Appendix A

The Basic Nuprl Type Theory

A.1 Syntax

Nuprl terms have the form \( \text{opid}\{p_1:F_1, \ldots, p_k:F_k\}(x_{11}, \ldots, x_{m1}, t_1; \ldots; x_{1n}, \ldots, x_{mn}, t_n) \).

\( \text{opid} \) is the operator identifier. The \( p_i:F_i \) are parameters. The \( x_{1i}, \ldots, x_{mi}, t_i \) are the bound subterms, expressing that the variables \( x_{1i}, \ldots, x_{mi} \) become bound in \( t_i \).

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 \text{tok}. Valid operators are listed in Nuprl’s \textit{operator table}, which contains the basic operators\(^2\) given in Figure A.1 as well as conservative language extensions defined by abstractions in the library.

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 \text{var}.
  
  Acceptable names are generated by the regular expression \([a–z A–Z 0–9 _ – %]^+\). The % character has a special use.

- **natural** : Natural numbers (including 0), implemented using the ML data type \text{int}.

  Acceptable numbers are generated by the regular expression \(0 + [1 – 9][0–9]^*\).

- **token** : Character strings, implemented using the ML data type \text{tok}.

  Acceptable strings can draw from any non-control characters in Nuprl’s font.

\(^1\)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 (\text{simplicrec} instead of \text{rec}) and the \text{less}-than predicate (lt instead of \text{less\_than}). A complete list of display forms for the basic terms can be found in the system’s \textit{core\_1} theory.
<table>
<thead>
<tr>
<th>Canonical</th>
<th>(Types)</th>
<th>(Members)</th>
<th>noncanonical</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{var}{x:v}()</td>
<td>\text{x}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{function}{}(&lt;S;x . T&gt;)</td>
<td>\text{lambda}{}(&lt;x . t&gt;)</td>
<td>\text{apply}{}(&lt;f . t&gt;)</td>
<td></td>
</tr>
<tr>
<td>\text{x}:S \rightarrow T, S \rightarrow T</td>
<td>\lambda x . t</td>
<td>\text{f} \rightarrow \text{t}</td>
<td></td>
</tr>
<tr>
<td>\text{product}{}(&lt;S;x . T&gt;)</td>
<td>\text{pair}{}(&lt;s ; t&gt;)</td>
<td>\text{spread}{}(&lt;e ; x , y . u&gt;)</td>
<td></td>
</tr>
<tr>
<td>\text{x}:S \times T, S \times T</td>
<td>\langle s , t \rangle</td>
<td>\text{let} \ (x , y) = \langle e \rangle \ \text{in} \ u</td>
<td></td>
</tr>
<tr>
<td>\text{union}{}(&lt;S;T&gt;)</td>
<td>\text{inl}{}(&lt;s&gt;) \text{, inr}{}(&lt;t&gt;)</td>
<td>\text{decide}{}(&lt;e ; x . u ; y . v&gt;)</td>
<td></td>
</tr>
<tr>
<td>\text{S + T}</td>
<td>\text{inl}(s) , \text{inr}(t)</td>
<td>\text{case} \ [ e ] \ \text{of} \ \text{inl}(x) \mapsto u \</td>
<td>\ \text{inr}(y) \mapsto v</td>
</tr>
<tr>
<td>\text{universe}{}(&lt;j:1&gt;)</td>
<td>\text{– All types of level} \ j –</td>
<td>\text{Ax}</td>
<td></td>
</tr>
<tr>
<td>\text{equal}{}(&lt;s ; t ; T&gt;)</td>
<td>\text{Axiom}{}()</td>
<td>\text{any}{}(&lt;e&gt;)</td>
<td></td>
</tr>
<tr>
<td>\text{s} = \text{t} \in \text{T}</td>
<td>\text{Ax}</td>
<td>\text{any}(e)</td>
<td></td>
</tr>
<tr>
<td>\text{weight}{}()</td>
<td>\text{Atom}{}()</td>
<td>\text{atom_eq}{}(&lt;u ; v ; s ; t&gt;)</td>
<td></td>
</tr>
<tr>
<td>\text{Atom}{}()</td>
<td>\text{token}{\text{token} : \text{t}}()</td>
<td>\text{if} \ \text{u} = \text{v} \ \text{then} \ s \ \text{else} \ t</td>
<td></td>
</tr>
<tr>
<td>\text{int}{}()</td>
<td>\text{natural_number}{\text{n} : \text{n}}()</td>
<td>\text{ind}{}(&lt;\text{u} ; \text{x} , \text{f} ; \text{s} ; \text{base} , \text{y} . \text{f} ; \text{y})t</td>
<td></td>
</tr>
<tr>
<td>\text{Z}</td>
<td>\text{n}</td>
<td>\text{ind}(&lt;\text{u} , \text{x} , \text{f} ; \text{s} ; \text{base} , \text{y} . \text{f} ; \text{y})t</td>
<td></td>
</tr>
<tr>
<td>\text{minus}{}(&lt;\text{natural_number}{\text{n} ; \text{n}}();</td>
<td>\text{minus}{}(&lt;\text{u} ; \text{add}{}(&lt;\text{u} ; \text{w}&gt;) , \text{sub}{}(&lt;\text{u} ; \text{v}&gt;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{– n}</td>
<td>\text{n}</td>
<td>\text{mul}{}(&lt;\text{u} ; \text{v}) , \text{div}{}(&lt;\text{u} ; \text{v}) , \text{rem}{}(&lt;\text{u} ; \text{v})</td>
<td></td>
</tr>
<tr>
<td>\text{less_than}{}(&lt;\text{u} ; \text{v})</td>
<td>\text{Axiom}{}()</td>
<td>\text{if} \ \text{u} = \text{v} \ \text{then} \ s \ \text{else} \ t</td>
<td></td>
</tr>
<tr>
<td>\text{u} &lt; \text{v}</td>
<td>\text{Ax}</td>
<td>\text{if} \ \text{u} = \text{v} \ \text{then} \ s \ \text{else} \ t</td>
<td></td>
</tr>
<tr>
<td>\text{list}{}(&lt;T&gt;)</td>
<td>\text{nil}{}() \text{, cons}{}(&lt;t ; l&gt;)</td>
<td>\text{list_ind}{}(&lt;s ; \text{base} ; x . l , \text{f}_\text{X}, \text{t})</td>
<td></td>
</tr>
<tr>
<td>\text{f} list</td>
<td>\text{t} : l</td>
<td>\text{list_ind}(&lt;s ; \text{base} ; x . l , \text{f}_\text{X}, \text{t})</td>
<td></td>
</tr>
<tr>
<td>\text{rec}{}(&lt;X ; T_X&gt;)</td>
<td>\text{– members defined by unrolling} \ T_X</td>
<td>\text{rec_ind}{}(&lt;e ; f , x . t&gt;)</td>
<td></td>
</tr>
<tr>
<td>\text{rectype} \ X = T_X</td>
<td>\text{– members of} \ S \ \text{that satisfy} \ P</td>
<td>\text{let}^* \ f(x) = \text{t} \ \text{in} \ f(e)</td>
<td></td>
</tr>
<tr>
<td>\text{set}{}(&lt;S ; x . T&gt;)</td>
<td>\text{– members of} \ S \ \text{that satisfy} \ P</td>
<td>\text{Ax}</td>
<td></td>
</tr>
<tr>
<td>\text{{x:S \mid T}, {S \mid T}}</td>
<td>\text{– members of} \ T[x] \ \text{–}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{isect}{}(&lt;S ; x . T&gt;)</td>
<td>\text{– Terms that belong to all} \ T[x] \ \text{–}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{quotient}{}(&lt;T ; x , y . E&gt;)</td>
<td>\text{– members of} \ T , \ \text{new equality} \ –</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{x , y : T} \ \text{E}</td>
<td>\text{–}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 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 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.
**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

\[ L ::= v \mid k \mid L \cdot i \mid L' \mid [L] \cdots [L] \]

