Appendix A

The Basic Nuprl Type Theory

A.1 Syntax

Nuprl terms have the form $\texttt{opid}\{p_1{:}F_1,\ldots,p_k{:}F_k\}(x^1_1,\ldots,x^1_{m_1}.t_1;\ \ldots;\ x^n_1,\ldots,x^n_{m_n}.t_n)$. We name the parts of a term as follows:

- $\texttt{opid}\{p_1{:}F_1,\ldots,p_k{:}F_k\}$ is the *operator*.

  The parts of the operator are:

  – *opid* is the *operator identifier*.

  – $p_j{:}F_j$ is the j-th *parameter*. p_j is the *parameter value* and F_j is the *parameter type*.

- The tuple $(m_1, \ldots, m_n)$, where $m_j{\geq}0$ is the *arity* of the term.
- $s_i = x^i_1,\ldots,x^i_{m_i}.t_i$ is the i-th *bound-term* of the term, which binds free occurrences of the *variables* $x^i_1,\ldots,x^i_{m_i}$ in t_i.

When writing terms, we sometimes omit the {} brackets around the parameter list if it is empty. Note that parameters are separated by commas while subterms are separated by semicolons.

A.1.1 Operator Identifiers

Operator identifiers are character strings drawn from the alphabet[1] `a`−`z` `A`−`Z` `0`−`9` `_` `-` `!`. An `!` at the start of a character string indicates that the term does not belong to Nuprl's object language. Operator identifiers are implemented using ML type `tok`. Valid operators are listed in Nuprl's *operator table*, which contains the basic operators[2] given in Table A.1 as well as conservative language extensions defined by abstractions in the library.

[1] We distinguish between the ASCII character `-` and the character range $x{-}y$, indicating the characters from x to y.

[2] Note that the operator identifier of a term is not always identical to the name a user has to type into Nuprl's term editor in order to generate the corresponding display template. The latter depends only on the information provided in the display form while the abstract name can only be used to enter terms in the abstract (expanded) mode. Different names are, for instance, used for the simple inductive types (simplerec instead of rec) and the less_than predicate (lt instead of `less_than`). A complete list of display forms for the basic terms can be found in the system's `core_1` theory.

Canonical (Types)	(Members)	**noncanonical**
	var{x:v}() x	
function{}(S; x.T) $x{:}S{\to}T$, $S{\to}T$	**lambda**{}(x.t) $\lambda x.t$	**apply**{}($\boxed{f}$;t) $\boxed{f}\ t$
product{}(S; x.T) $x{:}S{\times}T$, $S{\times}T$	**pair**{}(s;t) $\langle s,t\rangle$	**spread**{}($\boxed{e}$; $x,y.u$) let $\langle x,y\rangle = \boxed{e}$ in u
union{}(S;T) $S{+}T$	**inl**{}(s), **inr**{}(t) inl(s), inr(t)	**decide**{}($\boxed{e}$; $x.u$; $y.v$) case $\boxed{e}$ of inl(x) $\mapsto u$ $\mid$ inr(y) $\mapsto v$
universe{j:l}() $\mathbb{U}_j$	– *All types of level j* –	
equal{}(s;t;T) $s = t \in T$	**Axiom**{}() Ax	
void{}() void	– *No canonical elements* –	**any**{}(e) any(e)
Atom{}() Atom	**token**{*token*:t}() ¨*token*¨	**atom_eq**($\boxed{u}$; $\boxed{v}$; s; t) if $\boxed{u}{=}\boxed{v}$ then s else t
int{}() $\mathbb{Z}$	**natural_number**{n:n}() n	**ind**{}($\boxed{u}$,$x.f_x$; s; base,$y.f_y$)t ind($\boxed{u}$,$x.f_x$; s; base,$y.f_y$)t
	minus{}(**natural_number**{n:n}()) $-n$	**minus**{}($\boxed{u}$), **add**{}($\boxed{u}$;$\boxed{v}$), **sub**{}($\boxed{u}$;$\boxed{v}$) $-\boxed{u}$, $\boxed{u}{+}\boxed{v}$, $\boxed{u}{-}\boxed{v}$
		mul{}($\boxed{u}$;$\boxed{v}$), **div**{}($\boxed{u}$;$\boxed{v}$), **rem**{}($\boxed{u}$;$\boxed{v}$) $\boxed{u}{*}\boxed{v}$, $\boxed{u}{\div}\boxed{v}$, $\boxed{u}$ rem $\boxed{v}$
		int_eq($\boxed{u}$; $\boxed{v}$; s; t), **less**($\boxed{u}$; $\boxed{v}$; s; t) if $\boxed{u}{=}\boxed{v}$ then s else t, if $\boxed{u}{<}\boxed{v}$ then s else t
less_than{}(u;v) $u{<}v$	**Axiom**{}() Ax	
list{}(T) T list	**nil**{}(), **cons**{}(t;l) [], $t::l$	**list_ind**{}($\boxed{s}$; base; $x,l,f_{xl}.t$) list_ind($\boxed{s}$; base; $x,l,f_{xl}.t$)
rec{}($X.T_X$) rectype $X = T_X$	– *members defined by unrolling* T_X –	**rec_ind**{}($\boxed{e}$; $f,x.t$) let* $f(x) = t$ in $f(\boxed{e})$
set{}(S; $x.T$) $\{x{:}S \mid T\}$, $\{S \mid T\}$	– *members of* S *that satisfy* P –	
isect{}(S; $x.T$) $\cap x{:}S.T$	– *Terms that belong to all* $T[x]$ –	
quotient{}(T; $x,y.E$) $x,y : T//E$	– *members of* T, *new equality* –	

Table A.1: Basic operators of Nuprl's Type Theory. Terms are divided into canonical and noncanonical terms. Principal arguments in noncanonical terms are marked by a $\boxed{\text{box}}$. Standard display forms of terms are written below the abstract representation. The distinction between types and members is *not* a part of Nuprl's syntax.

A.1.2 Parameters

Parameters $p\!:\!F$ consist of a parameter name p and a parameter family F. The current parameter families and associated values are:

`variable` : Names of variables, implemented using the ML data type `var`.
Acceptable names are generated by the regular expression $[\texttt{a}-\texttt{z}\ \texttt{A}-\texttt{Z}\ \texttt{0}-\texttt{9}\ \_\ \texttt{-}\ \%]^+$. The `%` character has a special use.

`natural` : Natural numbers (including 0), implemented using the ML data type `int`.
Acceptable numbers are generated by the regular expression $\texttt{0} + [\texttt{1} - \texttt{9}][\texttt{0} - \texttt{9}]^*$.

`token` : Character strings, implemented using the ML data type `tok`.
Acceptable strings can draw from any non-control characters in Nuprl's font.

`string` : Character strings, implemented using the ML data type `string`.
Acceptable strings can draw from any non-control characters in Nuprl's font.

`level-expression` : Universe level expressions, implemented using the ML data type `level_exp`.
Universe level expressions are used to index universe levels in Nuprl's type theory. Their syntax is described by the grammar $L ::= v \mid k \mid L\ i \mid L' \mid [L|\cdots|L]$ where v denotes a level-expression variable (alphanumeric string), k a level expression constant (positive integer), and i a level expression increment (non-negative integer).

Level-expression variables are implicitly quantified over all positive integer levels. The expression $L\ i$ is interpreted as standing for levels $L + i$. L' is an abbreviation for L `1`. The expression $[L_1|\cdots|L_n]$ is interpreted as being the maximum of expressions $L_1 \cdots L_n$.

The names of parameter types are usually abbreviated to their first letters.

A.1.3 Binding Variables

Binding variables are character strings drawn from the same alphabet as variable parameters. To express terms without bindings, the empty string can be used as *null variable*. Null variables never bind. Binding variables are implemented using ML type `var`.

A.1.4 Injection of Variables and Numbers

In Nuprl, we consider variables and terms to be distinct. We have a special term kind, `variable{v}` for injecting variables into the term type. When we talk of the variable x *as a term*, we really mean the term `variable{`x`:v}`. In a similar way, when we talk of the number n *as a term*, we really mean the term `natural_number{`n`:n}`.

The injection is often made implicitly when it is clear from the context. Nuprl's editor atomatically converts variables and numbers into terms when they are typed into templates for terms.

A.1.5 Term Display

The *display* of NUPRL terms is not necessarily identical to their abstract form. Usually, they are presented in a more conventional notation, which is created by the display forms described in Chapter 7.2. In Table A.1 we present the standard display of NUPRL terms immediately below their abstract form.

Redex		Contractum
$(\boxed{\lambda x.u})\ t$	$\xrightarrow{\beta}$	$u[t/x]$
$\mathsf{let}\ \langle x,y\rangle = \boxed{\langle s,t\rangle}\ \mathsf{in}\ u$	$\xrightarrow{\beta}$	$u[s,t/x,y]$
$\mathsf{case}\ \boxed{\mathsf{inl}(s)}\ \mathsf{of}\ \mathsf{inl}(x)\mapsto u\ \mid\ \mathsf{inr}(y)\mapsto v$	$\xrightarrow{\beta}$	$u[s/x]$
$\mathsf{case}\ \boxed{\mathsf{inr}(t)}\ \mathsf{of}\ \mathsf{inl}(x)\mapsto u\ \mid\ \mathsf{inr}(y)\mapsto v$	$\xrightarrow{\beta}$	$v[t/y]$
$\mathsf{if}\ \boxed{a}\mathsf{=}\boxed{b}\ \mathsf{then}\ s\ \mathsf{else}\ t$	$\xrightarrow{\beta}$	*s, if* $a=b$*;* *t, otherwise*
$\mathsf{ind}(\boxed{0},x.f_x;\ s;\ base,y.f_y)t$	$\xrightarrow{\beta}$	$base$
$\mathsf{ind}(\boxed{i},x.f_x;\ s;\ base,y.f_y)t$	$\xrightarrow{\beta}$	$t[i,\mathsf{ind}(\boxed{i-1},x.f_x;\ s;\ base,y.f_y)t\ /y,f_y]$, $(i>0)$
$\mathsf{ind}(\boxed{-i},x.f_x;\ s;\ base,y.f_y)t$	$\xrightarrow{\beta}$	$s[-i,\mathsf{ind}(\boxed{-i+1},x.f_x;\ s;\ base,y.f_y)t\ /x,f_x]$, $(i>0)$
$\mathsf{-}\boxed{i}$	$\xrightarrow{\beta}$	*The negation of* i *(as number)*
$\boxed{i}\mathsf{+}\boxed{j}$	$\xrightarrow{\beta}$	*The sum of* i *and* j
$\boxed{i}\mathsf{-}\boxed{j}$	$\xrightarrow{\beta}$	*The difference of* i *and* j
$\boxed{i}\mathsf{*}\boxed{j}$	$\xrightarrow{\beta}$	*The product of* i *and* j
$\boxed{i}\div\boxed{j}$	$\xrightarrow{\beta}$	*0, if* j*=0;* *the integer division of* i *and* j*, otherwise*
$\boxed{i}\ \mathsf{rem}\ \boxed{j}$	$\xrightarrow{\beta}$	*0, if* j*=0;* *the division rest of* i *and* j*, otherwise*
$\mathsf{if}\ \boxed{i}\mathsf{=}\boxed{j}\ \mathsf{then}\ s\ \mathsf{else}\ t$	$\xrightarrow{\beta}$	*s, if* $i=j$*;* *t, otherwise*
$\mathsf{if}\ \boxed{i}\mathsf{<}\boxed{j}\ \mathsf{then}\ s\ \mathsf{else}\ t$	$\xrightarrow{\beta}$	*s, if* $i<j$*;* *t, otherwise*
$\mathsf{list\_ind}(\boxed{[\,]};\ base;\ x,l,f_{xl}.t)$	$\xrightarrow{\beta}$	$base$
$\mathsf{list\_ind}(\boxed{s::u};\ base;\ x,l,f_{xl}.t)$	$\xrightarrow{\beta}$	$t[s,u,\mathsf{list\_ind}(u;\ base;\ x,l,f_{xl}.t)/x,l,f_{xl}]$
$\mathsf{let}^*\ f(x)=t\ \mathsf{in}\ f(\boxed{e})$	$\xrightarrow{\beta}$	$t[\lambda y.\mathsf{let}^*\ f(x)=t\ \mathsf{in}\ f(y),e\ /\ f,x]$

