Companion Cube Calculator

Geneva Smith

October 14, 2017

1 Revision History

Date	Version	Notes
October 14, 2017	1.0.1	Added type information to the Table of Units; updated document template
October 4, 2017	1.0	Completed the initial SRS documentation

2 Reference Material

This section records information for easy reference.

2.1 Table of Units

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

2.2 Table of Symbols

The table that follows summarizes the symbols used in this document. The choice of symbols was made to be consistent with mathematical notations of functions, sets, and intervals. The symbols are listed in alphabetical order.

Symbol	Type	Description
$[a, b]$	$[\mathbb{R}, \mathbb{R}]$	Closed interval with endpoints a and b
(a, b)	$(\mathbb{R}, \mathbb{R})$	Open interval with endpoints a and b
b	$\mathbb{R}$	Base number; used to represent exponential functions (e.g. b^2)
$D(x)$	$[\mathbb{R}, \mathbb{R}]$	Closed domain of a variable x
$f(X)$	$\{[\mathbb{R}, \mathbb{R}]_0, ..., [\mathbb{R}, \mathbb{R}]_n\} \rightarrow [\mathbb{R}, \mathbb{R}]$	A function of X that produces a closed interval
M	-	A temporary variable for aiding in the readability of equations
$\mathbb{N}$	-	The set of natural numbers; assumes that 0 is included in this specification
$\mathbb{R}$	-	The set of real numbers
$R(f(X))$	$[\mathbb{R}, \mathbb{R}]$	Closed range of $f(X)$
n	$\mathbb{N}$	Power; used to represent exponential functions (e.g. 2^n)
X	$\{[\mathbb{R}, \mathbb{R}]_0, ..., [\mathbb{R}, \mathbb{R}]_n\}$	Set of $D(x)$ for variables x
Y	$\{[\mathbb{R}, \mathbb{R}]_0, ..., [\mathbb{R}, \mathbb{R}]_n\}$	Set of $D(y)$ for variables y

[Type information would make your SRS easier to understand. That is, giving the type of X, $f(X)$ etc. The type of x is really unclear. Is it reals? —SS]

2.3 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 (Games)
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
C^3	Companion Cube Calculator
T	Theoretical Model

Contents

1 Revision History **i**

2 Reference Material **ii**
2.1 Table of Units ii
2.2 Table of Symbols ii
2.3 Abbreviations and Acronyms iii

3 Introduction **1**
3.1 Purpose of Document 1
3.2 Scope of Requirements 1
3.3 Characteristics of Intended Reader 1
3.4 Organization of Document 2

4 General System Description **2**
4.1 System Context 2
4.2 User Characteristics 3
4.3 System Constraints 3

5 Specific System Description **3**
5.1 Problem Description 3
5.1.1 Terminology and Definitions 4
5.1.2 Physical System Description 4
5.1.3 Goal Statements 4
5.2 Solution Characteristics Specification 4
5.2.1 Assumptions 5
5.2.2 Theoretical Models 5
5.2.3 General Definitions 8
5.2.4 Data Definitions 8
5.2.5 Instance Models 8
5.2.6 Data Constraints 14
5.2.7 Properties of a Correct Solution 15

6 Requirements **15**
6.1 Functional Requirements 15
6.2 Non-Functional Requirements 16

7 Likely Changes **17**

8 Traceability Matrices and Graphs **18**

3 Introduction

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

3.1 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 concrete mathematical models driving the design. It also describes the assumptions and theoretical models used to influence the concrete models. 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 refine and expand the tool. Even though the Companion Cube Calculator was created specifically to aid in the development of the GLaDOS architecture, the user specified $f(X)$ is unitless. This means that the tool can be used for any $f(X)$ as long as any applicable units referred to in X are compatible.

Since the initial development of the C^3 tool will be limited to arithmetic operators, this document includes information that is useful to a developer looking to expand the abilities of the C^3 with additional mathematical models to suit their project, such as those proposed in LC3 and LC4.

3.2 Scope of Requirements

The C^3 calculates $R(f(X))$ of a user-defined $f(X)$ and $D(x), x \in X$. Each $D(x)$ must be specified by the user in the initial version of this tool, otherwise the unsupported interval $(-\infty, \infty)$ will be assumed (A1). [This is confusing. I would guess if the assumption were violated that the program would not calculate a result, but you seem to be suggesting otherwise. —SS]

For the initial version of this product, the mathematical operations allowed in an function will be limited to brackets (()) and the arithmetic operators ($+$, $-$, $\times$, $\div$, exponentiation). Future iterations can expand this list to include trigonometric functions (sin, cos, tan, arcsin, arccos, arctan), partial functions, and other useful mathematical constructs.