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 \( L \cdot i \) is interpreted as standing for \( L + i \). \( L' \) is an abbreviation for \( L \cdot 1 \). The expression \([L] \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 8. In Figure A.1 we present the standard display of Nuprl terms immediately below their abstract 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 noncanonical expression. If an argument does not have a value or if the resulting expression is not
<table>
<thead>
<tr>
<th>Redex</th>
<th>Contractum</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda x.u$ $t$</td>
<td>$\frac{\beta}{u[t/x]}$</td>
</tr>
<tr>
<td>let $\langle x,y \rangle = [s,t]$ in $u$</td>
<td>$\frac{\beta}{u[s,t/x,y]}$</td>
</tr>
<tr>
<td>case $\text{inl}(s)$ of $\text{inl}(x) \mapsto u \mid \text{inr}(y) \mapsto v$</td>
<td>$\frac{\beta}{u[s/x]}$</td>
</tr>
<tr>
<td>case $\text{inr}(t)$ of $\text{inl}(x) \mapsto u \mid \text{inr}(y) \mapsto v$</td>
<td>$\frac{\beta}{v[t/y]}$</td>
</tr>
<tr>
<td>if $a=b$ then $s$ else $t$</td>
<td>$\frac{\beta}{s}$, if $a=b$; $t$, otherwise</td>
</tr>
<tr>
<td>$\text{ind}(0,x.f_x; s; \text{base}, y.f_y)t$</td>
<td>$\frac{\beta}{\text{base}}$</td>
</tr>
<tr>
<td>$\text{ind}(\overline{i}, x.f_x; s; \text{base}, y.f_y)t$</td>
<td>$\frac{\beta}{t[i, \text{ind}(i-1, x.f_x; s; \text{base}, y.f_y)t / x / f_x]}$, $(i&gt;0)$</td>
</tr>
<tr>
<td>$\text{ind}(-i, x.f_x; s; \text{base}, y.f_y)t$</td>
<td>$\frac{\beta}{s[i, \text{ind}(-i+1, x.f_x; s; \text{base}, y.f_y)t / y / f_y]}$, $(i&gt;0)$</td>
</tr>
<tr>
<td>$i$</td>
<td>$\frac{\beta}{\text{The negation of } i \text{ (as number)}}$</td>
</tr>
<tr>
<td>$i+j$</td>
<td>$\frac{\beta}{\text{The sum of } i \text{ and } j}$</td>
</tr>
<tr>
<td>$i-j$</td>
<td>$\frac{\beta}{\text{The difference of } i \text{ and } j}$</td>
</tr>
<tr>
<td>$i\cdot j$</td>
<td>$\frac{\beta}{\text{The product of } i \text{ and } j}$</td>
</tr>
<tr>
<td>$i : j$</td>
<td>$\frac{\beta}{0}$, if $j=0$; the integer division of $i$ and $j$, otherwise</td>
</tr>
<tr>
<td>$i \text{ rem } j$</td>
<td>$\frac{\beta}{0}$, if $j=0$; the division rest of $i$ and $j$, otherwise</td>
</tr>
<tr>
<td>if $i\leq j$ then $s$ else $t$</td>
<td>$\frac{\beta}{s}$, if $i=j$; $t$, otherwise</td>
</tr>
<tr>
<td>if $i&lt;j$ then $s$ else $t$</td>
<td>$\frac{\beta}{s}$, if $i&lt;j$; $t$, otherwise</td>
</tr>
<tr>
<td>$\text{list_ind}([\overline{\epsilon}]; \text{base}; x, l, f_{xt}.t)$</td>
<td>$\frac{\beta}{\text{base}}$</td>
</tr>
<tr>
<td>$\text{list_ind}(s::u; \text{base}; x, l, f_{xt}.t)$</td>
<td>$\frac{\beta}{t[s, u, \text{list_ind}(u; \text{base}; x, l, f_{xt}.t) / x, l, f_{xt}]}$</td>
</tr>
<tr>
<td>let $^*f(x) = t$ in $f(e)$</td>
<td>$\frac{\beta}{t[\lambda y. \text{let}^*f(x) = t \text{ in } f(y), e / f, x]}$</td>
</tr>
</tbody>
</table>

Figure A.2: Redex–Contracta Table for Nuprl's Type Theory: the principal arguments must be in the corresponding canonical form

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 Figure A.1. The corresponding redex–contracta table is given in Figure A.2.

### 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.
if there are canonical terms

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
\[
\lambda x, t_1 = \lambda x, t_2 \in x : S \rightarrow T \\
\text{if } x : S \rightarrow T \text{ Type and } t_1[s_1/x_1] = t_2[s_2/x_2] \in T[s_1/x]
\]

for all \( s_1, s_2 \) with \( s_1 = s_2 \in S \).

\( (s_1, t_1) \in \langle s_2, t_2 \rangle \in x : S \times T \)

\( \text{if } x : S \times T \text{ Type, } s_1 = s_2 \in S, \) and \( t_1 = t_2 \in T[s_1/x] \).

\( \text{inf}(s_1) = \text{inf}(s_2) \in S + T \)

\( \text{if } S + T \text{ Type and } s_1 = s_2 \in S \).

\( \text{inr}(t_1) = \text{inr}(t_2) \in S + T \)

\( \text{if } S + T \text{ Type and } t_1 = t_2 \in T \).

\( \text{Ax} = \text{Ax} \in s + t \in T \)

\( \text{if } s = t \in T \).

\( s = t \in \text{void} \)

\( \text{never holds!} \)

\( \text{"token" = "token" } \in \text{Atom} \)

\( i = i \in \mathbb{Z} \)

\( \text{Ax} = \text{Ax} \in s < t \)

\( \text{if } s \rightarrow i \text{ and } t \rightarrow j \text{ for some integers } i, j \text{ with } i < j \)

\( [\] = [\] \in T \text{ list} \)

\( T \text{ Type} \)

\( t_1 : t_2 \in T \text{ list} \)

\( \text{if } T \text{ Type, } t_1 = t_2 \in T, \) and \( l_1 = l_2 \in T \text{ list} \)

\( s = t \in \text{rectype } X = T_X \)

\( \text{rectype } X = T_X \text{ Type and } s = t \in T_X[\text{rectype } X = T_X/X] \)

\( s = t \in \{ x : S \mid T \} \)

\( \{ x : S \mid T \} \text{ Type, } s = t \in S, \) and there is some term \( p \in T[s/x] \).

\( t_1 = t_2 \in \exists x : S \mid T \)

\( \exists x : S \mid T \text{ Type and } t_1 = t_2 \in T[s/x] \) for all \( s \in S \).

\( s = t \in x, y : T // E \)

\( x, y : T // E \text{ Type, } s = t, \) \( t \in T, \)

\( \text{and there is some term } p \in E[s, t, x, y] \).

\( x_1 : S_1 \rightarrow T_1 = x_2 : S_2 \rightarrow T_2 \in U \)

\( \text{if } S_1 = S_2 \in U \) \( s = t \in \mathbb{U} \) \( x_1, x_2 \text{ with } s_1 = s_2 \in S \)

\( T = S_2 \rightarrow T_2 \in U \)

\( \text{if } T = x_2 : S_2 \rightarrow T_2 \in U \) \( s = t \in \mathbb{U} \) \( x_1, x_2 \text{ with } s_1 = s_2 \in S \)

\( S_1 \rightarrow T_1 \in U \)

\( \text{if } x_1 : S_1 \rightarrow T_1 = T \in U \) \( s = t \in \mathbb{U} \) \( x_1, x_2 \text{ with } s_1 = s_2 \in S \)

\( x_1 : S_1 \times T_1 = x_2 : S_2 \times T_2 \in U \)

\( \text{if } S_1 = S_2 \in U \) \( s = t \in \mathbb{U} \) \( x_1, x_2 \text{ with } s_1 = s_2 \in S \)

\( T = S_2 \times T_2 \in U \)

\( \text{if } T = x_2 : S_2 \times T_2 \in U \) \( s = t \in \mathbb{U} \) \( x_1, x_2 \text{ with } s_1 = s_2 \in S \)

\( S_1 \times T_1 \in U \)

\( \text{if } x_1 : S_1 \times T_1 = T \in U \) \( s = t \in \mathbb{U} \) \( x_1, x_2 \text{ with } s_1 = s_2 \in S \)

\( S_1 + T_1 = S_2 + T_2 \in U \)

\( \text{if } S_1 = S_2 \in U \) \( s = t \in \mathbb{U} \) \( x_1, x_2 \text{ with } s_1 = s_2 \in S \)

\( i_1 \times j_1 = i_2 \times j_2 \in U \)

\( \text{if } i_1 = i_2 \in \mathbb{Z} \) \( s = t \in \mathbb{U} \) \( x_1, x_2 \text{ with } s_1 = s_2 \in S \)

\( T_1 : T_2 \in U \)

\( T_1 = T_2 \in U \)

\( \text{rectype } X_1 = T_X_1 = \text{rectype } X_2 = T_X_2 \)

\( \text{for all } X \in \mathbb{U} \)

\( \{ x_1 : S_1 \mid T_1 \} \) \( \{ x_2 : S_2 \mid T_2 \} \in \mathbb{U} \)

\( \text{there are terms } p_1, p_2 \text{ and a variable } x \text{ which occurs neither in } T_1 \text{ nor in } T_2 \) such that \( p_1 \in \forall x : S, T_1[x/x_1] = T_2[x/x_2] \) and \( p_2 \in \forall x : S, T_2[x/x_2] = T_1[x/x_1] \),

\( T = \{ S_1 \mid T_1 \} \in \mathbb{U} \)

\( \text{if } T = \{ x_2 : S_2 \mid T_2 \} \in \mathbb{U} \) \( s = t \in \mathbb{U} \) \( x_1, x_2 \text{ with } s_1 = s_2 \in S \)

\( \{ S_1 \mid T_1 \} = T \in \mathbb{U} \)

