [6 MARKS]

Give a CFG G for the following language, and justify briefly that $L(G) = L$ (in particular, state clearly what strings can be produced from each of your variables).

$$L = \{a^\ell b^m c^n : \ell, m, n \geq 0 \text{ and } (\ell \geq m \text{ or } \ell \geq n)\}$$

Use this page for rough work — clearly indicate any section(s) to be marked.

Question 2. [14 MARKS]

For each part of this question, your answers must follow good functional programming style—in particular, make use of patterns, local declarations, and higher-order functions where appropriate.

Part (a) [4 MARKS]

Write ML code to define a polymorphic “ternary tree” data type, where values are stored at the leaves and each internal node has exactly three children. Use constructor name **Leaf** for leaves and constructor name **Branch** for internal nodes.

Part (b) [4 MARKS]

Write ML code for a function `size = fn : 'a ttree -> int` that returns the total number of nodes (including leaves) in a ternary tree (as defined above).

Part (c) [6 MARKS]

Write ML code for a function `ttreemap = fn : ('a -> 'b) -> 'a ttree -> 'b ttree` that returns the ternary tree obtained by applying the function (given as first argument) to each value stored in the ternary tree (given as second argument).

Use this page for rough work — clearly indicate any section(s) to be marked.

Question 3. [14 MARKS]

For each part of this question, your answers must follow good functional programming style—in particular, make use of patterns, local declarations, and higher-order functions where appropriate.

Part (a) [6 MARKS]

Write efficient ML code for a function `unzip = fn : ('a * 'b) list -> 'a list * 'b list` that returns the list of first members and the list of second members of each pair in its argument. For example,

```
- unzip [(1,"blah"),(5,"hi"),(~2,""),(3,"x")];
val it = ([1,5,~2,3],[ "blah","hi","", "x"]) : int list * string list
```

Part (b) [4 MARKS]

Write ML code for a function `filter = fn : ('a -> bool) -> 'a list -> 'a list` that returns the sublist of every element (from the second argument) for which the function (given as first argument) evaluates to `true`. For example,

```
- filter (fn x => x >= 0) [1,2,~1,~6,0,~3,7];
val it = [1,2,0,7] : int list
```

Part (c) [4 MARKS]

Write ML code for a function `filterall = fn : ('a -> bool) list -> 'a list -> 'a list` that returns the sublist of every element (from the second argument) for which **all** functions (from the first argument) evaluate to `true`. For example,

```
- filterall [fn x => x >= 0, fn x => x mod 2 = 1] [1,2,~1,~6,0,~3,7];
val it = [1,7] : int list
```

(You may call function `filter` in your answer, even if you did not solve the previous part.)

On this page, please write nothing except your name.

Family Name(s): _____

Given Name(s): _____