Table A.2: Redex–Contracta Table for Nuprl's Type Theory: the $\boxed{\text{principal arguments}}$ must be in the corresponding canonical form

A.2 Semantics

A.2.1 Evaluation

Nuprl's semantics is based on a notion of values. Terms are divided into *canonical* forms, i.e. values, and *noncanonical* forms, i.e. terms that need to be evaluated. Evaluation in Nuprl is *lazy*: whether a term is canonical or not depends solely on its operator identifier but not on its subterms. In noncanonical forms, certain subterms are marked as *principal arguments*. If a principal argument is instantiated with a matching canonical form, the expression becomes *reducible* (i.e. a *redex*) and can be evaluated to its *contractum*, defined in a *redex–contracta table*.

Nuprl's *evaluation mechanism* first computes the values of all principal arguments of a non-canonical expression. If an argument does not have a value or if the resulting expression is not reducible, evaluation stops: the expression has no value. Otherwise the expression will be reduced according to redex–contracta table and the resulting term will be evaluated.

Canonical forms and noncanonical forms together with their principal arguments are given in Table A.1. The corresponding redex–contracta table is given in Table A.2.

$x_1{:}S_1{\to}T_1 = x_2{:}S_2{\to}T_2$ $T = S_2{\to}T_2$ $S_1{\to}T_1 = T$	if if if	$S_1{=}S_2$ and $T_1[s_1/x_1]{=}T_2[s_2/x_2]$ for all s_1, s_2 with $s_1{=}s_2 \in S_1$. $T = x_2{:}S_2{\to}T_2$ for some $x_2 \in \mathcal{V}$. $x_1{:}S_1{\to}T_1 = T$ for some $x_1 \in \mathcal{V}$.
$x_1{:}S_1{\times}T_1 = x_2{:}S_2{\times}T_2$ $T = S_2{\times}T_2$ $S_1{\times}T_1 = T$	if if if	$S_1{=}S_2$ and $T_1[s_1/x_1]{=}T_2[s_2/x_2]$ for all s_1, s_2 with $s_1{=}s_2 \in S_1$. $T = x_2{:}S_2{\times}T_2$ for some variable x_2. $x_1{:}S_1{\times}T_1 = T$ for some variable x_1.
$S_1{+}T_1 = S_2{+}T_2$	if	$S_1{=}S_2$ and $T_1{=}T_2$.
$\mathbb{U}_{j_1} = \mathbb{U}_{j_2}$	if	$j_1{=}j_2$ *(as natural number)*
$s_1{=}t_1 \in T_1 = s_2{=}t_2 \in T_2$	if	$T_1{=}T_2$, $s_1{=}s_2 \in T_1$, and $t_1{=}t_2 \in T_1$.
void = void Atom = Atom		
$\mathbb{Z} = \mathbb{Z}$ $i_1{<}j_1 = i_2{<}j_2$	 if	 $i_1 = i_2 \in \mathbb{Z}$ and $j_1 = j_2 \in \mathbb{Z}$
T_1 list $= T_2$ list	if	$T_1 = T_2$
rectype $X_1{=}T_1$ = rectype $X_2{=}T_2$	if	$T_1[X/X_1] = T_2[X/X_2]$ for all types X
$\{x_1{:}S_1 \mid T_1\} = \{x_2{:}S_2 \mid T_2\}$ $T = \{S_2 \mid T_2\}$ $\{S_1 \mid T_1\} = T$	if if if	$S_1{=}S_2$ and there are terms p_1, p_2 and a variable x, which occurs neither in T_1 nor in T_2, such that $p_1 \in \forall x{:}S_1.\ T_1[x/x_1] \Rightarrow T_2[x/x_2]$ and $p_2 \in \forall x{:}S_1.\ T_2[x/x_2] \Rightarrow T_1[x/x_1]$. $T = \{x_2{:}S_2 \mid T_2\}$ for some variable x_2. $\{x_1{:}S_1 \mid T_1\} = T$ for some variable x_1.
$\cap x_1{:}S_1.T_1 = \cap x_2{:}S_2.T_2$	if	$S_1{=}S_2$ and $T_1[s_1/x_1]{=}T_2[s_2/x_2]$ for all s_1, s_2 with $s_1{=}s_2 \in S_1$.
$x_1,y_1 : T_1//E_1 = x_2,y_2 : T_2//E_2$	if	$T_1 = T_2$ and there are terms p_1,p_2,r,s,t and variables x, y, z, which occur neither in E_1 nor in E_2, such that $p_1 \in \forall x{:}T_1.\forall y{:}T_1.\ E_1[x,y/x_1,y_1] \Rightarrow E_2[x,y/x_2,y_2]$, $p_2 \in \forall x{:}T_1.\forall y{:}T_1.\ E_2[x,y/x_2,y_2] \Rightarrow E_1[x,y/x_1,y_1]$, $r \in \forall x{:}T_1.\ E_1[x,x/x_1,y_1]$, $s \in \forall x{:}T_1.\forall y{:}T_1.\ E_1[x,y/x_1,y_1] \Rightarrow E_1[y,x/x_1,y_1]$, and $t \in \forall x{:}T_1.\forall y{:}T_1.\forall z{:}T_1.$ $E_1[x,y/x_1,y_1] \Rightarrow E_1[y,z/x_1,y_1] \Rightarrow E_1[x,z/x_1,y_1]$

Table A.3: Type semantics table for Nuprl

A.2.2 Judgments

The meaning of type theoretical expressions is given in the form of *judgments* about essential properties of the terms. Judgments are assertions of certain truths that form the foundation of type theory. We distinguish 4 types of judgments: *Typehood* (T Type), *Type Equality* ($S{=}T$), *Membership* ($t \in T$), and *Member Equality* ($s{=}t \in T$). The precise meaning of these judgments is defined as follows.

T Type if $T{=}T$.

$S{=}T$ if there are canonical terms S' and T' such that $S \xrightarrow{*} S'$, $T \xrightarrow{*} T'$ and $S'{=}T'$ follows from the *type semantics table*.

$t \in T$ if $t{=}t \in T$.

$s{=}t \in T$ if there are canonical terms s', t' and T' such that $s \xrightarrow{*} s'$, $t \xrightarrow{*} t'$, $T{=}T'$, and $s'{=}t' \in T'$ follows from the *member semantics table*.

Nuprl's type semantics table is given in Table A.3 and its member semantics table in Table A.4.