\( \text{if } \{ x_1 : S_1 \mid T_1 \} = T \in \mathbb{U} \) \( s = t \in \mathbb{U} \) \( x_1, x_2 \text{ with } s_1 = s_2 \in S \)

\( \exists x : S ; T = \exists x : S ; T_2 \in \mathbb{U} \)

\( \text{if } S_1 = S_2 \in U \) \( s = t \in \mathbb{U} \) \( x_1, x_2 \text{ with } s_1 = s_2 \in S \)

\( x, y : T // E \rightarrow x_2, y : T // E \)

\( \text{if } T = T_2 \in U \) \( s = t \in \mathbb{U} \) \( x_1, x_2 \text{ with } s_1 = s_2 \in S \)

\( \text{and there are terms } p_3, p_4, r, s, t \text{ and variables } x, y, z \text{ which occur neither in } E_2 \text{ nor in } E_1 \) such that \( p_3 \in \forall x : T_1, \forall y : T_1, E_1[x, y/x_1, y_1] \Rightarrow E_2[x, y/x_2, y_2], \)

\( p_4 \in \forall x : T_1, \forall y : T_1, E_2[x, y/x_2, y_2] \Rightarrow E_1[x, y/x_1, y_1], \)

\( r \in \forall x : T_1, E_1[x, x/x_1, y_1], \)

\( s \in \forall x : T_1, E_1[x, y/x_1, y_1] \Rightarrow E_1[y, x/x_1, y_1], \) and \( t \in \forall x : T_1, E_1[x, y/x_1, y_1] \Rightarrow E_1[y, z/x_1, y_1] \Rightarrow E_1[x, z/x_1, y_1], \)

\[ E_1[x, y/x_1, y_1] \Rightarrow E_1[y, z/x_1, y_1] \Rightarrow E_1[x, z/x_1, y_1] \]

<table>
<thead>
<tr>
<th>Figure A.4: Member semantics table for Nuprl</th>
</tr>
</thead>
</table>
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 parameters such as

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

Most of the elementary inference rules are subsumed by the one-step decomposition tactics
\textsc{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 \texttt{New, At, With, or Using}. 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 (with all required tacticals) and a maximal form with all tacticals
that may have an effect on the execution of the tactic. 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
\(a = a \in T\). \(-P\) stands for \(P \rightarrow \text{void}\), \(s \neq t\) for \(\neg(s = t \in \mathbb{Z})\), and \(s \leq t\) for \(\neg(t < s)\).
A.3.1 Functions

\[ \Gamma \vdash U_j \text{ } \text{ext } x : S \rightarrow T_j \]

by \ dependent\_functionFormation \ x \ S

\[ \Gamma \vdash S \in U_j \ [\alpha_j] \]
\[ \Gamma, x : S \vdash U_j \ [\text{ext } T] \]

\[ \Gamma \vdash x_1 : S_1 \rightarrow T_1 = x_2 : S_2 \rightarrow T_2 \in U_j \ [\alpha_j] \]
by \ functionEquality \ x

\[ \Gamma \vdash S_1 = S_2 \in U_j \ [\alpha_j] \]
\[ \Gamma, x : S_1 \vdash T_1[x/x_1] = T_2[x/x_2] \in U_j \ [\alpha_j] \]

\[ \Gamma \vdash \lambda x_1, t_1 = \lambda x_2, t_2 \in x : S \rightarrow T \ [\alpha_j] \]
by \ lambdaFormation \ j \ x'

\[ \Gamma, x' : S \vdash t_1[x'/x_1] = t_2[x'/x_2] \in T[x'/x] \ [\alpha_j] \]
\[ \Gamma \vdash S \in U_j \ [\alpha_j] \]

\[ \Gamma \vdash f, t_1 = f_2 t_2 \in T[t_1/x] \ [\alpha_j] \]
by \ applyEquality \ x : S \rightarrow T

\[ \Gamma \vdash f_1 = f_2 \in x : S \rightarrow T \ [\alpha_j] \]
\[ \Gamma \vdash t_1 = t_2 \in S \ [\alpha_j] \]

\[ \Gamma, f : x : S \rightarrow T, \ \Delta \vdash C \ [\text{ext } t[f s, y, z]] \]
by \ dependent\_functionElimination \ i \ s \ y \ z

\[ \Gamma, f : x : S \rightarrow T, \ \Delta \vdash s \in S \ [\alpha_j] \]
\[ \Gamma, f : x : S \rightarrow T, \ y : T[s/x], z : y = f s \in T[s/x], \ \Delta \vdash C \ [\text{ext } t] \]

\[ \Gamma \vdash (\lambda x. t) s = t_2 \in T \ [\alpha_j] \]
by \ applyReduce

\[ \Gamma \vdash t[s/x] = t_2 \in T \ [\alpha_j] \]

\[ \Gamma \vdash f_1 = f_2 \in x : S \rightarrow T \ [\text{ext } t] \]
by \ functionExtensionality \ j \ x_1 : S_1 \rightarrow T_1, x_2 : S_2 \rightarrow T_2 \ x'

\[ \Gamma, x' : S \vdash f_1 x' = f_2 x' \in T[x'/x] \ [\text{ext } t] \]
\[ \Gamma \vdash S \in U_j \ [\alpha_j] \]
\[ \Gamma \vdash f_1 \in x : S_1 \rightarrow T_1 \ [\alpha_j] \]
\[ \Gamma \vdash f_2 \in x : S_2 \rightarrow T_2 \ [\alpha_j] \]

<table>
<thead>
<tr>
<th>Rule</th>
<th>Minimal Tactic</th>
<th>Tactic with all tacticals</th>
</tr>
</thead>
<tbody>
<tr>
<td>dependent_functionFormation \ x \ S</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>independent_functionFormation</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>functionEquality \ x</td>
<td>EqCD</td>
<td>New [x] EqCD</td>
</tr>
<tr>
<td>independent_functionEquality</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>lambdaEquality \ j \ x'</td>
<td>EqCD</td>
<td>At U_j EqCD</td>
</tr>
<tr>
<td>lambdaFormation \ j \ x'</td>
<td>D 0</td>
<td>---</td>
</tr>
<tr>
<td>applyEquality \ x : S \rightarrow T</td>
<td>EqCD</td>
<td>With x : S \rightarrow T EqCD</td>
</tr>
<tr>
<td>independent_functionElimination \ i \ y</td>
<td>D i</td>
<td>---</td>
</tr>
<tr>
<td>dependent_functionElimination \ i \ s \ y \ z</td>
<td>D i</td>
<td>---</td>
</tr>
<tr>
<td>applyReduce</td>
<td>ReduceEquands 0</td>
<td>ReduceAtAddr [2] 0</td>
</tr>
<tr>
<td>functionExtensionality \ j \ x_1 : S_1 \rightarrow T_1, x_2 : S_2 \rightarrow T_2 \ x'</td>
<td>EqExtWith</td>
<td>Ext</td>
</tr>
</tbody>
</table>
A.3.2 Products

\[\Gamma \vdash \mathbf{U}_j^{\textit{ext}} S \times T\]

by \texttt{dependent_productFormation} \( x S \)
\[\Gamma \vdash S \in \mathbf{U}_j^{\textit{ext}} \]
\[\Gamma, x : S \vdash \mathbf{T}_j^{\textit{ext}} T\]

\[\Gamma \vdash x : S \times T_1 = x : S_2 \times T_2 \in \mathbf{U}_j^{\textit{ext}}\]

by \texttt{productEquality} \( x' \)
\[\Gamma \vdash S_1 = S_2 \in \mathbf{U}_j^{\textit{ext}}\]
\[\Gamma, x' : S_1 \vdash T_1[x'/x_1] = T_2[x'/x_2] \in \mathbf{U}_j^{\textit{ext}}\]

\[\Gamma \vdash \langle s_1, t_1 \rangle = \langle s_2, t_2 \rangle \in S \times T^{\textit{ext}}\]

by \texttt{dependent_pairEquality} \( j \)
\[\Gamma \vdash s_1 = s_2 \in S^{\textit{ext}}\]
\[\Gamma \vdash t_1 = t_2 \in T^{\textit{ext}}[s_1/x]\]
\[\Gamma, x' : S \vdash T[x'/x] \in \mathbf{U}_j^{\textit{ext}}\]

\[\Gamma \vdash \text{let } \langle x_1, y_1 \rangle = e_1 \text{ in } t_1 = \text{let } \langle x_2, y_2 \rangle = e_2 \text{ in } t_2 \in C[e_1/z]^{\textit{ext}}\]

by \texttt{spreadEquality} \( z \)
\[\Gamma \vdash e_1 = e_2 \in x : S \times T^{\textit{ext}}\]
\[\Gamma, x : S \vdash T[s/x], y : e = (s, t) \in x : S \times T^{\textit{ext}} \vdash t_1[s, t/x_1, y_1] = t_2[s, t/x_2, y_2] \in C[(s, t)/z]^{\textit{ext}}\]

\[\Gamma, z : x : S \times T, \Delta \vdash C^{\textit{ext}} \text{ let } (s, t) = z \text{ in } u\]

by \texttt{productElimination} \( i \)
\[\Gamma, z : x : S \times T, s : S, t : T[s/x] \vdash C[(s, t)/z]^{\textit{ext}} \text{ in } u\]

\[\Gamma \vdash \text{let } \langle x, y \rangle = (s, t) \text{ in } u = t_2 \in T^{\textit{ext}}\]

by \texttt{spreadReduce}
\[\Gamma \vdash u[s, t/x, y] = t_2 \in T^{\textit{ext}}\]

