# Galactic type theory

Andrej Bauer Christopher A. Stone

May 27, 2015

## 1 Universe indices

There is a pointed preorder $(\mathbb{P}, \leq, o)$ of *universe indices*, i.e., $\leq$ is reflexive and transitive, with a decidable equality, and a subset $\mathbb{F} \subseteq \mathbb{P}$ of *fibered indices*. We use lowercase Greek letters for universe indices.

FIX THIS PARAGRAPH, IT IS A MISLEADING EXAMPLE WHICH BREAKS THINGS. For example, we could take

$$\mathbb{P} = \mathbb{N} \times \{0, 1\}$$

with the lexicographic order, $o = (0, 0)$ and $\mathbb{F} = \mathbb{N} \times \{0\}$. Write $\mathcal{F}_i$ for the universe indexed by $(i, 0)$ and $\mathcal{U}_j$ for the universe indexed by $(i, 1)$. Then the poset structure of $\mathbb{P}$ reflects a hierachy of inclusions

$$\mathcal{F}_0 \leq \mathcal{U}_0 \leq \mathcal{F}_1 \leq \mathcal{U}_1 \leq \cdots$$

To express the fact that a universe may be a member of another one, and to calculate which dependent products may be formed and at what universe they land in, we assume given two partial functions

- $\mathsf{succ}(\cdot) : \mathbb{P} \rightharpoonup \mathbb{P}$
- $\mathsf{max}(\cdot, \cdot) : \mathbb{P} \times \mathbb{P} \rightharpoonup \mathbb{P}$

whose domains of definition are decidable and

- for all $\alpha, \beta \in \mathbb{F}$, if $\mathsf{max}(\alpha, \beta)$ is defined then $\mathsf{max}(\alpha, \beta) \in \mathbb{F}$,

That is, a dependent product of fibered types is again fibered, if it is defined, and each universe is itself fibered, as long as it is a member of a universe. It seems that these are non essential assumptions – a more flexibile scheme of universe indices would work just as well – but we are followng Voevodsky's HTS formulation.

The functions are *not* required to compute maximum or successor, but we use the suggestive notation because in the standard case they do compute maxima and succesors. For instance, we could express the fact that $\beta$ is the index of an impredicative universe by setting $\mathsf{max}(\alpha, \beta) = \beta$ for all $\alpha \in \mathbb{P}$.

## 2 The declarative formulation

In this section we give the formulation of galactic type theory in a declarative way which minimizes the number of judgments, is better suited for a semantic account, but is not susceptible to an algorithmic treatment.

### 2.1 Syntax

Contexts:

$$
\begin{array}{rcll}
\Gamma & ::= & \bullet & \text{empty context} \\
& \mid & \Gamma,\ x : T & \text{context extended with } x : T
\end{array}
$$

Types:

$$
\begin{array}{rcll}
T, U & ::= & \mathbb{U}_\alpha & \text{universe} \\
& \mid & \mathsf{El}^\alpha\, e & \text{type named by } e \\
& \mid & \mathsf{Unit} & \text{the unit type} \\
& \mid & \prod_{(x:T)} U & \text{product} \\
& \mid & \mathsf{Paths}_T(e_1, e_2) & \text{path type} \\
& \mid & \mathsf{Id}_T(e_1, e_2) & \text{equality type}
\end{array}
$$

Terms:

$$
\begin{array}{rcll}
e & ::= & x & \text{variable} \\
& \mid & \lambda x{:}T_1.T_2 \,.\, e & \lambda\text{-abstraction} \\
& \mid & e_1 \mathbin{@^{x:T_1.T_2}} e_2 & \text{application} \\
& \mid & \star & \text{the element of unit type} \\
& \mid & \mathsf{idpath}_T\, e & \text{identity path} \\
& \mid & \mathsf{J}_T([x\, y\, p \,.\, U], [z \,.\, e_1], e_2, e_3, e_4) & \text{path eliminator} \\
& \mid & \mathsf{refl}_T\, e & \text{reflexivity} \\
& \mid & \mathsf{G}_T([x\, y\, p \,.\, U], [z \,.\, e_1], e_2, e_3, e_4) & \text{equality eliminator} \\
& \mid & \mathsf{coerce}^{\alpha \mapsto \beta}\, e & \text{universe coercion} \\
& \mid & \mathsf{unit} & \text{the name of unit type} \\
& \mid & \pi^{\alpha,\beta} x : e_1 \,.\, e_2 & \text{the name of product type} \\
& \mid & u_\alpha & \text{the name of a universe} \\
& \mid & \mathsf{paths}^\alpha_{e_1}(e_2, e_3) & \text{the name of a path type} \\
& \mid & \mathsf{id}^\alpha_{e_1}(e_2, e_3) & \text{the name of an equality type}
\end{array}
$$

Note that $\lambda$-abstraction and application are tagged with extra types not usually seen in type theory. An abstraction $\lambda x{:}T_1.T_2 \,.\, e$ speficies not only the type $T_1$ of $x$ but also the type $T_2$ of $e$, where $x$ is bound in $T_2$ and $e$. Similarly, an application $e_1 \mathbin{@^{x:T_1.T_2}} e_2$ specifies that $e_1$ and $e_2$ have types $\prod_{(x:T_1)} T_2$ and $T_2$, respectively. This is necessary because in the presence of exotic equalities (think "$\mathsf{nat} \to \mathsf{bool} \equiv \mathsf{nat} \to \mathsf{nat}$") we must be *very* careful about $\beta$-reductions.

Following Voevodsky's HTS we include an equality eliminator. Note that it is not strictly necessary because every instance of it can be derived using the reflection rule EQ-REFLECTION. [XXX IS THIS TRUE?!]

## 2.2 Judgments

| | |
|---|---|
| $\Gamma\ \mathsf{ctx}$ | $\Gamma$ is a well formed context |
| $\Gamma \vdash T\ \mathsf{type}$ | $T$ is a type in context $\Gamma$ |
| $\Gamma \vdash T\ \mathsf{fibered}$ | $T$ is a fibered type in context $\Gamma$ |
| $\Gamma \vdash e : T$ | $e$ is a well formed term of type $T$ in context $\Gamma$ |
| $\Gamma \vdash T \equiv U$ | $T$ and $U$ are equal types in context $\Gamma$ |
| $\Gamma \vdash e_1 \equiv e_2 : T$ | $e_1$ and $e_2$ are equal terms of type $T$ in context $\Gamma$ |

## 2.3 Contexts

$$\textsc{ctx-empty}\quad \frac{}{\bullet\ \mathsf{ctx}} \qquad \textsc{ctx-extend}\quad \frac{\Gamma\ \mathsf{ctx} \qquad \Gamma \vdash T\ \mathsf{type}}{\Gamma,\, x : T\ \mathsf{ctx}}$$

## 2.4 Types

$$\textsc{ty-universe}\quad \frac{\Gamma\ \mathsf{ctx} \qquad \alpha \in \mathbb{P}}{\Gamma \vdash \mathbb{U}_\alpha\ \mathsf{type}} \qquad \textsc{ty-prod}\quad \frac{\Gamma \vdash T\ \mathsf{type} \qquad \Gamma,\, x : T \vdash U\ \mathsf{type}}{\Gamma \vdash \prod_{(x:T)} U\ \mathsf{type}} \qquad \textsc{ty-el}\quad \frac{\Gamma \vdash e : \mathbb{U}_\alpha}{\Gamma \vdash \mathsf{El}^\alpha\, e\ \mathsf{type}}$$

$$\textsc{ty-unit}\quad \frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \mathsf{Unit}\ \mathsf{type}} \qquad \textsc{ty-paths}\quad \frac{\Gamma \vdash T\ \mathsf{fibered} \qquad \Gamma \vdash e_1 : T \qquad \Gamma \vdash e_2 : T}{\Gamma \vdash \mathsf{Paths}_T(e_1, e_2)\ \mathsf{type}}$$

$$\textsc{ty-id}\quad \frac{\Gamma \vdash T\ \mathsf{type} \qquad \Gamma \vdash e_1 : T \qquad \Gamma \vdash e_2 : T}{\Gamma \vdash \mathsf{Id}_T(e_1, e_2)\ \mathsf{type}}$$

## 2.5 Fibered types

$$\textsc{fib-universe}\quad \frac{\Gamma\ \mathsf{ctx} \qquad \alpha \in \mathbb{P}}{\Gamma \vdash \mathbb{U}_\alpha\ \mathsf{fibered}} \qquad \textsc{fib-prod}\quad \frac{\Gamma \vdash T\ \mathsf{fibered} \qquad \Gamma,\, x : T \vdash U\ \mathsf{fibered}}{\Gamma \vdash \prod_{(x:T)} U\ \mathsf{fibered}}$$

$$\textsc{fib-el}\quad \frac{\Gamma \vdash e : \mathbb{U}_\alpha \qquad \alpha \in \mathbb{F}}{\Gamma \vdash \mathsf{El}^\alpha\, e\ \mathsf{fibered}} \qquad \textsc{fib-unit}\quad \frac{\Gamma\ \mathsf{ctx} \qquad o \in \mathbb{F}}{\Gamma \vdash \mathsf{Unit}\ \mathsf{fibered}}$$

$$\textsc{fib-paths}\quad \frac{\Gamma \vdash T\ \mathsf{fibered} \qquad \Gamma \vdash e_1 : T \qquad \Gamma \vdash e_2 : T}{\Gamma \vdash \mathsf{Paths}_T(e_1, e_2)\ \mathsf{fibered}}$$

## 2.6 Terms

### General terms

$$\textsc{term-eq}\quad \frac{\Gamma \vdash e : T \qquad \Gamma \vdash T \equiv U}{\Gamma \vdash e : U}$$

$$\textsc{term-var}\quad \frac{\Gamma\ \mathsf{ctx} \qquad (x{:}T) \in \Gamma}{\Gamma \vdash x : T}$$

### Products

$$\textsc{term-abs}\quad \frac{\Gamma, x{:}T \vdash e : U}{\Gamma \vdash (\lambda x{:}T.U\,.\,e) : \prod_{(x:T)} U}$$

$$\textsc{term-app}\quad \frac{\Gamma \vdash e_1 : \prod_{(x:T)} U \qquad \Gamma \vdash e_2 : T}{\Gamma \vdash e_1 \mathbin{@^{x:T.U}} e_2 : U[e_2/x]}$$

