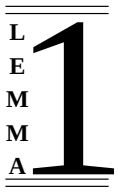


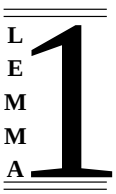
Lemma 1 Ltd.
c/o Interglossa
2nd Floor
31A Chain St.
Reading
Berks
RG1 2HX

ClawZ

Z Toolkit Specification

Version: 10.4
Date: 26 January 2004
Reference: LEMMA1/DAZ/DTD528
Pages: 57

Prepared by: R.B.Jones
Tel: +44 1344 648578
E-Mail: RBJones@RBJones.com



0 DOCUMENT CONTROL

0.1 Contents List

0.2 Document Cross References

[1] DS/FMU/IED/USR005. *ProofPower Description Manual.* Lemma 1 Ltd., `http://www.lemma-one.com`.

[2] LEMMA1/DAZ/IMP528. *ClawZ — Implementation of the ClawZ library.* R.B. Jones, Lemma 1 Ltd., `rbjones@rbjones.com`.

[3] LEMMA1/DAZ/PLN029. *Toolset Automation Enhancements — Proposal.* R.D. Arthan, Lemma 1 Ltd., `rda@lemma-one.com`.

[4] LEMMA1/DAZ/ZED505. *ClawZ - Z Library Specification.* R.B. Jones, Lemma 1 Ltd., `rbjones@rbjones.com`.

[5] LEMMA1/DAZ/ZED506. *ClawZ - Z Library Implementation.* R.B. Jones, Lemma 1 Ltd., `rbjones@rbjones.com`.

0.3 Changes History

Version 1.5 First issue to DERA.

The following notes relate to changes in this document to specifications previously located in the ClawZ library [5].

- Correction to definition of $:_m$ to cover negative ranges.
- Improved definition of *dot_product*.
- Widened domain of Matrix to permit unconditional rewrite.
- Definitions of *sum* and *product* revised to improve both readability and tractability.

- Definition of *bin*2*dec* amended for a reasonably compromise between readability and tractability.

Version 1.7 FEBRUARY 2002.

All arithmetic operators declared as left associative.

Version 1.9 JULY 2002.

Added section on Rounding Operators, and identified the matlab and fcn floor and ceil functions with floor and ceil defined here.

Version 10.1 AUGUST 2002.

Add subscript R to names of rounding operators.

Version 10.2 APRIL 2003.

Version 10.3 JANUARY 2004. Updates for changes in ProofPower version 2.7.3 (Z universal set is now called $\mathbb{U}$; "|" is now treated as a punctuation symbol).

Issue 10.4 JANUARY 2004. Incorporating new information about the syntax of exponentiation in Fcn expressions.

- Correction to definition of $:_m$.

0.4 Changes Forecast

None.

1 GENERAL

1.1 Scope

This document provides definitions of functions required either by the output from the ClawZ tool or in writing specifications of Simulink library blocks (which may be referred to in the output of ClawZ), together with the detailed design of support for reasoning about these functions, in the form of theorems, conversions, tactics and proof contexts.

1.2 Introduction

1.2.1 Purpose and Background

This document is one of the deliverables for the Toolset Automation project. For the relevant proposal see [3].

This document provides Z toolkit extensions required for ClawZ. The toolkit extensions are implemented as a ProofPower-Z specification.

Its purpose is to provide functions in terms of which the Simulink library blocks can be specified. The library blocks themselves are specified elsewhere, see [5].

It also specifies the basic proof support tools for the toolkit extensions. These tools comprise theorems, conversions and proof contexts and are defined here in the same style as is used for ProofPower proof facilities.

The theory *CLT_common* specified by this document was previously provided as a part of the ClawZ library in [4] and [5]. The revised and enhanced material is now described as the ClawZ Z toolkit to distinguish it from the ClawZ Z library remaining in [4] and [5] which makes use of this theory in providing formal specifications for Simulink library blocks.

In more detail, the structure of this document is as follows:

Section 2 discusses design issues and decisions which have been taken here;

Section 3 gives the Z specification aspects of the ClawZ toolkit;

Section 4 defines the ML structure which gives the interface for the facilities provided, defining the theorems and conversions implemented.

1.2.2 Dependencies

The theory defined in this document is a child of the theory *z_library* defined in ProofPower and of *z_reals* and *cn*, though dependency on "cn" is to be minimised. The module is expected to be

loaded into a ProofPower Compliance Tool database in typical use.

1.2.3 Possible Enhancements

Some aspects of the clawz library, for example the discrete components, are not yet supported by the ClawZ toolkit. There are opportunities for improving the performance of the facilities provided, if any of them should prove performance critical.

2 DESIGN ISSUES

2.1 Theory Hierarchy

The theory created by this document is called "CLT_common". Its parents are the theory "$z\_library$" which gives access to the Z toolkit as provided in ProofPower, "cn" giving access to facilities related to the Compliance Tool, and "z_reals".

2.2 Proof Facilities

The proof facilities follow a pattern which is common in ProofPower.

First of all, in section 4.2, theorems are presented which allow basic semi-automatic reasoning about the objects defined in a theory.

Secondly, in section 4.4, proof procedures which cannot be captured as theorems are given as derived inference rules, typically conversions (i.e. proof rules which support equational reasoning by taking a term tm as argument and proving a theorem of the form $\vdash\ tm\ =\ tm'$).

Finally, in section 4.5 the more generally applicable theorems and proof procedures are packaged together in proof contexts which make it convenient for a user to access a general purpose collection of simplifications customised for the theory. These proof contexts typically come in two flavours: component proof contexts containing only the simplifications for this theory for use in combination with other proof contexts; and complete proof contexts which contain a comprehensive collection of simplifications suitable for use when reasoning primarily about the constants introduced in the theory.

3 CLAWZ TOOLKIT

Specifications are in general preceded by some comments on their role and followed by remarks on methods of reasoning relevant to the particular global variables introduced. It should be understood that unless stated otherwise proof support mentioned will be available only in the *CLT_common* proof context.

3.1 Preamble

The following preamble creates the theory "CLT_common" as a child of "z_library" and makes the theories "z_reals" and "cn" parents.

SML

```
force_delete_theory "CLT_common" handle Fail _ => ();
open_theory "z_library";
push_pc "z_library";
val _ = set_flag ("z_type_check_only", false);
val _ = set_flag ("z_use_axioms", true);
new_theory "CLT_common";
new_parent"z_reals";
new_parent"cn";
```

3.2 Conditionals and Coercions

Note that though the following definition of the conditional is not in standard Z, the conditional it defines is in the Z standard. The conditional should be a part of the Z language, not of the Z toolkit, and would require special action from the parser to ensure correct treatment of logical operators in the condition. Since this feature has not been implemented the condition will sometimes have to be enclosed in a "Π" to prevent logical operators being construed as schema operations.

Z

```
fun if _ then _ else _
```

Z

$$[X]$$

$$\mathbf{if\ \_\ then\ \_\ else\ \_} : (BOOL \times X \times X) \rightarrow X$$

$$\forall\ b{:}BOOL;\ x,y{:}X \bullet$$
$$(b \Rightarrow (\mathit{if}\ b\ \mathit{then}\ x\ \mathit{else}\ y) = x)$$
$$\land\ (\neg\ b \Rightarrow (\mathit{if}\ b\ \mathit{then}\ x\ \mathit{else}\ y) = y)$$

Conditional clauses will be eliminated by rewriting or stripping if the condition evaluates to $true$ oe $false$. One way of forcing this to happen is to do a case split on the value of the conditional.

The following coercion is defined primarily for use in selecting elements from sequences using signal values.

z

$$\boldsymbol{r2z} : \mathbb{R} \pfun \mathbb{Z}$$

$$\forall\ i: \mathbb{Z} \bullet (real\ i \mapsto i) \in r2z$$

$r2z$ will be eliminated by rewriting if it is applied to an expression of the form $(real\ exp)$.

3.3 Slicing Vectors

The following function is used to select a slice from a vector in the synthesis of *Demux* and *BuseSelector* blocks. It is an integer variant of the similarly named matlab operator (defined below).

z

$$fun\ 2\ leftassoc\ \_:_z\_$$

z

$$\_\boldsymbol{:_z}\_ : (\mathbb{Z} \times \mathbb{Z}) \fun seq\ \mathbb{Z}$$

$$\begin{array}{l}\forall x,\ y: \mathbb{Z} \bullet \\ \quad x :_z y = \\ \quad \{i,z : \mathbb{Z} \\ \quad \mid \quad i = z - x + 1 \\ \quad \land \quad x \leq z \leq y\}\end{array}$$

3.4 Bounds

The following definitions of bounds are required for defining the various simulink $MinMax$ blocks.

z

$$rel\ \_\ lb_R\ \_$$

z

$$\_\ \boldsymbol{lb_R}\ \_ : \mathbb{R} \rightarrow \mathbb{P}\ \mathbb{R}$$

$$\forall\ r: \mathbb{R};\ sr: \mathbb{P}\ \mathbb{R} \bullet$$
$$r\ lb_R\ sr \Leftrightarrow (\forall\ x{:}sr \bullet\ r \leq_R x)$$

z

$$\boldsymbol{glb_R}: \mathbb{P}\ \mathbb{R} \pfun \mathbb{R}$$

$$\forall\ sr: \mathbb{P}\ \mathbb{R};\ glb: \mathbb{R}\bullet$$
$$(sr \mapsto glb) \in glb_R \Leftrightarrow$$
$$glb\ lb_R\ sr \land (\forall lb: \mathbb{R} \mid lb\ lb_R\ sr\bullet\ lb \leq_R glb)$$

z

$$rel\ \_\ ub_R\ \_$$

z

$$\_\ \boldsymbol{ub_R}\ \_: \mathbb{R} \rightarrow \mathbb{P}\ \mathbb{R}$$

$$\forall\ r: \mathbb{R};\ sr: \mathbb{P}\ \mathbb{R}\bullet$$
$$r\ ub_R\ sr \Leftrightarrow (\forall\ x{:}sr\bullet\ r \geq_R x)$$

z

$$\boldsymbol{lub_R}: \mathbb{P}\ \mathbb{R} \pfun \mathbb{R}$$

$$\forall\ sr: \mathbb{P}\ \mathbb{R};\ lub: \mathbb{R}\bullet$$
$$(sr \mapsto lub) \in lub_R \Leftrightarrow$$
$$lub\ ub_R\ sr \land (\forall ub: \mathbb{R} \mid ub\ ub_R\ sr\bullet\ ub \geq_R lub)$$

A variety of theorems are available for general reasoning about bounds. For the special case of equalities or inequalities involving bounds of set displays (including least upper and greatest lower) a package of theorems is available in the component proof context $'CLT\_Bounds$ which reduces these to inequalities suitable for solution by linear arithmetic. This proof context contains an automatic prover which applies the rewrites and then calls the linear arithmetic prover.

3.5 Relational Operators

These relational operations are used in defining the Simulink relational operator blocks.

z

$$rel\ is\_true_R\ \_$$

z

$$\boldsymbol{true_R}, \boldsymbol{false_R}: \mathbb{R};$$
$$\boldsymbol{is\_true_R}\ \_: \mathbb{P}\ \mathbb{R};$$
$$\boldsymbol{Boolean_R}: BOOL \to \mathbb{R}$$

$$true_R = real\ 1;$$
$$false_R = real\ 0;$$
$$(\forall x{:}\mathbb{U} \bullet is\_true_R\ x \Leftrightarrow \neg\ x = false_R);$$
$$\forall x{:}\mathbb{U} \bullet Boolean_R\ x = if\ x\ then\ true_R\ else\ false_R$$

z

$$[X, Y]$$

$$\boldsymbol{liftrel_R}: \mathbb{P}\ (X \times Y) \to (X \times Y \to \mathbb{R})$$

$$\forall\ r{:}\ \mathbb{P}\ (X \times Y);\ x{:}\ X;\ y{:}\ Y \bullet$$
$$liftrel_R\ r\ (x,y) = Boolean_R\ ((x,y) \in r)$$

z

$$fun\ 200\ leftassoc\ \_\ eq_R\ \_,\ \_\ noteq_R\ \_,\ \_\ less_R\ \_,\ \_\ less\_eq_R\ \_$$

z

$$fun\ 210\ leftassoc\ \_\ greater_R\ \_,\ \_\ greater\_eq_R\ \_$$

z

$$\_\ \boldsymbol{eq_R}\ \_: \mathbb{R} \times \mathbb{R} \to \mathbb{R};$$
$$\_\ \boldsymbol{noteq_R}\ \_: \mathbb{R} \times \mathbb{R} \to \mathbb{R};$$
$$\_\ \boldsymbol{less_R}\ \_: \mathbb{R} \times \mathbb{R} \to \mathbb{R};$$
$$\_\ \boldsymbol{less\_eq_R}\ \_: \mathbb{R} \times \mathbb{R} \to \mathbb{R};$$
$$\_\ \boldsymbol{greater_R}\ \_: \mathbb{R} \times \mathbb{R} \to \mathbb{R};$$
$$\_\ \boldsymbol{greater\_eq_R}\ \_: \mathbb{R} \times \mathbb{R} \to \mathbb{R}$$

$$(\_eq_R\_) = liftrel_R\ \{x,y{:}\mathbb{R} \mid x = y\};$$
$$(\_noteq_R\_) = liftrel_R\ \{x,y{:}\mathbb{R} \mid x \neq y\};$$
$$(\_less_R\_) = liftrel_R\ (\_<_R\_);$$
$$(\_less\_eq_R\_) = liftrel_R\ (\_\leq_R\_);$$
$$(\_greater_R\_) = liftrel_R\ (\_>_R\_);$$
$$(\_greater\_eq_R\_) = liftrel_R\ (\_\geq_R\_)$$

These operators are automatically elimated by rewriting or stripping leaving the corresponding relation over Z reals together with a coercion back to the appropriate real representative.

3.6 Rounding Functions

z

$$floor_R : \mathbb{R} \rightarrow real \ (\!|\ \mathbb{Z}\ |\!)$$

$$\forall\ x : \mathbb{R} \bullet real\ 0 \leq_R x -_R floor_R\ x <_R real\ 1$$

z

$$ceil_R : \mathbb{R} \rightarrow real \ (\!|\ \mathbb{Z}\ |\!)$$

$$\forall\ x : \mathbb{R} \bullet real\ 0 \leq_R ceil_R\ x -_R x <_R real\ 1$$

z

$$round_R : \mathbb{R} \rightarrow real \ (\!|\ \mathbb{Z}\ |\!)$$

$$\forall\ x : \mathbb{R} \bullet round_R\ x = if\ real\ 0 \leq_R x\ then\ floor_R\ (x +_R 5\ e \sim 1)\ else\ ceil_R\ (x -_R 5\ e \sim 1)$$

z

$$fix_R : \mathbb{R} \rightarrow real \ (\!|\ \mathbb{Z}\ |\!)$$

$$\forall\ x : \mathbb{R} \bullet fix_R\ x = if\ real\ 0 \leq_R x\ then\ floor_R\ x\ else\ ceil_R\ x$$

3.7 Logical Operators

The following definitions support the specification of the Simulink Logical Operator blocks.

z

$$fun\ 10\ leftassoc\ \_\ equiv_R\ \_$$

z

$$fun\ 20\ leftassoc\ \_\ xor_R\ \_$$

z

$$fun\ 30\ leftassoc\ \_\ or_R\ \_$$

z

$$fun\ 40\ leftassoc\ \_\ and_R\ \_$$

z

$$fun\ 50\ not_R\ \_$$

Logical operators always return $true_R$ for $true$, but will accept as true any non-zero real. Consequently, in the following specification and in some later theorems, when the value of an operator is

known to be the same truth value as one of its operands the operand is twice negated before being returned, to ensure that non standard truths are not propagated.

z

$$
\begin{array}{l}
\boldsymbol{not_R}\ \_ : \mathbb{R} \to \mathbb{R}; \\
\_\ \boldsymbol{and_R}\ \_ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}; \\
\_\ \boldsymbol{or_R}\ \_ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}; \\
\_\ \boldsymbol{equiv_R}\ \_ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}; \\
\_\ \boldsymbol{xor_R}\ \_ : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \\
\hline
\forall\ l : \mathbb{R} \bullet\ not_R\ l = if\ is\_true_R\ l\ then\ false_R\ else\ true_R; \\
\forall\ l,\ r : \mathbb{R} \bullet\ l\ and_R\ r = if\ is\_true_R\ l\ then\ not_R\ (not_R\ r)\ else\ false_R; \\
\forall\ l,\ r : \mathbb{R} \bullet\ l\ or_R\ r = if\ is\_true_R\ l\ then\ true_R\ else\ not_R\ (not_R\ r); \\
\forall\ l,\ r : \mathbb{R} \bullet\ l\ equiv_R\ r = if\ is\_true_R\ l\ then\ not_R\ (not_R\ r)\ else\ not_R\ r; \\
\forall\ l,\ r : \mathbb{R} \bullet\ l\ xor_R\ r = not_R\ (l\ equiv_R\ r)
\end{array}
$$

The supplied rewrites effect the translation of logical and relational expressions into Z formulae.

3.8 Matlab Operators

These function definitions are those of the Z names into which operators in Matlab expressions are translated.