The purpose of this product is to aid in the design and tuning of the GLaDOS architecture (Section 5.1), but it can be used in similar projects because it does not contain any GLaDOS-specific concepts or models.

3.3 Characteristics of Intended Reader

The intended reader of this document must understand elementary algebra, especially interval arithmetic, in the domain of real numbers ($\mathbb{R}$). An understanding of set notation is also required to understand the models presented in this document. 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.

3.4 Organization of Document

This document begins by describing the general description of the C^3 tool (Section 4), 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 address it (Section 5). This description is followed by the functional and non-functional requirements of the C^3 tool (Section 6) and suggested refinements and expansions for future iterations (Section 7). The last section in this document, traceability matrices and graphs (Section 8), visually describes the dependencies between different document components. This information can be referenced when making changes to the tool's core specifications.

4 General System Description

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

4.1 System Context

The C^3 tool is a stand-alone application for calculating $R(f(X))$ for a user-defined $f(X)$ given the known $D(x), x \in X$. 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 unsupported mathematical operations or special functions. The C^3 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 $f(X)$ that only contains numerical values and supported mathematical operations and symbols
  - Provide $D(x), \forall x \in X$
  - Ensure that $f(X)$ does not contain semantic errors
  - Determine if the C^3 outputs are appropriate for the intended application and make adjustments to the program inputs as required

- C^3 Responsibilities:
  - Allow the user to enter $f(X)$ and $D(x), \forall x \in X$ as system inputs
  - Detect data type mismatch, such as a string of characters instead of a floating point number, and communicate this to the user
  - Detect bracket mismatches in $f(X)$ and communicate this to the user
  - Calculate and output $R(f(X))$ using the user-provided $f(X)$ and $D(x), \forall x \in X$; if a result cannot be calculated, communicate this to the user

4.2 User Characteristics

The end user of C^3 is also the intended reader (Section 3.3) of this document.

4.3 System Constraints

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

5 Specific System Description

This section presents the problem description, which gives a high-level view of the motivation behind the C^3 tool. This is followed by the solution characteristics specification, presenting the assumptions, theories, definitions, and instance models. The solution characteristics are based in interval arithmetic and presented using set notation.

5.1 Problem Description

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 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 several $f(X)$, each with a different X. Each $f(X)$ must be normalized to a range of $[-1, 1]$, which can only be done if its $R(f(X))$ is known. The currently implemented $f(X)$ are not well-informed by observation or scientific research and must be subjected to an iterative process to address this problem. An automated method of calculating $R(f(X))$ for a proposed $f(X)$ will make this process faster and less error-prone.

5.1.1 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: A set of values for a variable x that are valid in $f(X)$
- Extended real interval: The set of all $\mathbb{R}$ that also contains $\pm\infty$
- Monotonic function: A function whose first derivative does not change its mathematical sign
- 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 [BEDMAS])
- Range: 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

5.1.2 Physical System Description

The C^3 tool does not have a physical system component because it exists independently of the context of $f(X)$.

5.1.3 Goal Statements

Given $f(X)$ and $D(x), x \in X$, the goal of the C^3 tool is:

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

[Your goal statement is more mathematical than usual. It is nice if the goal statement can be written with a minimal amount of mathematical notation, so that as large an audience as possible can understand it. —SS]

5.2 Solution Characteristics Specification

The instance models that govern C^3 tool are presented in Subsection 5.2.5. The information required to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

5.2.1 Assumptions

This section reduces the scope the original problem to help specify the theoretical models (Section 5.2.2). 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 [DD1, LC1, LC2, LC3, LC4].

A2: $R(f(X))$ is a closed, real interval [DD1, LC1, LC2, LC4].

A3: The mathematical operators used in $f(X)$ can be found in the set $\{+, -, \times, \div, b^x, x^n\}$ [LC4].

A4: Special mathematical functions (e.g. $\sin(x)$, $\cos(x)$, $\log(x)$, ...) are not in $f(X)$ [LC2, LC4].