### Paths

$$\textsc{term-idpath}\quad \frac{\Gamma \vdash e : T \qquad \Gamma \vdash T\ \mathsf{fibered}}{\Gamma \vdash \mathsf{idpath}_T\ e : \mathsf{Paths}_T(e, e)}$$

$$\textsc{term-J}\quad \frac{\begin{array}{c}\Gamma \vdash T\ \mathsf{fibered} \\ \Gamma, x{:}T, y{:}T, p{:}\mathsf{Paths}_T(x, y) \vdash U\ \mathsf{fibered} \\ \Gamma, z{:}T \vdash e_1 : U[z/x, z/y, (\mathsf{idpath}_T\ z)/p] \\ \Gamma \vdash e_3 : T \qquad \Gamma \vdash e_4 : T \qquad \Gamma \vdash e_2 : \mathsf{Paths}_T(e_3, e_4)\end{array}}{\Gamma \vdash \mathsf{J}_T([x\,y\,p\,.\,U], [z\,.\,e_1], e_2, e_3, e_4) : U[e_2/x, e_3/y, e_4/p]}$$

### Equality types

$$\textsc{term-refl}\quad \frac{\Gamma \vdash e : T}{\Gamma \vdash \mathsf{refl}_T\ e : \mathsf{Id}_T(e, e)}$$

$$\textsc{term-G}\quad \frac{\begin{array}{c}\Gamma \vdash T\ \mathsf{type} \\ \Gamma, x{:}T, y{:}T, p{:}\mathsf{Id}_T(x, y) \vdash U\ \mathsf{type} \\ \Gamma, z{:}T \vdash e_1 : U[z/x, z/y, (\mathsf{refl}_T\ z)/p] \\ \Gamma \vdash e_3 : T \qquad \Gamma \vdash e_4 : T \qquad \Gamma \vdash e_2 : \mathsf{Id}_T(e_3, e_4)\end{array}}{\Gamma \vdash \mathsf{G}_T([x\,y\,p\,.\,U], [z\,.\,e_1], e_2, e_3, e_4) : U[e_2/x, e_3/y, e_4/p]}$$

### Unit

$$\textsc{term-star}\quad \frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \star : \mathsf{Unit}}$$

**Names**

$$\frac{\Gamma \vdash e : \mathbb{U}_\alpha \qquad \alpha \le \beta}{\Gamma \vdash \mathsf{coerce}^{\alpha \mapsto \beta}\, e : \mathbb{U}_\beta}\ \text{TERM-COERCE} \qquad \frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \mathsf{unit} : \mathbb{U}_o}\ \text{NAME-UNIT} \qquad \frac{\Gamma\ \mathsf{ctx} \qquad \mathsf{succ}(\alpha) = \beta}{\Gamma \vdash u_\alpha : \mathbb{U}_\beta}\ \text{NAME-UNIVERSE}$$

$$\frac{\Gamma \vdash e_1 : \mathbb{U}_\alpha \qquad \Gamma, x : \mathsf{El}^\alpha\, e_1 \vdash e_2 : \mathbb{U}_\beta \qquad \mathsf{max}(\alpha, \beta) = \gamma}{\Gamma \vdash (\pi^{\alpha,\beta} x : e_1 \,.\, e_2) : \mathbb{U}_\gamma}\ \text{NAME-PROD}$$

$$\frac{\Gamma \vdash e_T : \mathbb{U}_\alpha \qquad \alpha \in \mathbb{F} \qquad \Gamma \vdash e_1 : \mathsf{El}^\alpha\, e_T \qquad \Gamma \vdash e_2 : \mathsf{El}^\alpha\, e_T}{\Gamma \vdash \mathsf{paths}^\alpha_{e_T}(e_1, e_2) : \mathbb{U}_\alpha}\ \text{NAME-PATHS}$$

$$\frac{\Gamma \vdash e_T : \mathbb{U}_\alpha \qquad \Gamma \vdash e_1 : \mathsf{El}^\alpha\, e_T \qquad \Gamma \vdash e_2 : \mathsf{El}^\alpha\, e_T}{\Gamma \vdash \mathsf{id}^\alpha_{e_T}(e_1, e_2) : \mathbb{U}_\alpha}\ \text{NAME-ID}$$

## 2.7 Type Equality

**General rules**

$$\frac{\Gamma \vdash T\ \mathsf{type}}{\Gamma \vdash T \equiv T}\ \text{TYEQ-REFL} \qquad \frac{\Gamma \vdash U \equiv T}{\Gamma \vdash T \equiv U}\ \text{TYEQ-SYM} \qquad \frac{\Gamma \vdash T \equiv T' \qquad \Gamma \vdash T' \equiv U}{\Gamma \vdash T \equiv U}\ \text{TYEQ-TRANS}$$

**Type formers are congruences**

$$\frac{\Gamma \vdash e_1 \equiv e_2 : \mathbb{U}_\alpha}{\Gamma \vdash \mathsf{El}^\alpha\, e_1 \equiv \mathsf{El}^\alpha\, e_2}\ \text{TYCONG-EL} \qquad \frac{\Gamma \vdash T_1 \equiv U_1 \qquad \Gamma, x : T_1 \vdash T_2 \equiv U_2}{\Gamma \vdash \prod_{(x:T_1)} T_2 \equiv \prod_{(x:U_1)} U_2}\ \text{TYCONG-PROD}$$

$$\frac{\begin{array}{c}\Gamma \vdash T\ \mathsf{fibered} \qquad \Gamma \vdash U\ \mathsf{fibered} \\ \Gamma \vdash T \equiv U \qquad \Gamma \vdash e_1 \equiv e_1' : T \qquad \Gamma \vdash e_2 \equiv e_2' : T\end{array}}{\Gamma \vdash \mathsf{Paths}_T(e_1, e_2) \equiv \mathsf{Paths}_U(e_1', e_2')}\ \text{TYCONG-PATHS}$$

$$\frac{\Gamma \vdash T \equiv U \qquad \Gamma \vdash e_1 \equiv e_1' : T \qquad \Gamma \vdash e_2 \equiv e_2' : T}{\Gamma \vdash \mathsf{Id}_T(e_1, e_2) \equiv \mathsf{Id}_U(e_1', e_2')}\ \text{TYCONG-ID}$$

**Computation of types from names**

TYEQ-EL-PI
$$\frac{\Gamma \vdash e_1 : \mathbb{U}_\alpha \qquad \Gamma,\, x : \mathsf{El}^\alpha\, e_1 \vdash e_2 : \mathbb{U}_\beta \qquad \mathsf{max}(\alpha, \beta) = \gamma}{\Gamma \vdash \mathsf{El}^\gamma\, (\pi^{\alpha,\beta} x : e_1 \,.\, e_2) \equiv \prod_{(x : \mathsf{El}^\alpha\, e_1)} \mathsf{El}^\beta\, e_2}$$

TYEQ-EL-UNIT
$$\frac{}{\Gamma \vdash \mathsf{El}^o\, \mathsf{unit} \equiv \mathsf{Unit}}$$

TYEQ-EL-COERCE
$$\frac{\Gamma \vdash e : \mathbb{U}_\alpha \qquad \alpha \leq \beta}{\Gamma \vdash \mathsf{El}^\beta\, (\mathsf{coerce}^{\alpha \mapsto \beta}\, e) \equiv \mathsf{El}^\alpha\, e}$$

TYEQ-EL-PATHS
$$\frac{\Gamma \vdash e_T : \mathbb{U}_\alpha \qquad \alpha \in \mathbb{F} \qquad \Gamma \vdash e_1 : \mathsf{El}^\alpha\, e_T \qquad \Gamma \vdash e_2 : \mathsf{El}^\alpha\, e_T}{\Gamma \vdash \mathsf{El}^\alpha\, (\mathsf{paths}^\alpha_{e_T}(e_1, e_2)) \equiv \mathsf{Paths}_{\mathsf{El}^\alpha\, e_T}(e_1, e_2)}$$

TYEQ-EL-ID
$$\frac{\Gamma \vdash e_T : \mathbb{U}_\alpha \qquad \Gamma \vdash e_1 : \mathsf{El}^\alpha\, e_T \qquad \Gamma \vdash e_2 : \mathsf{El}^\alpha\, e_T}{\Gamma \vdash \mathsf{El}^\alpha\, (\mathsf{id}^\alpha_{e_T}(e_1, e_2)) \equiv \mathsf{Id}_{\mathsf{El}^\alpha\, e_T}(e_1, e_2)}$$

## 2.8 Term Equality

General rules:

EQ-REFL
$$\frac{\Gamma \vdash e : T}{\Gamma \vdash e \equiv e : T}$$

EQ-SYM
$$\frac{\Gamma \vdash e_2 \equiv e_1 : T}{\Gamma \vdash e_1 \equiv e_2 : T}$$

EQ-TRANS
$$\frac{\Gamma \vdash e_1 \equiv e_2 : T \qquad \Gamma \vdash e_2 \equiv e_3 : T}{\Gamma \vdash e_1 \equiv e_3 : T}$$

EQ-EQ
$$\frac{\Gamma \vdash e_1 \equiv e_2 : T \qquad \Gamma \vdash T \equiv U}{\Gamma \vdash e_1 \equiv e_2 : U}$$

**Congruence rules for products**

CONG-ABS
$$\frac{\Gamma \vdash T_1 \equiv U_1 \qquad \Gamma,\, x : T_1 \vdash T_2 \equiv U_2 \qquad \Gamma,\, x : T_1 \vdash e_1 \equiv e_2 : T_2}{\Gamma \vdash (\lambda x{:}T_1.T_2 \,.\, e_1) \equiv (\lambda x{:}U_1.U_2 \,.\, e_2) : \prod_{(x:T_1)} T_2}$$

CONG-APP
$$\frac{\begin{array}{c}\Gamma \vdash T_1 \equiv U_1 \qquad \Gamma,\, x : T_1 \vdash T_2 \equiv U_2 \\ \Gamma \vdash e_1 \equiv e_1' : \prod_{(x:T_1)} T_2 \qquad \Gamma \vdash e_2 \equiv e_2' : T_1\end{array}}{\Gamma \vdash (e_1 \mathbin{@^{x:T_1.T_2}} e_2) \equiv (e_1' \mathbin{@^{x:U_1.U_2}} e_2') : T_2[e_2/x]}$$

The annotations on an application really do matter for determining when two terms are equal. For example, if $X, Y : \mathbb{U}_o$, $f : \mathsf{nat} \to X$ and $e : \mathsf{Id}_{\mathbb{U}_o}(\mathsf{nat}{\to}X, \mathsf{nat}{\to}Y)$, then $(f \mathbin{@^{\_:\mathsf{nat}.X}} 0) : X$ and $(f \mathbin{@^{\_:\mathsf{nat}.Y}} 0) : Y$, so the two identical-but-for-annotations terms have different types and thus cannot be equivalent.