$\lambda x_1.t_1 = \lambda x_2.t_2 \in x{:}S\to T$	if $x{:}S\to T$ Type and $t_1[s_1/x_1] = t_2[s_2/x_2] \in T[s_1/x]$ for all s_1,s_2 with $s_1{=}s_2{\in}S$.
$\langle s_1,t_1\rangle = \langle s_2,t_2\rangle \in x{:}S\times T$	if $x{:}S\times T$ Type, $s_1{=}s_2{\in}S$, and $t_1{=}t_2{\in}T[s_1/x]$.
$\mathsf{inl}(s_1) = \mathsf{inl}(s_2) \in S{+}T$	if $S{+}T$ Type and $s_1{=}s_2{\in}S$.
$\mathsf{inr}(t_1) = \mathsf{inr}(t_2) \in S{+}T$	if $S{+}T$ Type and $t_1{=}t_2{\in}T$
$\mathsf{Ax} = \mathsf{Ax} \in s{=}t\in T$	if $s = t \in T$
$s = t \in \mathsf{void}$	***never holds !***
$\ddot{}token\ddot{} = \ddot{}token\ddot{} \in \mathsf{Atom}$	
$i = i \in \mathbb{Z}$	
$\mathsf{Ax} = \mathsf{Ax} \in s{<}t$	if $s \xrightarrow{*} i$ and $t \xrightarrow{*} j$ for some integers i, j with $i<j$
$[] = [] \in T\,\mathsf{list}$	if T Type
$t_1{:}{:}l_1 = t_2{:}{:}l_2 \in T\,\mathsf{list}$	if T Type, $t_1{=}t_2{\in}T$, and $l_1{=}l_2{\in}T\,\mathsf{list}$
$s = t \in \mathsf{rectype}\ X = T_X$	if $\mathsf{rectype}\ X = T_X$ Type and $s = t \in T_X[\mathsf{rectype}\ X = T_X/X]$
$s = t \in \{x{:}S \mid T\}$	if $\{x{:}S \mid T\}$ Type, $s{=}t{\in}S$, and there is some term $p{\in}T[s/x]$
$t_1 = t_2 \in \cap x{:}S.T$	if $\cap x{:}S.T$ Type and $t_1{=}t_2{\in}T[s/x]$ for all $s{\in}S$.
$s = t \in x,y : T//E$	if $x,y : T//E$ Type, $s{\in}T$, $t{\in}T$, and there is some term $p{\in}\ E[s,t/x,y]$
$x_1{:}S_1{\to}T_1 = x_2{:}S_2{\to}T_2 \in \mathbb{U}_j$	if $S_1{=}S_2{\in}\mathbb{U}_j$ and $T_1[s_1/x_1]{=}T_2[s_2/x_2]{\in}\mathbb{U}_j$ for all s_1, s_2 with $s_1{=}s_2{\in}S_1$
$T = S_2{\to}T_2 \in \mathbb{U}_j$	if $T = x_2{:}S_2{\to}T_2 \in \mathbb{U}_j$ for some variable x_2
$S_1{\to}T_1 = T \in \mathbb{U}_j$	if $x_1{:}S_1{\to}T_1 = T \in \mathbb{U}_j$ for some variable x_1
$x_1{:}S_1{\times}T_1 = x_2{:}S_2{\times}T_2 \in \mathbb{U}_j$	if $S_1{=}S_2{\in}\mathbb{U}_j$ and $T_1[s_1/x_1]{=}T_2[s_2/x_2]{\in}\mathbb{U}_j$ for all s_1,s_2 with $s_1{=}s_2{\in}S_1$
$T = S_2{\times}T_2 \in \mathbb{U}_j$	if $T = x_2{:}S_2{\times}T_2 \in \mathbb{U}_j$ for some variable x_2
$S_1{\times}T_1 = T \in \mathbb{U}_j$	if $x_1{:}S_1{\times}T_1 = T \in \mathbb{U}_j$ for some variable x_1
$S_1{+}T_1 = S_2{+}T_2 \in \mathbb{U}_j$	if $S_1{=}S_2{\in}\mathbb{U}_j$ and $T_1{=}T_2{\in}\mathbb{U}_j$
$s_1{=}t_1 \in T_1 = s_2{=}t_2 \in T_2 \in \mathbb{U}_j$	if $T_1{=}T_2{\in}\mathbb{U}_j$, $s_1{=}s_2{\in}T_1$, and $t_1{=}t_2{\in}T_1$.
$\mathbb{U}_{j_1} = \mathbb{U}_{j_2} \in \mathbb{U}_j$	if $j_1{=}j_2{<}j$ *(as natural number)*
$\mathsf{void} = \mathsf{void} \in \mathbb{U}_j$	
$\mathsf{Atom} = \mathsf{Atom} \in \mathbb{U}_j$	
$\mathbb{Z} = \mathbb{Z} \in \mathbb{U}_j$	
$i_1{<}j_1 = i_2{<}j_2 \in \mathbb{U}_j$	if $i_1 = i_2 \in \mathbb{Z}$ und $j_1 = j_2 \in \mathbb{Z}$
$T_1\,\mathsf{list} = T_2\,\mathsf{list} \in \mathbb{U}_j$	if $T_1 = T_2 \in \mathbb{U}_j$
$\mathsf{rectype}\ X_1{=}T_1 = \mathsf{rectype}\ X_2{=}T_2 \in \mathbb{U}_j$	if $T_1[X/X_1] = T_2[X/X_2] \in \mathbb{U}_j$ for all $X \in \mathbb{U}_j$
$\{x_1{:}S_1 \mid T_1\} = \{x_2{:}S_2 \mid T_2\} \in \mathbb{U}_j$	if $S_1{=}S_2{\in}\mathbb{U}_j$ and there are terms p_1, p_2 and a variable x, which occurs neither in T_1 nor in T_2, such that $p_1 \in \forall x{:}S_1.\ T_1[x/x_1] \Rightarrow T_2[x/x_2]$ and $p_2 \in \forall x{:}S_1.\ T_2[x/x_2] \Rightarrow T_1[x/x_1]$.
$T = \{S_2 \mid T_2\} \in \mathbb{U}_j$	if $T = \{x_2{:}S_2 \mid T_2\} \in \mathbb{U}_j$ for some variable x_2.
$\{S_1 \mid T_1\} = T \in \mathbb{U}_j$	if $\{x_1{:}S_1 \mid T_1\} = T \in \mathbb{U}_j$ for some variable x_1.
$\cap x_1{:}S_1.T_1 = \cap x_2{:}S_2.T_2 \in \mathbb{U}_j$	if $S_1{=}S_2{\in}\mathbb{U}_j$ and $T_1[s_1/x_1]{=}T_2[s_2/x_2]{\in}\mathbb{U}_j$ for all s_1, s_2 with $s_1{=}s_2{\in}S_1$.
$x_1,y_1 : T_1//E_1 = x_2,y_2 : T_2//E_2 \in \mathbb{U}_j$	if $T_1 = T_2 \in \mathbb{U}_j$ and there are terms p_1,p_2,r,s,t and variables x,y,z, which occur neither in E_1 nor in E_2, such that $p_1 \in \forall x{:}T_1.\forall y{:}T_1.\ E_1[x,y/x_1,y_1] \Rightarrow E_2[x,y/x_2,y_2]$, $p_2 \in \forall x{:}T_1.\forall y{:}T_1.\ E_2[x,y/x_2,y_2] \Rightarrow E_1[x,y/x_1,y_1]$, $r \in \forall x{:}T_1.\ E_1[x,x/x_1,y_1]$, $s \in \forall x{:}T_1.\forall y{:}T_1.\ E_1[x,y/x_1,y_1] \Rightarrow E_1[y,x/x_1,y_1]$, and $t \in \forall x{:}T_1.\forall y{:}T_1.\forall z{:}T_1.$ $E_1[x,y/x_1,y_1] \Rightarrow E_1[y,z/x_1,y_1] \Rightarrow E_1[x,z/x_1,y_1]$

Table A.4: Member semantics table for Nuprl

A.3 Inference rules

Nuprl's inference rules describe the top-down refinement of proof sequents (see Chapter 6) and the bottom-up construction of extract terms. Rules are written in a top-down fashion, showing the goal sequent above the rule name and the subgoal sequents below it.

For each type there are rules for type formation and type equality, formation and equality of canonical members, equality of noncanonical forms, type decomposition in hypotheses (elimination), computation rules, and possibly additional rules. In addition to the rule name, a rule may need certain arguments (see Section 8.1.2) such as

- The position of a hypothesis to be used as in `hypothesis` $\boxed{i}$
- Names for newly created variables as in `functionEquality` $\boxed{x}$
- The universe level of a type as in `lambdaEquality` $\boxed{j}$ x'
- A term that instantiates a variable as in `dependent_pairFormation` j $\boxed{s}$ x'
- The type of some subterm in the goal as in `applyEquality` $\boxed{x{:}S{\rightarrow}T}$
- The dependency of a term $C[z]$ from a variable z as in `decideEquality` $\boxed{z\ \ C}$ $S{+}T$ s t y

Most of the elementary inference rules are subsumed by the one-step decomposition tactics `D`, `MemCD`, `EqCD`, `MemHD`, `EqHD`, `MemTypeCD`, `EqTypeCD`, `MemTypeHD`, `EqTypeHD`. These tactics try to determine the parameters of the corresponding rules from the context unless they are explicitly provided with the tacticals `New`, `At`, `With`, or `Using` (see Section 8.2.2). A user may choose to use these tacticals to support the tactics in situations where appropriate parameters cannot be found automatically or in order to enforce the use of, for instance, particular names for newly created variables.

In the following we present the basic inference rules of NUPRL's type theory as well as the tactics that can be used to perform the same one-step decomposition of proof goals. For the latter we describe both the minimal form, which only lists tactics that are needed for injecting required arguments, and a maximal form with all tacticals that may have an effect on the execution of the tactic. Some rules that are now considered obsolete are not covered by tactics and have to be converted explicitly into tactics using the function `refine` (see Section 8.1.3).

For integer and list induction we use the abstract terms instead of the lengthy display forms. Goals of the form $a \in T$ always abbreviate a=$a \in T$. $\neg P$ stands for $P{\rightarrow}\mathsf{void}$, $s{\neq}t$ for $\neg(s{=}t \in \mathbb{Z})$, and $s{\leq}t$ for $\neg(t{<}s)$.

A.3.1 Functions

$\Gamma \vdash \mathbb{U}_j \ \lfloor \textbf{ext } x{:}S{\to}T \rfloor$
by dependent_functionFormation x S
$\Gamma \vdash S \in \mathbb{U}_j \ \lfloor \mathsf{Ax} \rfloor$
$\Gamma,\ x{:}S \vdash \mathbb{U}_j \ \lfloor \textbf{ext } T \rfloor$

$\Gamma \vdash \mathbb{U}_j \ \lfloor \textbf{ext } S{\to}T \rfloor$
by independent_functionFormation
$\Gamma \vdash \mathbb{U}_j \ \lfloor \textbf{ext } S \rfloor$
$\Gamma \vdash \mathbb{U}_j \ \lfloor \textbf{ext } T \rfloor$

$\Gamma \vdash x_1{:}S_1{\to}T_1 = x_2{:}S_2{\to}T_2 \in \mathbb{U}_j \ \lfloor \mathsf{Ax} \rfloor$
by functionEquality x
$\Gamma \vdash S_1 = S_2 \in \mathbb{U}_j \ \lfloor \mathsf{Ax} \rfloor$
$\Gamma,\ x{:}S_1 \vdash T_1[x/x_1] = T_2[x/x_2] \in \mathbb{U}_j \ \lfloor \mathsf{Ax} \rfloor$

$\Gamma \vdash S_1{\to}T_1 = S_2{\to}T_2 \in \mathbb{U}_j \ \lfloor \mathsf{Ax} \rfloor$
by independent_functionEquality
$\Gamma \vdash S_1 = S_2 \in \mathbb{U}_j \ \lfloor \mathsf{Ax} \rfloor$
$\Gamma \vdash T_1 = T_2 \in \mathbb{U}_j \ \lfloor \mathsf{Ax} \rfloor$

$\Gamma \vdash \lambda x_1.t_1 = \lambda x_2.t_2 \in x{:}S{\to}T \ \lfloor \mathsf{Ax} \rfloor$
by lambdaEquality j x'
$\Gamma,\ x'{:}S \vdash t_1[x'/x_1] = t_2[x'/x_2] \in T[x'/x] \ \lfloor \mathsf{Ax} \rfloor$
$\Gamma \vdash S \in \mathbb{U}_j \ \lfloor \mathsf{Ax} \rfloor$

$\Gamma \vdash x{:}S{\to}T \ \lfloor \textbf{ext } \lambda x'.t \rfloor$
by lambdaFormation j x'
$\Gamma,\ x'{:}S \vdash T[x'/x] \ \lfloor \textbf{ext } t \rfloor$
$\Gamma \vdash S \in \mathbb{U}_j \ \lfloor \mathsf{Ax} \rfloor$

$\Gamma \vdash f_1 t_1 = f_2 t_2 \in T[t_1/x] \ \lfloor \mathsf{Ax} \rfloor$
by applyEquality $x{:}S{\to}T$
$\Gamma \vdash f_1 = f_2 \in x{:}S{\to}T \ \lfloor \mathsf{Ax} \rfloor$
$\Gamma \vdash t_1 = t_2 \in S \ \lfloor \mathsf{Ax} \rfloor$

$\Gamma,\ f{:}\,S{\to}T,\ \Delta \vdash C \ \lfloor \textbf{ext } t[f\,s,/y] \rfloor$
by independent_functionElimination i y
$\Gamma,\ f{:}\,S{\to}T,\ \Delta \vdash S \ \lfloor \textbf{ext } s \rfloor$
$\Gamma,\ f{:}\,S{\to}T,\ y{:}T,\ \Delta \vdash C \ \lfloor \textbf{ext } t \rfloor$

$\Gamma,\ f{:}\,x{:}S{\to}T,\ \Delta \vdash C \ \lfloor \textbf{ext } t[f\,s, \mathsf{Ax}/y, z] \rfloor$
by dependent_functionElimination i s y z
$\Gamma,\ f{:}\,x{:}S{\to}T,\ \Delta \vdash s \in S \ \lfloor \mathsf{Ax} \rfloor$
$\Gamma,\ f{:}\,x{:}S{\to}T,\ y{:}T[s/x],\ z{:}\,y{=}f\,s \in T[s/x],\ \Delta \vdash C \ \lfloor \textbf{ext } t \rfloor$

$\Gamma \vdash (\lambda x.t)\ s = t_2 \in T \ \lfloor \mathsf{Ax} \rfloor$
by applyReduce
$\Gamma \vdash t[s/x] = t_2 \in T \ \lfloor \mathsf{Ax} \rfloor$

