# Companion Cube Calculator

Geneva Smith

October 3, 2017

## Contents

**1 Reference Material** **iii**
- 1.1 Table of Units . . . iii
- 1.2 Table of Symbols . . . iii
- 1.3 Abbreviations and Acronyms . . . iv

**2 Introduction** **1**
- 2.1 Purpose of Document . . . 1
- 2.2 Scope of Requirements . . . 1
- 2.3 Characteristics of Intended Reader . . . 2
- 2.4 Organization of Document . . . 2

**3 General System Description** **2**
- 3.1 System Context . . . 2
- 3.2 User Characteristics . . . 3
- 3.3 System Constraints . . . 3

**4 Specific System Description** **3**
- 4.1 Problem Description . . . 3
  - 4.1.1 Terminology and Definitions . . . 4
  - 4.1.2 Physical System Description . . . 4
  - 4.1.3 Goal Statements . . . 5
- 4.2 Solution Characteristics Specification . . . 5
  - 4.2.1 Assumptions . . . 5
  - 4.2.2 Theoretical Models . . . 5
  - 4.2.3 General Definitions . . . 8
  - 4.2.4 Data Definitions . . . 8
  - 4.2.5 Instance Models . . . 8
  - 4.2.6 Data Constraints . . . 14
  - 4.2.7 Properties of a Correct Solution . . . 15

**5 Requirements** **15**
5.1 Functional Requirements . . . . . 15
5.2 Non-functional Requirements . . . . . 15

**6 Likely Changes** **17**

**7 Traceability Matrices and Graphs** **17**

Table 1: **Revision History**

| Date | Version | Notes |
|---|---|---|
| September 26, 2017 | 1.1 | Completed the first draft of the General System Description (Section 3) |
| September 25, 2017 | 1.0 | Completed the first draft of the Introduction (Section 2) |

# 1 Reference Material

This section records information for easy reference.

## 1.1 Table of Units

The tasks performed by the $C^3$ tool are unitless.

## 1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made to be consistent with the heat transfer literature and with existing documentation for solar water heating systems. The symbols are listed in alphabetical order.

| Symbol | Unit | Description |
|---|---|---|
| $b$ | - | Base number; used to represent exponential functions (e.g. $b^2$) |
| $D(x)$ | - | Domain of $x$ |
| $f(X)$ | - | A function of a variable set $X$ |
| $\mathbb{N}$ | - | The set of natural numbers; assumes that 0 is included in this specification |
| $\mathbb{R}$ | - | The set of real numbers |
| $R(f(X))$ | - | Range of $f(X)$ |
| $n$ | - | Power; used to represent exponential functions (e.g. $2^n$) |

## 1.3 Abbreviations and Acronyms

| Text | Description |
|---|---|
| A | Assumption |
| CRPG | Computer Role-Playing Game |
| DD | Data Definition |
| GD | General Definition |
| GS | Goal Statement |
| GUI | Graphical User Interface |
| IM | Instance Model |
| LC | Likely Change |
| NPC | Non-Player Character |
| PS | Physical System Description |
| R | Requirement |
| SRS | Software Requirements Specification |
| $C^3$ | Companion Cube Calculator |
| T | Theoretical Model |

# 2 Introduction

This document is an SRS for the Companion Cube Calculator ($C^3$), a mathematical tool which determines the range of a user-specified function given the domains of the function's variables. This tool is being developed to aid in the specification and refinement of GLaDOS, an emotion engine for Non-Player Characters (NPCs) in Computer Role-Playing Games (CRPG) as described by Smith (2017).

## 2.1 Purpose of Document

This document outlines the requirements identified for the development of the $C^3$ tool, including the product goals, product scope, and the mathematical models driving the design. It also describes the mathematical assumptions, theories, and models used to create the tool. The purpose of documenting this information is to aid in future use, maintenance, and development of the $C^3$ tool.

This document is intended for two reader types – those who wish to use the tool and those who wish to expand the tool. Even though the Companion Cube Calculator was created to aid in the development of a specific system (GLaDOS), the tested functions do not have any specific units. This means that it can be used for any function that exists in the domain of real numbers ($\mathbb{R}$). Therefore, this document can be used by a user who wishes to use the $C^3$ tool to determine the range of a specified mathematical function. Since the initial development of the $C^3$ tool will be limited to arithmetic operators, this document includes information that will be useful to a developer looking to expand the abilities of the $C^3$ with additional mathematical models such as trigonometry to suit their project.

## 2.2 Scope of Requirements

The Companion Cube Calculator calculates the mathematical range of a user-defined function using the defined variable domains for that function. In the cases where a variable domain is not provided, it will be assumed that the range is $(-\infty, \infty)$. For the initial version of this product, the mathematical operations allowed in an function will be limited to the four arithmetic operators ($+$, $-$, $\times$, $\div$, exponents). Future iterations can expand this list to include trigonometric functions (sin, cos, tan, arcsin, arccos, arctan), partial functions, and other useful operations and functions.

