
**SCHOOL OF MATHEMATICAL AND COMPUTER SCIENCES**

**Computer Science**

---

F28FS2

Formal Specification Mock exam

Semester 2 201415

---

**Sometime before 14 May 2015**

Duration: As long as you like (actual exam: 2 hours)

ANSWER THREE QUESTIONS

## Some words on using this mock paper

Every concept in this paper appears in the lecture notes and exercises. The difficulty level of a question is graduated from rather easy at the start, to rather hard towards the end.

The questions in this mock are harder, and often more open-ended, than what you will face in the exam. However, things seem more difficult during exams because of the stress.

If you understand and can do these questions, then you are certain to get a decent grade in the exam.

- You *must* attempt this entire paper *before* looking at the answers. Have you attempted the paper yet?

- Then look up answers.
- Then do the paper again.
- Repeat until perfect.

Good luck.

**1. (a)** Explain in English the meanings of the following sets:

1. $\{x : \mathbb{N} \bullet x\}$. (1)
2. $\{x : \mathbb{N} \mid x \geq 1\}$. (1)
3. $\{x : \mathbb{N} \bullet 2 * x\}$. (1)
4. $\{X : \mathbb{PN} \bullet \#X = 2\}$. (2)
5. $\{x, y : \mathbb{N} \bullet \{x, y\}\}$. (2)

**(b)** Write the following sets in Z notation:

1. Numbers divisible by 3. (1)
2. Numbers that are the sum of two distinct primes. Here and henceforth you may assume a set $prime : \mathbb{PN}$ of prime numbers. (2)
3. Numbers that are equal to the sum of two distinct primes, and also to the product of two distinct primes. (2)
   Comment on an ambiguity in this question. (1)

**(c)** 1. Explain in English the meaning of the Z type $iseq\,\mathbb{N}$. (1)

2. Give a precise description of the underlying sets implementation of $iseq\,\mathbb{N}$ in Z. Your answer need not necessarily be in mathematical notation, but you must demonstrate you understand how in full detail how this type is implemented. (2)

**(d)** Explain in English the meanings of the following sets:

1. $\{(x', x), (y', y) : \mathbb{N} \times \mathbb{N}_1) \mid x' * y = y' * x \bullet ((x', x), (y', y))\}$. (2)
2. $\bigcap\{X : \mathbb{PN} \mid 0 \in X \land \forall\, x{:}X \bullet x{+}2 \in X\}$. (2)

**2.** Assume an abstract types of *people* $[PERSON]$ and *rooms* $[ROOM]$.

Henceforth you may assume standard Z operations on sets and functions, such as sets union and intersection, sets subtraction, domain, range, domain and range (anti)restriction, functional and relation application, image, and inverse, and so forth.

**(a)** Write a schema $State$ with precisely one schema variable $inRoom : PERSON \pfun ROOM$. The state predicate should reflect that every person is in at most one room. (2)

**(b)** Write the schemas $\Delta State$ and $\Xi State$ in full. (2)

**(c)** Write a schema ***Move*** which inputs $p? : PERSON$ and $r? : ROOM$ and moves $p?$ to be in $r?$, provided that $p?$ is not already in $r?$. (4)

**(d)** Assume a type $MESSAGE ::= alreadyInRoom \mid success$. Totalise the schema $Move$ with an appropriate error message. (4)

**(e)** Write a schema $Outdoors$ with state variable including $outdoors! : \mathbb{P} PERSON$ that outputs the set of people who are not in any room; $Outdoors$ should not change the state. (2)

**(f)** Write a schema $FireAlert$ that empties all the rooms. (2)

**(g)** Write a schema $Mingle$ which shuffles people so that people who are indoors stay indoors, and people outdoors stay outdoors, but nobody is in the same room after the schema as they were before. (2)

**(h)** Assume a global constant $charlie : PERSON$. $charlie$ (family name Chaplin) is a fantastic conversationalist; people want to be around him.

Write a schema $Charlie$ which assumes that $charlie$ is in some room, and after the schema, at least one person has moved from every other nonempty room to be in the same room as $charlie?$. No other people move. (2)

**3. (a)** Your boy or girlfriend wants you to be assertive, yet sensitive; spontaneous, yet deep; popular, yet devoted only to him or her.

Translate this requirement into Z and comment on the possibilities of implementing this specification. (2)

**(b)** For each of the the following jobs, give a concrete example of where formal or semi-formal specification is important in that job, and one aspect of the job where formal specification is likely to be less useful:

1. Working the telephone in a Virgin Broadband call centre.
2. Teaching computer science at undergraduate level.
3. Parachute jumping instructor.
4. Chip designer. (8)

**(c)** Consider the following specification for hacking any account: "Open login screen. Type in the right password."

Using this example, explain the difference between *specification* and *implementation*. (2)

**(d)** Consider the following refined specification for hacking any account: "Open login screen. Type in every possible password until you find the right one."

Using this example, elaborate further on the difference between *specification* and *implementation*. (2)

**(e)** Microsoft used to be famous for shipping buggy code (less so now). Explain why this made economic sense in the 90s and early 2000s, and discuss why it is less acceptable now. (4)

**(f)** Give one example of a proper, full-fat, fully-formal specification that you encountered on your way in to the University today. (2)

**4. (a)** 1. Write the ML type of `mysucc`.
```
fun mysucc (x:int) = x+1;
```
(2)

2. State the value and type computed by
```
mysucc ~1;
```
(2)

**(b)** Consider the following Z specification:

*Factorial*

$$fact : \mathbb{N} \to \mathbb{N}$$

---

$$fact\, 0 = 1$$
$$\forall\, n : \mathbb{N}_1 \bullet fact\, n = n * (fact(n-1))$$

1. Implement `fact` in ML. (2)
2. State the type of `fact` and explain why it is different from the type specified in Z. (2)

**(c)** 1. Consider the ML function `real` and recall that `real (1:int)` evaluates to `1.0:real`. State the type of `real` and explain what it does. (2)

2. Explain why `real` is necessary. (1)
3. The *binomial* is specified in Z as follows, where $\mathbb{R}$ is the type of real numbers:

*Binomial*

$$bin : \mathbb{N} \to \mathbb{N} \to \mathbb{R}$$

---

$$\forall\, n, k{:}\mathbb{N} \bullet k \leq n \Rightarrow bin\, n\, k = (fact\, n)/((fact\, k)) * (fact\, (n-k))$$

a) State the type of an implementation of `bin` in ML. (2)

b) Implement `bin` in ML. You do not need to include error-handling code for out-of-bound inputs. (4)

4. Write an ML function `B` which inputs a number `n` and outputs the list $[\binom{n}{0}, \ldots, \binom{n}{n}]$. You do not need to include error-handling code. You are free to define any helper functions you find useful. (3)
State the ML type of `B`. (1)
5. Using `B` and a function `L : real list -> real`, which would normally be called `listsum` but we give it here a one-character name, write a function using only 3 non-whitespace characters that given $n : \mathbb{N}$ will calculate $\Sigma_{0 \leq k \leq n} \binom{n}{k} = \binom{n}{0} + \cdots + \binom{n}{n}$. (1)

**END OF PAPER**