The standard operations might have been OK, but we redefine them anyway, so that if there are any errors in the precedence they can be fixed in the library without having to change ClawZ.

z

$$
fun\ 3\ leftassoc\ \_.+_m\_,\ \_.-_m\_
$$

z

$$
fun\ 4\ leftassoc\ \_.*_m\_,\ \_./_m\_,\ \_.\backslash_m\_
$$

z

$$
\begin{array}{l}
\boldsymbol{\_\ .+_m\ \_},\ \boldsymbol{\_.-_m\_},\ \boldsymbol{\_.*_m\_},\ \boldsymbol{\_./_m\_},\ \boldsymbol{\_.\backslash_m\_} : (\mathbb{R} \times \mathbb{R}) \to \mathbb{R} \\
\hline
\quad (\_.+_m\_) = (\_+_R\_) \\
\wedge \quad (\_.-_m\_) = (\_-_R\_) \\
\wedge \quad (\_.*_m\_) = (\_*_R\_) \\
\wedge \quad (\_./_m\_) = (\_/_R\_) \\
\wedge \quad (\forall x,y\text{: } \mathbb{R} \bullet\ x\ .\backslash_m\ y = y\ /_R\ x)
\end{array}
$$

The following relations over reals (returning reals) are used as target for the corresponding operators in both matlab and fcn expressions:

z

$$fun\ 2\ leftassoc\ \_<_m\_,\ \_>_m\_,\ \_>=_m\_,\ \_<=_m\_,\ \_==_m\_,\ \_\sim=_m\_$$

z

$$\_\ \boldsymbol{<_m}\ \_,\ \_\boldsymbol{>_m}\_,\ \_\boldsymbol{>=_m}\_,\ \_\boldsymbol{<=_m}\_,\ \_\boldsymbol{==_m}\_,\ \_\boldsymbol{\sim=_m}\_ : (\mathbb{R} \times \mathbb{R}) \to \mathbb{R}$$

$$\begin{array}{ll} & (\_<_m\_) = (\_less_R\_) \\ \wedge & (\_>_m\_) = (\_greater_R\_) \\ \wedge & (\_>=_m\_) = (\_greater\_eq_R\_) \\ \wedge & (\_<=_m\_) = (\_less\_eq_R\_) \\ \wedge & (\_==_m\_) = (\_eq_R\_) \\ \wedge & (\_\sim=_m\_) = (\_noteq_R\_) \end{array}$$

The following logical operators are used for the corresponding operators in matlab expressions:

z

$$fun\ 1\ leftassoc\ \_and_m\_,\ \_or_m\_$$

z

$$\_\ \boldsymbol{and_m}\ \_,\ \_\boldsymbol{or_m}\_ : (\mathbb{R} \times \mathbb{R}) \to \mathbb{R}$$

$$\begin{array}{ll} & (\_and_m\_) = (\_and_R\_) \\ \wedge & (\_or_m\_) = (\_or_R\_) \end{array}$$

z

$$fun\ 5\ leftassoc\ \_.\hat{}_m\_$$

z

$$\_\boldsymbol{.\hat{}_m}\_ : (\mathbb{R} \times \mathbb{R}) \pfun \mathbb{R}$$

$$dom\ (\_.\hat{}_m\_) = \mathbb{R} \times (dom\ r2z);$$
$$\forall x{:}\mathbb{R};\ y{:}\ dom\ r2z \bullet\ x\ .\hat{}_m\ y = x\ \hat{}_Z\ (r2z\ y)$$

The following are unary operators occurring in matlab expressions:

z

$$fun\ 5\ mp_m\_,\ mm_m\_$$

z

$$fun\ 6\ \sim_m\_$$

z

$$\boldsymbol{mp_m}\ \_,\ \boldsymbol{mm_{m\_}},\ \boldsymbol{\sim_{m\_}} : \mathbb{R} \to \mathbb{R}$$

$$\begin{array}{ll} \forall x\colon \mathbb{R} \bullet & mp_m\ x = x \\ \land & mm_m\ x = (real\ 0) -_R x \\ \land & (\sim_m \_) = (not_R\ \_) \end{array}$$

z

$$fun\ 2\ leftassoc\ \_:_{m\_}$$

z

$$\_:_{m\_} : (\mathbb{R} \times \mathbb{R}) \to seq\ \mathbb{R}$$

$$\begin{array}{l} \forall x,\ y\colon \mathbb{R} \bullet \\ \quad x :_m y = \\ \quad \{z{:}\mathbb{R};\ i{:}\mathbb{Z} \\ \quad \mid \quad z = x +_R (real\ i) \\ \quad \land \quad (x \leq_R z \leq_R y) \\ \quad \bullet (i + 1,\ z)\} \end{array}$$

The matlab operators are eliminated in favour of the underlying operations over reals when an expression is rewritten in the proof context CLT_common.

3.9 Fcn Operators

These function definitions are those of the Z names into which operators in Fcn expressions are translated.

The standard operations over reals do not have high enough precedence for use in Fcn expressions.

z

$$fun\ 5\ leftassoc\ \_+_{\boldsymbol{f}\_},\ \_-_{\boldsymbol{f}\_}$$

z

$$fun\ 6\ leftassoc\ \_*_{\boldsymbol{f}\_},\ \_/_{\boldsymbol{f}\_}$$

z

$$\_ +_{\boldsymbol{f}}\ \_,\ \_-_{\boldsymbol{f}\_},\ \_*_{\boldsymbol{f}\_},\ \_/_{\boldsymbol{f}\_} : (\mathbb{R} \times \mathbb{R}) \to \mathbb{R}$$

$$\begin{array}{ll} & (\_+_f\_) = (\_+_R\_) \\ \land & (\_-_f\_) = (\_-_R\_) \\ \land & (\_*_f\_) = (\_*_R\_) \\ \land & (\_/_f\_) = (\_/_R\_) \end{array}$$

The following relations and operations over reals are provided to give a distinct priority to occurrences of these operations in Fcn expressions:

z

$$\textit{fun } 4 \textit{ leftassoc } \_<_f\_,\ \_>_f\_,\ \_>=_f\_,\ \_<=_f\_$$

z

$$\_<_f\_,\ \_>_f\_,\ \_>=_f\_,\ \_<=_f\_ : (\mathbb{R} \times \mathbb{R}) \to \mathbb{R}$$

$$(\_<_f\_) = (\_less_R\_)$$
$$\wedge \quad (\_>_f\_) = (\_greater_R\_)$$
$$\wedge \quad (\_>=_f\_) = (\_greater\_eq_R\_)$$
$$\wedge \quad (\_<=_f\_) = (\_less\_eq_R\_)$$

z

$$\textit{fun } 3 \textit{ leftassoc } \_=_f\_,\ \_\neq_f\_$$

z

$$\_=_f\_,\ \_\neq_f\_ : (\mathbb{R} \times \mathbb{R}) \to \mathbb{R}$$

$$(\_=_f\_) = (\_eq_R\_)$$
$$\wedge \quad (\_\neq_f\_) = (\_noteq_R\_)$$

z

$$\textit{fun } 2 \textit{ leftassoc } \_\mathbf{and}_f\_$$

z

$$\textit{fun } 1 \textit{ leftassoc } \_\mathbf{or}_f\_$$

z

$$\_\ \mathbf{and}_f\ \_,\ \_\mathbf{or}_f\_ : (\mathbb{R} \times \mathbb{R}) \to \mathbb{R}$$

$$(\_and_f\_) = (\_and_R\_)$$
$$\wedge \quad (\_or_f\_) = (\_or_R\_)$$

z

$$\textit{fun } 9 \textit{ leftassoc } \_\hat{}_f\_$$

z

$$\_\hat{}_f\_ : (\mathbb{R} \times \mathbb{R}) \pfun \mathbb{R}$$

$$dom\ (\_\hat{}_f\_) = \mathbb{R} \times (dom\ r2z);$$
$$\forall x{:}\mathbb{R};\ y{:}\ dom\ r2z \bullet\ x\ \hat{}_f\ y = x\ \hat{}_Z\ (r2z\ y)$$

The following are unary operators occurring in Fcn expressions:

z

$$fun\ 8\ \boldsymbol{mp_{f\_}}, \boldsymbol{mm_{f\_}}$$

z

$$fun\ 7\ \boldsymbol{not_{f\_}}$$

z

$$\boldsymbol{mp_{f}}\ \_, \boldsymbol{mm_{f\_}}, \boldsymbol{not_{f\_}} : \mathbb{R} \to \mathbb{R}$$

$$\begin{array}{ll} \forall x: \mathbb{R} \bullet & mp_f\ x = x \\ \land & mm_f\ x = (real\ 0) -_R x \\ \land & (not_{f\_}) = (not_R\ \_) \end{array}$$

The matlab operators are eliminated in favour of the underlying operations over reals when an expression is rewritten in the proof context CLT_common.

3.10 Matlab Functions

The following specifications are place holders for the various functions used in translation of Matlab expressions. They permit type checking but not reasoning which depends upon these functions.

z

$$fun\ 10\ abs_{m\_}, acos_{m\_}, asin_{m\_}, atan_{m\_}, ceil_{m\_}$$

z

$$fun\ 10\ cos_{m\_}, cosh_{m\_}, exp_{m\_}, fabs_{m\_}, floor_{m\_}$$

z

$$fun\ 10\ hypot_{m\_}, log_{m\_}, log10_{m\_}, sin_{m\_}, sinh_{m\_}$$

z

$$fun\ 10\ sqrt_{m\_}, tan_{m\_}, tanh_{m\_}$$

z

$$\boldsymbol{abs_{m\_}}, \boldsymbol{acos_{m\_}}, \boldsymbol{asin_{m\_}}, \boldsymbol{atan_{m\_}}, \boldsymbol{ceil_{m\_}} : \mathbb{R} \pfun \mathbb{R};$$
$$\boldsymbol{cos_{m\_}}, \boldsymbol{cosh_{m\_}}, \boldsymbol{exp_{m\_}}, \boldsymbol{fabs_{m\_}}, \boldsymbol{floor_{m\_}} : \mathbb{R} \pfun \mathbb{R};$$
$$\boldsymbol{hypot_{m\_}}, \boldsymbol{log_{m\_}}, \boldsymbol{log10_{m\_}}, \boldsymbol{sin_{m\_}}, \boldsymbol{sinh_{m\_}} : \mathbb{R} \pfun \mathbb{R};$$
$$\boldsymbol{sqrt_{m\_}}, \boldsymbol{tan_{m\_}}, \boldsymbol{tanh_{m\_}} : \mathbb{R} \pfun \mathbb{R}$$

$$(ceil_{m\_}) = ceil_R;$$
$$(floor_{m\_}) = floor_R$$

z

$$fun\ 10\ atan2_{m\_},\ power_{m\_},\ rem_{m\_}$$

z

$$\boldsymbol{atan2_{m\_}},\ \boldsymbol{power_{m\_}},\ \boldsymbol{rem_{m\_}} : \mathbb{R} \times \mathbb{R} \pfun \mathbb{R}$$

$$true$$

No proof support is available for the matlab functions.

3.11 Fcn Functions

The following specifications are place holders for the various functions used in translation of Fcn expressions. They permit type checking but not reasoning which depends upon these functions.

z

$$fun\ 10\ abs_{f\_},\ acos_{f\_},\ asin_{f\_},\ atan_{f\_},\ ceil_{f\_}$$

z

$$fun\ 10\ cos_{f\_},\ cosh_{f\_},\ exp_{f\_},\ fabs_{f\_},\ floor_{f\_}$$

z

$$fun\ 10\ hypot_{f\_},\ log_{f\_},\ log10_{f\_},\ sin_{f\_},\ sinh_{f\_}$$

z

$$fun\ 10\ sqrt_{f\_},\ tan_{f\_},\ tanh_{f\_},\ ln_{f\_},\ sgn_{f\_}$$

z

$$\begin{array}{l}\boldsymbol{abs_{f\_}},\ \boldsymbol{acos_{f\_}},\ \boldsymbol{asin_{f\_}},\ \boldsymbol{atan_{f\_}},\ \boldsymbol{ceil_{f\_}} : \mathbb{R} \pfun \mathbb{R};\\ \boldsymbol{cos_{f\_}},\ \boldsymbol{cosh_{f\_}},\ \boldsymbol{exp_{f\_}},\ \boldsymbol{fabs_{f\_}},\ \boldsymbol{floor_{f\_}} : \mathbb{R} \pfun \mathbb{R};\\ \boldsymbol{hypot_{f\_}},\ \boldsymbol{log_{f\_}},\ \boldsymbol{log10_{f\_}},\ \boldsymbol{sin_{f\_}},\ \boldsymbol{sinh_{f\_}} : \mathbb{R} \pfun \mathbb{R};\\ \boldsymbol{sqrt_{f\_}},\ \boldsymbol{tan_{f\_}},\ \boldsymbol{tanh_{f\_}},\ \boldsymbol{ln_{f\_}},\ \boldsymbol{sgn_{f\_}} : \mathbb{R} \pfun \mathbb{R}\end{array}$$

$$\begin{array}{l}(ceil_{f\_}) = ceil_R;\\ (floor_{f\_}) = floor_R\end{array}$$

z

$$fun\ 10\ atan2_{f\_},\ power_{f\_},\ rem_{f\_}$$

z

$$\boldsymbol{atan2_{f\_}},\ \boldsymbol{power_{f\_}},\ \boldsymbol{rem_{f\_}} : \mathbb{R} \times \mathbb{R} \pfun \mathbb{R}$$

$$true$$

No proof support is available for the fcn functions.

3.12 Math

3.12.1 Combinatorial Logic

bin2dec is required for the Combinatorial Logic block. The following definition is a reasonable compromise between simplicity and convenience for reasoning, and makes if feasible for *bin2dec* to be evaluated by rewriting.

z

$$\boldsymbol{bin2dec} : (seq\ \mathbb{R}) \rightarrow \mathbb{Z}$$

$$bin2dec\ \langle\rangle = 0;$$
$$\forall\ f\colon \mathbb{R};\ l\colon seq\ \mathbb{R} \bullet$$
$$bin2dec(\langle f \rangle \frown l) = (bin2dec\ l) + (if\ f = real\ 0\ then\ 0\ else\ 1) * (2\ **\ \#l)$$

When applied to a sequence display *bin2dec* is eliminated by rewriting in favour of an expression involving $**$.

The following definition is used in expressing constraints on the input to *DiscreteStateSpace* and *CombinatorialLogic* blocks.

z

$$rel\ \_\ Matrix\ \_$$

z

$$\_\ \boldsymbol{Matrix}\ \_ : (\mathbb{Z} \leftrightarrow (\mathbb{Z} \leftrightarrow \mathbb{R})) \leftrightarrow \mathbb{Z} \times \mathbb{Z}$$

$$\forall\ s : (\mathbb{Z} \leftrightarrow (\mathbb{Z} \leftrightarrow \mathbb{R}));\ m,\ n : \mathbb{Z} \bullet$$
$$s\ Matrix\ (m,\ n) \Leftrightarrow$$
$$s \in (seq\_) \land \#s = m$$
$$\land\ (\forall\ ss : ran\ s \bullet ss \in (seq\_) \land \#ss = n)$$

When applied to a pair of numeric literals and a sequence display of sequence displays and which is of the specified dimensions *Matrix* will be eliminated by rewriting.

3.12.2 Dot Product

$dot_product$ is required for several of the Simulink *Disrete* blocks.

z

$$\boldsymbol{dot\_product} : (seq\ \mathbb{R}) \times (seq\ \mathbb{R}) \pfun \mathbb{R}$$

$$dom\ dot\_product = \{In1?, In2? : seq\ \mathbb{R} \mid \#In1? = \#In2?\};$$

$$dot\_product\ (\langle\rangle, \langle\rangle) = real\ 0;$$

$$\forall\ h1,h2 : \mathbb{R};\ t1,t2 : seq\ \mathbb{R} \mid \#t1 = \#t2 \bullet$$
$$dot\_product(\langle h1\rangle \frown t1, \langle h2\rangle \frown t2) = h1 *_R h2 +_R dot\_product(t1, t2)$$

The *CLT_common* proof context will evaluate dot products of pairs of sequence displays. If all the elements are literals the result will be a literal, otherwise an expression involving real addition and multiplication.

3.12.3 Product

Vectorised product blocks are defined using the following function:

z

$$\boldsymbol{product} : (seq\ \mathbb{R}) \to \mathbb{R}$$

$$product\ \langle\rangle = real\ 1;$$
$$\forall\ h : \mathbb{R};\ t : seq\ \mathbb{R} \bullet$$
$$product(\langle h\rangle \frown t) = h *_R product\ t$$

Products of sequence displays will be expanded out by the proof context and will be fully evaluated if the elements are literals.

3.12.4 Sum

Vectorised sum blocks are defined using the following function:

z

$$\boldsymbol{sum} : (seq\ \mathbb{R}) \to \mathbb{R}$$

$$sum\ \langle\rangle = real\ 0;$$
$$\forall\ h : \mathbb{R};\ t : seq\ \mathbb{R} \bullet$$
$$sum\ (\langle h\rangle \frown t) = h +_R (sum\ t)$$

Sums of sequence displays will be expanded out by the proof context and will be fully evaluated if the elements are literals.

4 THE STRUCTURE

4.1 Preamble

SML

signature ***CLT_common*** = *sig*

Description This is the signature for the toolkit extensions required to support the theory *CLT_common*.

Theory Design

```
req_name "CLT_common" (Value "z_library");
req_language "Z";
req_parent "cn";
req_parent "z_reals";
req_parent "cache'clawzlib";
set_flag("tc_thms_only", true);
```

Description The theory *CLT_common* defines functions required by the specifications of Simulink library blocks, or by the Z specifications output by ClawZ. It is created in structure *CLT_common*. The specification of the theory name, and the language of the theory is defined using *req_name* and *req_language*. This conforms to the technique used in the rest of the ProofPower design documentation for specifying the requirement for theories.

4.2 Theorems

Theorems are provided which allow systematic expansion of the objects in the theory in terms of the Z toolkit operations. These theorems serve to support the proof procedures defined in section 4.4 below and may also be directly applied by the user.