[The notation selected for representing variables and sets is confusing. At first, I thought that x was an element of the interval X, but it turns out that x is a variable and X is a set of variables. Each of the variables has a function D defined that returns its domain. It would be more direct to say that f is a function of sequence of intervals and introduce a type of intervals. Slides 11 and 13 of `http://homes.esat.kuleuven.be/~optec/events/20090622_kozma.pdf` seems like a good place for inspiration. This is also why I suggested using types in the table of symbols. Type information would also help with your theories, data definitions and instance models. —SS]

5.2.2 Theoretical Models

This section focuses on the general equations and laws from interval arithmetic that the C^3 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	$X + Y = \{x + y \mid x \in X \land y \in Y\}$
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	$X - Y = \{x - y \mid x \in X \land y \in Y\}$
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	$X \times Y = \{x \times y \mid x \in X \land y \in Y\}$
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	$X \div Y = \{z \mid \exists x \in X, y \in Y \land y \neq 0, z = x \div y\}$
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 itself might not be an interval.
Source	Hickey et al. (2001)
Ref. By	IM4, IM5

[T1 to T4 are very similar. You could consider combining them and abstracting out the binary operator. You could add a constraint for the case where the binary operator is division. —SS]

Number	T5
Label	**Real Interval Exponents on a Constant Base Number**
Equation	$b^X = \{b^x \| x \in X \wedge b > 1\}$
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	$X^n = \begin{cases} \{x^n \| x \in X \wedge n \in \mathbb{N}\} & \text{n is odd} \\ \{[x_1^n, ..., x_z^n] \| x \in X \wedge n \in \mathbb{N}\} & \text{n is even} \wedge x_z \geq x_{z+1} \wedge x_1 > 0 \\ \{[x_z^n, ..., x_1^n] \| x \in X \wedge n \in \mathbb{N}\} & \text{n is even} \wedge x_z < x_{z+1} \wedge x_Z < 0 \\ \{[0, ..., max\{x_1^n, x_z^n\}] \| x \in X \wedge n \in \mathbb{N}\} & \text{ELSE} \end{cases}$
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: • When n is an odd number, the product interval is equal to the set of products x^n for each element x • 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 • 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 • 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

[This does not make sense to me. It does not match what I see at the wikipedia page. Is X an interval, or is X a set of intervals, or is X a set of variables? What is the subscript z? —SS]

5.2.3 General Definitions

No additional information is required to build the data definitions.

5.2.4 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 C^3 are unitless. These definitions are used to constrain the theoretical models so that they 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

5.2.5 Instance Models

This section transforms the problem defined in Section 5.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 5.2.4 to replace the abstract symbols in the models identified in 5.2.2.

The goal GS1 is solved by recursively solving instance models IM1, IM2, IM3, IM4, IM5, IM6, and IM7 and combining the results. 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 where the intermediary decomposition stages are commonly omitted by humans.