**Other rules for products**

$$\frac{\Gamma \vdash T_1 \equiv U_1 \qquad \Gamma,\, x : T_1 \vdash T_2 \equiv U_2 \qquad \Gamma,\, x : T_1 \vdash\ e_1\ :\ T_2 \qquad \Gamma \vdash\ e_2\ :\ U_1}{\Gamma \vdash \big((\lambda x{:}T_1 . T_2 \,.\, e_1)\ @^{x:U_1.U_2}\ e_2\big) \equiv e_1[e_2/x] : T_2[e_2/x]}\ \textsc{eq-beta}$$

$$\frac{\Gamma \vdash\ e_1\ :\ \prod_{(x:T)} U \qquad \Gamma \vdash\ e_2\ :\ \prod_{(x:T)} U \qquad \Gamma,\, x : T \vdash (e_1\ @^{x:T.U}\ x) \equiv (e_2\ @^{x:T.U}\ x) : U}{\Gamma \vdash e_1 \equiv e_2 : \prod_{(x:T)} U}\ \textsc{eq-ext}$$

**Eta rule for unit type**

$$\frac{\Gamma \vdash\ e\ :\ \mathsf{Unit}}{\Gamma \vdash e \equiv \star : \mathsf{Unit}}\ \textsc{eta-star}$$

**Congruence rules for paths**

$$\frac{\Gamma \vdash e_1 \equiv e_2 : T \qquad \Gamma \vdash T \equiv U \qquad \Gamma \vdash T\ \mathsf{fibered} \qquad \Gamma \vdash U\ \mathsf{fibered}}{\Gamma \vdash \mathsf{idpath}_T\ e_1 \equiv \mathsf{idpath}_U\ e_2 : \mathsf{Paths}_T(e_1, e_1)}\ \textsc{cong-idpath}$$

$$\frac{\begin{array}{c} \Gamma \vdash T\ \mathsf{fibered} \qquad \Gamma \vdash U\ \mathsf{fibered} \\ \Gamma,\, x : T,\, y : T,\, p : \mathsf{Paths}_T(x, y) \vdash P\ \mathsf{fibered} \\ \Gamma,\, x : U,\, y : U,\, p : \mathsf{Paths}_U(x, y) \vdash Q\ \mathsf{fibered} \\ \Gamma \vdash T \equiv U \qquad \Gamma,\, x : T,\, y : T,\, p : \mathsf{Paths}_T(x, y) \vdash P \equiv Q \\ \Gamma,\, z : T \vdash e_1 \equiv e_1' : P[z/x, z/y, (\mathsf{idpath}_T\ z)/p] \\ \Gamma \vdash e_3 \equiv e_3' : T \qquad \Gamma \vdash e_4 \equiv e_4' : T \qquad \Gamma \vdash e_2 \equiv e_2' : \mathsf{Paths}_T(e_3, e_4) \end{array}}{\Gamma \vdash \mathsf{J}_T([x\, y\, p \,.\, P], [z \,.\, e_1], e_2, e_3, e_4) \equiv \mathsf{J}_U([x\, y\, p \,.\, Q], [z \,.\, e_1'], e_2', e_3', e_4') : P[e_2/x, e_3/y, e_4/p]}\ \textsc{cong-J}$$

**Computation rule for paths**

$$\frac{\begin{array}{c} \Gamma \vdash T\ \mathsf{fibered} \qquad \Gamma,\, x : T,\, y : T,\, p : \mathsf{Paths}_T(x, y) \vdash U\ \mathsf{fibered} \\ \Gamma,\, z : T \vdash\ e_1\ :\ U[z/x, z/y, (\mathsf{idpath}_T\ z)/p] \qquad \Gamma \vdash\ e_2\ :\ T \end{array}}{\Gamma \vdash \mathsf{J}_T([x\, y\, p \,.\, U], [z \,.\, e_1], \mathsf{idpath}_T\ e_2, e_2, e_2) \equiv e_1[e_2/z] : U[e_2/x, e_2/x, (\mathsf{idpath}_T\ e_2)/p]}\ \textsc{eq-J}$$

**Congruence rules for equalities**

CONG-REFL
$$\frac{\Gamma \vdash e_1 \equiv e_2 : T \qquad \Gamma \vdash T \equiv U}{\Gamma \vdash \mathsf{refl}_T\ e_1 \equiv \mathsf{refl}_U\ e_2 : \mathsf{Id}_T(e_1, e_1)}$$

CONG-G
$$\frac{\begin{array}{c}\Gamma \vdash T \equiv U \qquad \Gamma,\, x\!:\!T,\, y\!:\!T,\, p\!:\!\mathsf{Id}_T(x,y) \vdash U \equiv V \\ \Gamma,\, z\!:\!T \vdash e_1 \equiv e_1' : U[z/x, z/y, (\mathsf{refl}_T\ z)/p] \\ \Gamma \vdash\ e_3\ :\ T \qquad \Gamma \vdash\ e_4\ :\ T \\ \Gamma \vdash e_3 \equiv e_3' : T \qquad \Gamma \vdash e_4 \equiv e_4' : T \qquad \Gamma \vdash e_2 \equiv e_2' : \mathsf{Id}_T(e_3, e_4)\end{array}}{\Gamma \vdash \mathsf{G}_T([x\,y\,p\,.\,U], [z\,.\,e_1], e_2, e_3, e_4) \equiv \mathsf{G}_U([x\,y\,p\,.\,V], [z\,.\,e_1'], e_2', e_3', e_4') : U[e_2/x, e_3/y, e_4/p]}$$

**Eta rules for equalities**

K-REFL
$$\frac{\Gamma \vdash\ e_1\ :\ \mathsf{Id}_T(e_2, e_2)}{\Gamma \vdash e_1 \equiv (\mathsf{refl}_T\ e_2) : \mathsf{Id}_T(e_2, e_2)}$$

**Equality reflection**

EQ-REFLECTION
$$\frac{\Gamma \vdash\ e\ :\ \mathsf{Id}_T(e_1, e_2)}{\Gamma \vdash e_1 \equiv e_2 : T}$$

**Computation rule for equality**

EQ-G
$$\frac{\begin{array}{c}\Gamma \vdash T\ \mathsf{type} \qquad \Gamma,\, x\!:\!T,\, y\!:\!T,\, p\!:\!\mathsf{Id}_T(x,y) \vdash U\ \mathsf{type} \\ \Gamma,\, z\!:\!T \vdash\ e_1\ :\ U[z/x, z/y, (\mathsf{refl}_T\ z)/p] \qquad \Gamma \vdash\ e_2\ :\ T\end{array}}{\Gamma \vdash \mathsf{G}_T([x\,y\,p\,.\,U], [z\,.\,e_1], \mathsf{refl}_T\ e_2, e_2, e_2) \equiv e_1[e_2/z] : U[e_2/x, e_2/x, (\mathsf{refl}_T\ e_2)/p]}$$

**Congruence rules for names**

CONG-NAME-PROD
$$\frac{\Gamma \vdash e_1 \equiv e_1' : \mathbb{U}_\alpha \qquad \Gamma,\, x\!:\!\mathsf{El}^\alpha\, e_1 \vdash e_2 \equiv e_2' : \mathbb{U}_\beta \qquad \mathsf{max}(\alpha, \beta) = \gamma}{\Gamma \vdash (\pi^{\alpha,\beta} x\!:\!e_1\,.\,e_2) \equiv (\pi^{\alpha,\beta} x\!:\!e_1'\,.\,e_2') : \mathbb{U}_\gamma}$$

CONG-NAME-UNIVERSE
$$\frac{\Gamma\ \mathsf{ctx} \qquad \mathsf{succ}(\alpha) = \gamma}{\Gamma \vdash u_\alpha \equiv u_\alpha : \mathbb{U}_\gamma}$$

CONG-NAME-PATHS
$$\frac{\alpha \in \mathbb{F} \qquad \Gamma \vdash e_1 \equiv e_1' : \mathbb{U}_\alpha \qquad \Gamma \vdash e_2 \equiv e_2' : \mathsf{El}^\alpha\, e_1 \qquad \Gamma \vdash e_3 \equiv e_3' : \mathsf{El}^\alpha\, e_1}{\Gamma \vdash \mathsf{paths}^\alpha_{e_1}(e_2, e_3) \equiv \mathsf{paths}^\alpha_{e_1'}(e_2', e_3') : \mathbb{U}_\alpha}$$

CONG-NAME-ID
$$\frac{\Gamma \vdash e_1 \equiv e_1' : \mathbb{U}_\alpha \qquad \Gamma \vdash e_2 \equiv e_2' : \mathsf{El}^\alpha\, e_1 \qquad \Gamma \vdash e_3 \equiv e_3' : \mathsf{El}^\alpha\, e_1}{\Gamma \vdash \mathsf{id}^\alpha_{e_1}(e_2, e_3) \equiv \mathsf{id}^\alpha_{e_1'}(e_2', e_3') : \mathbb{U}_\alpha}$$

**Congruence rule for coercions**

$$\frac{\Gamma \vdash e_1 \equiv e_2 : \mathbb{U}_\alpha \qquad \alpha \leq \beta}{\Gamma \vdash (\mathsf{coerce}^{\alpha \mapsto \beta}\, e_1) \equiv (\mathsf{coerce}^{\alpha \mapsto \beta}\, e_2) : \mathbb{U}_\beta}\ \textsc{cong-name-coerce}$$

**Functoriality of coercions**

$$\frac{\Gamma \vdash e : \mathbb{U}_\alpha}{\Gamma \vdash (\mathsf{coerce}^{\alpha \mapsto e}\,) \equiv e : \mathbb{U}_\alpha}\ \textsc{eq-name-coerce-trivial} \qquad \frac{\Gamma \vdash e : \mathbb{U}_\alpha \qquad \alpha \leq \beta \leq \gamma}{\Gamma \vdash \mathsf{coerce}^{\beta \mapsto \gamma}\,(\mathsf{coerce}^{\alpha \mapsto \beta}\, e)) \equiv \mathsf{coerce}^{\alpha \mapsto \gamma}\, e : \mathbb{U}_\gamma}\ \textsc{eq-name-coerce-trans}$$

# 3 Algorithmic formulation

We now define a bidirectional version of type theory amenable to algorithmic treatment. We replace the equality eliminator with equality and rewrite hints.