$\Gamma \vdash f_1 = f_2 \in x{:}S{\to}T \ \lfloor \textbf{ext } t \rfloor$
by functionExtensionality j $x_1{:}S_1{\to}T_1$ $x_2{:}S_2{\to}T_2$ x'
$\Gamma,\ x'{:}S \vdash f_1 x' = f_2 x' \in T[x'/x] \ \lfloor \textbf{ext } t \rfloor$
$\Gamma \vdash S \in \mathbb{U}_j \ \lfloor \mathsf{Ax} \rfloor$
$\Gamma \vdash f_1 \in x_1{:}S_1{\to}T_1 \ \lfloor \mathsf{Ax} \rfloor$
$\Gamma \vdash f_2 \in x_2{:}S_2{\to}T_2 \ \lfloor \mathsf{Ax} \rfloor$

Basic Inference Rule	*Corresponding Tactic* *with required arguments*	*with optional tacticals*
dependent_functionFormation x S	---	
independent_functionFormation	---	
functionEquality x	EqCD	New $[x]$ EqCD
independent_functionEquality	---	
lambdaEquality j x'	EqCD	At $\mathbb{U}_j$ EqCD
lambdaFormation j x'	D 0	
applyEquality $x{:}S{\to}T$	EqCD	With $x{:}S{\to}T$ EqCD
independent_functionElimination i y	D i	
dependent_functionElimination i s y z	D i	
applyReduce	ReduceEquands 0	ReduceAtAddr [2] 0
functionExtensionality j $x_1{:}S_1{\to}T_1$ $x_2{:}S_2{\to}T_2$ x'	EqExtWith	Ext

A.3.2 Products

$\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** $x{:}S{\times}T\rfloor$
by `dependent_productFormation` x S
$\Gamma \vdash S \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
$\Gamma,\ x{:}S \vdash \mathbb{U}_j$ $\lfloor$**ext** $T\rfloor$

$\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** $S{\times}T\rfloor$
by `independent_productFormation`
$\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** $S\rfloor$
$\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** $T\rfloor$

$\Gamma \vdash x_1{:}S_1{\times}T_1 = x_2{:}S_2{\times}T_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
by `productEquality` x'
$\Gamma \vdash S_1 = S_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
$\Gamma,\ x'{:}S_1 \vdash T_1[x'/x_1] = T_2[x'/x_2] \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash S_1{\times}T_1 = S_2{\times}T_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
by `independent_productEquality`
$\Gamma \vdash S_1 = S_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
$\Gamma \vdash T_1 = T_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash \langle s_1,t_1\rangle = \langle s_2,t_2\rangle \in x{:}S{\times}T$ $\lfloor$Ax$\rfloor$
by `dependent_pairEquality` j x'
$\Gamma \vdash s_1 = s_2 \in S$ $\lfloor$Ax$\rfloor$
$\Gamma \vdash t_1 = t_2 \in T[s_1/x]$ $\lfloor$Ax$\rfloor$
$\Gamma,\ x'{:}S \vdash T[x'/x] \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash x{:}S{\times}T$ $\lfloor$**ext** $\langle s,t\rangle\rfloor$
by `dependent_pairFormation` j s x'
$\Gamma \vdash s \in S$ $\lfloor$Ax$\rfloor$
$\Gamma \vdash T[s/x]$ $\lfloor$**ext** $t\rfloor$
$\Gamma,\ x'{:}S \vdash T[x'/x] \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash \langle s_1,t_1\rangle = \langle s_1,t_1\rangle \in S{\times}T$ $\lfloor$Ax$\rfloor$
by `independent_pairEquality`
$\Gamma \vdash s_1 = s_2 \in S$ $\lfloor$Ax$\rfloor$
$\Gamma \vdash t_1 = t_2 \in T$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash S{\times}T$ $\lfloor$**ext** $\langle s,t\rangle\rfloor$
by `independent_pairFormation`
$\Gamma \vdash S$ $\lfloor$**ext** $s\rfloor$
$\Gamma \vdash T$ $\lfloor$**ext** $t\rfloor$

$\Gamma \vdash \text{let } \langle x_1,y_1\rangle = e_1 \text{ in } t_1 = \text{let } \langle x_2,y_2\rangle = e_2 \text{ in } t_2 \in C[e_1/z]$ $\lfloor$Ax$\rfloor$
by `spreadEquality` z C $x{:}S{\times}T$ s t y
$\Gamma \vdash e_1 = e_2 \in x{:}S{\times}T$ $\lfloor$Ax$\rfloor$
$\Gamma,\ s{:}S,\ t{:}T[s/x],\ y{:}\,e_1{=}\langle s,t\rangle \in x{:}S{\times}T \vdash t_1[s,t/x_1,y_1] = t_2[s,t/x_2,y_2] \in C[\langle s,t\rangle/z]$ $\lfloor$Ax$\rfloor$

$\Gamma,\ z{:}\,x{:}S{\times}T,\ \Delta \vdash C$ $\lfloor$**ext** $\text{let } \langle s,t\rangle = z \text{ in } u\rfloor$
by `productElimination` i s t
$\Gamma,\ z{:}\,x{:}S{\times}T,\ s{:}S,\ t{:}T[s/x]\ \Delta[\langle s,t\rangle/z] \vdash C[\langle s,t\rangle/z]$ $\lfloor$**ext** $u\rfloor$

$\Gamma \vdash \text{let } \langle x,y\rangle = \langle s,t\rangle \text{ in } u = t_2 \in T$ $\lfloor$Ax$\rfloor$
by `spreadReduce`
$\Gamma \vdash u[s,t/x,y] = t_2 \in T$ $\lfloor$Ax$\rfloor$

Basic Inference Rule	*Corresponding Tactic* *with required arguments*	*with optional tacticals*
`dependent_productFormation` x S	---	
`independent_productFormation`	---	
`productEquality` x'	`EqCD`	
`independent_productEquality`	---	
`dependent_pairEquality` j x'	`EqCD`	
`dependent_pairEquality2` j x'	`EqCD`	
`dependent_pairFormation` j s x'	`With` s `(D 0)`	`At` $\mathbb{U}_j$ `(With` s `(New` $[x']$ `(D 0)))`
`independent_pairEquality`	`EqCD`	
`independent_pairFormation`	`D 0`	
`spreadEquality` z C $x{:}S{\times}T$ s t y	`EqCD`	
`productElimination` i s t	`D` i	
`spreadReduce`	`ReduceEquands 0`	`ReduceAtAddr [2] 0`

A.3.3 Disjoint Union

$\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** $S{+}T\rfloor$
by unionFormation
  $\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** $S\rfloor$
  $\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** $T\rfloor$

$\Gamma \vdash S_1{+}T_1 = S_2{+}T_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
by unionEquality
  $\Gamma \vdash S_1 = S_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
  $\Gamma \vdash T_1 = T_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash \mathsf{inl}(s_1) = \mathsf{inl}(s_2) \in S{+}T$ $\lfloor$Ax$\rfloor$
by inlEquality j
  $\Gamma \vdash s_1 = s_2 \in S$ $\lfloor$Ax$\rfloor$
  $\Gamma \vdash T \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash S{+}T$ $\lfloor$**ext** $\mathsf{inl}(s)\rfloor$
by inlFormation j
  $\Gamma \vdash S$ $\lfloor$**ext** $s\rfloor$
  $\Gamma \vdash T \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash \mathsf{inr}(t_1) = \mathsf{inr}(t_2) \in S{+}T$ $\lfloor$Ax$\rfloor$
by inrEquality j
  $\Gamma \vdash t_1 = t_2 \in T$ $\lfloor$Ax$\rfloor$
  $\Gamma \vdash S \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash S{+}T$ $\lfloor$**ext** $\mathsf{inr}(t)\rfloor$
by inrFormation j
  $\Gamma \vdash T$ $\lfloor$**ext** $t\rfloor$
  $\Gamma \vdash S \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash \mathsf{case}\ e_1\ \mathsf{of}\ \mathsf{inl}(x_1) \mapsto u_1 \mid \mathsf{inr}(y_1) \mapsto v_1 = \mathsf{case}\ e_2\ \mathsf{of}\ \mathsf{inl}(x_2) \mapsto u_2 \mid \mathsf{inr}(y_2) \mapsto v_2 \in C[e_1/z]$ $\lfloor$Ax$\rfloor$
by decideEquality z C $S{+}T$ s t y
  $\Gamma \vdash e_1 = e_2 \in S{+}T$ $\lfloor$Ax$\rfloor$
  $\Gamma,\ s{:}S,\ y{:}\ e_1{=}\mathsf{inl}(s) \in S{+}T \vdash u_1[s/x_1] = u_2[s/x_2] \in C[\mathsf{inl}(s)/z]$ $\lfloor$Ax$\rfloor$
  $\Gamma,\ t{:}T,\ y{:}\ e_1{=}\mathsf{inr}(t) \in S{+}T \vdash v_1[t/y_1] = v_2[t/y_2] \in C[\mathsf{inr}(t)/z]$ $\lfloor$Ax$\rfloor$

$\Gamma,\ z{:}S{+}T,\ \Delta \vdash C$ $\lfloor$**ext** $\mathsf{case}\ z\ \mathsf{of}\ \mathsf{inl}(x) \mapsto u \mid \mathsf{inr}(y) \mapsto v\rfloor$
by unionElimination i x y
  $\Gamma,\ z{:}S{+}T,\ x{:}S,\ \Delta[\mathsf{inl}(x)/z] \vdash C[\mathsf{inl}(x)/z]$ $\lfloor$**ext** $u\rfloor$
  $\Gamma,\ z{:}S{+}T,\ y{:}T,\ \Delta[\mathsf{inr}(y)/z] \vdash C[\mathsf{inr}(y)/z]$ $\lfloor$**ext** $v\rfloor$

$\Gamma \vdash \mathsf{case}\ \mathsf{inl}(s)\ \mathsf{of}\ \mathsf{inl}(x) \mapsto u \mid \mathsf{inr}(y) \mapsto v = t_2 \in T$ $\lfloor$Ax$\rfloor$
by decideReduceLeft
  $\Gamma \vdash u[s/x] = t_2 \in T$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash \mathsf{case}\ \mathsf{inr}(t)\ \mathsf{of}\ \mathsf{inl}(x) \mapsto u \mid \mathsf{inr}(y) \mapsto v = t_2 \in T$ $\lfloor$Ax$\rfloor$
by decideReduceRight
  $\Gamma \vdash v[t/y] = t_2 \in T$ $\lfloor$Ax$\rfloor$

Basic Inference Rule	*Corresponding Tactic* *with required arguments*	*with optional tacticals*
unionFormation	---	
unionEquality	EqCD	
inlEquality j	EqCD	
inlFormation j	Sel 1 D 0	
inrEquality j	EqCD	
inrFormation j	Sel 2 D 0	
decideEquality z C $S{+}T$ s t y	EqCD	Using [z,C] (With $S{+}T$ (New s t y EqCD))
unionElimination i x y	D i	
decideReduceLeft	ReduceEquands 0	ReduceAtAddr [2] 0
decideReduceRight	ReduceEquands 0	ReduceAtAddr [2] 0

A.3.4 Universes

$\Gamma \vdash \mathbb{U}_k$ ⌊**ext** $\mathbb{U}_j$⌋
by `universeFormation` j $*$

