

# EECS4312 Assignment 1

Firstname1 Lastname1 (cse99999cse.yorku.ca)

Firstname2 Lastname2 (cse99998cse.yorku.ca)

Firstname3 Lastname3 (cse99997cse.yorku.ca)

September 12, 2014

## Revisions

Date	Revision	Description
23 September 2012	1.0	Initial specification of system, R/E- descriptions, module specifications and function table.
30 September 2012		

## Contents

<b>1</b>	<b>What is a good requirement?</b>	<b>3</b>
<b>2</b>	<b>System Under Description</b>	<b>3</b>
<b>3</b>	<b>R-descriptions and E-descriptions</b>	<b>4</b>
3.1	Subsection heading . . . . .	4
3.2	Signum Function . . . . .	5
<b>4</b>	<b>Bibliography</b>	<b>6</b>

## List of Figures

1	PDF (right) displays better than jpeg or png (right) . . . . .	3
2	(a) Standard mathematical definition of $\textit{sgn}(x)$ . (b) Function table definition	6

## List of Tables

# 1 What is a good requirement?

A requirement is a separately verifiable contractual statement stating a need of the customer

- The customer has certain needs and goals. However, those needs are likely to be vague or incomplete. If you build an actual product, it is hard to see how you (or your customer) would check that the product you built satisfies the customer's needs. What is needed is a **precise requirements document**.
- A precise requirements document describes everything necessary to produce a safe and correct system—one that fulfills the needs of the customer—nothing more.
- The requirements document must provide the software developer with **all** the information needed for the system to be built.
- At the same time the specification must not over-constrain developers by venturing into design and implementation detail.
- The requirements document thus specifies *what* the system will do—not *how* the system will do it!

## 2 System Under Description

Some stuff. Here is a png figure (see folder “pics”)

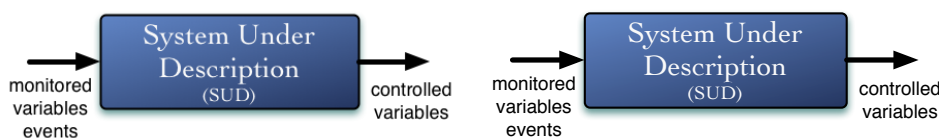


Figure 1: PDF (right) displays better than jpeg or png (right)

See Fig. 1 for how to put an image in a figure environment. The “htb” says put the figure first here, then top of the page, then bottom, depending on layout algorithm.

### 3 R-descriptions and E-descriptions

ENV1	Although it is physically possible for the trip and keypad commands to occur at the same time, the system is hardwired to respond sequentially (i.e. there is an arbitrary choice which one is dealt with first). Thus all signals arrive sequentially.	Refer to where your mathematical model uses this
------	---	--

REQ2	If the system is armed then a trip signal from the motion detector shall result in the siren being sounded.	Refer to mathematical model
------	---	-----------------------------

#### 3.1 Subsection heading

Other stuff with inline maths  $p \Rightarrow q$  and

$$p \wedge q \vee r \Rightarrow z \quad (1)$$

In (1), we see how to write a formula.

Here is a verbatim environment with smaller text:

```
=====
Use Case 1: uc1.txt: adding product types
=====
report:      ok
id:          0
products:
stock:
orders:
carts:
order_state:
->add_type("nuts")
```

```

report:      ok
id:          0
products:    nuts
..

```

Here is some formatted PVS

```

gate: THEORY
BEGIN
  p, q, r: bool
  not_gate(x: bool): bool = NOT x
  and_gate(x:bool, y:bool): bool = x AND y
  or_gate(x:bool, y:bool): bool = x OR y
  check1: CONJECTURE and_gate(p, not_gate(q)) => (p AND NOT q)
  check2: CONJECTURE and_gate(p, not_gate(q)) = (p AND NOT q)

  circuit_implementation(x:bool, y:bool): bool =
    and_gate(
      (or_gate( and_gate(x, not_gate(y))
                , and_gate(x,y))
      , y))

  and_conjecture: CONJECTURE
    and_gate(p,q) = circuit_implementation(p,q)

  or_conjecture: CONJECTURE
    or_gate(p,q) = circuit_implementation(p,q)
END gate

```

Some text

## 3.2 Signum Function

Let's start with a simple example. In mathematics, the sign function or signum (from *signum*, Latin for "sign") function  $sgn(x)$  — is a mathematical function that extracts the sign of a real number. In mathematical expressions the sign

The normal mathematical definition and the table layout are identical in meaning. However, the table layout is easier on the reader. Note that both description methods are **complete** and **disjoint**.

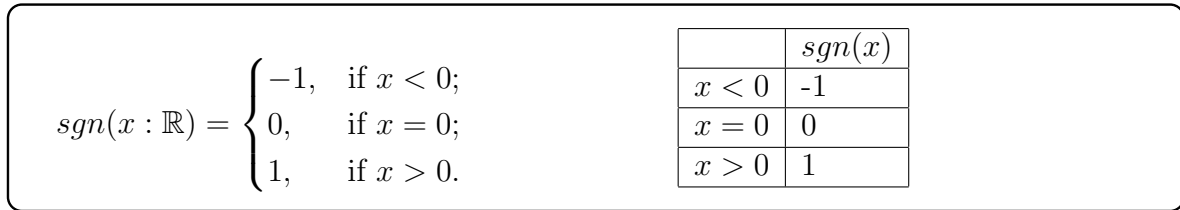


Figure 2: (a) Standard mathematical definition of  $\text{sgn}(x)$ . (b) Function table definition

## Completeness

The description of  $\text{sgn}$  is complete because the function is defined for all possible inputs  $x \in \mathbb{R}$ .

## Disjointness

The description is disjoint because each row in the table describes a case that is independent of the other rows — hence avoiding inconsistencies — where the same condition triggers different (and possibly contradictory) behaviours.

For example, in a computer controlling a robot, we would not want the press of a *move* button to signal to a robot to move right and move left at the same time.

The listing below shows how to describe the function in PVS.<sup>1</sup>

## 4 Bibliography

See this [2], this citeGS93 and this [1]. Run bibtex and then latex three times.

## References

- [1] J.M Spivey. *The Z notation*. Prentice Hall, 1992.
- [2] Axel van Lamsweerde. *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Wiley, 2009.

---

<sup>1</sup>See <https://wiki.eecs.yorku.ca/project/sel-students/p:tutorials:pvs:start>.