## 3.1 Syntax

Contexts:

$$\begin{array}{rcll} \Gamma & ::= & \bullet & \text{empty context} \\ & \mid & \Gamma,\, x : T & \text{context extended with } x : T \end{array}$$

Equality hints:

$$\begin{array}{rcll} \mathcal{H} & ::= & \circ & \text{empty hints} \\ & \mid & \mathcal{H}, (e_1 \equiv e_2)T & \text{extend hints with an equation} \\ & \mid & \mathcal{H}, (e_1 \rightsquigarrow e_2)T & \text{extend hints with a reduction} \end{array}$$

Types:

$$\begin{array}{rcll} T, U & ::= & \mathbb{U}_\alpha & \text{universe} \\ & \mid & \mathsf{El}^{\alpha}\, e & \text{type named by } e \\ & \mid & \mathsf{Unit} & \text{the unit type} \\ & \mid & \prod_{(x:T)} U & \text{product} \\ & \mid & \mathsf{Paths}_T(e_1, e_2) & \text{path type} \\ & \mid & \mathsf{Id}_T(e_1, e_2) & \text{equality type} \end{array}$$

Terms:

| | | |
|---|---|---|
| $e$ ::= | $x$ | variable |
| \| | $\mathsf{equation}\ e_1 : e_2 \equiv e_3\ \mathsf{in}\ e_4$ | use equality hint $e_1$ in $e_4$ |
| \| | $\mathsf{rewrite}\ e_1 : e_2 \equiv e_3\ \mathsf{in}\ e_4$ | use rewrite hint $e_1$ in $e_4$ |
| \| | $e :: T$ | ascribe type $T$ to term $e$ |
| \| | $\lambda x{:}T_1.T_2 \,.\, e$ | $\lambda$-abstraction |
| \| | $e_1 \mathbin{@^{x:T_1.T_2}} e_2$ | application |
| \| | $\star$ | the element of unit type |
| \| | $\mathsf{idpath}_T\ e$ | identity path |
| \| | $\mathsf{J}_T([x\, y\, p \,.\, U], [z \,.\, e_1], e_2, e_3, e_4)$ | path eliminator |
| \| | $\mathsf{refl}_T\ e$ | reflexivity |
| \| | $\mathsf{coerce}^{\alpha \mapsto \beta}\ e$ | universe coercion |
| \| | $\mathsf{unit}$ | the name of unit type |
| \| | $\pi^{\alpha,\beta} x : e_1 \,.\, e_2$ | the name of product type |
| \| | $u_\alpha$ | the name of a universe |
| \| | $\mathsf{paths}^{\alpha}_{e_1}(e_2, e_3)$ | the name of a path type |
| \| | $\mathsf{id}^{\alpha}_{e_1}(e_2, e_3)$ | the name of an equality type |

The annotations marked like this could be omitted from the initial input because they can be reconstructed on the fly during type checking/synthesis. They are extremely useful during equivalence checking.

## 3.2 Judgments

| | |
|---|---|
| $\Gamma \vdash \mathcal{H}\ \mathsf{hints}$ | $\mathcal{H}$ consists of legal hints |
| $\Gamma\,;\,\mathcal{H} \vdash T \Leftarrow \mathsf{type}$ | $T$ is a well-formed type |
| $\Gamma\,;\,\mathcal{H} \vdash T \Leftarrow \mathsf{fibered}$ | $T$ is a well-formed fibered type |
| $\Gamma\,;\,\mathcal{H} \vdash e \Leftarrow T$ | check that term $e$ has type $T$ |
| $\Gamma\,;\,\mathcal{H} \vdash e \Rightarrow T$ | synthesize type $T$ of term $e$ |
| $\Gamma\,;\,\mathcal{H} \vdash T \approx U$ | $T$ and $U$ are equal types |
| $\Gamma\,;\,\mathcal{H} \vdash T \sim U$ | $T$ and $U$ are equal normal types |
| $\Gamma\,;\,\mathcal{H} \vdash e_1 \approx e_2 \Leftarrow T$ | $e_1$ and $e_2$ are equal terms of type $T$ |
| $\Gamma\,;\,\mathcal{H} \vdash e_1 \simeq e_2 \Leftarrow T$ | $e_1$ and $e_2$ are equal terms of normal type $T$ |
| $\Gamma\,;\,\mathcal{H} \vdash e_1 \sim e_2$ | $e_1$ and $e_2$ are equal normal terms of same type |
| $\Gamma\,;\,\mathcal{H} \vdash T_1 \rightsquigarrow T_2 / \mathcal{H}'$ | type $T_1$ has a reduction step to $T_2$ yielding hints $\mathcal{H}'$ |
| $\Gamma\,;\,\mathcal{H} \vdash T_1 \rightsquigarrow^* T_2 \not\rightsquigarrow$ | type $T_1$ normalizes to $T_2$ |
| $\Gamma\,;\,\mathcal{H} \vdash e_1 \rightsquigarrow e_2 / \mathcal{H}'$ | term $e_1$ has a reduction step to $e_2$ yielding hints $\mathcal{H}'$ |
| $\Gamma\,;\,\mathcal{H} \vdash e_1 \rightsquigarrow^* e_2 \not\rightsquigarrow$ | term $e_1$ normalizes to $e_2$ |

## 3.3 Contexts with hints

$$\textsc{Hint-Empty}\ \dfrac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \circ\ \mathsf{hints}} \qquad \textsc{Hint-Eq}\ \dfrac{\Gamma \vdash \mathcal{H}\ \mathsf{hints} \qquad \Gamma \vdash e_1 \equiv e_2 : T}{\Gamma \vdash (\mathcal{H}, (e_1 \equiv e_2))\ \mathsf{hints}}$$

$$\textsc{Hint-Rw}\ \dfrac{\Gamma \vdash \mathcal{H}\ \mathsf{hints} \qquad \Gamma \vdash e_1 \equiv e_2 : T}{\Gamma \vdash (\mathcal{H}, (e_1 \rightsquigarrow e_2))\ \mathsf{hints}}$$

Note that this judgment is *not* invoked by the algorithm. It is used only in describing the conditions under which the algorithm is expected to work.

## 3.4 Well-formed types

$$\textsc{TyChk-Universe}\ \dfrac{\alpha \in \mathbb{P}}{\Gamma ; \mathcal{H} \vdash \mathbb{U}_\alpha \Leftarrow \mathsf{type}} \qquad \textsc{TyChk-Prod}\ \dfrac{\Gamma ; \mathcal{H} \vdash T \Leftarrow \mathsf{type} \qquad (\Gamma,\ x : T) ; \mathcal{H} \vdash U \Leftarrow \mathsf{type}}{\Gamma ; \mathcal{H} \vdash \prod_{(x:T)} U \Leftarrow \mathsf{type}}$$

$$\textsc{TyChk-El}\ \dfrac{\Gamma ; \mathcal{H} \vdash e \Rightarrow T \qquad \Gamma ; \mathcal{H} \vdash T \rightsquigarrow^* \mathbb{U}_\alpha \not\rightsquigarrow}{\Gamma ; \mathcal{H} \vdash \mathsf{El}^{\alpha}\ e \Leftarrow \mathsf{type}} \qquad \textsc{TyChk-Unit}\ \dfrac{}{\Gamma ; \mathcal{H} \vdash \mathsf{Unit} \Leftarrow \mathsf{type}}$$

$$\textsc{TyChk-Paths}\ \dfrac{\Gamma ; \mathcal{H} \vdash e_1 \Rightarrow T \qquad \mathsf{isfib}\ T = \mathsf{true} \qquad \Gamma ; \mathcal{H} \vdash e_2 \Leftarrow T}{\Gamma ; \mathcal{H} \vdash \mathsf{Paths}_{T}(e_1, e_2) \Leftarrow \mathsf{type}}$$

$$\textsc{TyChk-Id}\ \dfrac{\Gamma ; \mathcal{H} \vdash e_1 \Rightarrow T \qquad \Gamma ; \mathcal{H} \vdash e_2 \Leftarrow T}{\Gamma ; \mathcal{H} \vdash \mathsf{Id}_{T}(e_1, e_2) \Leftarrow \mathsf{type}}$$

## 3.5 Well-formed fibered types

$$\textsc{FibChk-Universe}\ \dfrac{\alpha \in \mathbb{P}}{\Gamma ; \mathcal{H} \vdash \mathbb{U}_\alpha \Leftarrow \mathsf{fibered}} \qquad \textsc{FibChk-Prod}\ \dfrac{\Gamma ; \mathcal{H} \vdash T \Leftarrow \mathsf{fibered} \qquad (\Gamma,\ x : T) ; \mathcal{H} \vdash U \Leftarrow \mathsf{fibered}}{\Gamma ; \mathcal{H} \vdash \prod_{(x:T)} U \Leftarrow \mathsf{fibered}}$$

$$\textsc{FibChk-El}\ \dfrac{\Gamma ; \mathcal{H} \vdash e \Rightarrow \mathbb{U}_\alpha \qquad \alpha \in \mathbb{F}}{\Gamma ; \mathcal{H} \vdash \mathsf{El}^{\alpha}\ e \Leftarrow \mathsf{fibered}} \qquad \textsc{FibChk-Unit}\ \dfrac{o \in \mathbb{F}}{\Gamma ; \mathcal{H} \vdash \mathsf{Unit} \Leftarrow \mathsf{fibered}}$$

$$\textsc{FibChk-Paths}\ \dfrac{\Gamma ; \mathcal{H} \vdash e_1 \Rightarrow T \qquad \mathsf{isfib}\ T = \mathsf{true} \qquad \Gamma ; \mathcal{H} \vdash e_2 \Leftarrow T}{\Gamma ; \mathcal{H} \vdash \mathsf{Paths}_{T}(e_1, e_2) \Leftarrow \mathsf{fibered}}$$