In following, some theorems, which are intended primarily for insertion in proof contexts rather than for direct use, are expressed in mixed language. In many cases unconditional rewriting over sequence or set displays is possible by this means but cannot be expressed in pure Z (and is not valid over sequences or sets in general). In this we exploit the fact that Z sequence and set displays are both represented in the underlying ProofPower HOL by the application of a semantic constant ($Z'\langle\rangle$ and $Z'Setd$ respectively) to an expression denoting a ProofPower HOL list (often formed using *Cons*, which adds an element to the head of a list).

4.2.1 Conditionals and Coercions

Theory Design

$$req\_thm(\text{"}\boldsymbol{cz\_if\_thm}\text{"}, ([], \llbracket \forall x, y{:}\mathbb{U}\bullet \mathit{if\ true\ then\ x\ else\ y} = x \wedge \mathit{if\ false\ then\ x\ else\ y} = y \rrbracket));$$

$$req\_thm(\text{"}\boldsymbol{cz\_r2z\_thm}\text{"}, ([], \llbracket \forall i{:}\mathbb{U}\bullet \mathit{r2z}\ (\mathit{real}\ i) = i \wedge \mathit{r2z}\ (\sim_R\ (\mathit{real}\ i)) = \sim i \rrbracket));$$

Description These theorems, permit conditional expressions and the partial inverse of *real*,to be eliminated.

4.2.2 Bounds

Theory Design

$$req\_thm(\text{"}\boldsymbol{cz\_ub\_thm}\text{"}, ([], \llbracket \forall r{:}\mathbb{U};\ sr{:}\mathbb{U}\bullet\ r\ ub_R\ sr \Leftrightarrow (\forall\ x{:}sr\bullet\ r \geq_R x) \rrbracket));$$

$$req\_thm(\text{"}\boldsymbol{cz\_lb\_thm}\text{"}, ([], \llbracket \forall r{:}\mathbb{U};\ sr{:}\mathbb{U}\bullet\ r\ lb_R\ sr \Leftrightarrow (\forall\ x{:}sr\bullet\ r \leq_R x) \rrbracket));$$

$$req\_thm(\text{"}\boldsymbol{cz\_ge\_ub\_trans\_thm}\text{"}, ([], \llbracket \forall x, y{:}\mathbb{U};\ z{:}\mathbb{U}\bullet\ y \leq_R x \wedge y\ ub_R\ z \Rightarrow x\ ub_R\ z \rrbracket));$$

$$req\_thm(\text{"}\boldsymbol{cz\_le\_lb\_trans\_thm}\text{"}, ([], \llbracket \forall x, y{:}\mathbb{U};\ z{:}\mathbb{U}\bullet\ x \leq_R y \wedge y\ lb_R\ z \Rightarrow x\ lb_R\ z \rrbracket));$$

$$req\_thm(\text{"}\boldsymbol{cz\_lb\_ub\_thm}\text{"}, ([], \llbracket \forall\ s : \mathbb{U};\ lb : \mathbb{U} \bullet lb\ lb_R\ s \Leftrightarrow \sim_R\ lb\ ub_R\ \{x : s \mid true \bullet \sim_R\ x\} \rrbracket));$$

$$req\_thm(\text{"}\boldsymbol{cz\_ub\_lb\_thm}\text{"}, ([], \llbracket \forall\ s : \mathbb{U};\ ub : \mathbb{U} \bullet ub\ ub_R\ s \Leftrightarrow \sim_R\ ub\ lb_R\ \{x : s \mid true \bullet \sim_R\ x\} \rrbracket));$$

$$req\_thm(\text{"}\boldsymbol{cz\_lub\_thm}\text{"}, ([], \llbracket \forall lub{:}\mathbb{U};\ sr{:}\mathbb{U}\bullet\ (sr \mapsto lub) \in lub_R \Leftrightarrow lub\ ub_R\ sr \wedge (\forall ub{:}\ \mathbb{U} \mid ub\ ub_R\ sr \bullet ub \geq_R lub) \rrbracket));$$

$$req\_thm(\text{"}\boldsymbol{cz\_glb\_thm}\text{"}, ([], \llbracket \forall glb{:}\mathbb{U};\ sr{:}\mathbb{U}\bullet\ (sr \mapsto glb) \in glb_R \Leftrightarrow glb\ lb_R\ sr \wedge (\forall lb{:}\ \mathbb{U} \mid lb\ lb_R\ sr \bullet lb \leq_R glb) \rrbracket));$$

$$req\_thm(\text{"}\boldsymbol{cz\_lub\_sup\_thm}\text{"}, ([], \llbracket \forall s\ l\bullet \llbracket s \mapsto l \in lub_R \rrbracket \Rightarrow l = Sup\ s \rrbracket));$$

$$req\_thm(\text{"}\boldsymbol{cz\_dom\_lub\_thm}\text{"}, ([], \llbracket \forall s{:}\mathbb{U}\bullet\ (\exists lub{:}\mathbb{U}\bullet\ (s \mapsto lub) \in lub_R) \Leftrightarrow \neg\ s = \{\} \wedge (\exists ub{:}\mathbb{U}\bullet\ ub\ ub_R\ s) \rrbracket));$$

$$req\_thm(\text{"}\boldsymbol{cz\_dom\_glb\_thm}\text{"}, ([], \llbracket \forall s{:}\mathbb{U}\bullet\ (\exists glb{:}\mathbb{U}\bullet\ (s \mapsto glb) \in glb_R) \Leftrightarrow \neg\ s = \{\} \wedge (\exists lb{:}\mathbb{U}\bullet\ lb\ lb_R\ s) \rrbracket));$$

$$req\_thm(\text{"}\boldsymbol{cz\_lub\_setd\_thm}\text{"}, ([], \llbracket \forall\ lub, x : \mathbb{U};\ s : \mathbb{U} \bullet lub = lub_R\ \ulcorner Z'Setd\ (Cons\ x\ s) \urcorner \Leftrightarrow lub\ ub_R\ \ulcorner Z'Setd\ (Cons\ x\ s) \urcorner \wedge (\forall\ ub : \mathbb{U} \mid ub\ ub_R\ \ulcorner Z'Setd\ (Cons\ x\ s) \urcorner \bullet ub \geq_R lub) \rrbracket));$$

$$req\_thm(\text{"}\boldsymbol{cz\_glb\_setd\_thm}\text{"}, ([], \llbracket \forall\ glb, x : \mathbb{U};\ s : \mathbb{U} \bullet glb = glb_R\ \ulcorner Z'Setd\ (Cons\ x\ s) \urcorner \Leftrightarrow glb\ lb_R\ \ulcorner Z'Setd\ (Cons\ x\ s) \urcorner \wedge (\forall\ lb : \mathbb{U} \mid lb\ lb_R\ \ulcorner Z'Setd\ (Cons\ x\ s) \urcorner \bullet lb \leq_R glb) \rrbracket));$$

Description A collection of results about upper, lower, greatest lower and lowest upper bounds

Theory Design

req_thm("**cz_ub_empty_thm**", ([], ⁅∀ *ub*:𝕌• *ub* ub_R {}⌝));
req_thm("**cz_lb_empty_thm**", ([], ⁅∀ *lb*:𝕌• *lb* lb_R {}⌝));

req_thm("**cz_lub_unit_thm**", ([], ⁅∀ *x*:𝕌• lub_R {*x*} = *x*⌝));
req_thm("**cz_glb_unit_thm**", ([], ⁅∀ *x*:𝕌• glb_R {*x*} = *x*⌝));

req_thm("**cz_ub_setd_thm**", ([], ⁅∀ *x*,*y*:𝕌; *t*:𝕌• *x* ub_R ⌜*Z'Setd* (*Cons y t*)⌝
⇔ *y* $\leq_R$ *x* ∧ *x* ub_R ⌜*Z'Setd t*⌝⌝));
req_thm("**cz_lb_setd_thm**", ([], ⁅∀ *lb*,*x*:𝕌; *s*:𝕌• *lb* lb_R ⌜*Z'Setd* (*Cons x s*)⌝
⇔ *lb* $\leq_R$ *x* ∧ *lb* lb_R ⌜*Z'Setd s*⌝⌝));

req_thm("**cz_lub_setd_le_thm**", ([], ⁅∀ *x*,*z*:𝕌; *t*:𝕌• lub_R ⌜*Z'Setd* (*Cons z t*)⌝ $\leq_R$ *x*
⇔ *x* ub_R ⌜*Z'Setd* (*Cons z t*)⌝⌝));
req_thm("**cz_le_glb_setd_thm**", ([], ⁅∀ *x*,*z*:𝕌; *t*:𝕌• *x* $\leq_R$ glb_R ⌜*Z'Setd* (*Cons z t*)⌝
⇔ *x* lb_R ⌜*Z'Setd* (*Cons z t*)⌝⌝));

req_thm("**cz_eq_lub_setd_thm**", ([], ⁅∀ *x*,*y*,*z*:𝕌; *t*:𝕌•
x = lub_R ⌜*Z'Setd* (*Cons y* (*Cons z t*))⌝
⇔ *x* = *y* ∧ *x* ub_R ⌜*Z'Setd* (*Cons z t*)⌝
∨ *y* $\leq_R$ *x* ∧ *x* = lub_R ⌜*Z'Setd* (*Cons z t*)⌝⌝));
req_thm("**cz_lub_setd_eq_thm**", ([], ⁅∀ *x*,*y*,*z*:𝕌; *t*:𝕌•
lub_R ⌜*Z'Setd* (*Cons y* (*Cons z t*))⌝ = *x*
⇔ *x* = *y* ∧ *x* ub_R ⌜*Z'Setd* (*Cons z t*)⌝
∨ *y* $\leq_R$ *x* ∧ *x* = lub_R ⌜*Z'Setd* (*Cons z t*)⌝⌝));
req_thm("**cz_le_lub_setd_thm**", ([], ⁅∀ *x*,*y*,*z*:𝕌; *t*:𝕌•
x $\leq_R$ lub_R ⌜*Z'Setd* (*Cons y* (*Cons z t*))⌝
⇔ *x* $\leq_R$ *y* ∨ *x* $\leq_R$ lub_R ⌜*Z'Setd* (*Cons z t*)⌝⌝));

req_thm("**cz_eq_glb_setd_thm**", ([], ⁅∀ *x*,*y*,*z*:𝕌; *t*:𝕌•
x = glb_R ⌜*Z'Setd* (*Cons y* (*Cons z t*))⌝
⇔ *x* = *y* ∧ *x* lb_R ⌜*Z'Setd* (*Cons z t*)⌝
∨ *x* $\leq_R$ *y* ∧ *x* = glb_R ⌜*Z'Setd* (*Cons z t*)⌝⌝));
req_thm("**cz_glb_setd_eq_thm**", ([], ⁅∀ *x*,*y*,*z*:𝕌; *t*:𝕌•
glb_R ⌜*Z'Setd* (*Cons y* (*Cons z t*))⌝ = *x*
⇔ *x* = *y* ∧ *x* lb_R ⌜*Z'Setd* (*Cons z t*)⌝
∨ *x* $\leq_R$ *y* ∧ *x* = glb_R ⌜*Z'Setd* (*Cons z t*)⌝⌝));
req_thm("**cz_glb_setd_le_thm**", ([], ⁅∀ *x*,*y*,*z*:𝕌; *t*:𝕌•
glb_R ⌜*Z'Setd* (*Cons y* (*Cons z t*))⌝ $\leq_R$ *x*
⇔ *y* $\leq_R$ *x* ∨ glb_R ⌜*Z'Setd* (*Cons z t*)⌝ $\leq_R$ *x*⌝));

Description These theorems give rules for eliminating talk about bounds of set displays in favour if inequalities, which may then be solvable using linear arithmetic.

The following illustrates the effects of rewriting with the above theorems.

Rewriting Example

$$
\begin{array}{l}
x = glb_R\ \{a, lub_R\{b,c\}\} \\
1) \quad ===> x = a \wedge x\ lb_R\ \{lub_R\ \{b,\ c\}\} \vee x \leq_R a \wedge x = glb_R\ \{lub_R\ \{b,\ c\}\} \\
2) \quad ===> x = a \wedge x \leq_R lub_R\ \{b,\ c\} \wedge x\ lb_R\ \{\} \vee x \leq_R a \wedge x = lub_R\ \{b,\ c\} \\
3) \quad ===> x = a \wedge (x \leq_R b \vee x \leq_R lub_R\ \{c\}) \wedge true \\
\qquad\qquad \vee x \leq_R a \wedge (x = b \wedge x\ ub_R\ \{c\} \vee b \leq_R x \wedge x = lub_R\ \{c\}) \\
4) \quad ===> x = a \wedge (x \leq_R b \vee x \leq_R c) \wedge true \\
\qquad\qquad \vee x \leq_R a \wedge (x = b \wedge c \leq_R x \wedge x\ ub_R\ \{\} \vee b \leq_R x \wedge x = c) \\
5) \quad ===> x = a \wedge (x \leq_R b \vee x \leq_R c) \wedge true \\
\qquad\qquad \vee x \leq_R a \wedge (x = b \wedge c \leq_R x \wedge true \vee b \leq_R x \wedge x = c)
\end{array}
$$

4.2.3 Relational Operators

Theory Design

$$
\begin{array}{l}
req\_thm(\texttt{"}\boldsymbol{liftrel\_thm}\texttt{"},\ ([],\ ^{\Gamma}_{\mathrm{Z}}\forall\ r : \mathbb{U};\ x : \mathbb{U};\ y : \mathbb{U} \bullet \\
\qquad liftrel_R\ r\ (x,\ y) = if\ (x,\ y) \in r\ then\ real\ 1\ else\ real\ 0^{\neg})); \\
\end{array}
$$

Description This theorem permits $liftrel_R$ to be eliminated.

Theory Design

$$
\begin{array}{l}
req\_thm(\texttt{"}\boldsymbol{cz\_relational\_clauses}\texttt{"},\ ([],\ ^{\Gamma}_{\mathrm{Z}} \\
\qquad \forall x,y{:}\mathbb{U} \bullet\ x\ eq_R\ y = Boolean_R\ (x = y) \\
\qquad \wedge\ x\ noteq_R\ y = Boolean_R\ (\Pi(\neg\ x = y)) \\
\qquad \wedge\ x\ less_R\ y = Boolean_R\ (x <_R y) \\
\qquad \wedge\ x\ less\_eq_R\ y = Boolean_R\ (x \leq_R y) \\
\qquad \wedge\ x\ greater_R\ y = Boolean_R\ (x >_R y) \\
\qquad \wedge\ x\ greater\_eq_R\ y = Boolean_R\ (x \geq_R y) \\
^{\neg}));
\end{array}
$$

Description The strategy for handling the relational operators is to convert them into expressions of the form $Boolean\, p$, where p is an atomic Z predicate or the negation of one.

These theorems support this strategy. They use the universal set $\mathbb{U}$ to make them easier to instantiate. The cast Π is a purely syntactic device used to allow a propositional connective in a function argument (see [1]).

4.2.4 Logical Operators

Theory Design

$$
\begin{array}{l}
\mathit{req\_thm}(\texttt{"}\boldsymbol{cz\_boolean\_clauses}\texttt{"},\ ([],\ \ulcorner\!\!\ulcorner \mathrm{Z} \\
\quad \mathit{true}_R = \mathit{Boolean}_R\ \mathit{true} \\
\quad \wedge\ \mathit{false}_R = \mathit{Boolean}_R\ \mathit{false} \\
\quad \wedge\ (\forall p{:}\mathbb{U} \bullet\ \mathit{not}_R\ (\mathit{Boolean}_R\ p) = \mathit{Boolean}_R\ (\Pi(\neg\ p))) \\
\quad \wedge\ (\forall p,q{:}\mathbb{U} \bullet\ (\mathit{Boolean}_R\ p)\ \mathit{and}_R\ (\mathit{Boolean}_R\ q) = \mathit{Boolean}_R\ (\Pi(p \wedge q))) \\
\quad \wedge\ (\forall p,q{:}\mathbb{U} \bullet\ (\mathit{Boolean}_R\ p)\ \mathit{or}_R\ (\mathit{Boolean}_R\ q) = \mathit{Boolean}_R\ (\Pi(p \vee q))) \\
\quad \wedge\ (\forall p,q{:}\mathbb{U} \bullet\ (\mathit{Boolean}_R\ p)\ \mathit{equiv}_R\ (\mathit{Boolean}_R\ q) = \mathit{Boolean}_R\ (\Pi(p \Leftrightarrow q))) \\
\quad \wedge\ (\forall p,q{:}\mathbb{U} \bullet\ (\mathit{Boolean}_R\ p)\ \mathit{xor}_R\ (\mathit{Boolean}_R\ q) = \mathit{Boolean}_R\ (\Pi(\neg(p \Leftrightarrow q)))) \\
\quad \wedge\ (\forall p,q{:}\mathbb{U} \bullet\ (\mathit{Boolean}_R\ p) = (\mathit{Boolean}_R\ q) \Leftrightarrow (p \Leftrightarrow q)) \\
\ulcorner\!\!\ulcorner));
\end{array}
$$

Description The strategy for handling the boolean operators is to convert an expression in not_R, and_R, or_R, $equiv_R$,and xor_R, into an expression of the form $Boolean_R\ p$, where p is constructed using the Z propositional operators. When such expressions appear as the operands of an equality, the whole predicate can be converted to a Z predicate not involving $Boolean$.

These theorems support this strategy. They use the universal set $\mathbb{U}$ to make them easier to instantiate. The cast Π is a purely syntactic device used to allow a propositional connective in a function argument (see [1]).

See Also *clawz_boolean_clauses*1, *clawz_boolean_clauses*2.