<table>
<thead>
<tr>
<th>Rule</th>
<th>Minimal Tactic</th>
<th>Tactic with all tacticals</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{dependent_productFormation} ( x S )</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>\texttt{independent_productFormation}</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>\texttt{productEquality} ( x' )</td>
<td>EqCD</td>
<td>EqCD</td>
</tr>
<tr>
<td>\texttt{independent_productEquality}</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>\texttt{dependent_pairEquality} ( j x' )</td>
<td>EqCD</td>
<td>EqCD</td>
</tr>
<tr>
<td>\texttt{dependent_pairEquality2} ( j x' )</td>
<td>EqCD</td>
<td>EqCD</td>
</tr>
<tr>
<td>\texttt{dependent_pairFormation} ( j s x' ) \texttt{with } s \texttt{ (D 0)}</td>
<td>\texttt{At } U_j^{\textit{ext}} \texttt{ (with } s \texttt{ (New [x'] (D 0)))}</td>
<td></td>
</tr>
<tr>
<td>\texttt{independent_pairEquality}</td>
<td>EqCD</td>
<td>EqCD</td>
</tr>
<tr>
<td>\texttt{independent_pairFormation}</td>
<td>D 0</td>
<td>D 0</td>
</tr>
<tr>
<td>\texttt{spreadEquality} ( z C x : S \times T s t y )</td>
<td>EqCD</td>
<td>EqCD</td>
</tr>
<tr>
<td>\texttt{productElimination} ( i s t )</td>
<td>D ( i )</td>
<td>D ( i )</td>
</tr>
<tr>
<td>\texttt{spreadReduce}</td>
<td>\texttt{ReduceEquands 0}</td>
<td>\texttt{ReduceAtAddr [2] 0}</td>
</tr>
</tbody>
</table>
A.3.3 Disjoint Union

$$\Gamma \vdash \bigcup_j \text{ext } S + T$$

by unionFormation

$$\Gamma \vdash \bigcup_j \text{ext } S$$
$$\Gamma \vdash \bigcup_j \text{ext } T$$

$$\Gamma \vdash \text{inl}(s_1) = \text{inl}(s_2) \in S + T \ \vdash \ A_{\mathcal{J}}$$
by inlEquality

$$\Gamma \vdash s_1 = s_2 \in S \ \vdash \ A_{\mathcal{J}}$$
$$\Gamma \vdash T \in \bigcup_j \ \vdash \ A_{\mathcal{J}}$$

$$\Gamma \vdash \text{inr}(t_1) = \text{inr}(t_2) \in S + T \ \vdash \ A_{\mathcal{J}}$$
by inrEquality

$$\Gamma \vdash t_1 = t_2 \in T \ \vdash \ A_{\mathcal{J}}$$
$$\Gamma \vdash S \in \bigcup_j \ \vdash \ A_{\mathcal{J}}$$

$$\Gamma \vdash \text{case } e_1 \text{ of } \text{inl}(x) \mapsto u_1 \ | \ \text{inr}(y) \mapsto v_1 = \text{case } e_2 \text{ of } \text{inl}(x) \mapsto u_2 \ | \ \text{inr}(y) \mapsto v_2 \in C[e_1/z] \ \vdash \ A_{\mathcal{J}}$$
by decideEquality $z \ C \ S + T \ s \ t \ y$

$$\Gamma \vdash e_1 = e_2 \in S + T \ \vdash \ A_{\mathcal{J}}$$
$$\Gamma, s:S, y:e_1=\text{inl}(s) \in S + T \vdash u_1[s/x_1] = u_2[s/x_2] \in C[\text{inl}(s)/z] \ \vdash \ A_{\mathcal{J}}$$
$$\Gamma, t:T, y:e_1=\text{inr}(t) \in S + T \vdash v_1[t/y_1] = v_2[t/y_2] \in C[\text{inr}(t)/z] \ \vdash \ A_{\mathcal{J}}$$

$$\Gamma, z:S + T, \Delta \vdash C \ \text{ext } z \ \text{case } z \text{ of } \text{inl}(x) \mapsto u \ | \ \text{inr}(y) \mapsto v \ \vdash \ A_{\mathcal{J}}$$
by unionElimination $i \ x \ y$

$$\Gamma, z:S + T, x:S, \Delta[\text{inl}(x)/z] \vdash C[\text{inl}(x)/z] \ \text{ext } u$$
$$\Gamma, z:S + T, y:T, \Delta[\text{inr}(y)/z] \vdash C[\text{inr}(y)/z] \ \text{ext } v$$

$$\Gamma \vdash \text{case } \text{inl}(s) \text{ of } \text{inl}(x) \mapsto u \ | \ \text{inr}(y) \mapsto v \ \vdash t_2 \in T \ \vdash \ A_{\mathcal{J}}$$
by decideReduceLeft

$$\Gamma \vdash u[s/x] = t_2 \in T \ \vdash \ A_{\mathcal{J}}$$

$$\Gamma \vdash \text{case } \text{inr}(t) \text{ of } \text{inl}(x) \mapsto u \ | \ \text{inr}(y) \mapsto v \ \vdash t_2 \in T \ \vdash \ A_{\mathcal{J}}$$
by decideReduceRight

$$\Gamma \vdash v[t/y] = t_2 \in T \ \vdash \ A_{\mathcal{J}}$$

<table>
<thead>
<tr>
<th>Rule</th>
<th>Minimal Tactic</th>
<th>Tactic with all tacticals</th>
</tr>
</thead>
<tbody>
<tr>
<td>unionFormation</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>unionEquality</td>
<td>EqCD</td>
<td></td>
</tr>
<tr>
<td>inlEquality $j$</td>
<td>EqCD</td>
<td></td>
</tr>
<tr>
<td>inlFormation $j$</td>
<td>Sel 1 D 0</td>
<td></td>
</tr>
<tr>
<td>inrEquality $j$</td>
<td>EqCD</td>
<td></td>
</tr>
<tr>
<td>inrFormation $j$</td>
<td>Sel 2 D 0</td>
<td></td>
</tr>
<tr>
<td>decideEquality $z \ C \ S + T \ s \ t \ y$</td>
<td>EqCD</td>
<td>Using $[z,C]$ (With $S + T$ (New $s \ t \ y$ EqCD)))</td>
</tr>
<tr>
<td>unionElimination $i \ x \ y$</td>
<td>D i</td>
<td></td>
</tr>
<tr>
<td>decideReduceLeft</td>
<td>ReduceEquands 0</td>
<td>ReduceAtAddr [2] 0</td>
</tr>
<tr>
<td>decideReduceRight</td>
<td>ReduceEquands 0</td>
<td>ReduceAtAddr [2] 0</td>
</tr>
</tbody>
</table>
A.3.4 Universes

\[ U_k \text{ ext} U_j \]
\[ \Gamma \vdash U_j = U_j \in U_k |\kappa| \]
\[ \text{by universeFormation } j \]

\[ T \in U_k |\kappa| \]
\[ \Gamma \vdash T \in U_j |\kappa| \]
\[ \text{by cumulativity } j \]

\[ \Gamma \vdash \text{ proviso } j < k \]

Rule | Minimal Tactic | Tactic with all tacticals
--- | --- | ---
universeFormation \( j \) | D 0 | 
universeEquality | EqCD | 
cumulativity \( j \) | Cumulativity \( j \) | 

A.3.5 Equality

\[ s = t \in T \]
\[ \Gamma \vdash s = t \in T_j \]
\[ \text{by equalityFormation } T \]
\[ \Gamma \vdash T \in U_j |\kappa| \]
\[ \Gamma \vdash \text{ ext } s \]
\[ \Gamma \vdash \text{ ext } t \]

\[ \Gamma \vdash A\!x = A\!x \in s = t \in T |\kappa| \]
\[ \text{by axiomEquality} \]
\[ \Gamma \vdash s = t \in T |\kappa| \]

\[ \Delta \vdash C \text{ ext } u_j \]
\[ \Gamma, z : s = t \in T, \Delta \vdash C \text{ ext } u_j \]
\[ \text{by equalityElimination } i \]

\[ \Delta, x : T, \Delta[\kappa/x] \vdash C[\kappa/x] \text{ ext } u_j \]
\[ \Gamma, x : T, \Delta \vdash x = x \in T |\kappa| \]
\[ \text{by hypothesisEquality } i \]

\[ \Gamma \vdash C[s/x] |\kappa| \]
\[ \text{by substitution } j \]
\[ \Gamma \vdash s = t \in T |\kappa| \]
\[ \Gamma \vdash C[t/x] |\kappa| \]
\[ \Gamma, x : T \vdash C \in U_j |\kappa| \]

\[ \Gamma \vdash s = t \in T |\kappa| \]
\[ \text{by equality} \]

\[ \text{Decision procedure for elementary equalities} \]