$\Gamma \vdash \mathbb{U}_j = \mathbb{U}_j \in \mathbb{U}_k$ ⌊Ax⌋
by `universeEquality` $*$

$\Gamma \vdash T \in \mathbb{U}_k$ ⌊Ax⌋
by `cumulativity` j $*$
$\Gamma \vdash T \in \mathbb{U}_j$ ⌊Ax⌋

$*$: *proviso* $j < k$

Basic Inference Rule	*Corresponding Tactic* *with required arguments with optional tacticals*
`universeFormation` j	`D 0`
`universeEquality`	`EqCD`
`cumulativity` j	`Cumulativity` j

A.3.5 Equality

$\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** $s = t \in T\rfloor$
by equalityFormation T
  $\Gamma \vdash T \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
  $\Gamma \vdash T$ $\lfloor$**ext** $s\rfloor$
  $\Gamma \vdash T$ $\lfloor$**ext** $t\rfloor$

$\Gamma \vdash s_1 = t_1 \in T_1 = s_2 = t_2 \in T_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
by equalityEquality
  $\Gamma \vdash T_1 = T_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
  $\Gamma \vdash s_1 = s_2 \in T_1$ $\lfloor$Ax$\rfloor$
  $\Gamma \vdash t_1 = t_2 \in T_1$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash$ Ax = Ax $\in s = t \in T$ $\lfloor$Ax$\rfloor$
by axiomEquality
  $\Gamma \vdash s = t \in T$ $\lfloor$Ax$\rfloor$

$\Gamma,\ z\!: s = t \in T,\ \Delta \vdash C$ $\lfloor$**ext** $u\rfloor$
by equalityElimination i
  $\Gamma,\ z\!: s = t \in T,\ \Delta[\mathsf{Ax}/z] \vdash C[\mathsf{Ax}/z]$ $\lfloor$**ext** $u\rfloor$

$\Gamma,\ x\!:\!T,\ \Delta \vdash x = x \in T$ $\lfloor$Ax$\rfloor$
by hypothesisEquality i

$\Gamma \vdash C[s/x]$ $\lfloor$**ext** $u\rfloor$
by substitution j $s{=}t \in T$ x C
  $\Gamma \vdash s{=}t \in T$ $\lfloor$Ax$\rfloor$
  $\Gamma \vdash C[t/x]$ $\lfloor$**ext** $u\rfloor$
  $\Gamma,\ x\!:\!T \vdash C \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash s = t \in T$ $\lfloor$Ax$\rfloor$
by equality

Decision procedure for elementary equalities

Basic Inference Rule	*Corresponding Tactic* *with required arguments*	*with optional tacticals*
equalityFormation T	---	
equalityEquality	EqCD	
axiomEquality	EqCD	
equalityElimination i	D i	
hypothesisEquality i	Declaration	NthDecl i
substitution j $s{=}t \in T$ x C	Subst $s{=}t \in T$ 0	At j (BasicSubst $s{=}t \in T$ x C)
equality	Eq	Eq

A.3.6 Void

$\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** void$\rfloor$
by voidFormation

$\Gamma \vdash$ void = void $\in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
by voidEquality

$\Gamma \vdash$ any(s) = any(t) $\in T$ $\lfloor$Ax$\rfloor$
by anyEquality
$\Gamma \vdash s = t \in$ void $\lfloor$Ax$\rfloor$

Γ, z: void, $\Delta \vdash C$ $\lfloor$**ext** any(z)$\rfloor$
by voidElimination i

Basic Inference Rule	*Corresponding Tactic* *with required arguments with optional tacticals*
voidFormation	---
voidEquality	EqCD
anyEquality	EqCD
voidElimination i	D i

A.3.7 Atom

$\Gamma \vdash \mathbb{U}_j$ ⌊**ext** Atom⌋
by atomFormation

$\Gamma \vdash$ Atom = Atom $\in \mathbb{U}_j$ ⌊Ax⌋
by atomEquality

$\Gamma \vdash$ "*token*" = "*token*" $\in$ Atom ⌊Ax⌋
by tokenEquality

$\Gamma \vdash$ Atom ⌊**ext** "*token*"⌋
by tokenFormation "*token*"

$\Gamma \vdash$ if u_1=v_1 then s_1 else t_1 = if u_2=v_2 then s_2 else t_2 $\in T$ ⌊Ax⌋
by atom_eqEquality v

- $\Gamma \vdash u_1 = u_2 \in$ Atom ⌊Ax⌋
- $\Gamma \vdash v_1 = v_2 \in$ Atom ⌊Ax⌋
- Γ, v: u_1=v_1∈Atom $\vdash s_1 = s_2 \in T$ ⌊Ax⌋
- Γ, v: ¬(u_1=v_1∈Atom) $\vdash t_1 = t_2 \in T$ ⌊Ax⌋

$\Gamma \vdash$ if u=v then s else t = $t_2 \in T$ ⌊Ax⌋
by atom_eqReduceTrue

- $\Gamma \vdash s = t_2 \in T$ ⌊Ax⌋
- $\Gamma \vdash u = v \in$ Atom ⌊Ax⌋

$\Gamma \vdash$ if u=v then s else t = $t_2 \in T$ ⌊Ax⌋
by atom_eqReduceFalse

- $\Gamma \vdash t = t_2 \in T$ ⌊Ax⌋
- $\Gamma \vdash$ ¬($u = v \in$ Atom) ⌊Ax⌋

Basic Inference Rule	*Corresponding Tactic* *with required arguments*	*with optional tacticals*
atomFormation	---	
atomEquality	EqCD	
tokenEquality	EqCD	
tokenFormation "*token*"	D 0	
atom_eqEquality v	EqCD	
atom_eqReduceTrue	PrimReduceFirstEquand ‘true‘	PrimReduceEquands ‘true‘ [1]
atom_eqReduceFalse	PrimReduceFirstEquand ‘false‘	PrimReduceEquands ‘false‘ [1]

A.3.8 Integers

$\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** $\mathbb{Z}\rfloor$
by `intFormation`

$\Gamma \vdash \mathbb{Z} \in \mathbb{U}_j$ $\lfloor \mathrm{Ax} \rfloor$
by `intEquality`

$\Gamma \vdash n = n \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$
by `natural_numberEquality`

$\Gamma \vdash \mathbb{Z}$ $\lfloor$**ext** $n\rfloor$
by `natural_numberFormation` n

$\Gamma \vdash -s_1 = -s_2 \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$
by `minusEquality`
$\Gamma \vdash s_1 = s_2 \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$

$\Gamma \vdash s_1 + t_1 = s_2 + t_2 \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$
by `addEquality`
$\Gamma \vdash s_1 = s_2 \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$
$\Gamma \vdash t_1 = t_2 \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$

$\Gamma \vdash \mathbb{Z}$ $\lfloor$**ext** $s+t\rfloor$
by `addFormation`
$\Gamma \vdash \mathbb{Z}$ $\lfloor$**ext** $s\rfloor$
$\Gamma \vdash \mathbb{Z}$ $\lfloor$**ext** $t\rfloor$

$\Gamma \vdash s_1 - t_1 = s_2 - t_2 \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$
by `subtractEquality`
$\Gamma \vdash s_1 = s_2 \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$
$\Gamma \vdash t_1 = t_2 \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$

$\Gamma \vdash \mathbb{Z}$ $\lfloor$**ext** $s-t\rfloor$
by `subtractFormation`
$\Gamma \vdash \mathbb{Z}$ $\lfloor$**ext** $s\rfloor$
$\Gamma \vdash \mathbb{Z}$ $\lfloor$**ext** $t\rfloor$

$\Gamma \vdash s_1 * t_1 = s_2 * t_2 \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$
by `multiplyEquality`
$\Gamma \vdash s_1 = s_2 \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$
$\Gamma \vdash t_1 = t_2 \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$

$\Gamma \vdash \mathbb{Z}$ $\lfloor$**ext** $s*t\rfloor$
by `multiplyFormation`
$\Gamma \vdash \mathbb{Z}$ $\lfloor$**ext** $s\rfloor$
$\Gamma \vdash \mathbb{Z}$ $\lfloor$**ext** $t\rfloor$

$\Gamma \vdash s_1 \div t_1 = s_2 \div t_2 \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$
by `divideEquality`
$\Gamma \vdash s_1 = s_2 \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$
$\Gamma \vdash t_1 = t_2 \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$
$\Gamma \vdash t_1 \neq 0$ $\lfloor \mathrm{Ax} \rfloor$

$\Gamma \vdash \mathbb{Z}$ $\lfloor$**ext** $s\div t\rfloor$
by `divideFormation`
$\Gamma \vdash \mathbb{Z}$ $\lfloor$**ext** $s\rfloor$
$\Gamma \vdash \mathbb{Z}$ $\lfloor$**ext** $t\rfloor$

$\Gamma \vdash s_1 \,\mathrm{rem}\, t_1 = s_2 \,\mathrm{rem}\, t_2 \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$
by `remainderEquality`
$\Gamma \vdash s_1 = s_2 \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$
$\Gamma \vdash t_1 = t_2 \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$
$\Gamma \vdash t_1 \neq 0$ $\lfloor \mathrm{Ax} \rfloor$

$\Gamma \vdash \mathbb{Z}$ $\lfloor$**ext** $s \,\mathrm{rem}\, t\rfloor$
by `remainderFormation`
$\Gamma \vdash \mathbb{Z}$ $\lfloor$**ext** $s\rfloor$
$\Gamma \vdash \mathbb{Z}$ $\lfloor$**ext** $t\rfloor$

$\Gamma \vdash 0 \leq s \,\mathrm{rem}\, t \wedge s \,\mathrm{rem}\, t < t$ $\lfloor \mathrm{Ax} \rfloor$
by `remainderBounds1`
$\Gamma \vdash 0 \leq s$ $\lfloor \mathrm{Ax} \rfloor$
$\Gamma \vdash 0 < t$ $\lfloor \mathrm{Ax} \rfloor$

$\Gamma \vdash 0 \leq s \,\mathrm{rem}\, t \wedge s \,\mathrm{rem}\, t < -t$ $\lfloor \mathrm{Ax} \rfloor$
by `remainderBounds2`
$\Gamma \vdash 0 \leq s$ $\lfloor \mathrm{Ax} \rfloor$
$\Gamma \vdash t < 0$ $\lfloor \mathrm{Ax} \rfloor$

$\Gamma \vdash s \,\mathrm{rem}\, t \leq 0 \wedge s \,\mathrm{rem}\, t > t$ $\lfloor \mathrm{Ax} \rfloor$
by `remainderBounds3`
$\Gamma \vdash s \leq 0$ $\lfloor \mathrm{Ax} \rfloor$
$\Gamma \vdash t < 0$ $\lfloor \mathrm{Ax} \rfloor$

$\Gamma \vdash s \,\mathrm{rem}\, t \leq 0 \wedge s \,\mathrm{rem}\, t > -t$ $\lfloor \mathrm{Ax} \rfloor$
by `remainderBounds4`
$\Gamma \vdash s \leq 0$ $\lfloor \mathrm{Ax} \rfloor$
$\Gamma \vdash 0 < t$ $\lfloor \mathrm{Ax} \rfloor$

