List of Figures

We adapt Cockx and Abel [2018] to handle Higher Inductive Types.

1 SYNTAX

1.1 Expressions

Expressions (Fig. 1) are given in spine-normal form, so that the head symbol of an application is easily accessible. Rather than adding the cubical types and operations described in Sect. **??** as new expression formers, we subsume them under f $\bar{e}$ and c $\bar{e_c}$. This happens also in the implementation of Agda, thanks to the pre-existing support for builtins and primitives.

We include a universe level ω for types like $\mathbb{I}$ which do not support transp or hcomp. Eliminations e include, beyond function application to u and projections $.\pi$, *path applications* $@_{u_0,u_1}\ v$. Path applications to interval element v are annotated by the endpoints u_0 and u_1 used for reduction, in case v becomes i0 or i1.

In contrast to ordinary applications, path applications of stuck terms can reduce; for instance, $x\ @_{u_0,u_1}$ i0 reduces to u_0. Thus, variable eliminations $x\ \bar{e}$ are not necessarily in weak head normal form. Thanks to HITs, this is not even the case for constructor applications; c $\bar{e}$ might also reduce!

$$
\begin{array}{lcll}
x, i & & & \text{variables} \\
\ell & ::= & n \mid \omega & \text{universe levels} \\
A, B, u, v & ::= & w & \text{weak head normal form} \\
& \mid & \mathsf{f}\ \bar{e} & \text{defined function or primitive applied to eliminations} \\
& \mid & x\ \bar{e} & \text{variable applied to eliminations} \\
& \mid & \mathsf{c}\ \bar{e}_{\mathsf{c}} & \text{constructor applied to eliminations} \\
W, w & ::= & (x : A) \to B & \text{dependent function type} \\
& \mid & \mathsf{Set}_\ell & \text{universe } \ell \\
& \mid & \mathsf{D}\ \bar{u} & \text{datatype fully applied to parameters} \\
& \mid & \mathsf{R}\ \bar{u} & \text{record type fully applied to parameters} \\
& \mid & \lambda x.\, u & \text{lambda abstraction} \\
e & ::= & e_{\mathsf{c}} & \text{elimination for constructors} \\
& \mid & .\pi & \text{projection} \\
e_{\mathsf{c}} & ::= & u & \text{application} \\
& \mid & @_{u_0, u_1}\ v & \text{path application}
\end{array}
$$

Fig. 1. Syntax of Terms.

Binary application $\boxed{u\ e}$ is defined as a partial function on the syntax, by β reduction $(\lambda x.u)\ v = u[v/x]$ in case of abstractions, or by accumulating eliminations $(x\ \bar{e})\ e = x\ (\bar{e}, e)$, $(\mathsf{f}\ \bar{e})\ e = \mathsf{f}\ (\bar{e}, e)$, $(\mathsf{c}\ \bar{e})\ e_{\mathsf{c}} = \mathsf{c}\ (\bar{e}, e_{\mathsf{c}})$, and otherwise it is undefined.

1.2 Patterns

Patterns are augmented with path application *copatterns*, also in the spine for constructors.

$$
\begin{array}{lcll}
p & ::= & x & \text{variable pattern} \\
& \mid & \mathsf{c}\ \bar{q}_{\mathsf{c}} & \text{fully applied constructor pattern} \\
& \mid & \lfloor \mathsf{c} \rfloor\ \bar{q}_{\mathsf{c}} & \text{forced constructor pattern} \\
& \mid & \lfloor u \rfloor & \text{forced argument} \\
& \mid & \emptyset & \text{absurd pattern} \\
\\
q & ::= & q_{\mathsf{c}} & \text{copattern for constructors} \\
& \mid & .\pi & \text{projection copattern} \\
q_{\mathsf{c}} & ::= & p & \text{application copattern} \\
& \mid & @_{u_0, u_1}\ i & \text{path application copattern}
\end{array}
$$

Note that we retain the boundary annotations of path applications even in patterns, since we convert copatterns to eliminations, denoted as $\lceil q \rceil$, during type and coverage checking for case trees.