The purpose of this product is to aid in the design and tuning of the GLaDOS architecture, a specialized game engine enables game designers to create Non-Player Characters (NPCs) that react to their environment by using models of emotion from psychology. One module in the architecture converts information from the environment into an internal representation that directs an NPC's decision-making. This task requires the specification of numerical functions with multiple variables which must be normalized to a range of $[-1, 1]$. In order to normalize the engine's calculation, the range of the function must be known. Currently, the encoded functions are not well-informed by observation or scientific research and must be subjected to an iterative process in order to address this problem. An automated method of calculating the range of a proposed function will make this process faster and less error-prone.

Although this product is being designed for a specific project, the concepts involved can be used in similar projects because it does not contain any project-specific concepts or models.

## 2.3 Characteristics of Intended Reader

The intended reader of this document must understand elementary algebra, especially interval arithmetic, in the domain of real numbers ($\mathbb{R}$). They must also have an understanding of domain and range with respect to a function in order to understand the outputs of the product and how it relates to its inputs.

## 2.4 Organization of Document

This document begins by describing the general description of the Companion Cube Calculator tool (Section 3), which includes the system context, constraints, and the intended users' characteristics. This is followed by a description of the problem to be solved, including the models and assumptions that are used to solve it (Section 4). This description is followed by the functional and non-functional requirements of the $C^3$ tool (Section 5) and the potential changes that will be made to them in future iterations (Section 6). The last section in this document, traceability matrices and graphs, visually describes the dependencies between different document components (Section 7). This information should be used by making changes to the decisions made for this version of the tool.

# 3 General System Description

This section identifies the interfaces between the system and its environment, describes the user characteristics and lists the system constraints.

## 3.1 System Context

The Companion Cube Calculator tool is a stand-alone application for calculating the range of a user-provided function given the known domains of the function's component variables. Therefore it is independent and self-contained with respect to external organizations, products, and technologies.

The user interface of the $C^3$ tool facilitates communication between the product and the user, and must contain the ability to exchange user inputs and system outputs. In general, the user is responsible for ensuring that they have provided a semantically correct function and that their inputs do not contain mathematical operations that are not implemented in the tool. The Companion Cube Calculator is responsible for providing an information exchange interface between itself and the user, performing mathematical calculations, and detecting syntactic errors in the system inputs.

- User Responsibilities:

  - Provide an function from the domain of real numbers ($\mathbb{R}$) that only contains mathematical operations that are recognized by the system
  - Provide the mathematical domains of the function's variables if they are known
  - Ensure that the function given to the tool does not contain semantic errors
  - Ensure that the products outputs are semantically correct with respect to the target application

- Companion Cube Calculator Responsibilities:
  - Allow the user to enter their function and known variable domains as system inputs
  - For variables with unknown domains, provide default values to be used in calculations
  - Detect data type mismatch, such as a string of characters instead of a floating point number
  - Detect bracket mismatches
  - Calculate and output the mathematical range corresponding to the user's inputs; if a result cannot be calculated, communicate this to the user

## 3.2 User Characteristics

The end user of Companion Cube Calculator is also the intended reader (Section 2.3) of this document.

## 3.3 System Constraints

The use of a Graphical User Interface (GUI) is useful for the $C^3$ tool because it can help the user visualize their function during design time and debugging. This has implications on the types of languages that this tool will be developed in and target platform choices. If a GUI is included in the $C^3$ design, only Windows platforms will be supported in the initial release.

# 4 Specific System Description

This section first presents the problem description, which gives a high-level view of the motivation behind the Companion Cube Calculator. This is followed by the solution characteristics specification, presenting the assumptions, theories, definitions, and instance models, which are grounded in interval arithmetic.

## 4.1 Problem Description

The GLaDOS architecture is a specialized game engine that enables game designers to create Non-Player Characters (NPCs) that react to their environment by using models of emotion from

psychology. A key component of the architecture is the primary appraisal module, where inputs from the environment are converted into emotion values. The functions currently used for this task are not well informed by any scientific research, and will be incrementally revised as more information is collected. These new functions are required to conform to the same mathematical range as the currently implemented functions. Ensuring that this requirement is maintained over consecutive iterations of the functions will become tedious and time consuming, distracting from the main task of developing more rigorous and explainable emotion functions.

The Companion Cube Calculator is a tool for calculating the range of a mathematical function based on its input variables, which will help make the function design process faster and less error-prone when compared to doing it by hand.

### 4.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

- Closed interval: A bounded set that includes its end points; this is expressed $[a, b]$
- Closed interval method: A mathematical method for determining the local maximum and minimum values of a function $f(x)$ using a closed domain interval $[a, b]$
- Connected set: a set of values that does not contain any disjoint members
- Domain ($D(x)$): the set of values for a variable $x$ that are valid in $f(X)$
- Extended real interval: the set of all real numbers that also contains $\pm\infty$
- Open interval: A bounded set of values that does not contain its end points; this is expressed $(a, b)$
- Order of Operations: Describes the precedence of mathematical operations when solving an equation (Brackets, Exponents, Division/Multiplication, Addition/Subtraction)
- Range ($R(f(X))$): the set of values that are produced by a function $f(X)$ with a set of input variables $X$
- Real interval: a closed, connected set of real numbers

### 4.1.2 Physical System Description

The Companion Cube Calculator tool does not have a physical system component because it exists independently of the context of $f(X)$.