Rule | Minimal Tactic | Tactic with all tacticals
--- | --- | ---
equalityFormation \( T \) | --- | 
equalityEquality | EqCD | 
axiomEquality | EqCD | 
equalityElimination \( i \) | D \( i \) | 
hypothesisEquality \( i \) | Declaration \( i \) | 
substitution \( j \) \( s = t \in T \) \( x \) \( C \) | Subst \( s = t \in T \) \( 0 \) \( \text{At } j \) \( \text{(BasicSubst } s = t \in T \text{ x } C) \)
equality | Eq | 

A.3.6 Void

\[ \Gamma \vdash \bigcup_j \text{void} \]

by voidFormation

\[ \Gamma \vdash \text{void} = \text{void} \in \bigcup_j \mathcal{A}_j \]

by voidEquality

\[ \Gamma \vdash \text{any}(s) = \text{any}(t) \in T \mathcal{A}_j \]

by anyEquality

\[ \Gamma, \Delta \vdash C \text{ext any}(z) \]

by voidElimination i

\[ \Gamma \vdash s = t \in \text{void} \]

Rule | Minimal Tactic | Tactic with all tacticals
--- | --- | ---
voidFormation | --- | ---
voidEquality | EqCD | EqCD
anyEquality | EqCD | EqCD
voidElimination i | D i | D i

A.3.7 Atom

\[ \Gamma \vdash \bigcup_j \text{Atom} \]

by atomFormation

\[ \Gamma \vdash \text{Atom} = \text{Atom} \in \bigcup_j \mathcal{A}_j \]

by atomEquality

\[ \Gamma \vdash "\text{token}" = "\text{token}" \in \text{Atom} \mathcal{A}_j \]

by tokenEquality

\[ \Gamma \vdash \text{if } u_1 = v_1 \text{ then } s_1 \text{ else } t_1 = \text{if } u_2 = v_2 \text{ then } s_2 \text{ else } t_2 \in T \mathcal{A}_j \]

by atom_eqEquality \( \triangledown \)

\( \Gamma \vdash u_1 = u_2 \in \text{Atom} \mathcal{A}_j \)

\( \Gamma \vdash v_1 = v_2 \in \text{Atom} \mathcal{A}_j \)

\( \Gamma, v: u_1 = v_1 \in \text{Atom} \vdash s_1 = s_2 \in T \mathcal{A}_j \)

\( \Gamma, v: u_1 = v_1 \in \text{Atom} \vdash t_1 = t_2 \in T \mathcal{A}_j \)

\[ \Gamma \vdash \text{if } u = v \text{ then } s \text{ else } t = t_2 \in T \mathcal{A}_j \]

by atom_eqReduceFalse

\( \Gamma \vdash s = t_2 \in T \mathcal{A}_j \)

\( \Gamma \vdash t = t_2 \in T \mathcal{A}_j \)

\( \Gamma \vdash u = v \in \text{Atom} \mathcal{A}_j \)

\( \Gamma \vdash \neg(u = v \in \text{Atom}) \mathcal{A}_j \)

Rule | Minimal Tactic | Tactic with all tacticals
--- | --- | ---
atomFormation | --- | ---
atomEquality | EqCD | EqCD
tokenEquality | EqCD | EqCD
tokenFormation "token" | D 0 | D 0
atom_eqEquality \( \triangledown \) | EqCD | EqCD
atom_eqReduceTrue | PrimReduceFirstEquand 'true' | PrimReduceEquands 'true' [1]
atom_eqReduceFalse | PrimReduceFirstEquand 'false' | PrimReduceEquands 'false' [1]
A.3.8 Integers

\[ \Gamma \vdash \bigcup_j \text{ext } \mathbb{Z}_j \]
by \text{intFormation}

\[ \Gamma \vdash n = n \in \mathbb{Z} \quad [\text{intEquality}] \]
by \text{natural_numberEquality}

\[ \Gamma \vdash -s_1 = -s_2 \in \mathbb{Z} \quad [\text{minusEquality}] \]
by \text{minusEquality}

\[ \Gamma \vdash s_1 \cdot s_2 = t_1 \cdot t_2 \in \mathbb{Z} \quad [\text{addEquality}] \]
by \text{addEquality}

\[ \Gamma \vdash s_1 = s_2 \in \mathbb{Z} \quad [\text{multiplyEquality}] \]
by \text{multiplyEquality}

\[ \Gamma \vdash s_1 = s_2 \in \mathbb{Z} \quad [\text{divideEquality}] \]
by \text{divideEquality}

\[ \Gamma \vdash s_1 \ mod s_2 = t_1 \ mod t_2 \in \mathbb{Z} \quad [\text{remainderEquality}] \]
by \text{remainderEquality}

\[ \Gamma \vdash 0 \leq s \mod t < t \quad [\text{remainderBounds1}] \]
by \text{remainderBounds1}

\[ \Gamma \vdash 0 \leq s \quad [\text{remainderBounds2}] \]
by \text{remainderBounds2}

\[ \Gamma \vdash 0 < t \quad [\text{remainderBounds3}] \]
by \text{remainderBounds3}

\[ \Gamma \vdash 0 \leq s \mod t < t \quad [\text{remainderBounds4}] \]
by \text{remainderBounds4}

\[ \Gamma \vdash s = (s \div t) \cdot t + (s \mod t) \quad [\text{divideRemainderSum}] \]
by \text{divideRemainderSum}

\[ \Gamma \vdash C \ ext t_j \]
by \text{arith} \ j

\[ \Gamma \vdash s_i \in \mathbb{Z} \quad [\text{arithmetic}] \]
by \text{arithmetic}

\[ \Gamma \vdash \mathbb{Z} = \bigcup_j \text{ext } \mathbb{Z}_j \quad [\text{intEquality}] \]
by \text{natural_numberEquality}

\[ \Gamma \vdash \mathbb{Z} \ ext n_j \]
by \text{natural_numberFormation} \ n

Decision procedure for elementary arithmetic

– subgoals for all non-arithmetical expressions \( s_i \) in \( C \) –
\[
\Gamma \vdash \text{ind}(u_i; x, f_x, s_i; \text{base}, y, f_y, t_i) = \text{ind}(u_i; x, f_x, s_i; \text{base}, y, f_y, t_i) \in T[u_1/z] \quad \text{by indEquality}
\]
\[
\Gamma \vdash (x, f_x, v) \in T \quad \text{by indEquality}
\]
\[
\Gamma, x:Z, v: x0, f_x: T[(x+1)/z] \vdash s_1[x, f_x/x_1, f_x] = s_2[x, f_x/x_2, f_x] \in T[x/z] \quad \text{by base}
\]
\[
\Gamma, x:Z, v: x0, f_x: T[(x-1)/z] \vdash t_1[x, f_x/y_1, f_y] = t_2[x, f_x/y_2, f_y] \in T[x/z]
\]
\[
\Gamma, z:Z, \Delta \vdash (x, f_x, s)[Ax/v]; \text{base}; x, f_x.t[ Ax/v)] \quad \text{by intElimination}
\]
\[
\Gamma, z:Z, \Delta, x:Z, v: x0, f_x: C[(x+1)/z] \vdash C[x/z] \quad \text{by ext s}
\]
\[
\Gamma, z:Z, \Delta \vdash C[0/z] \quad \text{by ext base}
\]
\[
\Gamma, z:Z, \Delta, x:Z, v: x0, f_x: C[(x-1)/z] \vdash C[x/z] \quad \text{by ext t}
\]
\[
\Gamma \vdash \text{ind}(i; x, f_x, s; \text{base}, y, f_y, t) = t_2 \in T \quad \text{by indRedcDown}
\]
\[
\Gamma \vdash i, \text{ind}(i+1; x, f_x, s; \text{base}, y, f_y, t)/x, f_x = t_2 \in T \quad \text{by indRedcUp}
\]
\[
\Gamma \vdash i < 0 \quad \text{by indRedcBase}
\]
\[
\Gamma \vdash \text{base} = t_2 \in T \quad \text{by int_eqEquality}
\]
\[
\Gamma \vdash \text{u} = v \quad \text{if } u = v \text{ then } s \text{ else } t \quad \text{by int_eqReduceTrue}
\]
\[
\Gamma \vdash \text{u} = v \quad \text{by int_eqReduceFalse}
\]
\[
\Gamma \vdash u = v \quad \text{by lessEquality}
\]
\[
\Gamma \vdash u < v \quad \text{by lessReduceTrue}
\]
\[
\Gamma \vdash u < v \quad \text{by lessReduceFalse}
\]