$\Gamma \vdash s = (s \div t) * t + (s \,\mathrm{rem}\, t)$ $\lfloor \mathrm{Ax} \rfloor$
by `divideRemainderSum`
$\Gamma \vdash s = s \in \mathbb{Z}$ $\lfloor \mathrm{Ax} \rfloor$
$\Gamma \vdash t \neq 0$ $\lfloor \mathrm{Ax} \rfloor$

$\Gamma \vdash \mathsf{ind}(u_1; x_1, f_{x_1}.s_1; base_1; y_1, f_{y_1}.t_1) = \mathsf{ind}(u_2; x_2, f_{x_2}.s_2; base_2; y_2, f_{y_2}.t_2) \in T[u_1/z]$ ⌊Ax⌋
by indEquality z T x f_x v

$\Gamma \vdash u_1 = u_2 \in \mathbb{Z}$ ⌊Ax⌋
$\Gamma,\ x{:}\mathbb{Z},\ v{:}\ x{<}0,\ f_x{:}T[(x{+}1)/z] \vdash s_1[x, f_x/x_1, f_{x1}] = s_2[x, f_x/x_2, f_{x2}] \in T[x/z]$ ⌊Ax⌋
$\Gamma \vdash base_1 = base_2 \in T[0/z]$ ⌊Ax⌋
$\Gamma,\ x{:}\mathbb{Z},\ v{:}\ 0{<}x,\ f_x{:}T[(x{-}1)/z] \vdash t_1[x, f_x/y_1, f_{y1}] = t_2[x, f_x/y_2, f_{y2}] \in T[x/z]$ ⌊Ax⌋

$\Gamma,\ z{:}\mathbb{Z},\ \Delta \vdash C$ ⌊**ext** $\mathsf{ind}(z;\ x, f_x.s[\mathsf{Ax}/v];\ base;\ x, f_x.t[\mathsf{Ax}/v])$⌋
by intElimination i x f_x v

$\Gamma,\ z{:}\mathbb{Z},\ \Delta,\ x{:}\mathbb{Z},\ v{:}\ x{<}0,\ f_x{:}C[(x{+}1)/z] \vdash C[x/z]$ ⌊**ext** s⌋
$\Gamma,\ z{:}\mathbb{Z},\ \Delta \vdash C[0/z]$ ⌊**ext** $base$⌋
$\Gamma,\ z{:}\mathbb{Z},\ \Delta,\ x{:}\mathbb{Z},\ v{:}\ 0{<}x,\ f_x{:}C[(x{-}1)/z] \vdash C[x/z]$ ⌊**ext** t⌋

$\Gamma \vdash \mathsf{ind}(i; x, f_x.s; base; y, f_y.t) = t_2 \in T$ ⌊Ax⌋
by indReduceDown

$\Gamma \vdash t[i, \mathsf{ind}(i{+}1; x, f_x.s; base; y, f_y.t)/x, f_x] = t_2 \in T$ ⌊Ax⌋
$\Gamma \vdash i < 0$ ⌊Ax⌋

$\Gamma \vdash \mathsf{ind}(i; x, f_x.s; base; y, f_y.t) = t_2 \in T$ ⌊Ax⌋
by indReduceUp

$\Gamma \vdash t[i, \mathsf{ind}(i{-}1; x, f_x.s; base; y, f_y.t)/y, f_y] = t_2 \in T$ ⌊Ax⌋
$\Gamma \vdash 0 < i$ ⌊Ax⌋

$\Gamma \vdash \mathsf{ind}(i; x, f_x.s; base; y, f_y.t) = t_2 \in T$ ⌊Ax⌋
by indReduceBase

$\Gamma \vdash base = t_2 \in T$ ⌊Ax⌋
$\Gamma \vdash i = 0 \in \mathbb{Z}$ ⌊Ax⌋

$\Gamma \vdash \mathsf{if}\ u_1{=}v_1\ \mathsf{then}\ s_1\ \mathsf{else}\ t_1 = \mathsf{if}\ u_2{=}v_2\ \mathsf{then}\ s_2\ \mathsf{else}\ t_2 \in T$ ⌊Ax⌋
by int_eqEquality

$\Gamma \vdash u_1 = u_2 \in \mathbb{Z}$ ⌊Ax⌋
$\Gamma \vdash v_1 = v_2 \in \mathbb{Z}$ ⌊Ax⌋
$\Gamma,\ v{:}\ u_1{=}v_1 \vdash s_1 = s_2 \in T$ ⌊Ax⌋
$\Gamma,\ v{:}\ u_1{\neq}v_1 \vdash t_1 = t_2 \in T$ ⌊Ax⌋

$\Gamma \vdash \mathsf{if}\ u{=}v\ \mathsf{then}\ s\ \mathsf{else}\ t = t_2 \in T$ ⌊Ax⌋
by int_eqReduceTrue

$\Gamma \vdash s = t_2 \in T$ ⌊Ax⌋
$\Gamma \vdash u = v \in \mathbb{Z}$ ⌊Ax⌋

$\Gamma \vdash \mathsf{if}\ u{=}v\ \mathsf{then}\ s\ \mathsf{else}\ t = t_2 \in T$ ⌊Ax⌋
by int_eqReduceFalse

$\Gamma \vdash t = t_2 \in T$ ⌊Ax⌋
$\Gamma \vdash u \neq v$ ⌊Ax⌋

$\Gamma \vdash \mathsf{if}\ u_1{<}v_1\ \mathsf{then}\ s\ \mathsf{else}\ t = \mathsf{if}\ u_2{<}v_2\ \mathsf{then}\ s_2\ \mathsf{else}\ t_2 \in T$ ⌊Ax⌋
by lessEquality

$\Gamma \vdash u_1 = u_2 \in \mathbb{Z}$ ⌊Ax⌋
$\Gamma \vdash v_1 = v_2 \in \mathbb{Z}$ ⌊Ax⌋
$\Gamma,\ v{:}\ u_1{<}v_1 \vdash s_1 = s_2 \in T$ ⌊Ax⌋
$\Gamma,\ v{:}\ u_1{\geq}v_1 \vdash t_1 = t_2 \in T$ ⌊Ax⌋

$\Gamma \vdash \mathsf{if}\ u{<}v\ \mathsf{then}\ s\ \mathsf{else}\ t = t_2 \in T$ ⌊Ax⌋
by lessReduceTrue

$\Gamma \vdash s = t_2 \in T$ ⌊Ax⌋
$\Gamma \vdash u < v$ ⌊Ax⌋

$\Gamma \vdash \mathsf{if}\ u{<}v\ \mathsf{then}\ s\ \mathsf{else}\ t = t_2 \in T$ ⌊Ax⌋
by lessReduceFalse

$\Gamma \vdash t = t_2 \in T$ ⌊Ax⌋
$\Gamma \vdash u \geq v$ ⌊Ax⌋

$\Gamma \vdash C$ $_{\lfloor}$**ext** $t_{\rfloor}$
by arith j
$\Gamma \vdash s_i \in \mathbb{Z}$ $_{\lfloor \mathrm{Ax} \rfloor}$

Decision procedure for elementary arithmetic

– subgoals for all non-arithmetical expressions s_i in C –

Basic Inference Rule	*Corresponding Tactic* *with required arguments*	*with optional tacticals*
intFormation	---	
intEquality	EqCD	
natural_numberEquality	EqCD	
natural_numberFormation n	With n (D 0)	
minusEquality	EqCD	
addEquality	EqCD	
addFormation	---	
subtractEquality	EqCD	
subtractFormation	---	
multiplyEquality	EqCD	
multiplyFormation	---	
divideEquality	EqCD	
divideFormation	---	
remainderEquality	EqCD	
remainderFormation	---	
remainderBounds1		
remainderBounds2		
remainderBounds3		
remainderBounds4		
divideRemainderSum		
indEquality z T x f_x v	StrongEqCD	
intElimination i x f_x v	D i	
indReduceDown	PrimReduceFirstEquand ‘down‘	PrimReduceEquands ‘down‘ [1]
indReduceUp	PrimReduceFirstEquand ‘up‘	PrimReduceEquands ‘up‘ [1]
indReduceBase	PrimReduceFirstEquand ‘base‘	PrimReduceEquands ‘base‘ [1]
int_eqEquality	EqCD	
int_eqReduceTrue	PrimReduceFirstEquand ‘true‘	PrimReduceEquands ‘true‘ [1]
int_eqReduceFalse	PrimReduceFirstEquand ‘false‘	PrimReduceEquands ‘false‘ [1]
lessEquality	EqCD	
lessReduceTrue	PrimReduceFirstEquand ‘true‘	PrimReduceEquands ‘true‘ [1]
lessReduceFalse	PrimReduceFirstEquand ‘false‘	PrimReduceEquands ‘false‘ [1]
arith j	Arith	

A.3.9 Less_Than Proposition

$\Gamma \vdash s_1 < t_1 = s_2 < t_2 \in \mathbb{U}_j$ ⌊Ax⌋
by less_thanEquality
  $\Gamma \vdash s_1 = s_2 \in \mathbb{Z}$ ⌊Ax⌋
  $\Gamma \vdash t_1 = t_2 \in \mathbb{Z}$ ⌊Ax⌋

$\Gamma \vdash \mathbb{U}_j$ ⌊**ext** $s < t$⌋
by less_thanFormation
  $\Gamma \vdash \mathbb{Z}$ ⌊**ext** s⌋
  $\Gamma \vdash \mathbb{Z}$ ⌊**ext** t⌋

$\Gamma \vdash$ Ax $\in s < t$ ⌊Ax⌋
by less_thanMember
  $\Gamma \vdash s < t$ ⌊Ax⌋

Basic Inference Rule	*Corresponding Tactic* *with required arguments with optional tacticals*
`less_thanEquality`	`EqCD`
`less_thanFormation`	`---`
`less_thanMember`	`EqCD`

A.3.10 Lists

$\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** T list$\rfloor$
by listFormation
$\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** $T\rfloor$

$\Gamma \vdash T_1$ list $=$ T_2 list $\in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
by listEquality
$\Gamma \vdash T_1 = T_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash$ [] = [] $\in T$ list $\lfloor$Ax$\rfloor$
by nilEquality j
$\Gamma \vdash T \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash T$ list $\lfloor$**ext** []$\rfloor$
by nilFormation j
$\Gamma \vdash T \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash t_1::l_1 = t_2::l_2 \in T$ list $\lfloor$Ax$\rfloor$
by consEquality
$\Gamma \vdash t_1 = t_2 \in T$ $\lfloor$Ax$\rfloor$
$\Gamma \vdash l_1 = l_2 \in T$ list $\lfloor$Ax$\rfloor$

$\Gamma \vdash T$ list $\lfloor$**ext** $t::l\rfloor$
by consFormation
$\Gamma \vdash T$ $\lfloor$**ext** $t\rfloor$
$\Gamma \vdash T$ list $\lfloor$**ext** $l\rfloor$