### 4.1.3 Goal Statements

Given an function and a set of variables, the goal statements are:

GS1: Calculate $R(f(X))$.

## 4.2 Solution Characteristics Specification

The instance models that govern Companion Cube Calculator are presented in Subsection 4.2.5. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

### 4.2.1 Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the theoretical model [T], general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

A1: $D(x)$ is a closed, real interval.

A2: $R(f(X))$ is a closed, real interval.

A3: The mathematical operators used in $f(X)$ can be found in the set $\{+, -, \times, \div, b^x, x^n\}$.

A4: Special mathematical functions (e.g. $\sin(x)$, $\cos(x)$, $\log(x)$, ...) are not in $f(X)$.

### 4.2.2 Theoretical Models

This section focuses on the general equations and laws from interval arithmetic that the Companion Cube Calculator is based on. The models presented here are for real intervals, which can be constrained to closed, real intervals using DD1 to satisfy A1 and A2. Additional models exist for interval arithmetic, but only the models named in A3 are mentioned.

| Number | T1 |
|---|---|
| Label | **Addition on Real Intervals** |
| Equation | $X + Y = \{x + y \mid x \in X \wedge y \in Y\}$ |
| Description | The summation interval of two real intervals $X$ and $Y$ is equal to the set of sums for each pairwise element $x$ and $y$. |
| Source | Hickey et al. (2001) |
| Ref. By | IM1 |

| Number | T2 |
|---|---|
| Label | **Subtraction on Real Intervals** |
| Equation | $X - Y = \{x - y \mid x \in X \wedge y \in Y\}$ |
| Description | The difference interval of two real intervals $X$ and $Y$ is equal to the set of differences for each pairwise element $x$ and $y$. |
| Source | Hickey et al. (2001) |
| Ref. By | IM2 |

| Number | T3 |
|---|---|
| Label | **Multiplication on Real Intervals** |
| Equation | $X \times Y = \{x \times y \mid x \in X \wedge y \in Y\}$ |
| Description | The product interval of two real intervals $X$ and $Y$ is equal to the set of products for each pairwise element $x$ and $y$. |
| Source | Hickey et al. (2001) |
| Ref. By | IM3 |

| Number | T4 |
|---|---|
| Label | **Division on Real Intervals** |
| Equation | $X \div Y = \{z \mid \exists x \in X, y \in Y \wedge y \neq 0, z = x \div y\}$ |
| Description | The quotient interval of two real intervals $X$ and $Y$ is equal to the set of quotients for each pairwise element $x$ and $y$ where $y \neq 0$. If $0 \in Y$, the quotient is not a real interval which contradicts A2. |
| Source | Hickey et al. (2001) |
| Ref. By | IM4, IM5 |

| Number | T5 |
|---|---|
| Label | **Real Interval Exponents on a Constant Base Number** |
| Equation | $b^X = \{b^x \mid x \in X \wedge b > 1\}$ |
| Description | The product interval of a constant number $b > 1$ raised to the power of a real interval $X$ is equal to the set of products $b^x$ for each interval element $x$. |
| Source | https://en.wikipedia.org/wiki/Interval_arithmetic |
| Ref. By | IM6 |

| Number | T6 |
|---|---|
| Label | **Constant Exponents on a Real Interval Base Number** |
| Equations | $X^n = \begin{cases} \{x^n \mid x \in X \wedge n \in \mathbb{N}\} & \text{n is odd} \\ \{[x_1^n, ..., x_z^n] \mid x \in X \wedge n \in \mathbb{N}\} & \text{n is even} \wedge x_z \geq x_{z+1} \wedge x_1 > 0 \\ \{[x_z^n, ..., x_1^n] \mid x \in X \wedge n \in \mathbb{N}\} & \text{n is even} \wedge x_z < x_{z+1} \wedge x_Z < 0 \\ \{[0, ..., max\{x_1^n, x_z^n\}] \mid x \in X \wedge n \in \mathbb{N}\} & \text{ELSE} \end{cases}$ |
| Description | The product interval of a real interval $X$ raised to the power of a constant, natural number $n$ depends on the properties of $X$ and $n$:<br>• When $n$ is an odd number, the product interval is equal to the set of products $x^n$ for each element $x$<br>• When $n$ is even, the function is monotonically decreasing, and $x_1 > 0$, the product interval is equal to the set of products $x^n$ for each element $x$<br>• When $n$ is even, the function is monotonically increasing, and $x_Z < 0$, the product interval is equal to the set of products $x^n$ for each element $x$ and ordered in the opposite direction of the original interval $X$<br>• Otherwise, the product interval is bounded by 0 and the maximum product of the first and last elements in the interval set |
| Source | https://en.wikipedia.org/wiki/Interval_arithmetic |
| Ref. By | IM7 |

### 4.2.3 General Definitions

No additional information is required to build the data definitions.