If we already know the given type is well-formed, there's an even simpler algorithm to see if it is fibered:

$$
\begin{aligned}
\mathsf{isfib}\ \mathbb{U}_\alpha &:= \mathsf{true}\\
\mathsf{isfib}\ \textstyle\prod_{(x:T)} U &:= \mathsf{isfib}\ T \ \wedge\ \mathsf{isfib}\ U\\
\mathsf{isfib}\ \mathsf{El}^\alpha\, e &:= \alpha \in \mathbb{F}\\
\mathsf{isfib}\ \mathsf{Unit} &:= o \in \mathbb{F}\\
\mathsf{isfib}\ \mathsf{Paths}_T(e_1, e_2) &:= \mathsf{true}\\
\mathsf{isfib}\ \mathsf{Id}_T(e_1, e_2) &:= \mathsf{false}
\end{aligned}
$$

## 3.6 Terms

### General rules

SYN-VAR
$$\frac{(x{:}T) \in \Gamma}{\Gamma\,;\,\mathcal{H} \vdash x \Rightarrow T}$$

SYN-ASCRIBE
$$\frac{\Gamma\,;\,\mathcal{H} \vdash T \Leftarrow \mathsf{type} \qquad \Gamma\,;\,\mathcal{H} \vdash e \Leftarrow T}{\Gamma\,;\,\mathcal{H} \vdash e :: T \Rightarrow T}$$

CHK-SYN
$$\frac{\Gamma\,;\,\mathcal{H} \vdash e \Rightarrow U \qquad \Gamma\,;\,\mathcal{H} \vdash U \approx T}{\Gamma\,;\,\mathcal{H} \vdash e \Leftarrow T}$$

### Hints

SYN-EQUATION-HINT
$$\frac{\begin{array}{c}\Gamma\,;\,\mathcal{H} \vdash e_1 \Rightarrow U' \qquad \Gamma\,;\,\mathcal{H} \vdash U' \rightsquigarrow^* \mathsf{Id}_U(e_2, e_3) \not\rightsquigarrow\\ \Gamma\,;\,(\mathcal{H}, (e_2 \equiv e_3)) \vdash e_4 \Rightarrow T\end{array}}{\Gamma\,;\,\mathcal{H} \vdash (\mathsf{equation}\ e_1 : e_2 \equiv e_3\ \mathsf{in}\ e_4) \Rightarrow T}$$

CHK-EQUATION-HINT
$$\frac{\begin{array}{c}\Gamma\,;\,\mathcal{H} \vdash e_1 \Rightarrow U' \qquad \Gamma\,;\,\mathcal{H} \vdash U' \rightsquigarrow^* \mathsf{Id}_U(e_2, e_3) \not\rightsquigarrow\\ \Gamma\,;\,(\mathcal{H}, (e_2 \equiv e_3)) \vdash e_4 \Leftarrow T\end{array}}{\Gamma\,;\,\mathcal{H} \vdash (\mathsf{equation}\ e_1 : e_2 \equiv e_3\ \mathsf{in}\ e_4) \Leftarrow T}$$

SYN-RW-HINT
$$\frac{\begin{array}{c}\Gamma\,;\,\mathcal{H} \vdash e_1 \Rightarrow U' \qquad \Gamma\,;\,\mathcal{H} \vdash U' \rightsquigarrow^* \mathsf{Id}_U(e_2, e_3) \not\rightsquigarrow\\ \Gamma\,;\,(\mathcal{H}, (e_2 \rightsquigarrow e_3)) \vdash e_4 \Rightarrow T\end{array}}{\Gamma\,;\,\mathcal{H} \vdash (\mathsf{rewrite}\ e_1 : e_2 \equiv e_3\ \mathsf{in}\ e_4) \Rightarrow T}$$

CHK-RW-HINT
$$\frac{\begin{array}{c}\Gamma\,;\,\mathcal{H} \vdash e_1 \Rightarrow U' \qquad \Gamma\,;\,\mathcal{H} \vdash U' \rightsquigarrow^* \mathsf{Id}_U(e_2, e_3) \not\rightsquigarrow\\ \Gamma\,;\,(\mathcal{H}, (e_2 \rightsquigarrow e_3)) \vdash e_4 \Leftarrow T\end{array}}{\Gamma\,;\,\mathcal{H} \vdash (\mathsf{rewrite}\ e_1 : e_2 \equiv e_3\ \mathsf{in}\ e_4) \Leftarrow T}$$

Rules CHK-EQUATION-HINT and CHK-REDUCE-HINT are among the few rules that perform checking differently than the default Rule CHK-SYN. When they apply, we try them first (despite Rule CHK-SYN being listed first in this section). In practice, if the rule fails, we will not backtrack and try to check without the hint.

**Products**

$$\frac{\Gamma\,;\,\mathcal{H} \vdash T \Leftarrow \mathsf{type} \qquad \Gamma,\, x : T; \mathcal{H} \vdash e \Rightarrow U}{\Gamma\,;\,\mathcal{H} \vdash (\lambda x{:}T.U\,.\,e) \Rightarrow \prod_{(x:T)} U}\ \textsc{Syn-Abs}$$

$$\frac{\Gamma\,;\,\mathcal{H} \vdash e_1 \Rightarrow T_1 \qquad \Gamma\,;\,\mathcal{H} \vdash T_1 \rightsquigarrow^* \prod_{(x:T)} U \not\rightsquigarrow \qquad \Gamma\,;\,\mathcal{H} \vdash e_2 \Leftarrow T}{\Gamma\,;\,\mathcal{H} \vdash e_1 \mathbin{@^{x:T.U}} e_2 \Rightarrow U[e_2/x]}\ \textsc{Syn-App}$$

**Unit type**

$$\frac{}{\Gamma\,;\,\mathcal{H} \vdash \star \Rightarrow \mathsf{Unit}}\ \textsc{Syn-Unit}$$

**Path type**

$$\frac{\Gamma\,;\,\mathcal{H} \vdash e \Rightarrow T}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{idpath}_T\ e \Rightarrow \mathsf{Paths}_T(e, e)}\ \textsc{Syn-IdPath}$$

$$\frac{\begin{array}{c}\Gamma\,;\,\mathcal{H} \vdash e_2 \Rightarrow T_2 \qquad \Gamma\,;\,\mathcal{H} \vdash T_2 \rightsquigarrow^* \mathsf{Paths}_T(e_3, e_4) \not\rightsquigarrow \\ \Gamma,\, x : T,\, y : T,\, p : \mathsf{Paths}_T(x, y); \mathcal{H} \vdash U \Leftarrow \mathsf{fibered} \\ \Gamma,\, z : T; \mathcal{H} \vdash e_1 \Leftarrow U[z/x, z/y, (\mathsf{idpath}_T\ z)/p]\end{array}}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{J}_T([x\,y\,p\,.\,U], [z\,.\,e_1], e_2, e_3, e_4) \Rightarrow U[e_3/x, e_4/y, e_2/p]}\ \textsc{Syn-J}$$

**Equality type**

$$\frac{\Gamma\,;\,\mathcal{H} \vdash e \Rightarrow T}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{refl}_T\ e \Rightarrow \mathsf{Id}_T(e, e)}\ \textsc{Syn-Refl}$$

**Names**

$$\frac{\text{SYN-NAME-UNIT}}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{unit} \Rightarrow \mathbb{U}_o}$$

$$\text{SYN-NAME-UNIVERSE}\quad \frac{\beta = \mathsf{succ}(\alpha)}{\Gamma\,;\,\mathcal{H} \vdash u_\alpha \Rightarrow \mathbb{U}_\beta}$$

$$\text{SYN-NAME-PROD}\quad \frac{\begin{array}{cc} \Gamma\,;\,\mathcal{H} \vdash e_1 \Rightarrow T_1 & \Gamma\,;\,\mathcal{H} \vdash T_1 \rightsquigarrow^* \mathbb{U}_\alpha \not\rightsquigarrow \\ (\Gamma,\, x : \mathsf{El}^\alpha\, e_1)\,;\,\mathcal{H} \vdash e_2 \Rightarrow T_2 & (\Gamma,\, x : \mathsf{El}^\alpha\, e_1)\,;\,\mathcal{H} \vdash T_2 \rightsquigarrow^* \mathbb{U}_\beta \not\rightsquigarrow \\ \multicolumn{2}{c}{\mathsf{max}(\alpha, \beta) = \gamma} \end{array}}{\Gamma\,;\,\mathcal{H} \vdash (\pi^{\alpha,\beta} x : e_1 \,.\, e_2) \Rightarrow \mathbb{U}_\gamma}$$

$$\text{SYN-NAME-COERCE}\quad \frac{\Gamma\,;\,\mathcal{H} \vdash e \Rightarrow T \qquad \Gamma\,;\,\mathcal{H} \vdash T \rightsquigarrow^* \mathbb{U}_\alpha \not\rightsquigarrow \qquad \alpha \leq \beta}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{coerce}^{\alpha \mapsto \beta}\, e \Rightarrow \mathbb{U}_\beta}$$

$$\text{SYN-NAME-PATHS}\quad \frac{\Gamma\,;\,\mathcal{H} \vdash e_2 \Rightarrow T_2 \qquad \mathsf{name\_of}\ T_2 = e_1 : \mathbb{U}_\alpha \qquad \Gamma\,;\,\mathcal{H} \vdash e_3 \Leftarrow T_2 \qquad \alpha \in \mathbb{F}}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{paths}^{\alpha}_{e_1}(e_2, e_3) \Rightarrow \mathbb{U}_\alpha}$$

$$\text{SYN-NAME-ID}\quad \frac{\Gamma\,;\,\mathcal{H} \vdash e_2 \Rightarrow T_2 \qquad \mathsf{name\_of}\ T_2 = e_1 : \mathbb{U}_\alpha \qquad \Gamma\,;\,\mathcal{H} \vdash e_3 \Leftarrow T_2}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{id}^{\alpha}_{e_1}(e_2, e_3) \Rightarrow \mathbb{U}_\alpha}$$

## 3.7 Type normalization

**Name reduction (possibly using hints)**

$$\text{TYNORM-EL}\quad \frac{\Gamma\,;\,\mathcal{H} \vdash e \rightsquigarrow e' / \mathcal{H}'}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{El}^\alpha\, e \rightsquigarrow \mathsf{El}^\alpha\, e' / \mathcal{H}'}$$