$\Gamma \vdash$ list_ind$(s_1;\, base_1;\, x_1, l_1, f_{xl1}.t_1)$ = list_ind$(s_2;\, base_2;\, x_2, l_2, f_{xl2}.t_2)$ $\in T[s_1/z]$ $\lfloor$Ax$\rfloor$
by list_indEquality z T S list x l f_{xl}
$\Gamma \vdash s_1 = s_2 \in S$ list $\lfloor$Ax$\rfloor$
$\Gamma \vdash base_1 = base_2 \in T[[]/z]$ $\lfloor$Ax$\rfloor$
$\Gamma,\ x{:}S,\ l{:}S$ list $,\ f_{xl}{:}T[l/z] \vdash t_1[x, l, f_{xl}/x_1, l_1, f_{xl1}] = t_2[x, l, f_{xl}/x_2, l_2, f_{xl2}] \in T[x::l/z]$ $\lfloor$Ax$\rfloor$

$\Gamma,\ z{:}T$ list $,\ \Delta \vdash C$ $\lfloor$**ext** list_ind$(z;\, base;\, x, l, f_{xl}.t)\rfloor$
by listElimination i f_{xl} x l
$\Gamma,\ z{:}T$ list $,\ \Delta \vdash C[[]/z]$ $\lfloor$**ext** $base\rfloor$
$\Gamma,\ z{:}T$ list $,\ \Delta,\ x{:}T,\ l{:}T$ list $,\ f_{xl}{:}C[l/z] \vdash C[x::l/z]$ $\lfloor$**ext** $t\rfloor$

$\Gamma \vdash$ list_ind$([];\, base;\, x, l, f_{xl}.t) = t_2 \in T$ $\lfloor$Ax$\rfloor$
by list_indReduceBase
$\Gamma \vdash base = t_2 \in T$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash$ list_ind$(s::u;\, base;\, x, l, f_{xl}.t) = t_2 \in T$ $\lfloor$Ax$\rfloor$
by list_indReduceUp
$\Gamma \vdash t[s, u,$ list_ind$(u;\, base;\, x, l, f_{xl}.t)/x, l, f_{xl}] = t_2 \in T$ $\lfloor$Ax$\rfloor$

Basic Inference Rule	*Corresponding Tactic* *with required arguments*	*with optional tacticals*
listFormation	---	
listEquality	EqCD	
nilEquality j	EqCD	
nilFormation j	D 0	
consEquality	EqCD	
consFormation	---	
list_indEquality z T S list x l f_{xl}	EqCD	
listElimination i f_{xl} x l	D i	
list_indReduceBase	ReduceEquands 0	ReduceAtAddr [2] 0
list_indReduceUp	ReduceEquands 0	ReduceAtAddr [2] 0

A.3.11 Inductive Types

$\Gamma \vdash \text{rectype } X_1 = T_{X1} \;=\; \text{rectype } X_2 = T_{X2} \;\in\; \mathbb{U}_j$ ⌊Ax⌋
by `recEquality` X
$\Gamma,\; X{:}\mathbb{U}_j \vdash T_{X1}[X/X_1] \;=\; T_{X2}[X/X_2] \;\in\; \mathbb{U}_j$ ⌊Ax⌋

$\Gamma \vdash s \;=\; t \;\in\; \text{rectype } X = T_X$ ⌊Ax⌋
by `rec_memberEquality` j
$\Gamma \vdash s \;=\; t \;\in\; T_X[\text{rectype } X = T_X/X]$ ⌊Ax⌋
$\Gamma \vdash \text{rectype } X = T_X \;\in\; \mathbb{U}_j$ ⌊Ax⌋

$\Gamma \vdash \text{rectype } X = T_X$ ⌊**ext** t⌋
by `rec_memberFormation` j
$\Gamma \vdash T_X[\text{rectype } X = T_X/X]$ ⌊**ext** t⌋
$\Gamma \vdash \text{rectype } X = T_X \;\in\; \mathbb{U}_j$ ⌊Ax⌋

$\Gamma \vdash \text{let}^*\, f_1(x_1) = t_1 \text{ in } f_1(e_1) \;=\; \text{let}^*\, f_2(x_2) = t_2 \text{ in } f_2(e_2) \;\in\; T[e_1/z]$ ⌊Ax⌋
by `rec_indEquality` z T $\text{rectype } X = T_X$ j P f x
$\Gamma \vdash e_1 \;=\; e_2 \;\in\; \text{rectype } X = T_X$ ⌊Ax⌋
$\Gamma \vdash \text{rectype } X = T_X \;\in\; \mathbb{U}_j$ ⌊Ax⌋
$\Gamma,\; P{:}\,(\text{rectype } X = T_X) \to \mathbb{P}_j,\; f{:}\,(x{:}\{x{:}\text{rectype } X = T_X \mid P(x)\} \to T[y/z]),$
$\quad x{:}\,T_X[\{x{:}\text{rectype } X = T_X \mid P(x)\}/X] \quad \vdash \quad t_1[f, x/f_1, x_1] \;=\; t_2[f, x/f_2, x_2] \in T[x/z]$ ⌊Ax⌋

$\Gamma,\; z{:}\,\text{rectype } X = T_X,\; \Delta \vdash C$ ⌊**ext** $\text{let}^*\, f(x) = t[\lambda y.\Lambda/P] \text{ in } f(z)$⌋
by `recElimination` i j P y f x
$\Gamma,\; z{:}\,\text{rectype } X = T_X,\; \Delta \vdash \text{rectype } X = T_X \;\in\; \mathbb{U}_j$ ⌊Ax⌋
$\Gamma,\; z{:}\,\text{rectype } X = T_X,\; \Delta,\; P{:}\,(\text{rectype } X = T_X) \to \mathbb{P}_j,\; f{:}\,(y{:}\{x{:}\text{rectype } X = T_X \mid P(x)\} \to C[y/z]),$
$\quad x{:}\,T_X[\{x{:}\text{rectype } X = T_X \mid P(x)\}/X] \quad \vdash \quad C[x/z]$ ⌊**ext** t⌋

$\Gamma,\; z{:}\,\text{rectype } X = T_X,\; \Delta \vdash C$ ⌊**ext** $t[z/x]$⌋
by `recUnrollElimination` i x v
$\Gamma,\; z{:}\,\text{rectype } X = T_X,\; \Delta,\; x{:}\,T_X[\text{rectype } X = T_X/X],\; v{:}\, z{=}x \in T_X[\text{rectype } X = T_X/X] \vdash C[x/z]$ ⌊**ext** t⌋

Basic Inference Rule	*Corresponding Tactic* *with required arguments with optional tacticals*
`recEquality` X	`---`
`rec_memberEquality` j	`EqTypeCD`
`rec_memberFormation` j	`---`
`rec_indEquality` z T $\text{rectype } X = T_X$ j P f x	`---`
`recElimination` i j P y f x	`RecTypeInduction` i
`recUnrollElimination` i x v	`D` i

A.3.12 Subset

$\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** $\{x{:}S \mid T\}\rfloor$
by dependent_setFormation S x
$\Gamma \vdash S \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
$\Gamma,\ x{:}S \vdash \mathbb{U}_j$ $\lfloor$**ext** $T\rfloor$

$\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** $\{S \mid T\}\rfloor$
by independent_setFormation
$\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** $S\rfloor$
$\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** $T\rfloor$

$\Gamma \vdash \{x_1{:}S_1 \mid T_1\} = \{x_2{:}S_2 \mid T_2\} \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
by setEquality x
$\Gamma \vdash S_1 = S_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
$\Gamma,\ x{:}S_1 \vdash T_1[x/x_1] = T_2[x/x_2] \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash s = t \in \{x{:}S \mid T\}$ $\lfloor$Ax$\rfloor$
by dependent_set_memberEquality j x'
$\Gamma \vdash s = t \in S$ $\lfloor$Ax$\rfloor$
$\Gamma \vdash T[s/x]$ $\lfloor$Ax$\rfloor$
$\Gamma,\ x'{:}S \vdash T[x'/x] \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash \{x{:}S \mid T\}$ $\lfloor$**ext** $s\rfloor$
by dependent_set_memberFormation j s x'
$\Gamma \vdash s \in S$ $\lfloor$Ax$\rfloor$
$\Gamma \vdash T[s/x]$ $\lfloor$Ax$\rfloor$
$\Gamma,\ x'{:}S \vdash T[x'/x] \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash s = t \in \{S \mid T\}$ $\lfloor$Ax$\rfloor$
by independent_set_memberEquality
$\Gamma \vdash s = t \in S$ $\lfloor$Ax$\rfloor$
$\Gamma \vdash T$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash \{S \mid T\}$ $\lfloor$**ext** $s\rfloor$
by independent_set_memberFormation
$\Gamma \vdash S$ $\lfloor$**ext** $s\rfloor$
$\Gamma \vdash T$ $\lfloor$Ax$\rfloor$

$\Gamma,\ z{:}\{x{:}S \mid T\},\ \Delta \vdash C$ $\lfloor$**ext** $(\lambda y.t)\ z\rfloor$
by setElimination i y v
$\Gamma,\ z{:}\{x{:}S \mid T\},\ y{:}S,\ [\![v]\!]{:}T[y/x],\ \Delta[y/z] \vdash C[y/z]$ $\lfloor$**ext** $t\rfloor$

Basic Inference Rule	*Corresponding Tactic* *with required arguments with optional tacticals*
dependent_setFormation S x	---
independent_setFormation	---
setEquality x	EqCD
dependent_set_memberEquality j x'	EqTypeCD
dependent_set_memberFormation j s x'	D 0
independent_set_memberEquality	EqTypeCD
independent_set_memberFormation	D 0
setElimination i y v	D i

A.3.13 Intersection

$\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** $\cap x{:}S.T\rfloor$
  by isectFormation x S
  $\Gamma \vdash S \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
  $\Gamma,\ x{:}S \vdash \mathbb{U}_j$ $\lfloor$**ext** $T\rfloor$

$\Gamma \vdash \cap x_1{:}S_1.T_1 = \cap x_2{:}S_2.T_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
  by isectEquality x
  $\Gamma \vdash S_1 = S_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
  $\Gamma,\ x{:}S_1 \vdash T_1[x/x_1] = T_2[x/x_2] \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash t_1 = t_2 \in \cap x{:}S.T$ $\lfloor$Ax$\rfloor$
  by isect_memberEquality j x'
  $\Gamma,\ x'{:}S \vdash t_1 = t_2 \in T[x'/x]$ $\lfloor$Ax$\rfloor$
  $\Gamma \vdash S \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash \cap x{:}S.T$ $\lfloor$**ext** $t\rfloor$
  by isect_memberFormation j x'
  $\Gamma,\ x'{:}S \vdash T[x'/x]$ $\lfloor$**ext** $t\rfloor$
  $\Gamma \vdash S \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash f_1 = f_2 \in T[t/x]$ $\lfloor$Ax$\rfloor$
  by isect_member_caseEquality $\cap x{:}S.T$ t
  $\Gamma \vdash f_1 = f_2 \in \cap x{:}S.T$ $\lfloor$Ax$\rfloor$
  $\Gamma \vdash t \in S$ $\lfloor$Ax$\rfloor$

$\Gamma,\ f{:}\cap x{:}S.T,\ \Delta \vdash C$ $\lfloor$**ext** $t[f, \mathsf{Ax}/y, z]\rfloor$
  by isectElimination i s y z
  $\Gamma,\ f{:}\cap x{:}S.T,\ \Delta \vdash s \in S$ $\lfloor$Ax$\rfloor$
  $\Gamma,\ f{:}\cap x{:}S.T,\ y{:}T[s/x],\ z{:}\ y{=}f \in T[s/x],\ \Delta \vdash C$ $\lfloor$**ext** $t\rfloor$