### 4.2.4 Data Definitions

This section collects and defines all the data needed to build the instance models. No dimensions are required because the tasks performed by Companion Cube Calculator are unitless. These definitions are used to constrain the theoretical models to such that thay can be translated into instance models on closed, real intervals to satisfy A1 and A2.

| Number | DD1 |
|---|---|
| Label | **Representation of a Closed, Real Interval** |
| Symbol | $[x, y]$ |
| SI Units | - |
| Equation | - |
| Description | The definition $[x, y]$ represents a closed, real interval (A1, A2) with endpoints $x$ and $y$. |
| Sources | - |
| Ref. By | IM1, IM2, IM3, IM4, IM5, IM6, IM7 |

### 4.2.5 Instance Models

This section transforms the problem defined in Section 4.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 4.2.4 to replace the abstract symbols in the models identified in Sections 4.2.2 and 4.2.3.

The goal GS1 is solved by recursively combining instance models IM1, IM2, IM3, IM4, IM5, IM6, and IM7. For example, if the user provided an equation of the form $a+b\times c$, it is decomposed into $a + B$ where $B = b \times c$. This approach is identical to a hand-written evaluation because a mechanized approach must explicitly state the intermediary steps, which are commonly omitted by humans, in order to complete the computational task.

These models can also be used as a specification for the closed interval method because of the closed, real interval assumption on $D(x)$ (A1).

| Number | IM1 |
|---|---|
| Label | **Addition of closed, real intervals** |
| Input | $[x_1, y_1]$, $[x_2, y_2]$ from DD1 |
| Output | $[x_{sum}, y_{sum}]$ such that<br>$x_{sum} = x_1 + x_2$, $y_{sum} = y_1 + y_2$,<br>$[x_{sum}, y_{sum}]$ is a closed, real interval (A2) |
| Description | $[x_1, y_1]$ and $[x_2, y_2]$ are the domains of two variables in the user's input function $f(X)$.<br>The combined boundary values $x_{sum}$ and $y_{sum}$ are determined using basic arithmetic addition.<br>The result, $[x_{sum}, y_{sum}]$, is guaranteed to be a closed, real interval (A2) due to the closed, real interval constraint on $[x_1, y_1]$ and $[x_2, y_2]$ (A1). |
| Sources | Hickey et al. (2001) |
| Ref. By | - |

| Number | IM2 |
|---|---|
| Label | **Subtraction of closed, real intervals** |
| Input | $[x_1, y_1]$, $[x_2, y_2]$ from DD1 |
| Output | $[x_{sub}, y_{sub}]$ such that<br>$x_{sub} = x_1 - x_2$, $y_{sub} = y_1 - y_2$,<br>$[x_{sub}, y_{sub}]$ is a closed, real interval (A2) |
| Description | $[x_1, y_1]$ and $[x_2, y_2]$ are the domains of two variables in the user's input function $f(X)$.<br>The combined boundary values $x_{sub}$ and $y_{sub}$ are determined using basic arithmetic subtraction.<br>The result, $[x_{sub}, y_{sub}]$, is guaranteed to be a closed, real interval (A2) due to the closed, real interval constraint on $[x_1, y_1]$ and $[x_2, y_2]$ (A1). |
| Sources | Hickey et al. (2001) |
| Ref. By | - |

| Number | IM3 |
|---|---|
| Label | **Multiplication of closed, real intervals** |
| Input | $[x_1, y_1]$, $[x_2, y_2]$ from DD1 |
| Output | $[x_{mul}, y_{mul}]$ such that<br>$x_{mul} = \min(M)$, $y_{mul} = \max(M)$, where $M = \{x_1 \times x_2, x_1 \times y_2, y_1 \times x_2, y_1 \times y_2\}$<br>$[x_{mul}, y_{mul}]$ is a closed, real interval (A2) |
| Description | $[x_1, y_1]$ and $[x_2, y_2]$ are the domains of two variables in the user's input function $f(X)$.<br>The combined boundary value $x_{mul}$ is determined by taking the minimum of all possible products calculated from the sets $[x_1, y_1]$ and $[x_2, y_2]$, where $[x_1, y_1]$ is the multiplier set and $[x_2, y_2]$ is the multiplicand set<br>The combined boundary value $y_{mul}$ is determined by taking the maximum of all possible products calculated from the sets $[x_1, y_1]$ and $[x_2, y_2]$, where $[x_1, y_1]$ is the multiplier set and $[x_2, y_2]$ is the multiplicand set<br>The result, $[x_{mul}, y_{mul}]$, is guaranteed to be a closed, real interval (A2) due to the closed, real interval constraint on $[x_1, y_1]$ and $[x_2, y_2]$ (A1). |
| Sources | Hickey et al. (2001) |
| Ref. By | - |

| Number | IM4 |
|---|---|
| Label | **Division of closed, real intervals (Divisor is a positive interval)** |
| Input | $[x_1, y_1]$, $[x_2, y_2]$ from DD1 where $0 \leq x_2 \leq y_2$ and $x_2 \neq 0$ |
| Output | $[x_{div}, y_{div}]$ such that<br>$x_{div} = x_1 \div y_2, \quad y_{div} = y_1 \div x_2 \quad 0 < x_1 \leq y_1$<br>$x_{div} = 0, \quad y_{div} = y_1 \div x_2 \quad x_1 = 0 \wedge x_1 \leq y_1$<br>$x_{div} = x_1 \div x_2, \quad y_{div} = y_1 \div x_2 \quad x_1 < 0 < y_1$<br>$x_{div} = x_1 \div x_2, \quad y_{div} = 0 \quad y_1 = 0, x_1 \leq y_1$<br>$x_{div} = x_1 \div x_2, \quad y_{div} = y_1 \div y_2 \quad x_1 \leq y_1 < 0$<br>$[x_{div}, y_{div}]$ is a closed, real interval (A2) |
| Description | $[x_1, y_1]$ and $[x_2, y_2]$ are the domains of two variables in the user's input function $f(X)$.<br>The boundary values $x_{div}$ and $y_{div}$ are determined by the monotonicity of the interval represented by $[x_1, y_1]$ and whether or not that interval contains 0.<br>The result, $[x_{div}, y_{div}]$, is guaranteed to be a closed, real interval (A2) due to the closed, real interval constraint on $[x_1, y_1]$ and $[x_2, y_2]$ (A1). |
| Sources | Hickey et al. (2001) |
| Ref. By | - |