164
A.3.9  Less-Than Proposition

\[ \Gamma \vdash s_1 < t_1 = s_2 < t_2 \in U_j \]  
\text{by less_thanEquality}  
\[ \Gamma \vdash s_1 = s_2 \in Z \]  
\text{by less_thanFormation}  
\[ \Gamma \vdash t_1 = t_2 \in Z \]  
\text{by less_thanFormation}  
\[ \Gamma \vdash \text{Ax} \in s < t \]  
\text{by less_thanMember}  
\[ \Gamma \vdash s < t \]  
\text{by less_thanMember}  

<table>
<thead>
<tr>
<th>Rule</th>
<th>Minimal Tactic</th>
<th>Tactic with all tacticals</th>
</tr>
</thead>
<tbody>
<tr>
<td>less_thanEquality</td>
<td>EqCD</td>
<td></td>
</tr>
<tr>
<td>less_thanFormation</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>less_thanMember</td>
<td>EqCD</td>
<td></td>
</tr>
</tbody>
</table>
### A.3.10 Lists

\[
\Gamma \vdash \mathsf{U}_j \cdot \mathsf{ext} T list_j \quad \Gamma \vdash T_j \cdot \text{list} = T_2 \cdot \text{list} \in \mathsf{U}_j \quad [\mathsf{\alpha}_j]
\]

**by listFormation**

\[
\Gamma \vdash \mathsf{U}_j \cdot \mathsf{ext} T
\]

**by listEquality**

\[
\Gamma \vdash T_1 = T_2 \in \mathsf{U}_j \quad [\mathsf{\alpha}_j]
\]

\[
\Gamma \vdash \emptyset = \emptyset \in \mathsf{T} \cdot \text{list} \quad [\mathsf{\alpha}_j]
\]

**by nilEquality \( j \)**

\[
\Gamma \vdash T \in \mathsf{U}_j \quad [\mathsf{\alpha}_j]
\]

\[
\Gamma \vdash t_1 :: l_1 = t_2 :: l_2 \in \mathsf{T} \cdot \text{list} \quad [\mathsf{\alpha}_j]
\]

**by consFormation**

\[
\Gamma \vdash t_1 = t_2 \in \mathsf{T} \quad [\mathsf{\alpha}_j]
\]

\[
\Gamma \vdash l_1 = l_2 \in \mathsf{T} \cdot \text{list} \quad [\mathsf{\alpha}_j]
\]

\[
\Gamma \vdash \mathsf{list} \cdot \text{ind} (s_1; \text{base}_1; x_1, l_1, f_{x_1} t_1) = \mathsf{list} \cdot \text{ind} (s_2; \text{base}_2; x_2, l_2, f_{x_2} t_2) \in \mathsf{T}[s_1/z] \quad [\mathsf{\alpha}_j]
\]

**by list\_indEquality \( z \cdot T \cdot \text{list} \cdot x \cdot l \cdot f_{x} t \)**

\[
\Gamma \vdash s_1 = s_2 \in \mathsf{S} \cdot \text{list} \quad [\mathsf{\alpha}_j]
\]

\[
\Gamma \vdash \text{base}_1 = \text{base}_2 \in \mathsf{T}[\emptyset/z] \quad [\mathsf{\alpha}_j]
\]

\[
\Gamma, x: \mathsf{S}, l: \mathsf{S} \cdot \text{list}, f_{x_1} t_1: \mathsf{T}[l/z] \vdash t_1[x_1, l, f_{x_1} x_1, l_1, f_{x_1} t_1] = t_2[x, l, f_{x_2} x_2, l_2, f_{x_2} t_2] \in \mathsf{T}[x::l/z] \quad [\mathsf{\alpha}_j]
\]

\[
\Gamma, z: \mathsf{T} \cdot \text{list}, \Delta \vdash C \cdot \mathsf{ext} \cdot \text{list} \cdot \text{ind} (z; \text{base}; x, l, f_{x} t)
\]

**by listElimination \( i \cdot f_{x} x \cdot l \)**

\[
\Gamma, z: \mathsf{T} \cdot \text{list}, \Delta \vdash C[\emptyset/z] \cdot \mathsf{ext} \cdot \text{base}
\]

\[
\Gamma, z: \mathsf{T} \cdot \text{list}, \Delta, x: \mathsf{T}, l: \mathsf{T} \cdot \text{list}, f_{x_1} t_1: C[l/z] \vdash C[x::l/z] \cdot \mathsf{ext} \cdot t_1
\]

\[
\Gamma \vdash \mathsf{list} \cdot \text{ind} (\emptyset; \text{base}; x, l, f_{x} t_1) = t_2 \in \mathsf{T} \quad [\mathsf{\alpha}_j]
\]

**by list\_indReduceBase**

\[
\Gamma \vdash \text{base} = t_2 \in \mathsf{T} \quad [\mathsf{\alpha}_j]
\]

\[
\Gamma \vdash \mathsf{list} \cdot \text{ind} (s; u; \text{base}; x, l, f_{x} t_1) = t_2 \in \mathsf{T} \quad [\mathsf{\alpha}_j]
\]

**by list\_indReduceUp**

\[
\Gamma \vdash t[s, u, \mathsf{list} \cdot \text{ind} (u; \text{base}; x, l, f_{x} t_1)/x, l, f_{x} t_1] = t_2 \in \mathsf{T} \quad [\mathsf{\alpha}_j]
\]

<table>
<thead>
<tr>
<th>Rule</th>
<th>Minimal Tactic</th>
<th>Tactic with all tacticals</th>
</tr>
</thead>
<tbody>
<tr>
<td>listFormation</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>listEquality</td>
<td>EqCD</td>
<td></td>
</tr>
<tr>
<td>nilEquality ( j )</td>
<td>EqCD</td>
<td></td>
</tr>
<tr>
<td>nilFormation ( j )</td>
<td>D 0</td>
<td></td>
</tr>
<tr>
<td>consEquality</td>
<td>EqCD</td>
<td></td>
</tr>
<tr>
<td>consFormation</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>list_indEquality ( z \cdot T \cdot \mathsf{S} \cdot \text{list} \cdot x \cdot l \cdot f_{x} t )</td>
<td>EqCD</td>
<td></td>
</tr>
<tr>
<td>listElimination ( i \cdot f_{x} x \cdot l )</td>
<td>D ( i )</td>
<td></td>
</tr>
<tr>
<td>list_indReduceBase</td>
<td>ReduceEquands 0</td>
<td>ReduceAtAddr ( [2] ) 0</td>
</tr>
<tr>
<td>list_indReduceUp</td>
<td>ReduceEquands 0</td>
<td>ReduceAtAddr ( [2] ) 0</td>
</tr>
</tbody>
</table>
A.3.11 Inductive Types

\[ \Gamma \vdash \text{recType} \ X \ j = T_{X1} = \text{recType} \ X_2 = T_{X2} \in \bigcup_j \{A_j\} \]

by \text{recEquality} \ X

\[ \Gamma, X : \bigcup_j \{X/X1\} = T_{X1} \bigcup_j \{X/X2\} \in \bigcup_j \{A_j\} \]

\[ \Gamma, s = t \in \text{recType} \ X = T_X \mid \text{ext} \ t \]

by \text{recMemberEquality} \ j

\[ \Gamma \vdash s = t \in \text{recType} \ X = T_{X/X} \mid \text{ext} \ t \]

\[ \Gamma \vdash \text{recType} \ X = T_X \in \bigcup_j \{A_j\} \]

\[ \Gamma \vdash \text{let}^* \ f, (x, ) = t_1 \text{ in } f_s (x) = t_2 \text{ in } f_s (x) = T[e_1] \mid \text{ext} \ t \]

by \text{recIndEquality} \ z \ T \ \text{recType} \ X = T_X \ j \ P \ f \ x

\[ \Gamma, z : \text{recType} \ X = T_X, \Delta \vdash C \mid \text{ext} \ t \]

\[ \Gamma, z : \text{recType} \ X = T_X, \Delta, P : (\text{recType} \ X = T_X) \rightarrow P, f : (y : \text{recType} \ X = T_X \mid P(x) \rightarrow C[y/z]) \]

by \text{recElimination} \ i \ j \ P \ y \ f \ x

\[ \Gamma, z : \text{recType} \ X = T_X, \Delta, x : T_X [\text{recType} \ X = T_X/X], v : z = x \in T_X [\text{recType} \ X = T_X/X] \vdash C[x/z] \mid \text{ext} \ t \]