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 $x_{sum} = x_1 + x_2$, $y_{sum} = y_1 + y_2$, $[x_{sum}, y_{sum}]$ is a closed, real interval (A2)
Description	$[x_1, y_1]$ and $[x_2, y_2]$ are the $D(x)$ for two $x \in X$ in the user's input function $f(X)$. The combined boundary values x_{sum} and y_{sum} are determined using basic arithmetic addition. 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 $x_{sub} = x_1 - x_2$, $y_{sub} = y_1 - y_2$, $[x_{sub}, y_{sub}]$ is a closed, real interval (A2)
Description	$[x_1, y_1]$ and $[x_2, y_2]$ are the the $D(x)$ for two $x \in X$ in the user's input function $f(X)$. The combined boundary values x_{sub} and y_{sub} are determined using basic arithmetic subtraction. 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 $x_{mul} = \min(M)$, $y_{mul} = \max(M)$, where $M = \{x_1 \times x_2, x_1 \times y_2, y_1 \times x_2, y_1 \times y_2\}$ $[x_{mul}, y_{mul}]$ is a closed, real interval (A2)
Description	$[x_1, y_1]$ and $[x_2, y_2]$ are the $D(x)$ for two $x \in X$ in the user's input function $f(X)$. 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 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 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 $$\begin{array}{lll} x_{div} = x_1 \div y_2, & y_{div} = y_1 \div x_2 & 0 < x_1 \leq y_1 \\ x_{div} = 0, & y_{div} = y_1 \div x_2 & x_1 = 0 \wedge x_1 \leq y_1 \\ x_{div} = x_1 \div x_2, & y_{div} = y_1 \div x_2 & x_1 < 0 < y_1 \\ x_{div} = x_1 \div x_2, & y_{div} = 0 & y_1 = 0, x_1 \leq y_1 \\ x_{div} = x_1 \div x_2, & y_{div} = y_1 \div y_2 & x_1 \leq y_1 < 0 \end{array}$$ $[x_{div}, y_{div}]$ is a closed, real interval (A2)
Description	$[x_1, y_1]$ and $[x_2, y_2]$ are the $D(x)$ for two $x \in X$ in the user's input function $f(X)$. 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. 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 $x_{div} = y_1 \div y_2, \quad y_{div} = x_1 \div x_2 \quad 0 < x_1 \leq y_1$ $x_{div} = y_1 \div y_2, \quad y_{div} = 0 \quad x_1 = 0 \wedge x_1 \leq y_1$ $x_{div} = y_1 \div y_2, \quad y_{div} = x_1 \div y_2 \quad x_1 < 0 < y_1$ $x_{div} = 0, \quad y_{div} = x_1 \div y_2 \quad y_1 = 0, x_1 \leq y_1$ $x_{div} = y_1 \div x_2, \quad y_{div} = x_1 \div y_2 \quad x_1 \leq y_1 < 0$ $[x_{div}, y_{div}]$ is a closed, real interval (A2)
Description	$[x_1, y_1]$ and $[x_2, y_2]$ are the $D(x)$ for two $x \in X$ in the user's input function $f(X)$. 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. 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 $x_{exp} = b^x$, $y_{exp} = b^y$ $[x_{exp}, y_{exp}]$ is a closed, real interval (A2)
Description	$[x, y]$ is the $D(x)$ for a $x \in X$ in the user's input function $f(X)$ which is used as an exponent on a constant base number b. The combined boundary values x_{exp} and y_{exp} are determined using basic arithmetic exponents. 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 $$\begin{aligned} x_{pow} &= x^n, & y_{pow} &= y^n & & n \text{ is odd} \\ x_{pow} &= x^n, & y_{pow} &= y^n & & n \text{ is even} \wedge (0 \leq x < y) \\ x_{pow} &= y^n, & y_{pow} &= x^n & & n \text{ is even} \wedge (0 \geq x > y) \\ x_{pow} &= 0, & y_{pow} &= \max(x^n, y^n) & & \text{ELSE} \end{aligned}$$ $[x_{pow}, y_{pow}]$ is a closed, real interval (A2)
Description	$[x, y]$ is the $D(x)$ for a $x \in X$ in the user's input function $f(X)$ raised to a constant power n. The combined boundary values x_{pow} and y_{pow} are determined by the divisibility of n by two and the monotonicity of $[x, y]$: • 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 • 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 • For all other cases, the interval is bounded by 0 and the maximum of the input bounds raised to the power of n 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	-

[You don't actually have an instance model to calculate the range of f. Something like slide 13 of the previously mentioned slides might be appropriate? —SS]

[You haven't mentioned directed rounding. Is that relevant for your problem. Of course it isn't a concern for real numbers and exact arithmetic, but mentioning the assumed differences from floating point arithmetic makes sense to me. —SS]

5.2.6 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)

5.2.7 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 $\{\forall x \in \mathbb{R} | a \leq x \leq b\}$.

6 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.

6.1 Functional Requirements

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

R2: The C^3 tool must convert each $D(x)$ 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 (BEDMAS).

R4: The C^3 tool must verify that the user inputs satisfy the input data constraints from 5.2.6.

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 form (DD1).

R7: The C^3 tool must verify that the program output satisfies the output data constraints from 5.2.6.

R8: The C^3 tool must show the verified $R(f(X))$ to the user. If this is not possible, communicate the reason to the user.

[It isn't clear how $D(x)$ will be entered. Why not have your input just be the input variables of type interval. The way you have written it, it appears that there are variables x and a function that returns the domain of a given x. If all you are looking for is upper and lower bounds for each variable, that would be easy to say. You talk about converting to interval form in R2, so maybe I'm missing something. Either way, it is confusing. R8 is ambiguous. Why wouldn't it be possible to verify R. What are the potential reasons that would be communicated to the user? —SS]

6.2 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 must be proven using mathematical induction.