| Number | IM5 |
|---|---|
| Label | **Division of closed, real intervals (Divisor is a negative interval)** |
| Input | $[x_1, y_1]$, $[x_2, y_2]$ from DD1 where $x_2 \leq y_2 \leq 0$ and $y_2 \neq 0$ |
| Output | $[x_{div}, y_{div}]$ such that<br>$x_{div} = y_1 \div y_2, \quad y_{div} = x_1 \div x_2 \quad 0 < x_1 \leq y_1$<br>$x_{div} = y_1 \div y_2, \quad y_{div} = 0 \quad x_1 = 0 \wedge x_1 \leq y_1$<br>$x_{div} = y_1 \div y_2, \quad y_{div} = x_1 \div y_2 \quad x_1 < 0 < y_1$<br>$x_{div} = 0, \quad y_{div} = x_1 \div y_2 \quad y_1 = 0, x_1 \leq y_1$<br>$x_{div} = y_1 \div x_2, \quad y_{div} = x_1 \div y_2 \quad x_1 \leq y_1 < 0$<br>$[x_{div}, y_{div}]$ is a closed, real interval (A2) |
| Description | $[x_1, y_1]$ and $[x_2, y_2]$ are the domains of two variables in the user's input function $f(X)$.<br>The boundary values $x_{div}$ and $y_{div}$ are determined by the monotonicity of the interval represented by $[x_1, y_1]$ and whether or not that interval contains 0.<br>The result, $[x_{div}, y_{div}]$, is guaranteed to be a closed, real interval (A2) due to the closed, real interval constraint on $[x_1, y_1]$ and $[x_2, y_2]$ (A1). |
| Sources | Hickey et al. (2001) |
| Ref. By | - |

| Number | IM6 |
|---|---|
| Label | **Closed, real intervals as Exponents** |
| Input | $[x, y]$ from DD1, $b$ |
| Output | $[x_{exp}, y_{exp}]$ such that<br>$x_{exp} = b^x$, $y_{exp} = b^y$<br>$[x_{exp}, y_{exp}]$ is a closed, real interval (A2) |
| Description | $[x, y]$ is the domain of one variable in the user's input function $f(X)$ which is used as an exponent on a constant base number $b$.<br>The combined boundary values $x_{exp}$ and $y_{exp}$ are determined using basic arithmetic exponents.<br>The result, $[x_{exp}, y_{exp}]$, is guaranteed to be a closed, real interval (A2) due to the closed, real interval constraint on $[x, y]$ and the data constraint on $b$ (A1). |
| Sources | https://en.wikipedia.org/wiki/Interval_arithmetic |
| Ref. By | - |

| Number | IM7 |
|---|---|
| Label | **Powers of closed, real intervals** |
| Input | $[x, y]$ from DD1, $n$ |
| Output | $[x_{pow}, y_{pow}]$ such that<br>$x_{pow} = x^n, \quad y_{pow} = y^n \quad n$ is odd<br>$x_{pow} = x^n, \quad y_{pow} = y^n \quad n$ is even $\wedge (0 \leq x < y)$<br>$x_{pow} = y^n, \quad y_{pow} = x^n \quad n$ is even $\wedge (0 \geq x > y)$<br>$x_{pow} = 0, \quad y_{pow} = \max(x^n, y^n) \quad$ ELSE<br>$[x_{pow}, y_{pow}]$ is a closed, real interval (A2) |
| Description | $[x, y]$ is the domain of one variable in the user's input function $f(X)$ raised to a constant power $n$.<br>The combined boundary values $x_{pow}$ and $y_{pow}$ are determined by the divisibility of $n$ by two and the monotonicity of $[x, y]$:<br>• For odd values of $n$ and any interval $[x, y]$, and even values of $n$ when $[x, y]$ is monotonically increasing, $[x_{pow}, y_{pow}]$ are determined by simply raising each of $x$ and $y$ to the power of $n$<br>• For even values of $n$ and $[x, y]$ is monotonically decreasing, the boundary values of $x$ and $y$ are reversed before being raised to the power of $n$<br>• For all other cases, the interval is bounded by 0 and the maximum of the input bounds raised to the power of $n$<br>The result, $[x_{exp}, y_{exp}]$, is guaranteed to be a closed, real interval (A2) due to the closed, real interval constraint on $[x, y]$ and the data constraint on $n$ (A1). |
| Sources | https://en.wikipedia.org/wiki/Interval_arithmetic |
| Ref. By | - |