**Conversion from name to type**

$$\textsc{tynorm-pi}\quad \frac{}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{El}^{\alpha}\,(\pi^{\beta,\gamma} x : e_1 \,.\, e_2) \rightsquigarrow \prod_{(x:\mathsf{El}^{\beta}\ e_1)} \mathsf{El}^{\gamma}\ e_2 / \mathcal{H}}$$

$$\textsc{tynorm-unit}\quad \frac{}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{El}^{\alpha}\ \mathsf{unit} \rightsquigarrow \mathsf{Unit}/\mathcal{H}} \qquad \textsc{tynorm-universe}\quad \frac{}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{El}^{\alpha}\ u_{\beta} \rightsquigarrow \mathbb{U}_{\beta}/\mathcal{H}}$$

$$\textsc{tynorm-coerce}\quad \frac{}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{El}^{\alpha}\,(\mathsf{coerce}^{\beta \mapsto \gamma}\ e) \rightsquigarrow \mathsf{El}^{\alpha}\ e/\mathcal{H}}$$

$$\textsc{tynorm-paths}\quad \frac{}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{El}^{\alpha}\,(\mathsf{paths}^{\beta}_{e_1}(e_2, e_3)) \rightsquigarrow \mathsf{Paths}_{\mathsf{El}^{\alpha}\ e_1}(e_2, e_3)/\mathcal{H}}$$

$$\textsc{tynorm-id}\quad \frac{}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{El}^{\alpha}\,(\mathsf{id}^{\beta}_{e_1}(e_2, e_3)) \rightsquigarrow \mathsf{Id}_{\mathsf{El}^{\alpha}\ e_1}(e_2, e_3)/\mathcal{H}}$$

If $(\mathsf{unit} \rightsquigarrow \mathsf{nat}) \in \mathcal{H}$, then rule ordering will give us $\Gamma\,;\,\mathcal{H} \vdash \mathsf{El}^{o}\ \mathsf{unit} \rightsquigarrow \mathsf{El}^{o}\ \mathsf{nat}/\mathcal{H}$. Only if we were to backtrack (and we don't, in the current implementation) would we consider $\Gamma\,;\,\mathcal{H} \vdash \mathsf{El}^{o}\ \mathsf{unit} \rightsquigarrow \mathsf{Unit}/\mathcal{H}$.

## 3.8 Type equality

**General Type equality**

$$\textsc{chk-tyeq-refl}\quad \frac{}{\Gamma\,;\,\mathcal{H} \vdash T \approx T}$$

$$\textsc{chk-tyeq-hnf}\quad \frac{\Gamma\,;\,\mathcal{H} \vdash T \rightsquigarrow^{*} T' \not\rightsquigarrow \qquad \Gamma\,;\,\mathcal{H} \vdash U \rightsquigarrow^{*} U' \not\rightsquigarrow \qquad \Gamma\,;\,\mathcal{H} \vdash T' \sim U'}{\Gamma\,;\,\mathcal{H} \vdash T \approx U}$$

Reflexivity is an optimization for the common case.

**Equality of head-normal forms**

CHK-TYEQ-PATH-REFL
$$\frac{}{\Gamma\,;\,\mathcal{H}\vdash T\sim T}$$

CHK-TYEQ-EL
$$\frac{\alpha=\beta \qquad \Gamma\,;\,\mathcal{H}\vdash e_1\approx e_2\Leftarrow\mathbb{U}_\alpha}{\Gamma\,;\,\mathcal{H}\vdash \mathsf{El}^{\alpha}\,e_1\sim\mathsf{El}^{\beta}\,e_2}$$

CHK-TYEQ-PROD
$$\frac{\Gamma\,;\,\mathcal{H}\vdash T_1\approx U_1 \qquad (\Gamma,\,x\,{:}\,T_1)\,;\,\mathcal{H}\vdash T_2\approx U_2}{\Gamma\,;\,\mathcal{H}\vdash \prod_{(x:T_1)}T_2\sim\prod_{(x:U_1)}U_2}$$

CHK-TYEQ-PATHS
$$\frac{\Gamma\,;\,\mathcal{H}\vdash T\approx U \qquad \Gamma\,;\,\mathcal{H}\vdash e_1\approx e_1'\Leftarrow T \qquad \Gamma\,;\,\mathcal{H}\vdash e_2\approx e_2'\Leftarrow T}{\Gamma\,;\,\mathcal{H}\vdash \mathsf{Paths}_T(e_1,e_2)\sim\mathsf{Paths}_U(e_1',e_2')}$$

CHK-TYEQ-ID
$$\frac{\Gamma\,;\,\mathcal{H}\vdash T\approx U \qquad \Gamma\,;\,\mathcal{H}\vdash e_1\approx e_1'\Leftarrow T \qquad \Gamma\,;\,\mathcal{H}\vdash e_2\approx e_2'\Leftarrow T}{\Gamma\,;\,\mathcal{H}\vdash \mathsf{Id}_T(e_1,e_2)\sim\mathsf{Id}_U(e_1',e_2')}$$

The reflexivity rule is not just an optimization, but also handles equivalence of base types and equivalence of universes.

## 3.9 Term normalization

**Normalization by hints**

NORM-HINT
$$\frac{(e\rightsquigarrow e')\in\mathcal{H} \qquad \Gamma\,;\,\mathcal{H}\vdash e'\rightsquigarrow e''/\mathcal{H}}{\Gamma\,;\,\mathcal{H}\vdash e\rightsquigarrow e''/\mathcal{H}}$$

In the case of multiple hints for reducing $e'$, we try the most-recently-added hint (last in the sequence $\mathcal{H}$) first.

**Redices**

NORM-EQUATION
$$\frac{}{\Gamma\,;\,\mathcal{H}\vdash(\mathsf{equation}\ e_1:e_2\equiv e_3\ \mathsf{in}\ e_4)\rightsquigarrow e_4/\mathcal{H}}$$

NORM-REWRITE
$$\frac{}{\Gamma\,;\,\mathcal{H}\vdash(\mathsf{rewrite}\ e_1:e_2\equiv e_3\ \mathsf{in}\ e_4)\rightsquigarrow e_4/\mathcal{H}}$$

NORM-ASCRIBE
$$\frac{}{\Gamma\,;\,\mathcal{H}\vdash(e::T)\rightsquigarrow e/}$$

NORM-APP-BETA
$$\frac{\Gamma\,;\,\mathcal{H}\vdash T_1\approx U_1 \qquad \Gamma,\,x\,{:}\,T_1\,;\,\mathcal{H}\vdash T_2\approx U_2}{\Gamma\,;\,\mathcal{H}\vdash(\lambda x{:}T_1.T_2\,.\,e_1)\ @^{x:U_1.U_2}\ e_2\rightsquigarrow e_1[e_2/x]/}$$

NORM-IDPATH
$$\frac{\Gamma\,;\,\mathcal{H}\vdash T\approx T'}{\Gamma\,;\,\mathcal{H}\vdash \mathsf{J}_T([x\,y\,p\,.\,U],[z\,.\,e_1],\mathsf{idpath}_{T'}\ e_2,e_3,e_4)\rightsquigarrow e_1[e_2/z]/}$$

$$\frac{\alpha = \beta}{\Gamma\,;\,\mathcal{H} \vdash (\mathsf{coerce}^{\alpha\mapsto\beta}\; e) \rightsquigarrow e/\mathcal{H}}\;\text{NORM-COERCE-TRIVIAL}$$

Rule NORM-COERCE-TRIVIAL makes no sense if we don't label universe coercion terms with both domain and codomain universes.

$$\frac{}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{coerce}^{\beta\mapsto\gamma}\;(\mathsf{coerce}^{\alpha\mapsto\beta}\; e) \rightsquigarrow \mathsf{coerce}^{\alpha\mapsto\gamma}\; e/\mathcal{H}}\;\text{NORM-COERCE-TRANS}$$

**Recursion**

$$\frac{\Gamma\,;\,\mathcal{H} \vdash e_1 \rightsquigarrow e_1'/\mathcal{H}'}{\Gamma\,;\,\mathcal{H} \vdash (e_1 \;@^{x:T.U}\; e_2) \rightsquigarrow (e_1' \;@^{x:T.U}\; e_2)/\mathcal{H}'}\;\text{NORM-APP}$$

$$\frac{\Gamma\,;\,\mathcal{H} \vdash e_2 \rightsquigarrow e_2'/\mathcal{H}'}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{J}_T([x\,y\,p\,.\,U],[z\,.\,e_1],e_2,e_3,e_4) \rightsquigarrow \mathsf{J}_T([x\,y\,p\,.\,U],[z\,.\,e_1],e_2',e_3,e_4)/\mathcal{H}'}\;\text{NORM-J}$$

By rule priority, if $e_1 \;@^{x:T.U}\; e_2$ has no hint and we take one step to $e_1' \;@^{x:T.U}\; e_2$, we will look again for a hint for that term before applying the other rules.

## 3.10 Term equality

For algorithmic purposes we should try to apply reflexivity and hints before doing anything else:

$$\frac{}{\Gamma\,;\,\mathcal{H} \vdash e \approx e \Leftarrow T}\;\text{CHK-EQ-REFL} \qquad \frac{(e_1 \equiv e_2) \in \mathcal{H}}{\Gamma\,;\,\mathcal{H} \vdash e_1 \approx e_2 \Leftarrow T}\;\text{CHK-EQ-HINT} \qquad \frac{(e_2 \equiv e_1) \in \mathcal{H}}{\Gamma\,;\,\mathcal{H} \vdash e_1 \approx e_2 \Leftarrow T}\;\text{CHK-EQ-HINT-SYM}$$

Otherwise, we check whether extensionality applies:

$$\frac{\Gamma\,;\,\mathcal{H} \vdash T \rightsquigarrow^* T' \not\rightsquigarrow \qquad \Gamma\,;\,\mathcal{H} \vdash e_1 \simeq e_2 \Leftarrow T'}{\Gamma\,;\,\mathcal{H} \vdash e_1 \approx e_2 \Leftarrow T}\;\text{CHK-EQ-EXT}$$

**Extensionality**

$$\frac{(\Gamma,\, x\!:\!T)\,;\,\mathcal{H} \vdash (e_1 \;@^{x:T.U}\; x) \approx (e_2 \;@^{x:T.U}\; x) \Leftarrow U}{\Gamma\,;\,\mathcal{H} \vdash e_1 \simeq e_2 \Leftarrow \prod_{(x:T)} U}\;\text{CHK-EQ-EXT-PROD} \qquad \frac{}{\Gamma\,;\,\mathcal{H} \vdash e_1 \simeq e_2 \Leftarrow \mathsf{Unit}}\;\text{CHK-EQ-EXT-UNIT}$$

$$\frac{}{\Gamma\,;\,\mathcal{H} \vdash e_1 \simeq e_2 \Leftarrow \mathsf{Id}_T(e_3,e_4)}\;\text{CHK-EQ-EXT-K} \qquad \frac{\Gamma\,;\,\mathcal{H} \vdash e_1 \rightsquigarrow^* e_1' \not\rightsquigarrow \qquad \Gamma\,;\,\mathcal{H} \vdash e_2 \rightsquigarrow^* e_2' \not\rightsquigarrow \qquad \Gamma\,;\,\mathcal{H} \vdash e_1' \sim e_2'}{\Gamma\,;\,\mathcal{H} \vdash e_1 \simeq e_2 \Leftarrow T}\;\text{CHK-EQ-EXT-WHNF}$$

In CHK-EQ-EXT-WHNF, we might want to check whether $e'_1$ and $e'_2$ are the same expressions before invoking the general comparison function.