Theory Design

$$
\begin{array}{l}
\mathit{req\_thm}(\texttt{"}\boldsymbol{cz\_boolean\_real\_clauses}\texttt{"},\ ([],\ \llbracket_{\mathrm{Z}}\ \mathit{not}_R\ (\mathit{real}\ 0) = \mathit{Boolean}_R\ \mathit{true} \\
\quad \wedge\ \mathit{not}_R\ (\mathit{real}\ 1) = \mathit{Boolean}_R\ \mathit{false} \\
\quad \wedge\ (\forall p{:}\mathbb{U} \bullet\ (\mathit{real}\ 1)\ \mathit{and}_R\ p = (\mathit{Boolean}_R\ \mathit{true})\ \mathit{and}_R\ p) \\
\quad \wedge\ (\forall p{:}\mathbb{U} \bullet\ p\ \mathit{and}_R\ (\mathit{real}\ 1) = p\ \mathit{and}_R\ (\mathit{Boolean}_R\ \mathit{true})) \\
\quad \wedge\ (\forall p{:}\mathbb{U} \bullet\ (\mathit{real}\ 0)\ \mathit{and}_R\ p = \mathit{Boolean}_R\ \mathit{false}) \\
\quad \wedge\ (\forall p{:}\mathbb{U} \bullet\ p\ \mathit{and}_R\ (\mathit{real}\ 0) = \mathit{Boolean}_R\ \mathit{false}) \\
\quad \wedge\ (\forall p{:}\mathbb{U} \bullet\ (\mathit{real}\ 1)\ \mathit{or}_R\ p = \mathit{Boolean}_R\ \mathit{true}) \\
\quad \wedge\ (\forall p{:}\mathbb{U} \bullet\ p\ \mathit{or}_R\ (\mathit{real}\ 1) = \mathit{Boolean}_R\ \mathit{true}) \\
\quad \wedge\ (\forall p{:}\mathbb{U} \bullet\ (\mathit{real}\ 0)\ \mathit{or}_R\ p = (\mathit{Boolean}_R\ \mathit{false})\ \mathit{or}_R\ p) \\
\quad \wedge\ (\forall p{:}\mathbb{U} \bullet\ p\ \mathit{or}_R\ (\mathit{real}\ 0) = p\ \mathit{or}_R\ (\mathit{Boolean}_R\ \mathit{false})) \\
\quad \wedge\ (\forall p{:}\mathbb{U} \bullet\ (\mathit{real}\ 1)\ \mathit{equiv}_R\ p = (\mathit{Boolean}_R\ \mathit{true})\ \mathit{equiv}_R\ p) \\
\quad \wedge\ (\forall p{:}\mathbb{U} \bullet\ p\ \mathit{equiv}_R\ (\mathit{real}\ 1) = p\ \mathit{equiv}_R\ (\mathit{Boolean}_R\ \mathit{true})) \\
\quad \wedge\ (\forall p{:}\mathbb{U} \bullet\ (\mathit{real}\ 0)\ \mathit{equiv}_R\ p = \mathit{not}_R\ p) \\
\quad \wedge\ (\forall p{:}\mathbb{U} \bullet\ p\ \mathit{equiv}_R\ (\mathit{real}\ 0) = \mathit{not}_R\ p) \\
\quad \wedge\ (\forall p{:}\mathbb{U} \bullet\ (\mathit{real}\ 1)\ \mathit{xor}_R\ p = \mathit{not}_R\ p) \\
\quad \wedge\ (\forall p{:}\mathbb{U} \bullet\ p\ \mathit{xor}_R\ (\mathit{real}\ 1) = \mathit{not}_R\ p) \\
\quad \wedge\ (\forall p{:}\mathbb{U} \bullet\ (\mathit{real}\ 0)\ \mathit{xor}_R\ p = (\mathit{Boolean}_R\ \mathit{false})\ \mathit{xor}_R\ p) \\
\quad \wedge\ (\forall p{:}\mathbb{U} \bullet\ p\ \mathit{xor}_R\ (\mathit{real}\ 0) = p\ \mathit{xor}_R\ (\mathit{Boolean}_R\ \mathit{false})) \\
\rceil_{\mathrm{Z}}));
\end{array}
$$

Description General replacement of real literals 0 and 1 by their boolean equivalents is not desirable, but without this theorem their use in boolean expressions would inhibit translation into Z predicates.

This theorem eliminates occurrences of *real* 0 or *real* 1 when they appear as operands of the clawz boolean operations.

Theory Design

$$
\begin{array}{l}
\mathit{req\_thm}(\texttt{"}\boldsymbol{cz\_boolean\_clauses1}\texttt{"},\ ([],\ \llbracket_{\mathrm{Z}} \\
\quad (\forall p{:}\mathbb{U} \bullet\ \mathit{not}_R\ p = \mathit{Boolean}_R\ (\Pi(p = \mathit{false}_R))) \\
\quad \wedge\ (\forall p,q{:}\mathbb{U} \bullet\ p\ \mathit{and}_R\ q = \mathit{Boolean}_R\ (\Pi(\neg\ p = \mathit{false}_R \wedge \neg\ q = \mathit{false}_R))) \\
\quad \wedge\ (\forall p,q{:}\mathbb{U} \bullet\ p\ \mathit{or}_R\ q = \mathit{Boolean}_R\ (\Pi(\neg\ p = \mathit{false}_R \vee \neg\ q = \mathit{false}_R))) \\
\quad \wedge\ (\forall p,q{:}\mathbb{U} \bullet\ p\ \mathit{equiv}_R\ q = \mathit{Boolean}_R\ (\Pi(p = \mathit{false}_R \Leftrightarrow q = \mathit{false}_R))) \\
\quad \wedge\ (\forall p,q{:}\mathbb{U} \bullet\ p\ \mathit{xor}_R\ q = \mathit{Boolean}_R\ (\Pi(p = \mathit{false}_R \Leftrightarrow \neg\ q = \mathit{false}_R))) \\
\rceil_{\mathrm{Z}}));
\end{array}
$$

Description The strategy for handling the boolean operators is to convert an expression in *not*, *and*, *or*, and *xor*, into an expression of the form $\mathit{Boolean}\, p$, where p is constructed using the Z propositional operators. When such expressions appear as the operands of an equality, the whole predicate can be converted to a Z predicate not involving *Boolean* (except applied to *true*).

This theorem, with *cn_boolean_clauses*2, support this strategy when one or both of the arguments of a boolean operator are not of the form *Boolean* x for some x.

Theory Design

$$
\begin{array}{l}
\mathit{req\_thm}(\texttt{"}\boldsymbol{cz\_boolean\_clauses2}\texttt{"},\ ([],\ ⌜\ \mathrm{Z} \\
\quad (\forall b{:}\mathbb{U};\ r{:}\mathbb{U} \bullet\ r\ \mathit{and}_R\ (\mathit{Boolean}_R\ b) = \mathit{Boolean}_R\ (\Pi(\neg\ r = \mathit{false}_R \wedge b))) \\
\quad \wedge\ (\forall b{:}\mathbb{U};\ r{:}\mathbb{U} \bullet\ (\mathit{Boolean}_R\ b)\ \mathit{and}_R\ r = \mathit{Boolean}_R\ (\Pi(b \wedge \neg\ r = \mathit{false}_R))) \\
\quad \wedge\ (\forall b{:}\mathbb{U};\ r{:}\mathbb{U} \bullet\ r\ \mathit{or}_R\ (\mathit{Boolean}_R\ b) = \mathit{Boolean}_R\ (\Pi(\neg\ r = \mathit{false}_R \vee b))) \\
\quad \wedge\ (\forall b{:}\mathbb{U};\ r{:}\mathbb{U} \bullet\ (\mathit{Boolean}_R\ b)\ \mathit{or}_R\ r = \mathit{Boolean}_R\ (\Pi(b \vee \neg\ r = \mathit{false}_R))) \\
\quad \wedge\ (\forall b{:}\mathbb{U};\ r{:}\mathbb{U} \bullet\ r\ \mathit{equiv}_R\ (\mathit{Boolean}_R\ b) = \mathit{Boolean}_R\ (\Pi(\neg\ r = \mathit{false}_R \Leftrightarrow b))) \\
\quad \wedge\ (\forall b{:}\mathbb{U};\ r{:}\mathbb{U} \bullet\ (\mathit{Boolean}_R\ b)\ \mathit{equiv}_R\ r = \mathit{Boolean}_R\ (\Pi(b \Leftrightarrow \neg\ r = \mathit{false}_R))) \\
\quad \wedge\ (\forall b{:}\mathbb{U};\ r{:}\mathbb{U} \bullet\ r\ \mathit{xor}_R\ (\mathit{Boolean}_R\ b) = \mathit{Boolean}_R\ (\Pi(r = \mathit{false}_R \Leftrightarrow b))) \\
\quad \wedge\ (\forall b{:}\mathbb{U};\ r{:}\mathbb{U} \bullet\ (\mathit{Boolean}_R\ b)\ \mathit{xor}_R\ r = \mathit{Boolean}_R\ (\Pi(b \Leftrightarrow r = \mathit{false}_R))) \\
⌝));
\end{array}
$$

Description See *cn_boolean_clauses*1 for use.

The effect of rewriting with the theorems presented in this section and the previous section is shown in the following example, in which we show how the expression:

$$N\ \mathit{eq}_R\ (\mathit{real}\ 0)\ \mathit{or}_R\ \mathit{not}_R\ N\ \mathit{less}_R\ (\mathit{real}\ 4) = \mathit{true}_R$$

is rewritten as "$N\ =\ (\mathit{real}\ 0)\ \vee\ \neg\ N\ <_R\ (\mathit{real}\ 4)$".

Rewriting Example

$$
\begin{array}{ll}
& N\ \mathit{eq}_R\ (\mathit{real}\ 0)\ \mathit{or}_R\ \mathit{not}_R\ N\ \mathit{less}_R\ (\mathit{real}\ 4) = \mathit{true}_R \\
1) & \quad ===>\ \mathit{Boolean}_R\ (N = (\mathit{real}\ 0))\ \mathit{or}_R\ \mathit{not}_R\ \mathit{Boolean}_R\ (N <_R (\mathit{real}\ 4)) = \mathit{Boolean}_R\ \mathit{true} \\
2) & \quad ===>\ \mathit{Boolean}_R\ (N = (\mathit{real}\ 0))\ \mathit{or}_R\ \mathit{Boolean}_R\ (\neg\ N <_R (\mathit{real}\ 4)) = \mathit{Boolean}_R\ \mathit{true} \\
3) & \quad ===>\ \mathit{Boolean}_R\ (N = (\mathit{real}\ 0) \vee \neg\ N <_R (\mathit{real}\ 4)) = \mathit{Boolean}_R\ \mathit{true} \\
4) & \quad ===>\ N = (\mathit{real}\ 0) \vee \neg\ N <_R (\mathit{real}\ 4) \Leftrightarrow \mathit{true} \\
5) & \quad ===>\ N = (\mathit{real}\ 0) \vee \neg\ N <_R (\mathit{real}\ 4)
\end{array}
$$

Here in step 1 the theorem of this section starts thing off by turning the ClawZ atomic predicates into Z expressions involving $\mathit{Boolean}_R$. In steps 2 and 3, the theorems of the previous section turn the vocabulary of the theory *CLT_common* into the Z toolkit vocabulary using the argument of $\mathit{Boolean}_R$ to accumulate the result. At step 4 $\mathit{Boolean}_R$ is eliminated. Finally a standard ProofPower-Z simplification removes the unnecessary "$\Leftrightarrow\ \mathit{true}$".

4.2.5 Matlab Operators

Theory Design

$$
\begin{array}{l}
req\_thm(\texttt{"}\boldsymbol{cz\_matlab\_clauses}\texttt{"},\ ([],\ \ulcorner_{\mathrm{Z}} \\
\quad (\_.+_m\_) = (\_+_R\_) \\
\wedge \quad (\_.-_m\_) = (\_-_R\_) \\
\wedge \quad (\_.*_m\_) = (\_*_R\_) \\
\wedge \quad (\_./_m\_) = (\_/_R\_) \\
\wedge \quad (\forall x,y: \mathbb{U} \bullet\ x\ .\backslash_m\ y = y\ /_R\ x) \\
\wedge \quad (\_<_m\_) = (\_less_R\_) \\
\wedge \quad (\_>_m\_) = (\_greater_R\_) \\
\wedge \quad (\_>=_m\_) = (\_greater\_eq_R\_) \\
\wedge \quad (\_<=_m\_) = (\_less\_eq_R\_) \\
\wedge \quad (\_==_m\_) = (\_eq_R\_) \\
\wedge \quad (\_\sim=_m\_) = (\_noteq_R\_) \\
\wedge \quad (\_and_m\_) = (\_and_R\_) \\
\wedge \quad (\_or_m\_) = (\_or_R\_) \\
\wedge \quad (\forall r{:}\mathbb{U}; z{:}\mathbb{U} \bullet\ r\ .\hat{}_m\ (real\ z) = r\ \hat{}_Z\ z) \\
\wedge \quad (\forall x{:}\ \mathbb{U} \bullet\ mp_m\ x = x) \\
\wedge \quad (\forall x{:}\ \mathbb{U} \bullet\ mm_m\ x = (real\ 0)\ -_R\ x) \\
\wedge \quad (\sim_m\_) = (not_R\ \_) \\
\urcorner));
\end{array}
$$

Description This theorem permits the elimination of Matlab operators.

4.2.6 Fcn Operators

Theory Design

$$
\begin{array}{l}
\mathit{req\_thm}(\texttt{"}\boldsymbol{cz\_fcn\_clauses}\texttt{"},\ ([],\ ⌜_{\mathrm{Z}}\\
\quad (\_+_f\_) = (\_+_R\_)\\
\wedge\quad (\_-_f\_) = (\_-_R\_)\\
\wedge\quad (\_*_f\_) = (\_*_R\_)\\
\wedge\quad (\_/_f\_) = (\_/_R\_)\\
\wedge\quad (\_<_f\_) = (\_less_R\_)\\
\wedge\quad (\_>_f\_) = (\_greater_R\_)\\
\wedge\quad (\_>=_f\_) = (\_greater\_eq_R\_)\\
\wedge\quad (\_<=_f\_) = (\_less\_eq_R\_)\\
\wedge\quad (\_=_f\_) = (\_eq_R\_)\\
\wedge\quad (\_\neq_f\_) = (\_noteq_R\_)\\
\wedge\quad (\_and_f\_) = (\_and_R\_)\\
\wedge\quad (\_or_f\_) = (\_or_R\_)\\
\wedge\quad (\forall r{:}\mathbb{U}; z{:}\mathbb{U}\bullet\ r\ \hat{}_f\ (\mathit{real}\ z) = r\ \hat{}_Z\ z)\\
\wedge\quad (\forall x{:}\ \mathbb{U}\bullet\ mp_f\ x = x)\\
\wedge\quad (\forall x{:}\ \mathbb{U}\bullet\ mm_f\ x = (\mathit{real}\ 0)\ -_R\ x)\\
\wedge\quad (not_f\_) = (not_R\ \_)\\
⌝));
\end{array}
$$

Description This theorem permits the elimination of Fcn operators.

4.2.7 Product and Sum

Theory Design

$$
\begin{array}{l}
\mathit{req\_thm}(\texttt{"}\boldsymbol{cz\_bin2dec\_thm}\texttt{"},\ ([],\ ⌜_{\mathrm{Z}}\mathit{bin2dec}\ \langle\rangle = 0 \wedge (\forall\ f{:}\ \mathbb{U};\ l{:}\ \mathbb{U}\bullet\\
\quad \mathit{bin2dec}(⌜\texttt{\$"}Z'\langle\rangle\texttt{"}\ (\mathit{Cons}\ f\ l)⌝) = (\mathit{bin2dec}\ ⌜\texttt{\$"}Z'\langle\rangle\texttt{"}\ l⌝) +\\
\quad (\mathit{if}\ f = \mathit{real}\ 0\ \mathit{then}\ 0\ \mathit{else}\ 1) * (2\ **\ \#\ ⌜\texttt{\$"}Z'\langle\rangle\texttt{"}\ l⌝))\\
⌝));\\
\mathit{req\_thm}(\texttt{"}\boldsymbol{cz\_dot\_product\_empty\_thm}\texttt{"},\ ([],\ ⌜_{\mathrm{Z}}\mathit{dot\_product}\ (\langle\rangle,\langle\rangle) = \mathit{real}\ 0⌝));\\
\mathit{req\_thm}(\texttt{"}\boldsymbol{cz\_dot\_product\_thm}\texttt{"},\ ([],\ ⌜_{\mathrm{Z}}\forall\ h1,h2{:}\mathbb{U};\ t1,t2{:}\mathbb{U}\bullet\ ⌜\mathit{Length}\ t1⌝ = ⌜\mathit{Length}\ t2⌝ \Rightarrow\\
\quad \mathit{dot\_product}\ (⌜\texttt{\$"}Z'\langle\rangle\texttt{"}\ (\mathit{Cons}\ h1\ t1)⌝, ⌜\texttt{\$"}Z'\langle\rangle\texttt{"}\ (\mathit{Cons}\ h2\ t2)⌝)\\
\quad = h1\ *_R\ h2\ +_R\ (\mathit{dot\_product}\ (⌜\texttt{\$"}Z'\langle\rangle\texttt{"}\ t1⌝, ⌜\texttt{\$"}Z'\langle\rangle\texttt{"}\ t2⌝))\\
⌝));\\
\mathit{req\_thm}(\texttt{"}\boldsymbol{cz\_product\_thm}\texttt{"},\ ([],\ ⌜_{\mathrm{Z}}\mathit{product}\ \langle\rangle = \mathit{real}\ 1 \wedge (\forall\ h{:}\mathbb{U};\ t{:}\mathbb{U}\bullet\\
\quad \mathit{product}\ ⌜\texttt{\$"}Z'\langle\rangle\texttt{"}\ (\mathit{Cons}\ h\ t)⌝ = h\ *_R\ (\mathit{product}\ ⌜\texttt{\$"}Z'\langle\rangle\texttt{"}\ t⌝))\\
⌝));\\
\mathit{req\_thm}(\texttt{"}\boldsymbol{cz\_sum\_thm}\texttt{"},\ ([],\ ⌜_{\mathrm{Z}}\mathit{sum}\ \langle\rangle = \mathit{real}\ 0 \wedge (\forall\ h{:}\mathbb{U};\ t{:}\mathbb{U}\bullet\\
\quad \mathit{sum}\ ⌜\texttt{\$"}Z'\langle\rangle\texttt{"}\ (\mathit{Cons}\ h\ t)⌝ = h\ +_R\ (\mathit{sum}\ ⌜\texttt{\$"}Z'\langle\rangle\texttt{"}\ t⌝))\\
⌝));
\end{array}
$$

Description These theorems supports evaluation of "sum" on sequence displays.