#### 4.2.6 Data Constraints

The inputs to the $C^3$ tool are subject to the following data constraints:

- The function $f(X)$ contains only the mathematical operators in the set $\{()\} \cup \{+, -, \times, \div, b^x, x^n\}$ (A3)
- If $f(X)$ contains $b^x$, then $b > 1 \in \mathbb{R}$

- If $f(X)$ contains $x^n$, then $n \in \mathbb{N}$
- For every variable $x \in X$, $D(x)$ is defined and is a closed, real interval (A1)

The output of the $C^3$ tool is subject to the following data constraint:

- The interval $R(f(X))$ is defined and is a closed, real interval (A2)

### 4.2.7 Properties of a Correct Solution

The interval of $R(f(X))$ produced by the $C^3$ tool must exhibit the properties of a closed, real interval (A2). This can be verified mathematically by checking if $R(f(X)) = [a, b]$ satisfies $\forall x \in \mathbb{R} | a \leq x \leq b$.

# 5 Requirements

This section provides the functional requirements, the business tasks that the $C^3$ tool is expected to complete, and the non-functional requirements, the qualities that the $C^3$ tool is expected to exhibit.

## 5.1 Functional Requirements

R1: The $C^3$ tool must accept an equation $f(X)$ from the user and a domain for each $x \in X$. These inputs can be entered directly into the tool or read from a file.

R2: The $C^3$ tool must convert each domain $x \in X$ into interval form (DD1).

R3: The $C^3$ tool must decompose $f(X)$ into a series of two-operand equations following the standard order of operations rules.

R4: The $C^3$ tool must verify that the user inputs satisfy the input data constraints from 4.2.6.

R5: The $C^3$ tool must solve each equation identified in R3 using IM1, IM2, IM3, IM4, IM5, IM6, and IM7.

R6: The $C^3$ tool must verify that the program produces a $R(f(X))$ in interval notation (DD1).

R7: The $C^3$ tool must verify that the program output satisfies the output data constraints from 4.2.6.

R8: The $C^3$ tool must show the verified $R(f(X))$ to the user.

## 5.2 Non-functional Requirements

**Correctness**

- The $C^3$ tool must be correct in its decomposition of $f(X)$ into smaller equations.
  **Fit Criterion**: The decomposition process can be proven using mathematical induction.

**Reliability**

- The $C^3$ tool must calculate the value of $R(f(X))$ with an error rate relative to the floating point precision available on the host machine.
  **Fit Criterion**: Given the floating point error on a test machine, verify that the output produced by the program is within the error range when compared to a manually calculated result.

**Robustness**

- The $C^3$ tool must be able to recognize violated data constraints (4.2.6) and report them to the user.

**Performance**

Time and space performance is not a priority because the intended users will be using the $C^3$ tool in an environment with unrestricted time resources and the space required to perform its calculations will be relatively small in reasonable implementations.

**Verifiability**

- The $C^3$ tool must be verifiable with respect to the correctness of its calculations.
  **Fit Criterion**: The calculation procedures used by the $C^3$ tool must be written such that they can be verified using mathematical proofs.

**Usability**

- The user must be able to enter $f(X)$ using the standard mathematical format.
- The user must be able to enter values for $D(x)$ using standard interval notation.
- The program must output the value $R(f(X))$ using standard interval notation.
- If the design of the tool includes a GUI (3.3), it must be organized such that it aids in the user's understanding of the order of program inputs.
  **Fit Criterion**: The intended user should know what inputs are required at each processing stage without referring to the product documentation.

**Maintainability**

- The evolvability of the $C^3$ tool must allow the addition of open, real intervals identified in LC1.
- The evolvability of the $C^3$ tool must allow the addition of mathematical operators and special functions identified in LC3 and LC4.

**Reusability**

Reusability is not a priority because there are currently no future products that will rely on elements from the $C^3$ tool.

**Portability**

The portability of the $C^3$ tool is not a priority because it is expected that the intended users of the product will not expect it to run on many different platforms. One commonly used platform is sufficient.

# 6 Likely Changes

LC1: Open, real intervals are supported (A1, A2)

LC2: Determine $D(x)$ for any $x \in X$ if it is not part of the input set (A1, A2)

LC3: Addition of the trigonometric functions $\sin(x)$ and $\cos(x)$ (A1, A4)

LC4: Addition of operators and trigonometric functions whose range includes $\pm\infty$ (A1, A2, A3, A4)

# 7 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an "X" may have to be modified as well. Table 3 shows the dependencies of theoretical models, data definitions, and instance models with each other. Items from the likely changes section are not included in this table because implementing the likely changes will require the creation of new models and cannot be adequately addressed by changing the existing models. Table 4 shows the dependencies of instance models, requirements, and data constraints on each other. Table 2 shows the dependencies of theoretical models, data definitions, instance models, and likely changes on the assumptions. Since there is only one goal statement (GS1) in this specification, it can be assumed that changing the goal will impact every item contained within this document.