### Whnf equivalence

CHK-EQ-WHNF-REFLEXIVITY
$$\frac{}{\Gamma\,;\,\mathcal{H} \vdash e \sim e}$$

CHK-EQ-WHNF-EQUATION
$$\frac{(e_1 \equiv e_2) \in \mathcal{H}}{\Gamma\,;\,\mathcal{H} \vdash e_1 \sim e_2}$$

CHK-EQ-WHNF-VAR
$$\frac{}{\Gamma\,;\,\mathcal{H} \vdash x \sim x}$$

CHK-EQ-WHNF-APP
$$\frac{\begin{array}{cc}\Gamma\,;\,\mathcal{H} \vdash T_1 \approx U_1 & (\Gamma,\, x\!:\!T_1)\,;\,\mathcal{H} \vdash T_2 \approx U_2 \\ \Gamma\,;\,\mathcal{H} \vdash e_1 \sim e'_1 & \Gamma\,;\,\mathcal{H} \vdash e_2 \approx e'_2 \Leftarrow T_1\end{array}}{\Gamma\,;\,\mathcal{H} \vdash (e_1 \mathbin{@^{x:T_1.T_2}} e_2) \sim (e'_1 \mathbin{@^{x:U_1.U_2}} e'_2)}$$

CHK-EQ-WHNF-IDPATH
$$\frac{\Gamma\,;\,\mathcal{H} \vdash T \approx U \qquad \Gamma\,;\,\mathcal{H} \vdash e_1 \approx e_2 \Leftarrow T}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{idpath}_T\, e_1 \sim \mathsf{idpath}_U\, e_2}$$

CHK-EQ-WHNF-J
$$\frac{\begin{array}{c}\Gamma\,;\,\mathcal{H} \vdash T \approx T' \qquad (\Gamma,\, x\!:\!T,\, y\!:\!T,\, p\!:\!\mathsf{Paths}_T(x,y))\,;\,\mathcal{H} \vdash U \approx U' \\ (\Gamma,\, z\!:\!T)\,;\,\mathcal{H} \vdash e_1 \approx e'_1 \Leftarrow P[z/x, z/y, (\mathsf{idpath}_T\, z)/p] \\ \Gamma\,;\,\mathcal{H} \vdash e_3 \approx e'_3 \Leftarrow T \qquad \Gamma\,;\,\mathcal{H} \vdash e_4 \approx e'_4 \Leftarrow T \qquad \Gamma\,;\,\mathcal{H} \vdash e_2 \sim e'_2\end{array}}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{J}_T([x\,y\,p\,.\,U], [z\,.\,e_1], e_2, e_3, e_4) \sim \mathsf{J}_{T'}([x\,y\,p\,.\,U'], [z\,.\,e'_1], e'_2, e'_3, e'_4)}$$

CHK-EQ-WHNF-REFL
$$\frac{\Gamma\,;\,\mathcal{H} \vdash T \approx U \qquad \Gamma\,;\,\mathcal{H} \vdash e_1 \approx e_2 \Leftarrow T}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{refl}_T\, e_1 \sim \mathsf{refl}_U\, e_2}$$

**Whnf equivalence of names**

CHK-EQ-WHNF-PROD
$$\frac{\alpha = \alpha' \qquad \beta = \beta' \qquad \Gamma\,;\,\mathcal{H} \vdash e_1 \approx e_1' \Leftarrow \mathbb{U}_\alpha \qquad (\Gamma,\, x : \mathsf{El}^\alpha\, e_1)\,;\,\mathcal{H} \vdash e_2 \approx e_2' \Leftarrow \mathbb{U}_\beta}{\Gamma\,;\,\mathcal{H} \vdash (\pi^{\alpha,\beta} x : e_1\,.\,e_2) \sim (\pi^{\alpha',\beta'} x : e_1'\,.\,e_2')}$$

CHK-EQ-WHNF-UNIVERSE
$$\frac{\alpha = \beta}{\Gamma\,;\,\mathcal{H} \vdash u_\alpha \sim u_\beta}$$

CHK-EQ-WHNF-PATHS
$$\frac{\alpha = \alpha' \qquad \Gamma\,;\,\mathcal{H} \vdash e_1 \approx e_1' \Leftarrow \mathbb{U}_\alpha \qquad \Gamma\,;\,\mathcal{H} \vdash e_2 \approx e_2' \Leftarrow \mathsf{El}^\alpha\, e_1 \qquad \Gamma\,;\,\mathcal{H} \vdash e_3 \approx e_3' \Leftarrow \mathsf{El}^\alpha\, e_1}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{paths}^\alpha_{e_1}(e_2, e_3) \sim \mathsf{paths}^{\alpha'}_{e_1'}(e_2', e_3')}$$

CHK-EQ-WHNF-ID
$$\frac{\alpha = \alpha' \qquad \Gamma\,;\,\mathcal{H} \vdash e_1 \approx e_1' \Leftarrow \mathbb{U}_\alpha \qquad \Gamma\,;\,\mathcal{H} \vdash e_2 \approx e_2' \Leftarrow \mathsf{El}^\alpha\, e_1 \qquad \Gamma\,;\,\mathcal{H} \vdash e_3 \approx e_3' \Leftarrow \mathsf{El}^\alpha\, e_1}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{id}^\alpha_{e_1}(e_2, e_3) \sim \mathsf{id}^{\alpha'}_{e_1'}(e_2', e_3')}$$

CHK-EQ-WHNF-COERCE
$$\frac{\alpha = \alpha' \qquad \Gamma\,;\,\mathcal{H} \vdash e_1 \approx e_1' \Leftarrow \mathbb{U}_\alpha}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{coerce}^{\alpha \mapsto \beta}\, e_1 \sim \mathsf{coerce}^{\alpha' \mapsto \beta'}\, e_1'}$$

**Last-ditch whnf equivalence rules** These rules should not be necessary (due to eta), but sufficiently nasty hints might get us here.

CHK-EQ-WHNF-ABS
$$\frac{\Gamma\,;\,\mathcal{H} \vdash T_1 \approx U_1 \qquad (\Gamma,\, x : T_1)\,;\,\mathcal{H} \vdash T_2 \approx U_2 \qquad (\Gamma,\, x : T_1)\,;\,\mathcal{H} \vdash e_1 \approx e_2 \Leftarrow T_2}{\Gamma\,;\,\mathcal{H} \vdash (\lambda x{:}T_1.T_2\,.\,e_1) \sim (\lambda x{:}U_1.U_2\,.\,e_2)}$$

CHK-EQ-WHNF-UNIT-RIGHT
$$\frac{}{\Gamma\,;\,\mathcal{H} \vdash e \sim \star}$$

CHK-EQ-WHNF-UNIT-LEFT
$$\frac{}{\Gamma\,;\,\mathcal{H} \vdash \star \sim e}$$

CHK-EQ-WHNF-REFL-LEFT
$$\frac{}{\Gamma\,;\,\mathcal{H} \vdash \mathsf{refl}_T\, e_1 \sim e_2}$$

CHK-EQ-WHNF-REFL-RIGHT
$$\frac{}{\Gamma\,;\,\mathcal{H} \vdash e_1 \sim \mathsf{refl}_T\, e_2}$$

We rely heavily on the precondition that the two terms being compared are already known to be well-formed with the same type. Rule CHK-EQ-WHNF-ABS still has to conservatively re-check the equivalence of the type annotations; in the absence of Pi injectivity, just because we know the two lambda abstractions have judgmentally equal Pi types, there's no guarantee that they are componentwise equal.

## 3.11 Normalization

We provide a normalization procedure which ignores hints. It is used to compute whether a universally quantified equation applies in a given situation. We introduce the following judgments:

$$
\begin{array}{ll}
T \hookrightarrow T' & \text{type } T \text{ normalizes to } T' \\
e \hookrightarrow e' & \text{term } e \text{ normalizes to } e'
\end{array}
$$

Normalization preserves well-formedness and types.