4.3 ML Bindings for Theorems

SML

```
val cz_ub_thm : THM;
val cz_lb_thm : THM;
val cz_ge_ub_trans_thm : THM;
val cz_le_lb_trans_thm : THM;
val cz_lb_ub_thm : THM;
val cz_ub_lb_thm : THM;
val cz_lub_thm : THM;
val cz_glb_thm : THM;
val cz_lub_sup_thm :THM;
val cz_dom_lub_thm : THM;
val cz_dom_glb_thm : THM;
val cz_lub_setd_thm : THM;
val cz_glb_setd_thm : THM;
val cz_ub_empty_thm : THM;
val cz_lb_empty_thm : THM;
val cz_lub_unit_thm : THM;
val cz_glb_unit_thm : THM;
val cz_ub_setd_thm : THM;
val cz_lb_setd_thm : THM;
val cz_lub_setd_le_thm : THM;
val cz_le_glb_setd_thm : THM;
val cz_eq_lub_setd_thm : THM;
val cz_lub_setd_eq_thm : THM;
val cz_le_lub_setd_thm : THM;
val cz_eq_glb_setd_thm : THM;
val cz_glb_setd_eq_thm : THM;
val cz_glb_setd_le_thm : THM;
```

Description These are the ML names for the theorems in the theory "*CLT_common*", which contains extensions to the Z toolkit required to support the ClawZ tool output. These theorems concern upper and lower, least upper and greatest lower bounds.

SML

val ***cz_if_thm*** : *THM*;
val ***cz_r2z_thm*** : *THM*;
val ***cz_boolean_clauses*** : *THM*;
val ***cz_boolean_clauses1*** : *THM*;
val ***cz_boolean_clauses2*** : *THM*;
val ***cz_boolean_real_clauses*** : *THM*;
val ***cz_relational_clauses*** : *THM*;
val ***cz_matlab_clauses*** : *THM*;
val ***cz_fcn_clauses*** : *THM*;
val ***cz_bin2dec_thm*** : *THM*;
val ***cz_dot_product_empty_thm*** : *THM*;
val ***cz_dot_product_thm*** : *THM*;
val ***cz_product_thm*** : *THM*;
val ***cz_sum_thm*** : *THM*;

Description These are the ML names for the theorems in the theory "*CLT_common*", which contains extensions to the Z toolkit required to support the ClawZ tool output. These theorems cover expansion of conditionals, boolean expressions, relations, matlab and fcn expressions, binary to decimal, dot product, and sequence product and sum evaluation.

4.4 ML Bindings for Proof Procedures

SML

val ***cz_matrix_conv*** : *CONV*;

Description Conversions for the clawz library. *cz_matrix_conv* applies to terms of the form $smatrix(n, m)$, where s is a sequence display of n sequence displays each of length m, and n and m are numeric literals. In this case the result is ⌜ᶻs matrix (n,m) ⇔ true⌝, any other case fails.

cz_dot_product_conv

Description applies to terms of the form $dot\_product(s1, s2)$, where $s1$ and $s2$ are non-empty sequence displays of equal length. It does the first step in evaluation of such a product, but pulling out the first element of each list and returning the sum of their product and a shorter dot product. This conversion is not suitable for use by end users because it only solves part of the problem (of evaluating dot products) and is therefore supplied only in proof contexts (in which context the whole problem is solved). An improved version suitable for direct use by end users may be made available in later releases.

Errors

528000 ?0 is not a rectangular matrix display of size ?1 by ?2
528001 ?0 is not of the form ⌜ᶻ*md Matrix (r,c)*⌝ *where md is a sequence display of r sequence displays each of length c.*
528002 ?0 is not of the form ⌜ᶻ*dot_product (d1, d2)*⌝ *where d1 and d2 are sequence displays of equal length.*

4.5 PROOF CONTEXTS

NOTE: The proof contexts $'CLT\_common$ etc. are actually defined in [2]. They are also described here for convenience of the user reading this document, and to make it unnecessary for users to consult [2].

SML

(∗ *Proof Context*: ***'CLT_common*** ∗)

Description Component proof contexts for the theory *CLT_common* which supports reasoning about specifications produced by ClawZ.

The main purpose of the 'CLT_common proof context is to automate the elimination of the vocabulary of the theory *CLT_common* in favour of plain Z toolkit constructs wherever this is possible without introducing excessive complexity.

The 'CLT_bounds proof context provides extra facilities for reducing claims about bounds of set displays to inequalities for solution using linear arithmetic.

Vocabulary concerning upper and lower bounds is left untouched.

Contents

Rewriting:

cz_if_thm
cz_r2z_thm
cz_boolean_clauses
cz_boolean_real_clauses
cz_relational_clauses
cz_matlab_clauses
cz_fcn_clauses
cz_product_thm
cz_sum_thm
cz_matrix_conv
cz_dot_product_empty_thm
cz_dot_product_conv

Stripping theorems:

cz_if_thm
cz_r2z_thm
cz_boolean_clauses
cz_boolean_real_clauses
cz_relational_clauses
cz_matlab_clauses
cz_fcn_clauses
cz_product_thm
cz_sum_thm
cz_matrix_conv
cz_dot_product_empty_thm
cz_dot_product_conv

See Also 'CLT_bounds

SML

```
(* Proof Context: 'CLT_bounds *)
```

Description Component proof context for the theory *CLT_common* which supports the ClawZ library.

The purpose of the proof context is to automate the proof of claims about the bounds of set displays.

This proof context will typically be used in conjunction with the Z real linear arithmetic proof context *'z_ℝ_lin_arith*.

```
set_merge_pcs["'CLT_common", "'z_reals", "z_library"];
```

Contents

Rewriting and stripping:

```
    cz_ub_empty_thm,
    cz_lb_empty_thm,
    cz_lub_unit_thm,
    cz_glb_unit_thm,
    cz_ub_setd_thm,
    cz_lb_setd_thm,
    cz_lub_setd_le_thm,
    cz_le_glb_setd_thm,
    cz_eq_lub_setd_thm,
    cz_lub_setd_eq_thm,
    cz_le_lub_setd_thm,
    cz_eq_glb_setd_thm,
    cz_glb_setd_eq_thm,
    cz_glb_setd_le_thm
```

Automatic proof:

```
    fn thms => rewrite_tac[]
    THEN PC_T1 "'z_ℝ_lin_arith" asm_proof_tac thms
```

See Also *'CLT_common*, *'CLT_seq*

SML

(∗ *Proof Context*: **'CLT_seq** ∗)

Description The purpose of the proof context is to automate the proof of claims about sequence displays. This includes:

- evaluation of the length of sequence displays.
- evaluation of the domain and range of sequence displays.
- evaluation of selection from sequence displays by application to a numeric literal.

Contents

Rewriting:

z_seqd_∈_seq_thm,
z_dom_seqd_thm,
z_ran_seqd_thm,
z_seqd_⌢_⟨⟩_clauses,
z_size_seqd_conv,
z_seqd_app_conv

Stripping:

z_seqd_∈_seq_thm,
z_dom_seqd_thm,
z_ran_seqd_thm,
z_seqd_⌢_⟨⟩_clauses,
z_size_seqd_conv,
z_seqd_app_conv

See Also *'CLT_common*, *'CLT_bounds*

4.6 Epilogue

SML

```
end (* end of signature CLT_common *);
```

SML

```
reset_flag ("z_type_check_only");
reset_flag ("z_use_axioms");
reset_flag ("standard_z_paras");
```

5 TEST POLICY

The tests include most of the endproof goals supplied by DERA.

6 THE THEORY *CLT_common*

7 THE Z THEORY CLT_common

7.1 Parents

cache′ clawzlib z_reals
cn z_library

7.2 Global Variables

$(\boldsymbol{if\ \_\ then\ \_\ else\ \_})[\boldsymbol{X}]$ $BOOL \times X \times X \leftrightarrow X$

$\boldsymbol{r2z}$ $\mathbb{R} \leftrightarrow \mathbb{Z}$

$(\_ :_z \_)$ $\mathbb{Z} \times \mathbb{Z} \leftrightarrow \mathbb{Z} \leftrightarrow \mathbb{Z}$

$(\_\ \boldsymbol{lb_R}\ \_)$ $\mathbb{R} \leftrightarrow \mathbb{P}\,\mathbb{R}$

$\boldsymbol{glb_R}$ $\mathbb{P}\,\mathbb{R} \leftrightarrow \mathbb{R}$

$(\_\ \boldsymbol{ub_R}\ \_)$ $\mathbb{R} \leftrightarrow \mathbb{P}\,\mathbb{R}$

$\boldsymbol{lub_R}$ $\mathbb{P}\,\mathbb{R} \leftrightarrow \mathbb{R}$

$\boldsymbol{Boolean_R}$ $BOOL \leftrightarrow \mathbb{R}$

$(\boldsymbol{is\_true_R}\ \_)$ $\mathbb{P}\,\mathbb{R}$

$\boldsymbol{true_R}$ $\mathbb{R}$

$\boldsymbol{false_R}$ $\mathbb{R}$

$\boldsymbol{liftrel_R}[\boldsymbol{X}, \boldsymbol{Y}]$ $(X \leftrightarrow Y) \leftrightarrow X \times Y \leftrightarrow \mathbb{R}$

$(\_\ \boldsymbol{greater\_eq_R}\ \_)$ $\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$

$(\_\ \boldsymbol{greater_R}\ \_)$ $\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$

$(\_\ \boldsymbol{less\_eq_R}\ \_)$ $\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$

$(\_\ \boldsymbol{less_R}\ \_)$ $\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$

$(\_\ \boldsymbol{noteq_R}\ \_)$ $\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$

$(\_\ \boldsymbol{eq_R}\ \_)$ $\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$

$\boldsymbol{floor_R}$ $\mathbb{R} \leftrightarrow \mathbb{R}$

$\boldsymbol{ceil_R}$ $\mathbb{R} \leftrightarrow \mathbb{R}$

$\boldsymbol{round_R}$ $\mathbb{R} \leftrightarrow \mathbb{R}$

$\boldsymbol{fix_R}$ $\mathbb{R} \leftrightarrow \mathbb{R}$

$(\_\ \boldsymbol{xor_R}\ \_)$ $\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$

$(\_\ \boldsymbol{equiv_R}\ \_)$ $\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$

$(\_\ \boldsymbol{or_R}\ \_)$ $\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$

$(\_\ and_R\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(not_R\ \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ .+_m\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ .-_m\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ .*_m\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ ./_m\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ .\backslash_m\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ <_m\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ >_m\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ >=_m\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ <=_m\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ ==_m\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ \sim=_m\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ and_m\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ or_m\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ .\hat{}_m\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(mp_m\ \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(mm_m\ \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(\sim_m\ \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ :_m\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{Z} \leftrightarrow \mathbb{R}$
$(\_\ +_f\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ -_f\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ *_f\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ /_f\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ <_f\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ >_f\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ >=_f\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ <=_f\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ =_f\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ \neq_f\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ and_f\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ or_f\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(\_\ \hat{}_f\ \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(mp_f\ \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(mm_f\ \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(not_f\ \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(sqrt_m\ \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(tan_m\ \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(tanh_m\ \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(hypot_m\ \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(log_m\ \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(log10_m\ \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(sin_m\ \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$

$(sinh_m \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(cos_m \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(cosh_m \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(exp_m \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(fabs_m \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(floor_m \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(abs_m \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(acos_m \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(asin_m \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(atan_m \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(ceil_m \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(atan2_m \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(power_m \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(rem_m \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(sqrt_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(tan_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(tanh_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(ln_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(sgn_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(hypot_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(log_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(log10_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(sin_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(sinh_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(cos_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(cosh_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(exp_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(fabs_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(floor_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(abs_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(acos_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(asin_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(atan_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(ceil_f \_)$	$\mathbb{R} \leftrightarrow \mathbb{R}$
$(atan2_f \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(power_f \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$(rem_f \_)$	$\mathbb{R} \times \mathbb{R} \leftrightarrow \mathbb{R}$
$bin2dec$	$(\mathbb{Z} \leftrightarrow \mathbb{R}) \leftrightarrow \mathbb{Z}$
$(\_\ Matrix\ \_)$	$(\mathbb{Z} \leftrightarrow \mathbb{Z} \leftrightarrow \mathbb{R}) \leftrightarrow \mathbb{Z} \times \mathbb{Z}$

dot_product $(\mathbb{Z} \leftrightarrow \mathbb{R}) \times (\mathbb{Z} \leftrightarrow \mathbb{R}) \leftrightarrow \mathbb{R}$
product $(\mathbb{Z} \leftrightarrow \mathbb{R}) \leftrightarrow \mathbb{R}$
sum $(\mathbb{Z} \leftrightarrow \mathbb{R}) \leftrightarrow \mathbb{R}$

7.3 Fixity

fun 0 rightassoc
$(if\ \_\ then\ \_\ else\ \_)$

fun 1 leftassoc
$(\_\ and_m\ \_)$ $(\_\ or_f\ \_)$ $(\_\ or_m\ \_)$

fun 2 leftassoc
$(\_\ and_f\ \_)$ $(\_\ :_z\ \_)$ $(\_\ <_m\ \_)$ $(\_\ >=_m\ \_)$ $(\_\ \sim=_m\ \_)$
$(\_\ :_m\ \_)$ $(\_\ <=_m\ \_)$ $(\_\ ==_m\ \_)$ $(\_\ >_m\ \_)$

fun 3 leftassoc
$(\_\ .+_m\ \_)$ $(\_\ .-_m\ \_)$ $(\_\ =_f\ \_)$ $(\_\ \neq_f\ \_)$

fun 4 leftassoc
$(\_\ .*_m\ \_)$ $(\_\ .\backslash_m\ \_)$ $(\_\ <_f\ \_)$ $(\_\ >_f\ \_)$
$(\_\ ./_m\ \_)$ $(\_\ <=_f\ \_)$ $(\_\ >=_f\ \_)$

fun 5 leftassoc
$(\_\ +_f\ \_)$ $(\_\ -_f\ \_)$ $(\_\ .\hat{}_m\ \_)$

fun 5 rightassoc
$(mm_m\ \_)$ $(mp_m\ \_)$

fun 6 leftassoc
$(\_\ *_f\ \_)$ $(\_\ /_f\ \_)$

fun 6 rightassoc
$(\sim_m\ \_)$

fun 7 rightassoc
$(not_f\ \_)$

fun 8 rightassoc
$(mm_f\ \_)$ $(mp_f\ \_)$

fun* 9 *leftassoc

(_ $\hat{}_f$ _)

fun* 10 *leftassoc

(_ $equiv_R$ _)

fun* 10 *rightassoc

(abs_f _)	($atan_m$ _)	($fabs_f$ _)	(log_f _)	(sin_f _)
(abs_m _)	($ceil_f$ _)	($fabs_m$ _)	(log_m _)	(sin_m _)
($acos_f$ _)	($ceil_m$ _)	($floor_f$ _)	($power_f$ _)	($sqrt_f$ _)
($acos_m$ _)	($cosh_f$ _)	($floor_m$ _)	($power_m$ _)	($sqrt_m$ _)
($asin_f$ _)	($cosh_m$ _)	($hypot_f$ _)	(rem_f _)	($tanh_f$ _)
($asin_m$ _)	(cos_f _)	($hypot_m$ _)	(rem_m _)	($tanh_m$ _)
($atan2_f$ _)	(cos_m _)	(ln_f _)	(sgn_f _)	(tan_f _)
($atan2_m$ _)	(exp_f _)	($log10_f$ _)	($sinh_f$ _)	(tan_m _)
($atan_f$ _)	(exp_m _)	($log10_m$ _)	($sinh_m$ _)	

fun* 20 *leftassoc

(_ xor_R _)

fun* 30 *leftassoc

(_ or_R _)

fun* 40 *leftassoc

(_ and_R _)

fun* 50 *rightassoc

(not_R _)

fun* 200 *leftassoc

(_ eq_R _)	(_ $less_R$ _)
(_ $less\_eq_R$ _)	(_ $noteq_R$ _)

fun* 210 *leftassoc

(_ $greater\_eq_R$ _) (_ $greater_R$ _)

rel (is_true_R _) (_ *Matrix* _)
(_ lb_R _) (_ ub_R _)