We write $\boxed{\mathrm{PV}(\bar{q})}$ for the set of variables appearing as variable patterns x or as i in a path application copattern. We will also often drop the subscript from e_{c} and q_{c}.

$$
\lceil x \rceil = x \qquad \begin{array}{lll} \lceil \mathsf{c}\ \bar{q} \rceil = \mathsf{c}\ \lceil \bar{q} \rceil & & \lceil \lfloor u \rfloor \rceil = u \\ \lceil \lfloor \mathsf{c} \rfloor\ \bar{q} \rceil = \mathsf{c}\ \lceil \bar{q} \rceil & & \lceil .\pi \rceil = .\pi \end{array} \tag{1}
$$

1.3 Contexts and Telescopes

$$\begin{array}{rcll} \Gamma & ::= & \epsilon & \text{empty context} \\ & \mid & \Gamma(x : A) & \text{context extension} \end{array} \qquad \begin{array}{rcll} \Delta & ::= & \epsilon & \text{empty telescope} \\ & \mid & (x : A)\Delta & \text{non-empty telescope} \end{array} \tag{2}$$

We write Φ for telescopes where all the types A are $\mathbb{I}$.

1.4 Declarations

$$\begin{array}{rcll} s & ::= & \ominus & \text{status: unchecked} \\ & \mid & \oplus & \text{status: checked} \\ \\ \mathit{decl}^s & ::= & \mathsf{data}\ \mathsf{D}\ \Delta : \mathsf{Set}_n\ \mathsf{where}\ \overline{\mathit{con}} & \text{datatype declaration} \\ & \mid & \mathsf{record}\ \mathit{self} : \mathsf{R}\ \Delta : \mathsf{Set}_n\ \mathsf{where}\ \overline{\mathit{field}} & \text{record declaration} \\ & \mid & \mathsf{definition}\ \mathsf{f} : A\ \mathsf{where}\ \overline{\mathit{cls}^s} & \text{function declaration} \\ \\ \mathit{con} & ::= & \mathsf{c}\ \Delta\ [\bar{i} \mid b] & \text{constructor declaration} \\ b & ::= & \epsilon \mid (u_0, u_1)\ b & \text{boundary terms} \\ \\ \mathit{field} & ::= & \pi : A & \text{field declaration} \\ \\ \mathit{cls}^{\ominus} & ::= & \bar{q} \hookrightarrow \mathit{rhs} & \text{unchecked clause} \\ \mathit{cls}^{\oplus} & ::= & \Delta \vdash \bar{q} \hookrightarrow u : B & \text{checked clause} \\ \\ \mathit{rhs} & ::= & u & \text{clause body: expression} \\ & \mid & \mathsf{impossible} & \text{empty body for absurd pattern} \\ \\ \Sigma & ::= & \overline{\mathit{decl}^{\oplus}} & \text{signature} \end{array}$$

Fig. 2. Declarations.

2 TYPING

Our type theory employs the following basic typing and equality judgments, which are relative to a signature Σ.

$\Sigma \vdash \Gamma$	context Γ is well-formed
$\Sigma; \Gamma \vdash_\ell \Delta$	in context Γ, telescope Δ is well-formed and ℓ-bounded
$\Sigma; \Gamma \vdash u : A$	in context Γ, term u has type A
$\Sigma; \Gamma \vdash \bar{u} : \Delta$	in context Γ, term list $\bar{u}$ instantiates telescope Δ
$\Sigma; \Gamma \mid u : A \vdash \bar{e} : B$	in context Γ, head u of type A is eliminated via $\bar{e}$ to type B
$\Sigma; \Gamma \vdash u = v : A$	in context Γ, terms u and v are equal of type A
$\Sigma; \Gamma \vdash \bar{u} = \bar{v} : \Delta$	in context Γ, term lists $\bar{u}$ and $\bar{v}$ are equal instantiations of Δ
$\Sigma; \Gamma \mid u : A \vdash \bar{e} = \bar{e}' : B$	$\bar{e}$ and $\bar{e}'$ are equal eliminations of head $u : A$ to type B in Γ

In all these judgements, the signature Σ is fixed, thus we usually omit it, e.g. in the inferences rules.