Basic Inference Rule	*Corresponding Tactic* *with required arguments*	*with optional tacticals*
isectFormation x S	---	
isectEquality x	EqCD	
isect_memberEquality j x'	EqTypeCD	
isect_memberFormation j x'	D 0	
isect_member_caseEquality $\cap x{:}S.T$ t	GenTypeCD $t = x \in S$	
isectElimination i s y z	With s (D i)	

A.3.14 Quotient Type

$\Gamma \vdash \mathbb{U}_j$ $\lfloor$**ext** $x,y : T//E\rfloor$
by quotientFormation T E x y z v v'
$\Gamma \vdash T \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
$\Gamma,\ x{:}T,\ y{:}T \vdash E \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
$\Gamma,\ x{:}T, \vdash E[x,x/x,y]$ $\lfloor$Ax$\rfloor$
$\Gamma,\ x{:}T,\ y{:}T,\ v{:}\,E[x,y/x,y] \vdash E[y,x/x,y]$ $\lfloor$Ax$\rfloor$
$\Gamma,\ x{:}T,\ y{:}T,\ z{:}T,\ v{:}\,E[x,y/x,y],\ v'{:}\,E[y,z/x,y] \vdash E[x,z/x,y]$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash x_1,y_1 : T_1//E_1 = x_2,y_2 : T_2//E_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
by quotientWeakEquality x y z v v'
$\Gamma \vdash T_1 = T_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
$\Gamma,\ x{:}T_1,\ y{:}T_1 \vdash E_1[x,y/x_1,y_1] = E_2[x,y/x_2,y_2] \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
$\Gamma,\ x{:}T_1 \vdash E_1[x,x/x_1,y_1]$ $\lfloor$Ax$\rfloor$
$\Gamma,\ x{:}T_1,\ y{:}T_1,\ v{:}\,E_1[x,y/x_1,y_1] \vdash E_1[y,x/x_1,y_1]$ $\lfloor$Ax$\rfloor$
$\Gamma,\ x{:}T_1,\ y{:}T_1,\ z{:}T_1,\ v{:}\,E_1[x,y/x_1,y_1],\ v'{:}\,E_1[y,z/x_1,y_1] \vdash E_1[x,z/x_1,y_1]$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash x_1,y_1 : T_1//E_1 = x_2,y_2 : T_2//E_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
by quotientEquality
$\Gamma \vdash x_1,y_1 : T_1//E_1 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
$\Gamma \vdash x_2,y_2 : T_2//E_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
$\Gamma \vdash T_1 = T_2 \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
$\Gamma,\ v{:}\,T_1{=}T_2{\in}\mathbb{U}_j,\ x{:}T_1,\ y{:}T_1 \vdash E_1[x,y/x_1,y_1] \Rightarrow E_2[x,y/x_2,y_2]$ $\lfloor$Ax$\rfloor$
$\Gamma,\ v{:}\,T_1{=}T_2{\in}\mathbb{U}_j,\ x{:}T_1,\ y{:}T_1 \vdash E_2[x,y/x_2,y_2] \Rightarrow E_1[x,y/x_1,y_1]$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash s = t \in x,y : T//E$ $\lfloor$Ax$\rfloor$
by quotient_memberWeakEquality j
$\Gamma \vdash x,y : T//E \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
$\Gamma \vdash s = t \in T$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash x,y : T//E$ $\lfloor$**ext** $t\rfloor$
by quotient_memberFormation j
$\Gamma \vdash x,y : T//E \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
$\Gamma \vdash T$ $\lfloor$**ext** $t\rfloor$

$\Gamma \vdash s = t \in x,y : T//E$ $\lfloor$Ax$\rfloor$
by quotient_memberEquality j
$\Gamma \vdash x,y : T//E \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
$\Gamma \vdash s \in T$ $\lfloor$Ax$\rfloor$
$\Gamma \vdash t \in T$ $\lfloor$Ax$\rfloor$
$\Gamma \vdash E[s,t/x,y]$ $\lfloor$Ax$\rfloor$

$\Gamma,\ v{:}\,s{=}t\in x,y : T//E,\ \Delta \vdash C$ $\lfloor$**ext** $u\rfloor$
by quotient_equalityElimination i j v'
$\Gamma,\ v{:}\,s{=}t\in x,y : T//E,\ [\![v']\!]{:}E[s,t/x,y],\ \Delta \vdash C$ $\lfloor$**ext** $u\rfloor$
$\Gamma,\ v{:}\,s = t\in x,y : T//E,\ \Delta \vdash E[s,t/x,y] \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma,\ z{:}\,x,y : T//E,\ \Delta \vdash s = t \in S$ $\lfloor$Ax$\rfloor$
by quotientElimination i j x' y' v
$\Gamma,\ z{:}\,x,y : T//E,\ \Delta,\ x'{:}T,\ y'{:}T \vdash E[x',y'/x,y] \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
$\Gamma,\ z{:}\,x,y : T//E,\ \Delta \vdash S \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$
$\Gamma,\ z{:}\,x,y : T//E,\ \Delta,\ x'{:}T,\ y'{:}T,\ v{:}\,E[x',y'/x,y] \vdash s[x'/z] = t[y'/z] \in S[x'/z]$ $\lfloor$Ax$\rfloor$

$\Gamma,\ z{:}\,x,y : T/\!/E\ ,\ \Delta \vdash s = t \in S$ ⌊Ax⌋

by quotientElimination_2 i j x' y' v

$\Gamma,\ z{:}\,x,y : T/\!/E\ ,\ \Delta,\ x'{:}T,\ y'{:}T \vdash E[x',y'/x,y] \in \mathbb{U}_j$ ⌊Ax⌋

$\Gamma,\ z{:}\,x,y : T/\!/E\ ,\ \Delta \vdash S \in \mathbb{U}_j$ ⌊Ax⌋

$\Gamma,\ z{:}\,x,y : T/\!/E\ ,\ x'{:}T,\ y'{:}T,\ v{:}\,E[x',y'/x,y],\ \Delta[x'/z] \vdash s[x'/z] = t[y'/z] \in S[x'/z]$ ⌊Ax⌋

Basic Inference Rule	*Corresponding Tactic* *with required arguments*	*with optional tacticals*
quotientFormation T E x y z v v'	---	
quotientWeakEquality x y z v v'	EqCD	
quotientEquality	QuotEqCD	
quotient_memberWeakEquality j	WeakEqTypeCD	
quotient_memberFormation j	D 0	
quotient_memberEquality j	EqTypeCD	
quotient_equalityElimination i j v'	EqTypeD i	
quotientElimination i j x' y' v	D i	
quotientElimination_2 i j x' y' v	QuotD i	QuotientHD' $[x';y']$ i

A.3.15 Direct Computation

$\Gamma \vdash C \quad \lfloor \textbf{ext } t \rfloor$
by `direct_computation` *tagC*
$\Gamma \vdash C{\downarrow}_{tagC} \quad \lfloor \textbf{ext } t \rfloor$

$\Gamma \vdash C \quad \lfloor \textbf{ext } t \rfloor$
by `reverse_direct_computation` *tagC*
$\Gamma, \vdash C{\uparrow}_{tagC} \quad \lfloor \textbf{ext } t \rfloor$

$\Gamma,\ z{:}\,T,\ \Delta \vdash C \quad \lfloor \textbf{ext } t \rfloor$
by `direct_computation_hypothesis` i *tagT*
$\Gamma,\ z{:}\,T{\downarrow}_{tagT},\ \Delta \vdash C \quad \lfloor \textbf{ext } t \rfloor$

$\Gamma,\ z{:}\,T,\ \Delta \vdash C \quad \lfloor \textbf{ext } t \rfloor$
by `reverse_direct_computation_hypothesis` i *tagT*
$\Gamma,\ z{:}\,T{\uparrow}_{tagT},\ \Delta \vdash C \quad \lfloor \textbf{ext } t \rfloor$

Basic Inference Rule	*Corresponding Tactic*	
	with required arguments	*with optional tacticals*
`direct_computation` *tagC*	`ComputeWithTaggedTerm` *tagC* `0`	
`reverse_direct_computation` *tagC*	`RevComputeWithTaggedTerm` *tagC* `0`	
`direct_computation_hypothesis` i *tagT*	`ComputeWithTaggedTerm` *tagT* i	
`reverse_direct_computation_hypothesis` i *tagT*	`RevComputeWithTaggedTerm` *tagT* i	

A.3.16 Miscellaneous

Γ, $x{:}T$, $\Delta \vdash T$ $\lfloor$**ext** $x\rfloor$
by hypothesis i

Γ, $x{:}T$, $\Delta \vdash C$ $\lfloor$**ext** $t\rfloor$
by thin i
Γ, $\Delta \vdash C$ $\lfloor$**ext** $t\rfloor$

Γ, $\Delta \vdash C$ $\lfloor$**ext** $(\lambda x.t)\ s\rfloor$
by cut i T x
Γ, $\Delta \vdash T$ $\lfloor$**ext** $s\rfloor$
Γ, $x{:}T$, $\Delta \vdash C$ $\lfloor$**ext** $t\rfloor$

$\Gamma \vdash T$ $\lfloor$**ext** $t\rfloor$
by introduction t
$\Gamma \vdash t \in T$ $\lfloor$Ax$\rfloor$

Γ, $z{:}T$, $\Delta \vdash C$ $\lfloor$**ext** $t\rfloor$
by hyp_replacement i S j
Γ, $z{:}S$, $\Delta \vdash C$ $\lfloor$**ext** $t\rfloor$
Γ, $z{:}T$, $\Delta \vdash T = S \in \mathbb{U}_j$ $\lfloor$Ax$\rfloor$

$\Gamma \vdash C$ $\lfloor$**ext** $t\rfloor$
by lemma "*theorem-name*"

$\Gamma \vdash t \in T$ $\lfloor$Ax$\rfloor$
by extract "*theorem-name*"

$\Gamma \vdash C$ $\lfloor$**ext** $t[\sigma]\rfloor$
by instantiate Γ' C' σ
$\Gamma' \vdash C'$ $\lfloor$**ext** $t\rfloor$

$\Gamma \vdash C$ $\lfloor$**ext** $t[y/x]\rfloor$
by rename y x
$\Gamma[x/y] \vdash C[x/y]$ $\lfloor$**ext** $t\rfloor$

$\Gamma \vdash C$ $\lfloor$Ax$\rfloor$
by because

Basic Inference Rule	*Corresponding Tactic* *with required arguments*	*with optional tacticals*
hypothesis i	Hypothesis	NthHyp i
thin i	Thin i	
cut i T x	Assert T	AssertDeclAtHyp i T x
introduction t	UseWitness t	UseWitness t
hyp_replacement i S j	SubstClause S i	
lemma "*theorem-name*"	Lemma "*theorem-name*"	Lemma "*theorem-name*"
extract "*theorem-name*"		
instantiate Γ' C' σ		
rename y x	RenameVar x i *	
because	Fiat	Fiat

*: y is the variable declared in hypothesis i.