7.4 Axioms

if _ then _ else _
$$\vdash [X]((\mathit{if}\ \_\ \mathit{then}\ \_\ \mathit{else}\ \_)[X] \in \mathit{BOOL} \times X \times X \to X$$
$$\land\ (\forall\ b : \mathit{BOOL};\ x,\ y : X$$
$$\bullet\ (b \Rightarrow (\mathit{if}\ \_\ \mathit{then}\ \_\ \mathit{else}\ \_)[X]\ (b,\ x,\ y) = x)$$
$$\land\ (\neg$$
$$b$$
$$\Rightarrow (\mathit{if}\ \_\ \mathit{then}\ \_\ \mathit{else}\ \_)[X]\ (b,\ x,\ y) = y)))$$

r2z
$$\vdash \mathit{r2z} \in \mathbb{R} \pfun \mathbb{Z} \land (\forall\ i : \mathbb{Z} \bullet \mathit{real}\ i \mapsto i \in \mathit{r2z})$$

_ :z _
$$\vdash (\_ :_z \_) \in \mathbb{Z} \times \mathbb{Z} \to \mathit{seq}\ \mathbb{Z}$$
$$\land\ (\forall\ x,\ y : \mathbb{Z}$$
$$\bullet\ x :_z y$$
$$= \{i,\ z : \mathbb{Z}$$
$$\mid i = z - x + 1 \land x \leq z \land z \leq y\})$$

_ lbR _
$$\vdash (\_\ \mathit{lb}_R\ \_) \in \mathbb{R} \rightarrow \mathbb{P}\ \mathbb{R}$$
$$\land\ (\forall\ r : \mathbb{R};\ \mathit{sr} : \mathbb{P}\ \mathbb{R}$$
$$\bullet\ r\ \mathit{lb}_R\ \mathit{sr} \Leftrightarrow (\forall\ x : \mathit{sr} \bullet r \leq_R x))$$

glbR
$$\vdash \mathit{glb}_R \in \mathbb{P}\ \mathbb{R} \pfun \mathbb{R}$$
$$\land\ (\forall\ \mathit{sr} : \mathbb{P}\ \mathbb{R};\ \mathit{glb} : \mathbb{R}$$
$$\bullet\ \mathit{sr} \mapsto \mathit{glb} \in \mathit{glb}_R$$
$$\Leftrightarrow \mathit{glb}\ \mathit{lb}_R\ \mathit{sr}$$
$$\land\ (\forall\ \mathit{lb} : \mathbb{R} \mid \mathit{lb}\ \mathit{lb}_R\ \mathit{sr} \bullet \mathit{lb} \leq_R \mathit{glb}))$$

_ ubR _
$$\vdash (\_\ \mathit{ub}_R\ \_) \in \mathbb{R} \rightarrow \mathbb{P}\ \mathbb{R}$$
$$\land\ (\forall\ r : \mathbb{R};\ \mathit{sr} : \mathbb{P}\ \mathbb{R}$$
$$\bullet\ r\ \mathit{ub}_R\ \mathit{sr} \Leftrightarrow (\forall\ x : \mathit{sr} \bullet r \geq_R x))$$

lubR
$$\vdash \mathit{lub}_R \in \mathbb{P}\ \mathbb{R} \pfun \mathbb{R}$$
$$\land\ (\forall\ \mathit{sr} : \mathbb{P}\ \mathbb{R};\ \mathit{lub} : \mathbb{R}$$
$$\bullet\ \mathit{sr} \mapsto \mathit{lub} \in \mathit{lub}_R$$
$$\Leftrightarrow \mathit{lub}\ \mathit{ub}_R\ \mathit{sr}$$
$$\land\ (\forall\ \mathit{ub} : \mathbb{R} \mid \mathit{ub}\ \mathit{ub}_R\ \mathit{sr} \bullet \mathit{ub} \geq_R \mathit{lub}))$$

BooleanR

is_trueR _

trueR

falseR
$$\vdash (\{\mathit{true}_R,\ \mathit{false}_R\} \subseteq \mathbb{R}$$
$$\land\ (\mathit{is\_true}_R\ \_) \in \mathbb{P}\ \mathbb{R}$$
$$\land\ \mathit{Boolean}_R \in \mathit{BOOL} \to \mathbb{R})$$
$$\land\ \mathit{true}_R = \mathit{real}\ 1$$
$$\land\ \mathit{false}_R = \mathit{real}\ 0$$
$$\land\ (\forall\ x : \mathbb{U} \bullet \mathit{is\_true}_R\ x \Leftrightarrow \neg\ x = \mathit{false}_R)$$
$$\land\ (\forall\ x : \mathbb{U}$$
$$\bullet\ \mathit{Boolean}_R\ x = \mathit{if}\ x\ \mathit{then}\ \mathit{true}_R\ \mathit{else}\ \mathit{false}_R)$$

liftrelR
$$\vdash [X,$$
$$Y](\mathit{liftrel}_R[X,\ Y] \in \mathbb{P}\ (X \times Y) \to X \times Y \to \mathbb{R}$$
$$\land\ (\forall\ r : \mathbb{P}\ (X \times Y);\ x : X;\ y : Y$$
$$\bullet\ (\mathit{liftrel}_R[X,\ Y]\ r)\ (x,\ y)$$
$$= \mathit{Boolean}_R\ ((x,\ y) \in r)))$$

_ greater_eqR _

_ greaterR _

$\_\ \boldsymbol{less\_eq_R}\ \_$

$\_\ \boldsymbol{less_R}\ \_$

$\_\ \boldsymbol{noteq_R}\ \_$

$\_\ \boldsymbol{eq_R}\ \_$ $\vdash ((\_\ eq_R\ \_) \in \mathbb{R} \times \mathbb{R} \to \mathbb{R}$
$\wedge\ (\_\ noteq_R\ \_) \in \mathbb{R} \times \mathbb{R} \to \mathbb{R}$
$\wedge\ (\_\ less_R\ \_) \in \mathbb{R} \times \mathbb{R} \to \mathbb{R}$
$\wedge\ (\_\ less\_eq_R\ \_) \in \mathbb{R} \times \mathbb{R} \to \mathbb{R}$
$\wedge\ (\_\ greater_R\ \_) \in \mathbb{R} \times \mathbb{R} \to \mathbb{R}$
$\wedge\ (\_\ greater\_eq_R\ \_) \in \mathbb{R} \times \mathbb{R} \to \mathbb{R})$
$\wedge\ (\_\ eq_R\ \_) = liftrel_R\ \{x, y : \mathbb{R} \mid x = y\}$
$\wedge\ (\_\ noteq_R\ \_) = liftrel_R\ \{x, y : \mathbb{R} \mid x \neq y\}$
$\wedge\ (\_\ less_R\ \_) = liftrel_R\ (\_ <_R \_)$
$\wedge\ (\_\ less\_eq_R\ \_) = liftrel_R\ (\_ \leq_R \_)$
$\wedge\ (\_\ greater_R\ \_) = liftrel_R\ (\_ >_R \_)$
$\wedge\ (\_\ greater\_eq_R\ \_) = liftrel_R\ (\_ \geq_R \_)$

$\boldsymbol{floor_R}$ $\vdash floor_R \in \mathbb{R} \to real\ (\!|\ \mathbb{Z}\ |\!)$
$\wedge\ (\forall\ x : \mathbb{R}$
$\bullet\ real\ 0 \leq_R x -_R floor_R\ x$
$\wedge\ x -_R floor_R\ x <_R real\ 1)$

$\boldsymbol{ceil_R}$ $\vdash ceil_R \in \mathbb{R} \to real\ (\!|\ \mathbb{Z}\ |\!)$
$\wedge\ (\forall\ x : \mathbb{R}$
$\bullet\ real\ 0 \leq_R ceil_R\ x -_R x$
$\wedge\ ceil_R\ x -_R x <_R real\ 1)$

$\boldsymbol{round_R}$ $\vdash round_R \in \mathbb{R} \to real\ (\!|\ \mathbb{Z}\ |\!)$
$\wedge\ (\forall\ x : \mathbb{R}$
$\bullet\ round_R\ x$
$= if$
$real\ 0 \leq_R x\ then$
$floor_R\ (x +_R 5\ e\ (\sim 1))\ else$
$ceil_R\ (x -_R 5\ e\ (\sim 1)))$

$\boldsymbol{fix_R}$ $\vdash fix_R \in \mathbb{R} \to real\ (\!|\ \mathbb{Z}\ |\!)$
$\wedge\ (\forall\ x : \mathbb{R}$
$\bullet\ fix_R\ x$
$= if\ real\ 0 \leq_R x\ then\ floor_R\ x\ else\ ceil_R\ x)$

$\_\ \boldsymbol{xor_R}\ \_$

$\_\ \boldsymbol{equiv_R}\ \_$

$\_\ \boldsymbol{or_R}\ \_$

$\_\ \boldsymbol{and_R}\ \_$

$\boldsymbol{not_R}\ \_$ $\vdash ((not_R\ \_) \in \mathbb{R} \to \mathbb{R}$
$\wedge\ (\_\ and_R\ \_) \in \mathbb{R} \times \mathbb{R} \to \mathbb{R}$
$\wedge\ (\_\ or_R\ \_) \in \mathbb{R} \times \mathbb{R} \to \mathbb{R}$
$\wedge\ (\_\ equiv_R\ \_) \in \mathbb{R} \times \mathbb{R} \to \mathbb{R}$
$\wedge\ (\_\ xor_R\ \_) \in \mathbb{R} \times \mathbb{R} \to \mathbb{R})$
$\wedge\ (\forall\ l : \mathbb{R}$
$\bullet\ not_R\ l$
$= if\ is\_true_R\ l\ then\ false_R\ else\ true_R)$
$\wedge\ (\forall\ l, r : \mathbb{R}$
$\bullet\ l\ and_R\ r$

$$
\begin{array}{l}
\quad = \mathit{if} \\
\qquad \mathit{is\_true}_R\ l\ \mathit{then} \\
\qquad \mathit{not}_R\ \mathit{not}_R\ r\ \mathit{else} \\
\qquad \mathit{false}_R) \\
\land\ (\forall\ l,\ r : \mathbb{R} \\
\quad \bullet\ l\ \mathit{or}_R\ r \\
\quad = \mathit{if} \\
\qquad \mathit{is\_true}_R\ l\ \mathit{then} \\
\qquad \mathit{true}_R\ \mathit{else} \\
\qquad \mathit{not}_R\ \mathit{not}_R\ r) \\
\land\ (\forall\ l,\ r : \mathbb{R} \\
\quad \bullet\ l\ \mathit{equiv}_R\ r \\
\quad = \mathit{if} \\
\qquad \mathit{is\_true}_R\ l\ \mathit{then} \\
\qquad \mathit{not}_R\ \mathit{not}_R\ r\ \mathit{else} \\
\qquad \mathit{not}_R\ r) \\
\land\ (\forall\ l,\ r : \mathbb{R} \bullet l\ \mathit{xor}_R\ r = \mathit{not}_R\ (l\ \mathit{equiv}_R\ r))
\end{array}
$$

$$
\begin{array}{l}
\_ \cdot +_{\boldsymbol{m}} \_ \\
\_ \cdot -_{\boldsymbol{m}} \_ \\
\_ \cdot *_{\boldsymbol{m}} \_ \\
\_ \cdot /_{\boldsymbol{m}} \_ \\
\_ \cdot \backslash_{\boldsymbol{m}} \_ \qquad \vdash \{(\_ .+_m \_), \\
\qquad\qquad\qquad (\_ .-_m \_), \\
\qquad\qquad\qquad (\_ .*_m \_), \\
\qquad\qquad\qquad (\_ ./_m \_), \\
\qquad\qquad\qquad (\_ .\backslash_m \_)\} \\
\qquad\qquad\qquad \subseteq \mathbb{R} \times \mathbb{R} \to \mathbb{R} \\
\qquad\qquad \land\ (\_ .+_m \_) = (\_ +_R \_) \\
\qquad\qquad \land\ (\_ .-_m \_) = (\_ -_R \_) \\
\qquad\qquad \land\ (\_ .*_m \_) = (\_ *_R \_) \\
\qquad\qquad \land\ (\_ ./_m \_) = (\_ /_R \_) \\
\qquad\qquad \land\ (\forall\ x,\ y : \mathbb{R} \bullet x\ .\backslash_m\ y = y\ /_R\ x)
\end{array}
$$

$$
\begin{array}{l}
\_ <_{\boldsymbol{m}} \_ \\
\_ >_{\boldsymbol{m}} \_ \\
\_ >=_{\boldsymbol{m}} \_ \\
\_ <=_{\boldsymbol{m}} \_ \\
\_ ==_{\boldsymbol{m}} \_ \\
\_ \sim=_{\boldsymbol{m}} \_ \qquad \vdash \{(\_ <_m \_), \\
\qquad\qquad\qquad (\_ >_m \_), \\
\qquad\qquad\qquad (\_ >=_m \_), \\
\qquad\qquad\qquad (\_ <=_m \_), \\
\qquad\qquad\qquad (\_ ==_m \_), \\
\qquad\qquad\qquad (\_ \sim=_m \_)\} \\
\qquad\qquad\qquad \subseteq \mathbb{R} \times \mathbb{R} \to \mathbb{R} \\
\qquad\qquad \land\ (\_ <_m \_) = (\_\ \mathit{less}_R\ \_) \\
\qquad\qquad \land\ (\_ >_m \_) = (\_\ \mathit{greater}_R\ \_) \\
\qquad\qquad \land\ (\_ >=_m \_) = (\_\ \mathit{greater\_eq}_R\ \_) \\
\qquad\qquad \land\ (\_ <=_m \_) = (\_\ \mathit{less\_eq}_R\ \_)
\end{array}
$$

$$
\begin{array}{ll}
 & \quad \land\ (\_ ==_m \_) = (\_\ eq_R\ \_) \\
 & \quad \land\ (\_ \sim=_m \_) = (\_\ noteq_R\ \_) \\
\_\ \boldsymbol{and_m}\ \_ & \\
\_\ \boldsymbol{or_m}\ \_ & \vdash \{(\_\ and_m\ \_), (\_\ or_m\ \_)\} \subseteq \mathbb{R} \times \mathbb{R} \to \mathbb{R} \\
 & \quad \land\ (\_\ and_m\ \_) = (\_\ and_R\ \_) \\
 & \quad \land\ (\_\ or_m\ \_) = (\_\ or_R\ \_) \\
\_\ \boldsymbol{.\hat{}_m}\ \_ & \vdash (\_\ .\hat{}_m\ \_) \in \mathbb{R} \times \mathbb{R} \pfun \mathbb{R} \\
 & \quad \land\ dom\ (\_\ .\hat{}_m\ \_) = \mathbb{R} \times dom\ r2z \\
 & \quad \land\ (\forall\ x : \mathbb{R};\ y : dom\ r2z \bullet x\ .\hat{}_m\ y = x\ \hat{}_Z\ r2z\ y) \\
\boldsymbol{mp_m}\ \_ & \\
\boldsymbol{mm_m}\ \_ & \\
\boldsymbol{\sim_m}\ \_ & \vdash \{(mp_m\ \_), (mm_m\ \_), (\sim_m\ \_)\} \subseteq \mathbb{R} \to \mathbb{R} \\
 & \quad \land\ (\forall\ x : \mathbb{R} \\
 & \qquad \bullet\ mp_m\ x = x \\
 & \qquad\ \land\ mm_m\ x = real\ 0\ -_R\ x \\
 & \qquad\ \land\ (\sim_m\ \_) = (not_R\ \_)) \\
\_\ \boldsymbol{:_m}\ \_ & \vdash (\_\ :_m\ \_) \in \mathbb{R} \times \mathbb{R} \to seq\ \mathbb{R} \\
 & \quad \land\ (\forall\ x,\ y : \mathbb{R} \\
 & \qquad \bullet\ x :_m y \\
 & \qquad\ = \{z : \mathbb{R};\ i : \mathbb{Z} \\
 & \qquad\quad \mid z = x +_R real\ i \land x \leq_R z \land z \leq_R y \\
 & \qquad\quad \bullet\ (i + 1,\ z)\}) \\
\_\ \boldsymbol{+_f}\ \_ & \\
\_\ \boldsymbol{-_f}\ \_ & \\
\_\ \boldsymbol{*_f}\ \_ & \\
\_\ \boldsymbol{/_f}\ \_ & \vdash \{(\_ +_f \_), (\_ -_f \_), (\_ *_f \_), (\_ /_f \_)\} \\
 & \qquad \subseteq \mathbb{R} \times \mathbb{R} \to \mathbb{R} \\
 & \quad \land\ (\_ +_f \_) = (\_ +_R \_) \\
 & \quad \land\ (\_ -_f \_) = (\_ -_R \_) \\
 & \quad \land\ (\_ *_f \_) = (\_ *_R \_) \\
 & \quad \land\ (\_ /_f \_) = (\_ /_R \_) \\
\_\ \boldsymbol{<_f}\ \_ & \\
\_\ \boldsymbol{>_f}\ \_ & \\
\_\ \boldsymbol{>=_f}\ \_ & \\
\_\ \boldsymbol{<=_f}\ \_ & \vdash \{(\_ <_f \_), (\_ >_f \_), (\_ >=_f \_), (\_ <=_f \_)\} \\
 & \qquad \subseteq \mathbb{R} \times \mathbb{R} \to \mathbb{R} \\
 & \quad \land\ (\_ <_f \_) = (\_\ less_R\ \_) \\
 & \quad \land\ (\_ >_f \_) = (\_\ greater_R\ \_) \\
 & \quad \land\ (\_ >=_f \_) = (\_\ greater\_eq_R\ \_) \\
 & \quad \land\ (\_ <=_f \_) = (\_\ less\_eq_R\ \_) \\
\_\ \boldsymbol{=_f}\ \_ & \\
\_\ \boldsymbol{\neq_f}\ \_ & \vdash \{(\_ =_f \_), (\_ \neq_f \_)\} \subseteq \mathbb{R} \times \mathbb{R} \to \mathbb{R} \\
 & \quad \land\ (\_ =_f \_) = (\_\ eq_R\ \_) \\
 & \quad \land\ (\_ \neq_f \_) = (\_\ noteq_R\ \_) \\
\_\ \boldsymbol{and_f}\ \_ & \\
\_\ \boldsymbol{or_f}\ \_ & \vdash \{(\_\ and_f\ \_), (\_\ or_f\ \_)\} \subseteq \mathbb{R} \times \mathbb{R} \to \mathbb{R} \\
 & \quad \land\ (\_\ and_f\ \_) = (\_\ and_R\ \_) \\
 & \quad \land\ (\_\ or_f\ \_) = (\_\ or_R\ \_)
\end{array}
$$