| | A1 | A2 | A3 | A4 |
|---|---|---|---|---|
| T1 | | | | |
| T2 | | | | |
| T3 | | | | |
| T4 | | | | |
| T5 | | | | |
| T6 | | | | |
| DD1 | X | X | | |
| IM1 | | | | |
| IM2 | | | | |
| IM3 | | | | |
| IM4 | | | | |
| IM5 | | | | |
| IM6 | | | | |
| IM7 | | | | |
| LC1 | X | X | | |
| LC2 | X | X | | |
| LC3 | X | | | X |
| LC4 | X | X | X | X |

Table 2: Traceability Matrix Showing the Connections Between Assumptions and Other Items

| | T1 | T2 | T3 | T4 | T5 | T6 | DD1 | IM1 | IM2 | IM3 | IM4 | IM5 | IM6 | IM7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T1 | | | | | | | | | | | | | | |
| T2 | | | | | | | | | | | | | | |
| T3 | | | | | | | | | | | | | | |
| T4 | | | | | | | | | | | | | | |
| T5 | | | | | | | | | | | | | | |
| T6 | | | | | | | | | | | | | | |
| DD1 | | | | | | | | | | | | | | |
| IM1 | X | | | | | | X | | | | | | | |
| IM2 | | X | | | | | X | | | | | | | |
| IM3 | | | X | | | | X | | | | | | | |
| IM4 | | | | X | | | X | | | | | | | |
| IM5 | | | | X | | | X | | | | | | | |
| IM6 | | | | | X | | X | | | | | | | |
| IM7 | | | | | | X | X | | | | | | | |

Table 3: Traceability Matrix Showing the Connections Between Items of Different Sections

| | IM1 | IM2 | IM3 | IM4 | IM5 | IM6 | IM7 | DD1 | 4.2.6 | R3 |
|---|---|---|---|---|---|---|---|---|---|---|
| IM1 | | | | | | | | | | |
| IM2 | | | | | | | | | | |
| IM3 | | | | | | | | | | |
| IM4 | | | | | | | | | | |
| IM5 | | | | | | | | | | |
| IM6 | | | | | | | | | | |
| IM7 | | | | | | | | | | |
| DD1 | | | | | | | | | | |
| R1 | | | | | | | | | | |
| R2 | | | | | | | | X | | |
| R3 | | | | | | | | | | |
| R4 | | | | | | | | | X | |
| R5 | X | X | X | X | X | X | X | | | X |
| R6 | | | | | | | | X | | |
| R7 | | | | | | | | | X | |
| R8 | | | | | | | | | | |

Table 4: Traceability Matrix Showing the Connections Between Requirements and Instance Models

The purpose of the traceability graphs is also to provide easy references on what has to be additionally modified if a certain component is changed. The arrows in the graphs represent dependencies. The component at the tail of an arrow is depended on by the component at the head of that arrow. Therefore, if a component is changed, the components that it points to should also be changed. Figure 1 is a companion to Table 2 and 3, showing the dependencies of theoretical models, data definitions, instance models, likely changes, and assumptions on each other. Diagram elements representing assumption nodes and traces are coloured to aid in its readability. Figure 2 is a companion to Table 4, showing the dependencies of instance models, requirements, and data constraints on each other.