<table>
<thead>
<tr>
<th>Rule</th>
<th>Minimal Tactic</th>
<th>Tactic with all tacticals</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{recEquality} \ X</td>
<td>EqCD</td>
<td></td>
</tr>
<tr>
<td>\text{recMemberEquality} \ j</td>
<td>EqTypeCD</td>
<td></td>
</tr>
<tr>
<td>\text{recMemberFormation} \ j</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>\text{recIndEquality} \ z \ T</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>\text{recElimination} \ i \ j \ P \ y \ f \ x</td>
<td>RecTypeInduction \ i</td>
<td></td>
</tr>
<tr>
<td>\text{recUnrollElimination} \ i \ x \ v</td>
<td>D \ i</td>
<td></td>
</tr>
</tbody>
</table>
A.3.12 Subset

\[ \Gamma \vdash \bigcup_j \text{ext} \{x : S \mid T\} \]

by dependent_setFormation \( S \ x \)

\[ \Gamma \vdash S \in \bigcup_j [A] \]

\[ \Gamma, x:S \vdash \bigcup_j \text{ext} \ T \]

\[ \Gamma \vdash \{x_1:S_1 \mid T_1\} = \{x_2:S_2 \mid T_2\} \in \bigcup_j [A] \]

by setEquality \( x \)

\[ \Gamma \vdash S_1 = S_2 \in \bigcup_j [A] \]

\[ \Gamma, x:S_1 \vdash T_1[x/x_1] = T_2[x/x_2] \in \bigcup_j [A] \]

\[ \Gamma \vdash s = t \in \{x:S \mid T\} \ [A] \]

by dependent_set_memberEquality \( j \ x' \)

\[ \Gamma \vdash s = t \in S \ [A] \]

\[ \Gamma \vdash T[s/x] \ [A] \]

\[ \Gamma, x':S \vdash T[x'/x] \in \bigcup_j [A] \]

\[ \Gamma \vdash s = t \in \{S \mid T\} \ [A] \]

by independent_set_memberEquality

\[ \Gamma \vdash s = t \in S \ [A] \]

\[ \Gamma \vdash T[s/x] \ [A] \]

\[ \Gamma, x:S \vdash T[x'/x] \in \bigcup_j [A] \]

\[ \Gamma, z:\{x:S \mid T\}, \Delta \vdash C \text{ ext } (\lambda y.t) \ z \]

by setElimination \( i \ y \ v \)

\[ \Gamma, z:\{x:S \mid T\}, y:S, [v] : T[y/x], \Delta[y/z] \vdash C[y/z] \text{ ext } t \]

<table>
<thead>
<tr>
<th>Rule</th>
<th>Minimal Tactic</th>
<th>Tactic with all tacticals</th>
</tr>
</thead>
<tbody>
<tr>
<td>dependent_setFormation ( S \ x )</td>
<td>(--)</td>
<td>(--)</td>
</tr>
<tr>
<td>independent_setFormation</td>
<td>(--)</td>
<td>(--)</td>
</tr>
<tr>
<td>setEquality ( x )</td>
<td>EqCD</td>
<td>EqCD</td>
</tr>
<tr>
<td>dependent_set_memberEquality ( j \ x' )</td>
<td>EqTypeCD</td>
<td>EqTypeCD</td>
</tr>
<tr>
<td>dependent_set_memberFormation ( j \ s \ x' )</td>
<td>D 0</td>
<td>D 0</td>
</tr>
<tr>
<td>independent_set_memberEquality</td>
<td>EqTypeCD</td>
<td>EqTypeCD</td>
</tr>
<tr>
<td>independent_set_memberFormation</td>
<td>D 0</td>
<td>D 0</td>
</tr>
<tr>
<td>setElimination ( i \ y \ v )</td>
<td>D ( i )</td>
<td>D ( i )</td>
</tr>
</tbody>
</table>

168
A.3.13 Intersection

\[ \Gamma \vdash \bigcup_{j} \text{ext } \cap x:S.T \]
by \text{isectFormation } x \ S
\( \Gamma, x:S \vdash \bigcup_{j} \text{ext } T \)

\[ \Gamma \vdash t_1 = t_2 \in \cap x:S.T \ |_{A} \]
by \text{isect.memberEquality } j \ x'
\( \Gamma, x':S \vdash t_1 = t_2 \in T[x'/x] \ |_{A} \)
\( \Gamma \vdash S \in \bigcup_{j} \ |_{A} \)

\[ \Gamma \vdash f_1 = f_2 \in T[t/x] \ |_{A} \]
by \text{isect.memberCaseEquality } \cap x:S.T \ t
\( \Gamma \vdash f_1 = f_2 \in \cap x:S.T \ |_{A} \)
\( \Gamma \vdash t \in S \ |_{A} \)

\( \Gamma, f: \cap x:S.T, \Delta \vdash C \ \text{ext } t[f,Ax/y,z] \)
by \text{isectElimination } i \ s \ y \ z
\( \Gamma, f: \cap x:S.T, \Delta \vdash s \in S \ |_{A} \)
\( \Gamma, f: \cap x:S.T, y:T[s/x], \ z:y=f \in T[s/x], \Delta \vdash C \ \text{ext } t \)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Minimal Tactic</th>
<th>Tactic with all tacticals</th>
</tr>
</thead>
<tbody>
<tr>
<td>isectFormation  x \ S</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>isectEquality x</td>
<td>EqCD</td>
<td></td>
</tr>
<tr>
<td>isect.memberEquality j \ x'</td>
<td>EqTypeCD</td>
<td></td>
</tr>
<tr>
<td>isect.memberFormation j \ x'</td>
<td>D O</td>
<td></td>
</tr>
<tr>
<td>isect.member_caseEquality \cap x:S.T \ t</td>
<td>GenTypeCD t = x \in S</td>
<td></td>
</tr>
<tr>
<td>isectElimination i \ s \ y \ z</td>
<td>D i</td>
<td></td>
</tr>
</tbody>
</table>
A.3.14 Quotient Type

Γ ⊢ \text{ext } x, y : T // E_{i}

\text{by quotientFormation } T \ E \ x \ y \ z \ v \ v'

Γ ⊢ T \in U_j \ \forall \alpha

Γ, x : T, y : T \vdash E \in U_j \ \forall \alpha

Γ, x : T, v : E[x, y/x, y] \vdash E[y, x, y] \ \forall \alpha

Γ, x : T, y : T, v : E[x, y/x, y], v' : E[y, z/x, y] \vdash E[x, z/x, y] \ \forall \alpha

Γ ⊢ x, y : T // E_{i} = x, y : T // E_{j} \in U_j \ \forall \alpha

\text{by quotientWeakEquality } x \ y \ z \ v \ v'

Γ ⊢ T_1 = T_2 \in U_j \ \forall \alpha

Γ, x : T_1, y : T_1 \vdash E_1[x, y/x_1, y_1] = E_2[x, y/x_2, y_2] \in U_j \ \forall \alpha

Γ, x : T_1 \vdash E_1[x, x_1, y_1] \ \forall \alpha

Γ, x : T_1, y : T_1, v : E_1[x, y/x_1, y_1] \vdash E_1[y, x_1, y_1] \ \forall \alpha

Γ, x : T_1, y : T_1, z : T_1, v, v' : E_1[y, z/x_1, y_1], v' : E_1[y, z/x_1, y_1] \vdash E_1[x, z/x_1, y_1] \ \forall \alpha

Γ ⊢ x_1, y_1 : T // E_{i} = x_2, y_2 : T // E_{j} \in U_j \ \forall \alpha

\text{by quotientEquality } x \ y \ z \ v \ v'

Γ ⊢ \text{ext } y \ T // E_{i} \ \forall \alpha

Γ ⊢ x, y : T // E \in U_j \ \forall \alpha

Γ ⊢ s = t \in T \ \forall \alpha

Γ ⊢ s = t \in x, y : T // E \ \forall \alpha

\text{by quotient.memberEquality } j

Γ ⊢ x, y : T // E \in U_j \ \forall \alpha

Γ ⊢ s = t \in T \ \forall \alpha

Γ ⊢ E[s, t/x, y] \ \forall \alpha

Γ, v : s = t \in x, y : T // E, \ \Delta \vdash C \ \text{ext } u_j

\text{by quotient.equalityElimination } i \ j \ v \ v'

Γ, v : s = t \in x, y : T // E, \ \Delta \vdash C \ \text{ext } u_j

Γ, v, s = t \in x, y : T // E, \ \Delta, x' : T, y' : T \vdash E[x', y'/x, y] \in U_j \ \forall \alpha

Γ, z : x, y : T // E, \ \Delta \vdash s = t \in S \ \forall \alpha

\text{by quotientElimination } 1 \ j \ x' \ y' \ v

Γ, z : x, y : T // E, \ \Delta, x' : T, y' : T \vdash E[x', y'/x, y] \in U_j \ \forall \alpha

Γ, z : x, y : T // E, \ \Delta \vdash S \in U_j \ \forall \alpha

Γ, 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] \ \forall \alpha

Γ, z : x, y : T // E, \ \Delta \vdash s = t \in S \ \forall \alpha

\text{by quotientElimination } 2 \ j \ x' \ y' \ v

Γ, z : x, y : T // E, \ \Delta, x' : T, y' : T \vdash E[x', y'/x, y] \in U_j \ \forall \alpha