Reliability

- The C^3 tool must calculate the interval $R(f(X))$ 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 range of a manually calculated result $\pm$ the floating point error.

Robustness

- The C^3 tool must be able to recognize violated data constraints (5.2.6) and report them to the user.

- The C^3 tool must inform the user when it encounters any unspecified state.

Performance

Time and space performance is not a priority in the C^3 tool specification. It is predicted that the intended users will be using the 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 implemented such that they can be verified using mathematical proofs.

Usability

- The user must be able to enter $f(X)$ using standard mathematical notation.
- The user must be able to enter values for $D(x)$ using standard interval notation.
- If the user provides a file as input, they must be able to include both $f(X)$ and each $D(x)$, $x \in X$ in the same file.
- The program must output the value $R(f(X))$ using standard interval notation.
- If the design of the tool includes a GUI (4.3), 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 (LC1).
- The evolvability of the C^3 tool must allow the addition of mathematical operators and special functions (LC3, LC4).

Reusability

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

Portability

The portability of the C^3 tool is not a priority because it is expected that the intended users will use Windows platforms for their analysis.

7 Likely Changes

LC1: Support for open, real intervals (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)

8 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 2 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 3 shows the dependencies of instance models, requirements, and data constraints on each other. Table 1 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 1: 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 2: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM1	IM2	IM3	IM4	IM5	IM6	IM7	DD1	5.2.6	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 3: Traceability Matrix Showing the Connections Between Requirements and Instance Models

The purpose of the traceability graphs is to provide an alternate visualization of the information presented in the traceability tables. The arrows in the graphs represent dependencies where 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 1 and 2, 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 3, showing the dependencies of instance models, requirements, and data constraints on each other.