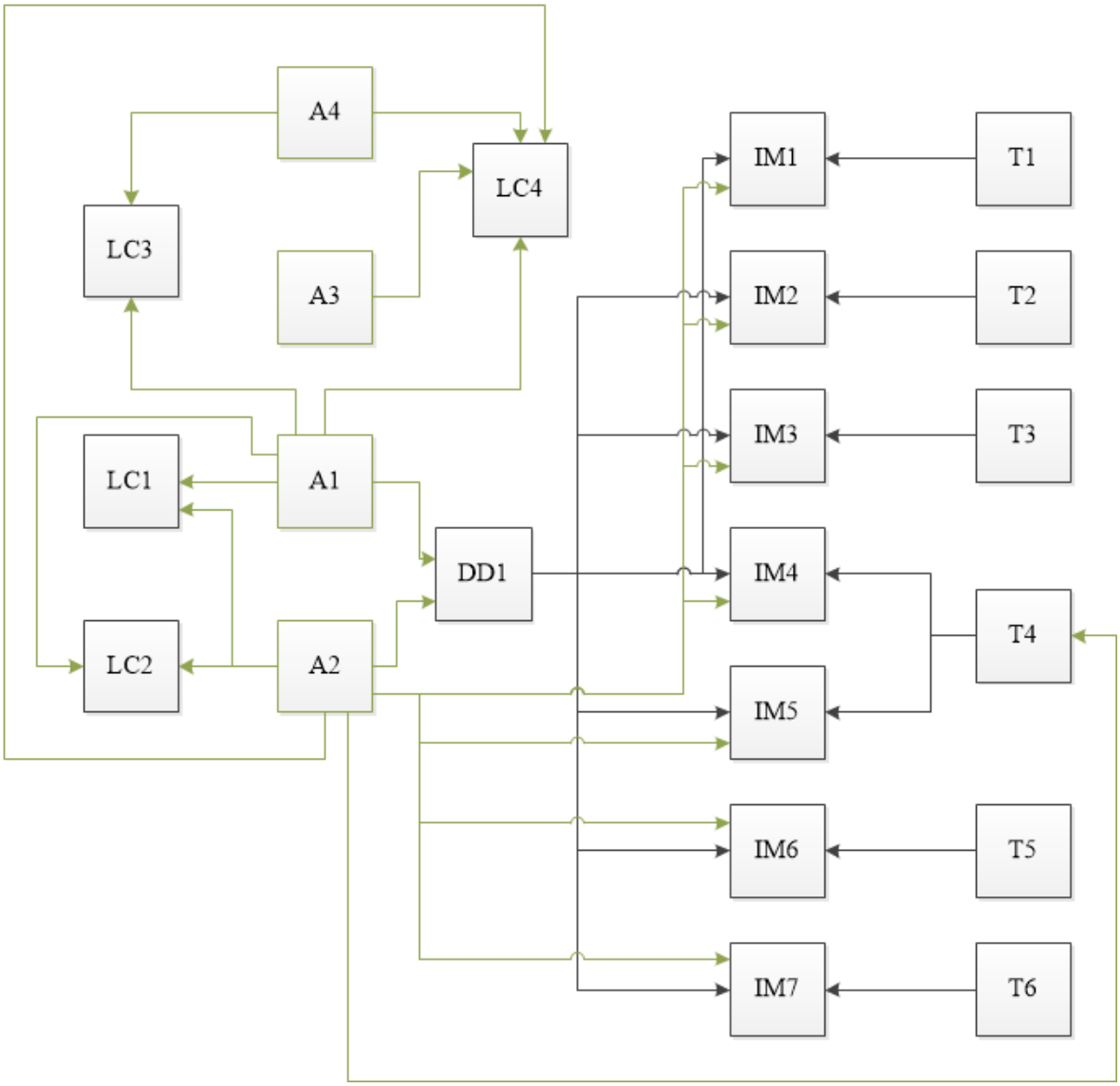


Figure 1: Traceability Matrix Showing the Connections Between Items of Different Sections

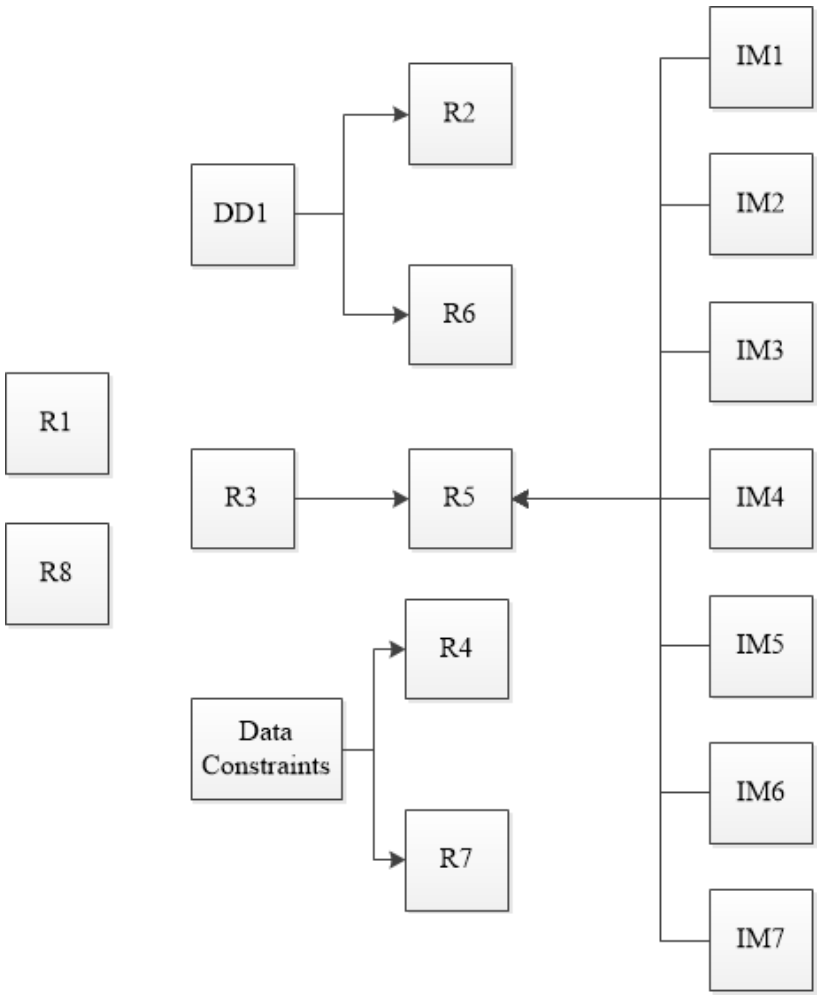


Figure 2: Traceability Matrix Showing the Connections Between Requirements, Instance Models, and Data Constraints

# References

T. Hickey, Q. Ju, and M. H. Van Emden. Interval arithmetic: From principles to implementation. *J. ACM*, 48(5):1038–1068, September 2001. URL http://doi.acm.org/10.1145/502102.502106.

Geneva Smith. GLaDOS: Integrating Emotion-Based Behaviours into Non-Player Characters in Computer Role-Playing Games. M.A.Sc. (Software Engineering) Thesis, McMaster University, April 2017. URL http://hdl.handle.net/11375/21369.