$\_ \mathbin{\hat{}}_{\boldsymbol{f}} \_$

$$\vdash (\_ \mathbin{\hat{}}_f \_) \in \mathbb{R} \times \mathbb{R} \pfun \mathbb{R}$$
$$\land\ dom\ (\_ \mathbin{\hat{}}_f \_) = \mathbb{R} \times dom\ r2z$$
$$\land\ (\forall\ x : \mathbb{R};\ y : dom\ r2z \bullet x \mathbin{\hat{}}_f y = x \mathbin{\hat{}}_Z r2z\ y)$$

$\boldsymbol{mp_f}\ \_$

$\boldsymbol{mm_f}\ \_$

$\boldsymbol{not_f}\ \_$

$$\vdash \{(mp_f\ \_), (mm_f\ \_), (not_f\ \_)\} \subseteq \mathbb{R} \to \mathbb{R}$$
$$\land\ (\forall\ x : \mathbb{R}$$
$$\bullet\ mp_f\ x = x$$
$$\land\ mm_f\ x = real\ 0 -_R x$$
$$\land\ (not_f\ \_) = (not_R\ \_))$$

$\boldsymbol{sqrt_m}\ \_$

$\boldsymbol{tan_m}\ \_$

$\boldsymbol{tanh_m}\ \_$

$\boldsymbol{hypot_m}\ \_$

$\boldsymbol{log_m}\ \_$

$\boldsymbol{log10_m}\ \_$

$\boldsymbol{sin_m}\ \_$

$\boldsymbol{sinh_m}\ \_$

$\boldsymbol{cos_m}\ \_$

$\boldsymbol{cosh_m}\ \_$

$\boldsymbol{exp_m}\ \_$

$\boldsymbol{fabs_m}\ \_$

$\boldsymbol{floor_m}\ \_$

$\boldsymbol{abs_m}\ \_$

$\boldsymbol{acos_m}\ \_$

$\boldsymbol{asin_m}\ \_$

$\boldsymbol{atan_m}\ \_$

$\boldsymbol{ceil_m}\ \_$

$$\vdash (\{(abs_m\ \_),$$
$$(acos_m\ \_),$$
$$(asin_m\ \_),$$
$$(atan_m\ \_),$$
$$(ceil_m\ \_)\}$$
$$\subseteq \mathbb{R} \pfun \mathbb{R}$$
$$\land\ \{(cos_m\ \_),$$
$$(cosh_m\ \_),$$
$$(exp_m\ \_),$$
$$(fabs_m\ \_),$$
$$(floor_m\ \_)\}$$
$$\subseteq \mathbb{R} \pfun \mathbb{R}$$
$$\land\ \{(hypot_m\ \_),$$
$$(log_m\ \_),$$
$$(log10_m\ \_),$$
$$(sin_m\ \_),$$
$$(sinh_m\ \_)\}$$
$$\subseteq \mathbb{R} \pfun \mathbb{R}$$
$$\land\ \{(sqrt_m\ \_), (tan_m\ \_), (tanh_m\ \_)\} \subseteq \mathbb{R} \pfun \mathbb{R})$$
$$\land\ (ceil_m\ \_) = ceil_R$$
$$\land\ (floor_m\ \_) = floor_R$$

$$
\begin{array}{l}
\boldsymbol{atan2_m}\ \_ \\
\boldsymbol{power_m}\ \_ \\
\boldsymbol{rem_m}\ \_ \quad \vdash \{(atan2_m\ \_),\ (power_m\ \_),\ (rem_m\ \_)\} \subseteq \mathbb{R} \times \mathbb{R} \pfun \mathbb{R} \\
\qquad\qquad \land\ true \\
\boldsymbol{sqrt_f}\ \_ \\
\boldsymbol{tan_f}\ \_ \\
\boldsymbol{tanh_f}\ \_ \\
\boldsymbol{ln_f}\ \_ \\
\boldsymbol{sgn_f}\ \_ \\
\boldsymbol{hypot_f}\ \_ \\
\boldsymbol{log_f}\ \_ \\
\boldsymbol{log10_f}\ \_ \\
\boldsymbol{sin_f}\ \_ \\
\boldsymbol{sinh_f}\ \_ \\
\boldsymbol{cos_f}\ \_ \\
\boldsymbol{cosh_f}\ \_ \\
\boldsymbol{exp_f}\ \_ \\
\boldsymbol{fabs_f}\ \_ \\
\boldsymbol{floor_f}\ \_ \\
\boldsymbol{abs_f}\ \_ \\
\boldsymbol{acos_f}\ \_ \\
\boldsymbol{asin_f}\ \_ \\
\boldsymbol{atan_f}\ \_ \\
\boldsymbol{ceil_f}\ \_ \quad \vdash (\{(abs_f\ \_), \\
\qquad\qquad\qquad (acos_f\ \_), \\
\qquad\qquad\qquad (asin_f\ \_), \\
\qquad\qquad\qquad (atan_f\ \_), \\
\qquad\qquad\qquad (ceil_f\ \_)\} \\
\qquad\qquad\qquad \subseteq \mathbb{R} \pfun \mathbb{R} \\
\qquad\qquad \land\ \{(cos_f\ \_), \\
\qquad\qquad\qquad (cosh_f\ \_), \\
\qquad\qquad\qquad (exp_f\ \_), \\
\qquad\qquad\qquad (fabs_f\ \_), \\
\qquad\qquad\qquad (floor_f\ \_)\} \\
\qquad\qquad\qquad \subseteq \mathbb{R} \pfun \mathbb{R} \\
\qquad\qquad \land\ \{(hypot_f\ \_), \\
\qquad\qquad\qquad (log_f\ \_), \\
\qquad\qquad\qquad (log10_f\ \_), \\
\qquad\qquad\qquad (sin_f\ \_), \\
\qquad\qquad\qquad (sinh_f\ \_)\} \\
\qquad\qquad\qquad \subseteq \mathbb{R} \pfun \mathbb{R} \\
\qquad\qquad \land\ \{(sqrt_f\ \_), \\
\qquad\qquad\qquad (tan_f\ \_), \\
\qquad\qquad\qquad (tanh_f\ \_), \\
\qquad\qquad\qquad (ln_f\ \_), \\
\qquad\qquad\qquad (sgn_f\ \_)\} \\
\qquad\qquad\qquad \subseteq \mathbb{R} \pfun \mathbb{R}) \\
\qquad\qquad \land\ (ceil_f\ \_) = ceil_R
\end{array}
$$

$$\wedge\ (floor_f\ \_) = floor_R$$

$\boldsymbol{atan2_f\ \_}$

$\boldsymbol{power_f\ \_}$

$\boldsymbol{rem_f\ \_}$ $\vdash \{(atan2_f\ \_), (power_f\ \_), (rem_f\ \_)\} \subseteq \mathbb{R} \times \mathbb{R} \pfun \mathbb{R}$

$\wedge\ true$

$\boldsymbol{bin2dec}$ $\vdash bin2dec \in seq\ \mathbb{R} \to \mathbb{Z}$

$\wedge\ bin2dec\ \langle\rangle = 0$

$\wedge\ (\forall\ f : \mathbb{R};\ l : seq\ \mathbb{R}$

$\bullet\ bin2dec\ (\langle f\rangle \frown l)$

$= bin2dec\ l$

$+\ (if\ f = real\ 0\ then\ 0\ else\ 1) * 2\ **\ \#\ l)$

$\boldsymbol{\_\ Matrix\ \_}$ $\vdash (\_\ Matrix\ \_) \in (\mathbb{Z} \rel \mathbb{Z} \rel \mathbb{R}) \rel \mathbb{Z} \times \mathbb{Z}$

$\wedge\ (\forall\ s : \mathbb{Z} \rel \mathbb{Z} \rel \mathbb{R};\ m,\ n : \mathbb{Z}$

$\bullet\ s\ Matrix\ (m,\ n)$

$\Leftrightarrow\ s \in (seq\ \_)$

$\wedge\ \#\ s = m$

$\wedge\ (\forall\ ss : ran\ s \bullet ss \in (seq\ \_) \wedge \#\ ss = n))$

$\boldsymbol{dot\_product}$

$\vdash dot\_product \in (seq\ \mathbb{R}) \times (seq\ \mathbb{R}) \pfun \mathbb{R}$

$\wedge\ dom\ dot\_product$

$= \{In1?,\ In2? : seq\ \mathbb{R}$

$\mid \#\ In1? = \#\ In2?\}$

$\wedge\ dot\_product\ (\langle\rangle,\ \langle\rangle) = real\ 0$

$\wedge\ (\forall\ h1,\ h2 : \mathbb{R};\ t1,\ t2 : seq\ \mathbb{R}$

$\mid \#\ t1 = \#\ t2$

$\bullet\ dot\_product\ (\langle h1\rangle \frown t1,\ \langle h2\rangle \frown t2)$

$= h1 *_R h2 +_R dot\_product\ (t1,\ t2))$

$\boldsymbol{product}$ $\vdash product \in seq\ \mathbb{R} \to \mathbb{R}$

$\wedge\ product\ \langle\rangle = real\ 1$

$\wedge\ (\forall\ h : \mathbb{R};\ t : seq\ \mathbb{R}$

$\bullet\ product\ (\langle h\rangle \frown t) = h *_R product\ t)$

$\boldsymbol{sum}$ $\vdash sum \in seq\ \mathbb{R} \to \mathbb{R}$

$\wedge\ sum\ \langle\rangle = real\ 0$

$\wedge\ (\forall\ h : \mathbb{R};\ t : seq\ \mathbb{R}$

$\bullet\ sum\ (\langle h\rangle \frown t) = h +_R sum\ t)$

7.5 Theorems

$\boldsymbol{cz\_if\_thm}$ $\vdash \forall\ x,\ y : \mathbb{U}$

$\bullet\ if\ true\ then\ x\ else\ y = x$

$\wedge\ if\ false\ then\ x\ else\ y = y$

$\boldsymbol{cz\_r2z\_thm}$ $\vdash \forall\ i : \mathbb{U} \bullet r2z\ (real\ i) = i \wedge r2z\ (\sim_R real\ i) = \sim i$

$\boldsymbol{cz\_ub\_thm}$ $\vdash \forall\ r : \mathbb{U};\ sr : \mathbb{U} \bullet r\ ub_R\ sr \Leftrightarrow (\forall\ x : sr \bullet r \geq_R x)$

$\boldsymbol{cz\_lb\_thm}$ $\vdash \forall\ r : \mathbb{U};\ sr : \mathbb{U} \bullet r\ lb_R\ sr \Leftrightarrow (\forall\ x : sr \bullet r \leq_R x)$

$\boldsymbol{cz\_lub\_thm}$ $\vdash \forall\ lub : \mathbb{U};\ sr : \mathbb{U}$

$\bullet\ sr \mapsto lub \in lub_R$

$\Leftrightarrow\ lub\ ub_R\ sr$

$$\wedge\ (\forall\ ub : \mathbb{U} \mid ub\ ub_R\ sr \bullet ub \geq_R lub)$$

cz_glb_thm
$$\vdash \forall\ glb : \mathbb{U};\ sr : \mathbb{U}$$
$$\bullet\ sr \mapsto glb \in glb_R$$
$$\Leftrightarrow\ glb\ lb_R\ sr$$
$$\wedge\ (\forall\ lb : \mathbb{U} \mid lb\ lb_R\ sr \bullet lb \leq_R glb)$$

cz_lub_sup_thm
$$\vdash \ulcorner \forall\ s\ l \bullet \llbracket s \rrbracket \mapsto l \urcorner \in \llbracket lub_R \rrbracket \urcorner \Rightarrow l = Sup\ s \urcorner$$

cz_ub_lb_thm
$$\vdash \forall\ s : \mathbb{U};\ ub : \mathbb{U}$$
$$\bullet\ ub\ ub_R\ s \Leftrightarrow \sim_R\ ub\ lb_R\ \{x : s \mid true \bullet \sim_R\ x\}$$

cz_lb_ub_thm
$$\vdash \forall\ s : \mathbb{U};\ lb : \mathbb{U}$$
$$\bullet\ lb\ lb_R\ s \Leftrightarrow \sim_R\ lb\ ub_R\ \{x : s \mid true \bullet \sim_R\ x\}$$

cz_dom_lub_thm
$$\vdash \forall\ s : \mathbb{U}$$
$$\bullet\ (\exists\ lub : \mathbb{U} \bullet s \mapsto lub \in lub_R)$$
$$\Leftrightarrow\ \neg\ s = \{\} \wedge (\exists\ ub : \mathbb{U} \bullet ub\ ub_R\ s)$$

cz_dom_glb_thm
$$\vdash \forall\ s : \mathbb{U}$$
$$\bullet\ (\exists\ glb : \mathbb{U} \bullet s \mapsto glb \in glb_R)$$
$$\Leftrightarrow\ \neg\ s = \{\} \wedge (\exists\ lb : \mathbb{U} \bullet lb\ lb_R\ s)$$

cz_le_lb_trans_thm
$$\vdash \forall\ x,\ y : \mathbb{U};\ z : \mathbb{U} \bullet x \leq_R y \wedge y\ lb_R\ z \Rightarrow x\ lb_R\ z$$

cz_ge_ub_trans_thm
$$\vdash \forall\ x,\ y : \mathbb{U};\ z : \mathbb{U} \bullet y \leq_R x \wedge y\ ub_R\ z \Rightarrow x\ ub_R\ z$$

cz_lb_empty_thm
$$\vdash \forall\ lb : \mathbb{U} \bullet lb\ lb_R\ \{\}$$

cz_ub_empty_thm
$$\vdash \forall\ ub : \mathbb{U} \bullet ub\ ub_R\ \{\}$$

cz_lub_unit_thm
$$\vdash \forall\ x : \mathbb{U} \bullet lub_R\ \{x\} = x$$

cz_glb_unit_thm
$$\vdash \forall\ x : \mathbb{U} \bullet glb_R\ \{x\} = x$$

cz_lb_setd_thm
$$\vdash \forall\ lb,\ x : \mathbb{U};\ s : \mathbb{U}$$
$$\bullet\ lb\ lb_R\ \ulcorner Z'Setd\ (Cons\ x\ s) \urcorner$$
$$\Leftrightarrow\ lb \leq_R x \wedge lb\ lb_R\ \ulcorner Z'Setd\ s \urcorner$$

cz_ub_setd_thm
$$\vdash \forall\ x,\ y : \mathbb{U};\ t : \mathbb{U}$$
$$\bullet\ x\ ub_R\ \ulcorner Z'Setd\ (Cons\ y\ t) \urcorner$$
$$\Leftrightarrow\ y \leq_R x \wedge x\ ub_R\ \ulcorner Z'Setd\ t \urcorner$$

cz_glb_setd_thm
$$\vdash \forall\ glb,\ x : \mathbb{U};\ s : \mathbb{U}$$
$$\bullet\ glb = glb_R\ \ulcorner Z'Setd\ (Cons\ x\ s) \urcorner$$
$$\Leftrightarrow\ glb\ lb_R\ \ulcorner Z'Setd\ (Cons\ x\ s) \urcorner$$
$$\wedge\ (\forall\ lb : \mathbb{U}$$
$$\mid lb\ lb_R\ \ulcorner Z'Setd\ (Cons\ x\ s) \urcorner$$
$$\bullet\ lb \leq_R glb)$$

cz_lub_setd_thm

$$\vdash \forall\ lub,\ x : \mathbb{U};\ s : \mathbb{U}$$
$$\bullet\ lub = lub_R\ \ulcorner Z'Setd\ (Cons\ x\ s)\urcorner$$
$$\Leftrightarrow lub\ ub_R\ \ulcorner Z'Setd\ (Cons\ x\ s)\urcorner$$
$$\wedge\ (\forall\ ub : \mathbb{U}$$
$$\mid ub\ ub_R\ \ulcorner Z'Setd\ (Cons\ x\ s)\urcorner$$
$$\bullet\ ub \geq_R lub)$$

cz_eq_glb_setd_thm

$$\vdash \forall\ x,\ y,\ z : \mathbb{U};\ t : \mathbb{U}$$
$$\bullet\ x = glb_R\ \ulcorner Z'Setd\ (Cons\ y\ (Cons\ z\ t))\urcorner$$
$$\Leftrightarrow x = y \wedge x\ lb_R\ \ulcorner Z'Setd\ (Cons\ z\ t)\urcorner$$
$$\vee\ x \leq_R y \wedge x = glb_R\ \ulcorner Z'Setd\ (Cons\ z\ t)\urcorner$$

cz_glb_setd_eq_thm

$$\vdash \forall\ x,\ y,\ z : \mathbb{U};\ t : \mathbb{U}$$
$$\bullet\ glb_R\ \ulcorner Z'Setd\ (Cons\ y\ (Cons\ z\ t))\urcorner = x$$
$$\Leftrightarrow x = y \wedge x\ lb_R\ \ulcorner Z'Setd\ (Cons\ z\ t)\urcorner$$
$$\vee\ x \leq_R y \wedge x = glb_R\ \ulcorner Z'Setd\ (Cons\ z\ t)\urcorner$$