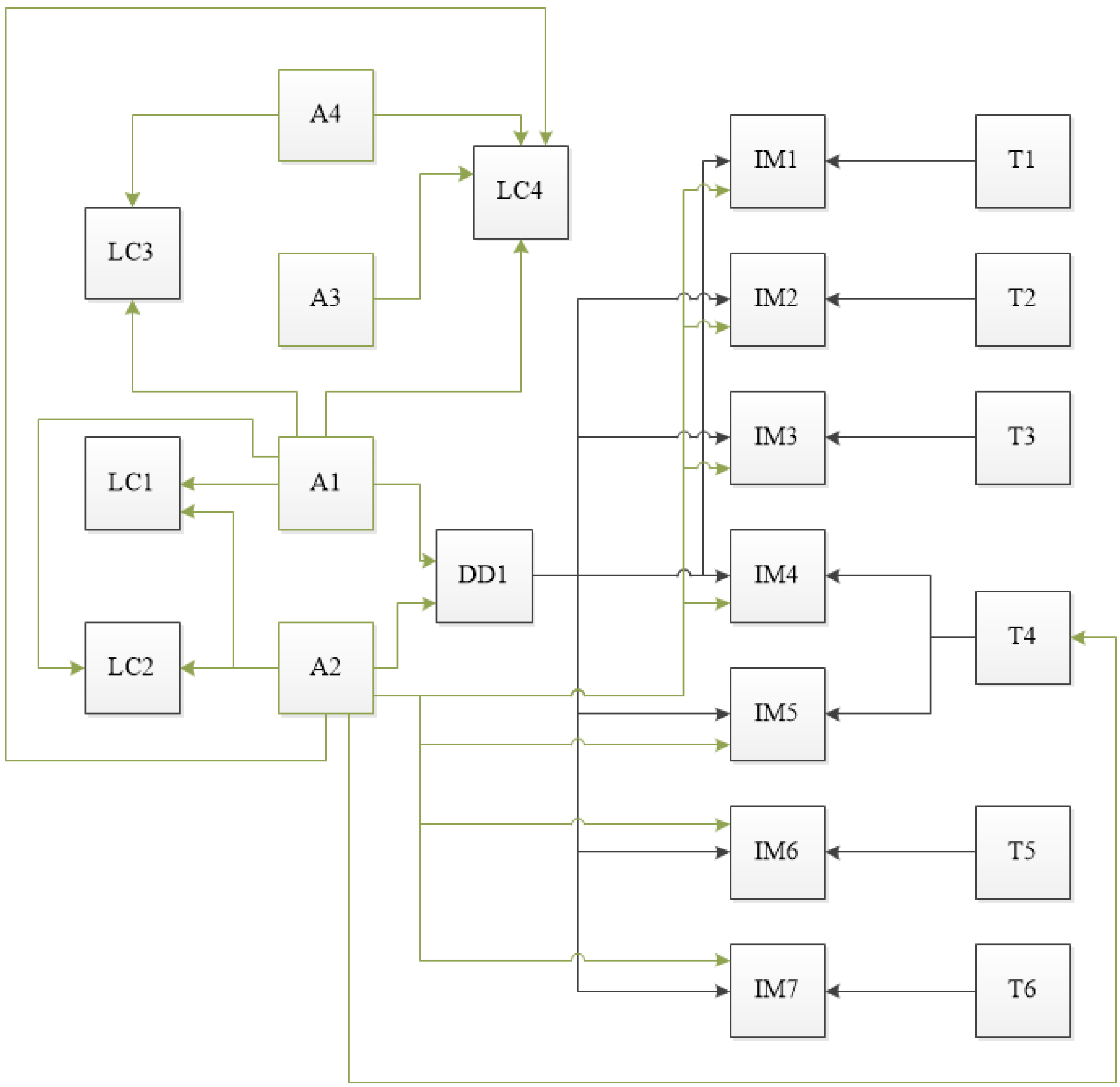

Figure 1: Traceability Matrix Showing the Connections Between Items of Different Sections

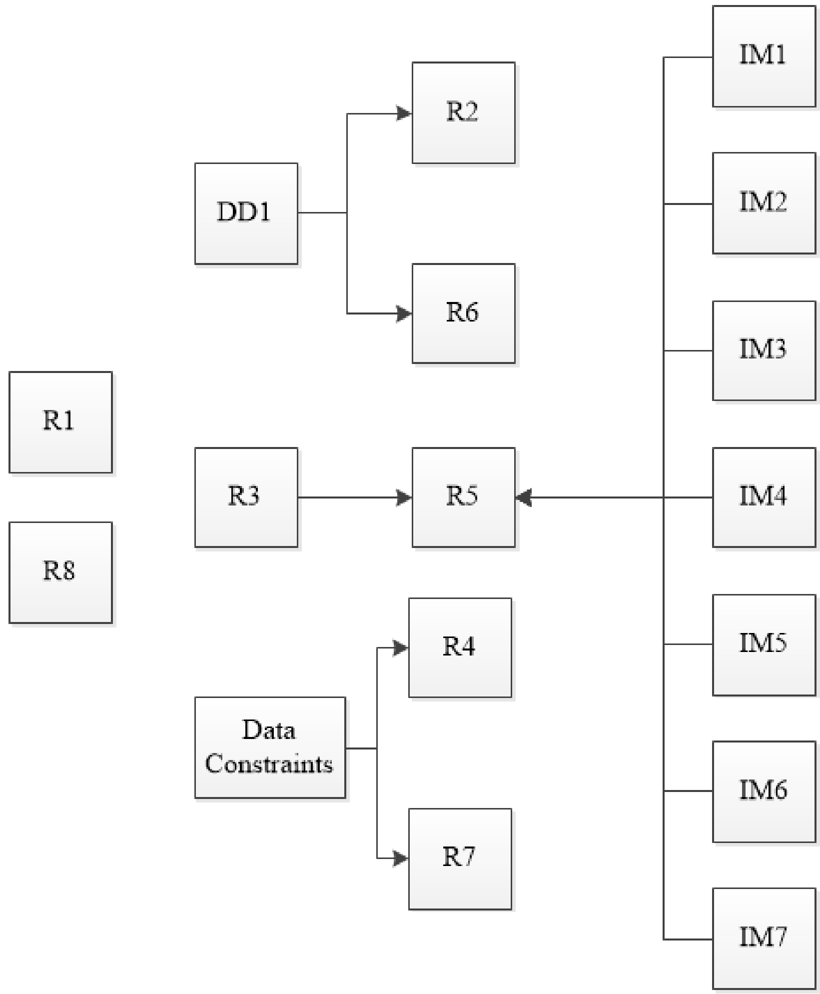

Figure 2: Traceability Matrix Showing the Connections Between Requirements, Instance Models, and Data Constraints

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.