**Types**

$$\frac{}{\mathbb{U}_\alpha \hookrightarrow \mathbb{U}_\alpha}\ \textsc{norm-ty-universe}$$

$$\frac{e \hookrightarrow \mathsf{coerce}^{\beta \mapsto \gamma}\, e' \qquad \gamma = \alpha \qquad \mathsf{El}^{\beta}\, e' \hookrightarrow T}{\mathsf{El}^{\alpha}\, e \hookrightarrow T}\ \textsc{norm-ty-el-coerce}$$

$$\frac{e \hookrightarrow \mathsf{unit} \qquad \alpha = o}{\mathsf{El}^{\alpha}\, e \hookrightarrow \mathsf{Unit}}\ \textsc{norm-ty-el-unit}$$

$$\frac{\begin{array}{c} e \hookrightarrow \pi^{\beta,\gamma} x : e_1 \,.\, e_2 \qquad \mathsf{max}(\beta, \gamma) = \alpha \\ \mathsf{El}^{\beta}\, e_1 \hookrightarrow T_1 \qquad \mathsf{El}^{\gamma}\, e_2 \hookrightarrow T_2 \end{array}}{\mathsf{El}^{\alpha}\, e \hookrightarrow \prod_{(x:T_1)} T_2}\ \textsc{norm-ty-el-prod}$$

$$\frac{e \hookrightarrow u_\beta \qquad \mathsf{succ}(\beta) = \alpha}{\mathsf{El}^{\alpha}\, e \hookrightarrow \mathbb{U}_\beta}\ \textsc{norm-ty-el-universe}$$

$$\frac{\begin{array}{c} e \hookrightarrow \mathsf{paths}^{\beta}_{e_1}(e_2, e_3) \qquad \beta = \alpha \\ \mathsf{El}^{\alpha}\, e_1 \hookrightarrow T_1 \qquad e_2 \hookrightarrow e_2' \qquad e_3 \hookrightarrow e_3' \end{array}}{\mathsf{El}^{\alpha}\, e \hookrightarrow \mathsf{Paths}_{T_1}(e_2', e_3')}\ \textsc{norm-ty-el-paths}$$

$$\frac{\begin{array}{c} e \hookrightarrow \mathsf{id}^{\beta}_{e_1}(e_2, e_3) \qquad \beta = \alpha \\ \mathsf{El}^{\alpha}\, e_1 \hookrightarrow T_1 \qquad e_2 \hookrightarrow e_2' \qquad e_3 \hookrightarrow e_3' \end{array}}{\mathsf{El}^{\alpha}\, e \hookrightarrow \mathsf{Id}_{T_1}(e_2', e_3')}\ \textsc{norm-ty-el-id}$$

$$\frac{e \hookrightarrow e'}{\mathsf{El}^{\alpha}\, e \hookrightarrow \mathsf{El}^{\alpha}\, e'}\ \textsc{norm-ty-el-other}$$

$$\frac{}{\mathsf{Unit} \hookrightarrow \mathsf{Unit}}\ \textsc{norm-ty-unit}$$

$$\frac{T \hookrightarrow T' \qquad U \hookrightarrow U'}{\prod_{(x:T)} U \hookrightarrow \prod_{(x:T')} U'}\ \textsc{norm-ty-prod}$$

$$\frac{T \hookrightarrow T' \qquad e_1 \hookrightarrow e_1' \qquad e_2 \hookrightarrow e_2'}{\mathsf{Paths}_T(e_1, e_2) \hookrightarrow \mathsf{Paths}_{T'}(e_1', e_2')}\ \textsc{norm-ty-paths}$$

$$\frac{T \hookrightarrow T' \qquad e_1 \hookrightarrow e_1' \qquad e_2 \hookrightarrow e_2'}{\mathsf{Id}_T(e_1, e_2) \hookrightarrow \mathsf{Id}_{T'}(e_1', e_2')}\ \textsc{norm-ty-id}$$

**General terms**

$$\frac{}{x \hookrightarrow x}\ \textsc{norm-var}$$

$$\frac{e_4 \hookrightarrow e_4'}{\mathsf{equation}\ e_1 : e_2 \equiv e_3\ \mathsf{in}\ e_4 \hookrightarrow e_4'}\ \textsc{norm-equation}$$

$$\frac{e_4 \hookrightarrow e_4'}{\mathsf{rewrite}\ e_1 : e_2 \equiv e_3\ \mathsf{in}\ e_4 \hookrightarrow e_4'}\ \textsc{norm-rewrite}$$

$$\frac{e \hookrightarrow e'}{e :: T \hookrightarrow e'}\ \textsc{norm-ascribe}$$

$$\frac{}{\star \hookrightarrow \star}\ \textsc{norm-star}$$

**Application and $\lambda$-abstraction**

$$\frac{T_1 \hookrightarrow T_1' \qquad T_2 \hookrightarrow T_2' \qquad e \hookrightarrow e'}{\lambda x{:}T_1.T_2 \,.\, e \hookrightarrow \lambda x{:}T_1'.T_2' \,.\, e'}\ \textsc{norm-abs}$$

$$\frac{\begin{array}{c} T_1 \hookrightarrow T_1' \qquad T_2 \hookrightarrow T_2' \qquad e_1 \hookrightarrow \lambda x{:}U_1.U_2 \,.\, e_1' \\ T_1' \equiv U_1 \qquad T_2' \equiv U_2 \qquad e_2 \hookrightarrow e_2' \qquad e_1'[e_2'/x] \hookrightarrow e' \end{array}}{e_1 \mathbin{@^{x:T_1.T_2}} e_2 \hookrightarrow e'}\ \textsc{norm-app-redex}$$

$$\frac{T_1 \hookrightarrow T_1' \qquad T_2 \hookrightarrow T_2' \qquad e_1 \hookrightarrow e_1' \qquad e_2 \hookrightarrow e_2'}{e_1 \mathbin{@^{x:T_1.T_2}} e_2 \hookrightarrow e_1' \mathbin{@^{x:T_1'.T_2'}} e_2'}\ \textsc{norm-app-other}$$

**Paths and equality**

$$\frac{T \hookrightarrow T' \qquad e \hookrightarrow e'}{\mathsf{idpath}_T\ e \hookrightarrow \mathsf{idpath}_{T'}\ e'}\ \textsc{norm-idpath} \qquad \frac{T \hookrightarrow T' \qquad e \hookrightarrow e'}{\mathsf{refl}_T\ e \hookrightarrow \mathsf{refl}_{T'}\ e'}\ \textsc{norm-refl}$$

$$\frac{\begin{array}{c} T \hookrightarrow T' \qquad e_2 \hookrightarrow \mathsf{idpath}_{T''}\ e_2' \qquad T' \equiv T'' \\ e_1 \hookrightarrow e_1' \qquad e_1'[e_2'/z] \hookrightarrow e' \end{array}}{\mathsf{J}_T([x\,y\,p \,.\, U], [z \,.\, e_1], e_2, e_3, e_4) \hookrightarrow e'}\ \textsc{norm-J-redex}$$

$$\frac{\begin{array}{c} T \hookrightarrow T' \qquad e_2 \hookrightarrow e_2' \qquad U \hookrightarrow U' \\ e_1 \hookrightarrow e_1' \qquad e_3 \hookrightarrow e_3' \qquad e_4 \hookrightarrow e_4' \end{array}}{\mathsf{J}_T([x\,y\,p \,.\, U], [z \,.\, e_1], e_2, e_3, e_4) \hookrightarrow \mathsf{J}_{T'}([x\,y\,p \,.\, U'], [z \,.\, e_1'], e_2', e_3', e_4')}\ \textsc{norm-J-other}$$

### Coercions

$$\textsc{norm-coerce-trivial}\quad \frac{\alpha = \beta \qquad e \hookrightarrow e'}{\mathsf{coerce}^{\alpha\mapsto\beta}\, e \hookrightarrow e'}$$

$$\textsc{norm-coerce-trans}\quad \frac{e \hookrightarrow \mathsf{coerce}^{\gamma\mapsto\delta}\, e' \qquad \alpha = \delta \qquad \mathsf{coerce}^{\gamma\mapsto\beta}\, e' \hookrightarrow e''}{\mathsf{coerce}^{\alpha\mapsto\beta}\, e \hookrightarrow e''}$$

$$\textsc{norm-coerce-pi}\quad \frac{\begin{array}{c} e \hookrightarrow \pi^{\gamma,\delta} x : e_1 \,.\, e_2 \qquad \alpha = \mathsf{max}(\gamma,\delta) \qquad \gamma \leq \alpha \qquad \delta \leq \alpha \\ \mathsf{coerce}^{\gamma\mapsto\beta}\, e_1 \hookrightarrow e_1' \qquad \mathsf{coerce}^{\delta\mapsto\beta}\, e_2 \hookrightarrow e_2' \end{array}}{\mathsf{coerce}^{\alpha\mapsto\beta}\, e \hookrightarrow \pi^{\beta,\beta} x : e_1' \,.\, e_2'}$$

$$\textsc{norm-coerce-paths}\quad \frac{\begin{array}{c} e \hookrightarrow \mathsf{paths}^{\gamma}_{e_1}(e_2, e_3) \qquad \alpha = \gamma \\ \mathsf{coerce}^{\alpha\mapsto\beta}\, e_1 \hookrightarrow e_1' \\ e_2 \hookrightarrow e_2' \qquad e_3 \hookrightarrow e_3' \end{array}}{\mathsf{coerce}^{\alpha\mapsto\beta}\, e \hookrightarrow \mathsf{paths}^{\beta}_{e_1'}(e_2', e_3')}$$

$$\textsc{norm-coerce-id}\quad \frac{\begin{array}{c} e \hookrightarrow \mathsf{id}^{\gamma}_{e_1}(e_2, e_3) \qquad \alpha = \gamma \\ \mathsf{coerce}^{\alpha\mapsto\beta}\, e_1 \hookrightarrow e_1' \\ e_2 \hookrightarrow e_2' \qquad e_3 \hookrightarrow e_3' \end{array}}{\mathsf{coerce}^{\alpha\mapsto\beta}\, e \hookrightarrow \mathsf{id}^{\beta}_{e_1'}(e_2', e_3')}$$

$$\textsc{norm-coerce-other}\quad \frac{e \hookrightarrow e'}{\mathsf{coerce}^{\alpha\mapsto\beta}\, e \hookrightarrow \mathsf{coerce}^{\alpha\mapsto\beta}\, e'}$$

### Names

$$\textsc{norm-name-unit}\quad \frac{}{\mathsf{unit} \hookrightarrow \mathsf{unit}}$$

$$\textsc{norm-name-prod}\quad \frac{e_1 \hookrightarrow e_1' \qquad e_2 \hookrightarrow e_2'}{\pi^{\alpha,\beta} x : e_1 \,.\, e_2 \hookrightarrow \pi^{\alpha,\beta} x : e_1' \,.\, e_2'}$$

$$\textsc{norm-name-universe}\quad \frac{}{u_\alpha \hookrightarrow u_\alpha}$$

$$\textsc{norm-name-paths}\quad \frac{e_1 \hookrightarrow e_1' \qquad e_2 \hookrightarrow e_2' \qquad e_3 \hookrightarrow e_3'}{\mathsf{paths}^{\alpha}_{e_1}(e_2, e_3) \hookrightarrow \mathsf{paths}^{\alpha}_{e_1'}(e_2', e_3')}$$

$$\textsc{norm-name-id}\quad \frac{e_1 \hookrightarrow e_1' \qquad e_2 \hookrightarrow e_2' \qquad e_3 \hookrightarrow e_3'}{\mathsf{id}^{\alpha}_{e_1}(e_2, e_3) \hookrightarrow \mathsf{id}^{\alpha}_{e_1'}(e_2', e_3')}$$