cz_eq_lub_setd_thm

$$\vdash \forall\ x,\ y,\ z : \mathbb{U};\ t : \mathbb{U}$$
$$\bullet\ x = lub_R\ \ulcorner Z'Setd\ (Cons\ y\ (Cons\ z\ t))\urcorner$$
$$\Leftrightarrow x = y \wedge x\ ub_R\ \ulcorner Z'Setd\ (Cons\ z\ t)\urcorner$$
$$\vee\ y \leq_R x \wedge x = lub_R\ \ulcorner Z'Setd\ (Cons\ z\ t)\urcorner$$

cz_lub_setd_eq_thm

$$\vdash \forall\ x,\ y,\ z : \mathbb{U};\ t : \mathbb{U}$$
$$\bullet\ lub_R\ \ulcorner Z'Setd\ (Cons\ y\ (Cons\ z\ t))\urcorner = x$$
$$\Leftrightarrow x = y \wedge x\ ub_R\ \ulcorner Z'Setd\ (Cons\ z\ t)\urcorner$$
$$\vee\ y \leq_R x \wedge x = lub_R\ \ulcorner Z'Setd\ (Cons\ z\ t)\urcorner$$

cz_le_glb_setd_thm

$$\vdash \forall\ x,\ z : \mathbb{U};\ t : \mathbb{U}$$
$$\bullet\ x \leq_R glb_R\ \ulcorner Z'Setd\ (Cons\ z\ t)\urcorner$$
$$\Leftrightarrow x\ lb_R\ \ulcorner Z'Setd\ (Cons\ z\ t)\urcorner$$

cz_lub_setd_le_thm

$$\vdash \forall\ x,\ z : \mathbb{U};\ t : \mathbb{U}$$
$$\bullet\ lub_R\ \ulcorner Z'Setd\ (Cons\ z\ t)\urcorner \leq_R x$$
$$\Leftrightarrow x\ ub_R\ \ulcorner Z'Setd\ (Cons\ z\ t)\urcorner$$

cz_glb_setd_le_thm

$$\vdash \forall\ x,\ y,\ z : \mathbb{U};\ t : \mathbb{U}$$
$$\bullet\ glb_R\ \ulcorner Z'Setd\ (Cons\ y\ (Cons\ z\ t))\urcorner \leq_R x$$
$$\Leftrightarrow y \leq_R x \vee glb_R\ \ulcorner Z'Setd\ (Cons\ z\ t)\urcorner \leq_R x$$

cz_le_lub_setd_thm

$$\vdash \forall\ x,\ y,\ z : \mathbb{U};\ t : \mathbb{U}$$
$$\bullet\ x \leq_R lub_R\ \ulcorner Z'Setd\ (Cons\ y\ (Cons\ z\ t))\urcorner$$
$$\Leftrightarrow x \leq_R y \vee x \leq_R lub_R\ \ulcorner Z'Setd\ (Cons\ z\ t)\urcorner$$

liftrel_thm

$$\vdash \forall\ r : \mathbb{U};\ x : \mathbb{U};\ y : \mathbb{U}$$
$$\bullet\ (liftrel_R\ r)\ (x,\ y)$$
$$= if\ (x,\ y) \in r\ then\ real\ 1\ else\ real\ 0$$

cz_relational_clauses

$$
\begin{array}{l}
\vdash \forall\ x,\ y : \mathbb{U} \\
\quad \bullet\ x\ eq_R\ y = Boolean_R\ (x = y) \\
\quad\ \wedge\ x\ noteq_R\ y = Boolean_R\ (\neg\ x = y) \\
\quad\ \wedge\ x\ less_R\ y = Boolean_R\ (x <_R y) \\
\quad\ \wedge\ x\ less\_eq_R\ y = Boolean_R\ (x \leq_R y) \\
\quad\ \wedge\ x\ greater_R\ y = Boolean_R\ (x >_R y) \\
\quad\ \wedge\ x\ greater\_eq_R\ y = Boolean_R\ (x \geq_R y)
\end{array}
$$

cz_boolean_clauses

$$
\begin{array}{l}
\vdash\ true_R = Boolean_R\ true \\
\quad \wedge\ false_R = Boolean_R\ false \\
\quad \wedge\ (\forall\ p : \mathbb{U} \bullet not_R\ Boolean_R\ p = Boolean_R\ (\neg\ p)) \\
\quad \wedge\ (\forall\ p,\ q : \mathbb{U} \\
\quad\quad \bullet\ Boolean_R\ p\ and_R\ Boolean_R\ q \\
\quad\quad\ = Boolean_R\ (p \wedge q)) \\
\quad \wedge\ (\forall\ p,\ q : \mathbb{U} \\
\quad\quad \bullet\ Boolean_R\ p\ or_R\ Boolean_R\ q \\
\quad\quad\ = Boolean_R\ (p \vee q)) \\
\quad \wedge\ (\forall\ p,\ q : \mathbb{U} \\
\quad\quad \bullet\ Boolean_R\ p\ equiv_R\ Boolean_R\ q \\
\quad\quad\ = Boolean_R\ (p \Leftrightarrow q)) \\
\quad \wedge\ (\forall\ p,\ q : \mathbb{U} \\
\quad\quad \bullet\ Boolean_R\ p\ xor_R\ Boolean_R\ q \\
\quad\quad\ = Boolean_R\ (\neg\ (p \Leftrightarrow q))) \\
\quad \wedge\ (\forall\ p,\ q : \mathbb{U} \bullet Boolean_R\ p = Boolean_R\ q \Leftrightarrow p \Leftrightarrow q)
\end{array}
$$

***cz_boolean_clauses*1**

$$
\begin{array}{l}
\vdash\ (\forall\ p : \mathbb{U} \bullet not_R\ p = Boolean_R\ (p = false_R)) \\
\quad \wedge\ (\forall\ p,\ q : \mathbb{U} \\
\quad\quad \bullet\ p\ and_R\ q \\
\quad\quad\ = Boolean_R\ (\neg\ p = false_R \wedge \neg\ q = false_R)) \\
\quad \wedge\ (\forall\ p,\ q : \mathbb{U} \\
\quad\quad \bullet\ p\ or_R\ q \\
\quad\quad\ = Boolean_R\ (\neg\ p = false_R \vee \neg\ q = false_R)) \\
\quad \wedge\ (\forall\ p,\ q : \mathbb{U} \\
\quad\quad \bullet\ p\ equiv_R\ q \\
\quad\quad\ = Boolean_R\ (p = false_R \Leftrightarrow q = false_R)) \\
\quad \wedge\ (\forall\ p,\ q : \mathbb{U} \\
\quad\quad \bullet\ p\ xor_R\ q \\
\quad\quad\ = Boolean_R\ (p = false_R \Leftrightarrow \neg\ q = false_R))
\end{array}
$$

cz_boolean_real_clauses

$$
\begin{array}{l}
\vdash\ not_R\ real\ 0 = Boolean_R\ true \\
\quad \wedge\ not_R\ real\ 1 = Boolean_R\ false \\
\quad \wedge\ (\forall\ p : \mathbb{U} \\
\quad\quad \bullet\ real\ 1\ and_R\ p = Boolean_R\ true\ and_R\ p) \\
\quad \wedge\ (\forall\ p : \mathbb{U} \\
\quad\quad \bullet\ p\ and_R\ real\ 1 = p\ and_R\ Boolean_R\ true) \\
\quad \wedge\ (\forall\ p : \mathbb{U} \bullet real\ 0\ and_R\ p = Boolean_R\ false) \\
\quad \wedge\ (\forall\ p : \mathbb{U} \bullet p\ and_R\ real\ 0 = Boolean_R\ false) \\
\quad \wedge\ (\forall\ p : \mathbb{U} \bullet real\ 1\ or_R\ p = Boolean_R\ true)
\end{array}
$$

$$
\begin{array}{l}
\quad \land\ (\forall\ p : \mathbb{U} \bullet p\ or_R\ real\ 1 = Boolean_R\ true) \\
\quad \land\ (\forall\ p : \mathbb{U} \\
\qquad \bullet\ real\ 0\ or_R\ p = Boolean_R\ false\ or_R\ p) \\
\quad \land\ (\forall\ p : \mathbb{U} \\
\qquad \bullet\ p\ or_R\ real\ 0 = p\ or_R\ Boolean_R\ false) \\
\quad \land\ (\forall\ p : \mathbb{U} \\
\qquad \bullet\ real\ 1\ equiv_R\ p = Boolean_R\ true\ equiv_R\ p) \\
\quad \land\ (\forall\ p : \mathbb{U} \\
\qquad \bullet\ p\ equiv_R\ real\ 1 = p\ equiv_R\ Boolean_R\ true) \\
\quad \land\ (\forall\ p : \mathbb{U} \bullet real\ 0\ equiv_R\ p = not_R\ p) \\
\quad \land\ (\forall\ p : \mathbb{U} \bullet p\ equiv_R\ real\ 0 = not_R\ p) \\
\quad \land\ (\forall\ p : \mathbb{U} \bullet real\ 1\ xor_R\ p = not_R\ p) \\
\quad \land\ (\forall\ p : \mathbb{U} \bullet p\ xor_R\ real\ 1 = not_R\ p) \\
\quad \land\ (\forall\ p : \mathbb{U} \\
\qquad \bullet\ real\ 0\ xor_R\ p = Boolean_R\ false\ xor_R\ p) \\
\quad \land\ (\forall\ p : \mathbb{U} \\
\qquad \bullet\ p\ xor_R\ real\ 0 = p\ xor_R\ Boolean_R\ false)
\end{array}
$$

cz_boolean_clauses2

$$
\begin{array}{l}
\vdash\ (\forall\ b : \mathbb{U};\ r : \mathbb{U} \\
\qquad \bullet\ r\ and_R\ Boolean_R\ b \\
\qquad\quad = Boolean_R\ (\neg\ r = false_R \land b)) \\
\quad \land\ (\forall\ b : \mathbb{U};\ r : \mathbb{U} \\
\qquad \bullet\ Boolean_R\ b\ and_R\ r \\
\qquad\quad = Boolean_R\ (b \land \neg\ r = false_R)) \\
\quad \land\ (\forall\ b : \mathbb{U};\ r : \mathbb{U} \\
\qquad \bullet\ r\ or_R\ Boolean_R\ b \\
\qquad\quad = Boolean_R\ (\neg\ r = false_R \lor b)) \\
\quad \land\ (\forall\ b : \mathbb{U};\ r : \mathbb{U} \\
\qquad \bullet\ Boolean_R\ b\ or_R\ r \\
\qquad\quad = Boolean_R\ (b \lor \neg\ r = false_R)) \\
\quad \land\ (\forall\ b : \mathbb{U};\ r : \mathbb{U} \\
\qquad \bullet\ r\ equiv_R\ Boolean_R\ b \\
\qquad\quad = Boolean_R\ (\neg\ r = false_R \Leftrightarrow b)) \\
\quad \land\ (\forall\ b : \mathbb{U};\ r : \mathbb{U} \\
\qquad \bullet\ Boolean_R\ b\ equiv_R\ r \\
\qquad\quad = Boolean_R\ (b \Leftrightarrow \neg\ r = false_R)) \\
\quad \land\ (\forall\ b : \mathbb{U};\ r : \mathbb{U} \\
\qquad \bullet\ r\ xor_R\ Boolean_R\ b \\
\qquad\quad = Boolean_R\ (r = false_R \Leftrightarrow b)) \\
\quad \land\ (\forall\ b : \mathbb{U};\ r : \mathbb{U} \\
\qquad \bullet\ Boolean_R\ b\ xor_R\ r \\
\qquad\quad = Boolean_R\ (b \Leftrightarrow r = false_R))
\end{array}
$$

cz_matlab_clauses

$$
\begin{array}{l}
\vdash\ (\_\ .+_m\ \_) = (\_\ +_R\ \_) \\
\quad \land\ (\_\ .-_m\ \_) = (\_\ -_R\ \_) \\
\quad \land\ (\_\ .*_m\ \_) = (\_\ *_R\ \_) \\
\quad \land\ (\_\ ./_m\ \_) = (\_\ /_R\ \_) \\
\quad \land\ (\forall\ x,\ y : \mathbb{U} \bullet x\ .\backslash_m\ y = y\ /_R\ x)
\end{array}
$$

$$
\begin{array}{l}
\quad \wedge\ (\_ <_m \_) = (\_\ less_R\ \_) \\
\quad \wedge\ (\_ >_m \_) = (\_\ greater_R\ \_) \\
\quad \wedge\ (\_ >=_m \_) = (\_\ greater\_eq_R\ \_) \\
\quad \wedge\ (\_ <=_m \_) = (\_\ less\_eq_R\ \_) \\
\quad \wedge\ (\_ ==_m \_) = (\_\ eq_R\ \_) \\
\quad \wedge\ (\_ \sim=_m \_) = (\_\ noteq_R\ \_) \\
\quad \wedge\ (\_\ and_m\ \_) = (\_\ and_R\ \_) \\
\quad \wedge\ (\_\ or_m\ \_) = (\_\ or_R\ \_) \\
\quad \wedge\ (\forall\ r : \mathbb{U};\ z : \mathbb{U} \bullet r\ .\hat{}_m\ real\ z = r\ \hat{}_Z\ z) \\
\quad \wedge\ (\forall\ x : \mathbb{U} \bullet mp_m\ x = x) \\
\quad \wedge\ (\forall\ x : \mathbb{U} \bullet mm_m\ x = real\ 0\ -_R\ x) \\
\quad \wedge\ (\sim_m \_) = (not_R\ \_)
\end{array}
$$

cz_fcn_clauses

$$
\begin{array}{l}
\vdash\ (\_ +_f \_) = (\_ +_R \_) \\
\quad \wedge\ (\_ -_f \_) = (\_ -_R \_) \\
\quad \wedge\ (\_ *_f \_) = (\_ *_R \_) \\
\quad \wedge\ (\_ /_f \_) = (\_ /_R \_) \\
\quad \wedge\ (\_ <_f \_) = (\_\ less_R\ \_) \\
\quad \wedge\ (\_ >_f \_) = (\_\ greater_R\ \_) \\
\quad \wedge\ (\_ >=_f \_) = (\_\ greater\_eq_R\ \_) \\
\quad \wedge\ (\_ <=_f \_) = (\_\ less\_eq_R\ \_) \\
\quad \wedge\ (\_ =_f \_) = (\_\ eq_R\ \_) \\
\quad \wedge\ (\_ \neq_f \_) = (\_\ noteq_R\ \_) \\
\quad \wedge\ (\_\ and_f\ \_) = (\_\ and_R\ \_) \\
\quad \wedge\ (\_\ or_f\ \_) = (\_\ or_R\ \_) \\
\quad \wedge\ (\forall\ r : \mathbb{U};\ z : \mathbb{U} \bullet r\ \hat{}_f\ real\ z = r\ \hat{}_Z\ z) \\
\quad \wedge\ (\forall\ x : \mathbb{U} \bullet mp_f\ x = x) \\
\quad \wedge\ (\forall\ x : \mathbb{U} \bullet mm_f\ x = real\ 0\ -_R\ x) \\
\quad \wedge\ (not_f\ \_) = (not_R\ \_)
\end{array}
$$

cz_bin2dec_thm

$$
\begin{array}{l}
\vdash\ bin2dec\ \langle\rangle = 0 \\
\quad \wedge\ (\forall\ f : \mathbb{U};\ l : \mathbb{U} \\
\quad\ \bullet\ bin2dec\ \ulcorner\$"Z'\langle\rangle"\ (Cons\ f\ l)\urcorner \\
\quad\quad = bin2dec\ \ulcorner\$"Z'\langle\rangle"\ l\urcorner \\
\quad\quad\quad + (if\ f = real\ 0\ then\ 0\ else\ 1) \\
\quad\quad\quad\quad * 2\ **\ \#\ \ulcorner\$"Z'\langle\rangle"\ l\urcorner)
\end{array}
$$

cz_dot_product_empty_thm

$$
\vdash\ dot\_product\ (\langle\rangle, \langle\rangle) = real\ 0
$$

cz_dot_product_thm

$$
\begin{array}{l}
\vdash\ \forall\ h1,\ h2 : \mathbb{U};\ t1,\ t2 : \mathbb{U} \\
\quad \bullet\ \ulcorner Length\ t1\urcorner = \ulcorner Length\ t2\urcorner \\
\quad\ \Rightarrow dot\_product \\
\quad\quad\quad (\ulcorner\$"Z'\langle\rangle"\ (Cons\ h1\ t1)\urcorner, \\
\quad\quad\quad\ \ulcorner\$"Z'\langle\rangle"\ (Cons\ h2\ t2)\urcorner) \\
\quad\quad = h1\ *_R\ h2 \\
\quad\quad\quad +_R\ dot\_product \\
\quad\quad\quad\quad (\ulcorner\$"Z'\langle\rangle"\ t1\urcorner, \ulcorner\$"Z'\langle\rangle"\ t2\urcorner)
\end{array}
$$

cz_product_thm

$$\vdash product\ \langle\rangle = real\ 1 \\ \land (\forall\ h : \mathbb{U};\ t : \mathbb{U} \\ \bullet\ product\ \ulcorner\$"\ Z'\langle\rangle"\ (Cons\ h\ t)\urcorner \\ = h *_R product\ \ulcorner\$"\ Z'\langle\rangle"\ t\urcorner)$$

cz_sum_thm $\vdash sum\ \langle\rangle = real\ 0 \\ \land (\forall\ h : \mathbb{U};\ t : \mathbb{U} \\ \bullet\ sum\ \ulcorner\$"\ Z'\langle\rangle"\ (Cons\ h\ t)\urcorner \\ = h +_R sum\ \ulcorner\$"\ Z'\langle\rangle"\ t\urcorner)$

8 INDEX