Γ, z : x, y : T // E, \ \Delta \vdash S \in U_j \ \forall \alpha

Γ, z : x, y : T // E, \ \Delta, x' : T, y' : T, v : E[x', y'/x, y], \ \Delta [x'/z] \vdash s[x'/z] = t[y'/z] \in S[x'/z] \ \forall \alpha

170
### A.3.15 Direct Computation

- \[ \Gamma \vdash C \quad \text{ext } t_j \]
  - by direct\_computation \( \text{tag}C \)
  - \( \Gamma \vdash C \downarrow_{\text{tag}C} \quad \text{ext } t_j \)

- \[ \Gamma \vdash C \quad \text{ext } t_j \]
  - by reverse\_direct\_computation \( \text{tag}C \)
  - \( \Gamma, z : T, \Delta \vdash C \quad \text{ext } t_j \)

- \[ \Gamma, z : T, \Delta \vdash C \quad \text{ext } t_j \]
  - by direct\_computation\_hypothesis \( i \ \text{tag}T \)
  - \( \Gamma, z : T \downarrow_{\text{tag}T}, \Delta \vdash C \quad \text{ext } t_j \)

- \[ \Gamma, z : T, \Delta \vdash C \quad \text{ext } t_j \]
  - by reverse\_direct\_computation\_hypothesis \( i \ \text{tag}T \)
  - \( \Gamma, z : T \downarrow_{\text{tag}T}, \Delta \vdash C \quad \text{ext } t_j \)
A.3.16 Miscellaneous

\[ \Gamma, x:T, \Delta \vdash T \text{ by hypothesis } i \]
\[ \Gamma, \Delta \vdash C \text{ by cut } i \]
\[ \Gamma, \Delta \vdash T \text{ by introduction } i \]
\[ \Gamma, x:T, \Delta \vdash C \text{ by thin } i \]
\[ \Gamma \vdash t \text{ by extract "theorem-name" } \]
\[ \Gamma \vdash C \text{ by lemma "theorem-name" } \]
\[ \Gamma \vdash t \in T \text{ by } \texttt{extract "theorem-name"} \]
\[ \Gamma \vdash C \text{ by instantiate } \Gamma' \]
\[ \Gamma' \vdash C' \text{ by } \texttt{rename } y x \]
\[ \Gamma \vdash C \text{ by because } \]
\[ \Gamma \vdash t \in T \text{ by } \texttt{rename } y x \]

<table>
<thead>
<tr>
<th>Rule</th>
<th>Minimal Tactic</th>
<th>Tactic with all tacticals</th>
</tr>
</thead>
<tbody>
<tr>
<td>hypothesis } i</td>
<td>Hypothesis</td>
<td>NthHyp } i</td>
</tr>
<tr>
<td>thin } i</td>
<td>Thin } i</td>
<td></td>
</tr>
<tr>
<td>cut } i T x</td>
<td>Assert } T</td>
<td>AssertDeclAtHyp } i T x</td>
</tr>
<tr>
<td>introduction } t</td>
<td>UseWitness } t</td>
<td>UseWitness } t</td>
</tr>
<tr>
<td>hyp_replacement } i S j</td>
<td>SubstClause } S i</td>
<td>SubstClause } S i</td>
</tr>
<tr>
<td>lemma &quot;theorem-name&quot;</td>
<td>Lemma &quot;theorem-name&quot;</td>
<td>Lemma &quot;theorem-name&quot;</td>
</tr>
<tr>
<td>extract &quot;theorem-name&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>instantiate } \Gamma' C' } \sigma</td>
<td>RenameVar } x i *</td>
<td>RenameVar } x i *</td>
</tr>
<tr>
<td>because }</td>
<td>FiAt</td>
<td>FiAt</td>
</tr>
</tbody>
</table>

*: y is the variable declared in hypothesis } i.

A.4 Term Structure of Rules

Rule definitions are expressed as ‘terms’ in the sense described in Chapter 5. They are normally stored in library objects of kind rule.

Figure A.5 shows the basic tree structure of rule terms. I have included kinds of ‘virtual tree nodes’ in this description. These virtual nodes do not correspond to term constructors used in building up rules, but they do help in explaining the structure of rule terms.

Italic type is used in Figure A.5 for kinds of tree nodes that correspond to terms and term slots. Roman type is used for tree nodes that correspond to text strings and text slots. }\null indicates a linebreak in a display form, }\null indicates a display form with ‘zero width’, and }\null is the usual invisible space. The suffix * character on a node-kind names indicate a sequence of nodes. The }\null-term node kind is for terms in Nuprl’s object language.

Names for the term constructors that are suitable for entering the constructors are shown in...
Table A.5: Structure of Rules

<table>
<thead>
<tr>
<th>node name</th>
<th>alternatives / structure</th>
<th>physical node name</th>
<th>alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>rule-def</td>
<td>goal \BY rule \ subgoal*</td>
<td>simple-rule</td>
<td>rldef</td>
</tr>
<tr>
<td>goal</td>
<td>goal \BY rule \ constraint \ subgoal*</td>
<td>constrained-rule</td>
<td>crldef</td>
</tr>
<tr>
<td>subgoal*</td>
<td>subgoal \ ... \ subgoal</td>
<td>No Subgoals</td>
<td></td>
</tr>
<tr>
<td>subgoal</td>
<td>segment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sequent</td>
<td>hyp-item* + concl</td>
<td>no-ext-sequent</td>
<td>seq</td>
</tr>
<tr>
<td></td>
<td>hyp-item* + concl ext extract</td>
<td>ext-sequent</td>
<td>eseq</td>
</tr>
<tr>
<td>hyp-item*</td>
<td>hyp-item , ... , hyp-item \null</td>
<td>normal-declaration</td>
<td>decl</td>
</tr>
<tr>
<td>hyp-item</td>
<td>hyp \ hyp-list</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hyp</td>
<td>variable : subst-term</td>
<td>normal-declaration</td>
<td>decl</td>
</tr>
<tr>
<td></td>
<td>[variable : subst-term]</td>
<td>hidden-declaration</td>
<td>hdecl</td>
</tr>
<tr>
<td>hyp-list</td>
<td>variable \ substitution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>concl</td>
<td>subst-term</td>
<td></td>
<td></td>
</tr>
<tr>
<td>extract</td>
<td>subst-term</td>
<td></td>
<td></td>
</tr>
<tr>
<td>constraint</td>
<td>Let term = subst-term</td>
<td>matching-let</td>
<td>mlet</td>
</tr>
<tr>
<td></td>
<td>Let arg* = Call{lisp-fun-name}</td>
<td>lisp-let</td>
<td>llet</td>
</tr>
<tr>
<td>rule</td>
<td>rule-name arg*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arg*</td>
<td>arg \ ... \ arg</td>
<td>()</td>
<td></td>
</tr>
<tr>
<td>arg</td>
<td>variable</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>#</td>
<td>hyp-index</td>
<td>hypi</td>
</tr>
<tr>
<td></td>
<td>#hyp-num</td>
<td>hyp-index-n</td>
<td>hypind</td>
</tr>
<tr>
<td></td>
<td>level{level}</td>
<td>level-exp</td>
<td>level</td>
</tr>
<tr>
<td></td>
<td>parm-sub{parm-sub}</td>
<td>parm-substitution</td>
<td>parmsg</td>
</tr>
<tr>
<td>subst-term</td>
<td>prl-term</td>
<td>substitution</td>
<td>subst</td>
</tr>
<tr>
<td>substitution</td>
<td>variable[prl-term/variable]</td>
<td>subst-1</td>
<td>subst</td>
</tr>
<tr>
<td></td>
<td>variable[prltm,prlmtm/var,]</td>
<td>subst-2</td>
<td>subst2</td>
</tr>
<tr>
<td></td>
<td>variable[prlmtm,prlmtm/var,]</td>
<td>subst-3</td>
<td>subst3</td>
</tr>
<tr>
<td></td>
<td>variable[parm-sub]</td>
<td>subst-parms</td>
<td>substp</td>
</tr>
<tr>
<td>variable</td>
<td>variable</td>
<td>var-name</td>
<td></td>
</tr>
<tr>
<td>nat-number</td>
<td>natural</td>
<td>nat-digits</td>
<td></td>
</tr>
</tbody>
</table>

Figure A.5: Structure of Rules
the ‘alias’ column. The term constructors for sequences do not need to be entered explicitly by name. The editor recognizes the context of each sequence, and sequence items can be added and deleted using the (C-0) and (C-C) sequence commands.

For rule terms to be well-formed, there are several extra constraints on their structure. These include:

- **substitution** terms can only occur in
  - the concl or hyp-item term of a subgoal,
  - the right-hand subterm of a matching-constraint,
  - extract term of a goal.

- There should be a hyp-index term as a rule arg for each hyp-list hyp-item that is followed by a hyp hyp-item in the goal of the rule definition.

- Adjacent hyp-items should not be both variables.

- The extract of a subgoal (if it exists) should always be a variable.