We further define some shorthands for type-level judgements when we do not care about the universe level ℓ:

$\Sigma; \Gamma \vdash \Delta \iff \exists \ell.\ \Sigma; \Gamma \vdash_\ell \Delta$		well-formed telescope
$\Sigma; \Gamma \vdash A \iff \exists \ell.\ \Sigma; \Gamma \vdash A : \mathsf{Set}_\ell$		well-formed type
$\Sigma; \Gamma \vdash A = B \iff \exists \ell.\ \Sigma; \Gamma \vdash A = B : \mathsf{Set}_\ell$		equal types

We do not here include typing and equality rules for all the primitives. Good references are [Cohen et al. 2018; Huber 2017] or even the online documentation of Cubical Agda[1].

$\boxed{\Gamma \vdash u : A}$ Entails $\vdash \Gamma$ and $\Gamma \vdash A$.

Types.

$$\frac{\vdash \Gamma}{\Gamma \vdash \mathsf{Set}_n : \mathsf{Set}_{n+1}} \qquad \frac{\Gamma \vdash A : \mathsf{Set}_\ell \qquad \Gamma(x : A) \vdash B : \mathsf{Set}_{\ell'}}{\Gamma \vdash (x : A) \to B : \mathsf{Set}_{\max(\ell,\ell')}}$$

$$\frac{\mathsf{data}\ \mathsf{D}\ \Delta : \mathsf{Set}_\ell \in \Sigma \qquad \Gamma \vdash \bar{u} : \Delta}{\Gamma \vdash \mathsf{D}\ \bar{u} : \mathsf{Set}_\ell} \qquad \frac{\mathsf{record}\ \mathsf{R}\ \Delta : \mathsf{Set}_\ell \in \Sigma \qquad \Gamma \vdash \bar{u} : \Delta}{\Gamma \vdash \mathsf{R}\ \bar{u} : \mathsf{Set}_\ell}$$

$$\frac{\vdash \Gamma}{\Gamma \vdash \mathbb{I} : \mathsf{Set}_\omega} \qquad \frac{\Gamma \vdash A : \mathbb{I} \to \mathsf{Set}_n \qquad \Gamma \vdash u : A\ \mathsf{i0} \qquad \Gamma \vdash v : A\ \mathsf{i1}}{\Gamma \vdash \mathsf{PathP}\ A\ u\ v : \mathsf{Set}_n}$$

$$\frac{\Gamma \vdash A : \mathsf{Set}_\ell \qquad \Gamma \vdash u : A \qquad \Gamma \vdash v : A}{\Gamma \vdash \mathsf{Eq}_A\ u\ v : \mathsf{Set}_\ell}$$

Heads ($h ::= x\ \epsilon \mid \mathsf{f}\ \epsilon$) and applications $h\ \bar{e}$.

$$\frac{\vdash \Gamma \qquad x : A \in \Gamma}{\Gamma \vdash x\ \epsilon : A} \qquad \frac{\vdash \Gamma \qquad \mathsf{definition}\ \mathsf{f} : A \in \Sigma}{\Gamma \vdash \mathsf{f}\ \epsilon : A} \qquad \frac{\Gamma \vdash h : A \qquad \Gamma \mid h : A \vdash \bar{e} : C}{\Gamma \vdash h\ \bar{e} : C}$$

Values.

$$\frac{\begin{array}{c}\mathsf{constructor}\ \mathsf{c}\ \Delta_c\ [\bar{i} \mid b] : \mathsf{D}\ \Delta \in \Sigma \qquad \Gamma \vdash \bar{u} : \Delta \\ \Gamma \mid \mathsf{c}\ \epsilon : (\Delta_c[\bar{i} \mid b] \to \mathsf{D}\ \Delta)[\bar{u} \,/\, \Delta] \vdash \bar{e} : \mathsf{D}\ \bar{u}\end{array}}{\Gamma \vdash \mathsf{c}\ \bar{e} : \mathsf{D}\ \bar{u}}$$

$$\frac{\Gamma \vdash u : A}{\Gamma \vdash \mathsf{refl} : \mathsf{Eq}_A\ u\ u} \qquad \frac{\Gamma(x : A) \vdash u : B}{\Gamma \vdash \lambda x.\, u : (x : A) \to B}$$

Conversion.

$$\frac{\Gamma \vdash u : A \qquad \Gamma \vdash A = B}{\Gamma \vdash u : B}$$

Fig. 3. Typing rules for expressions.

[1] https://agda.readthedocs.io/en/latest/language/cubical.html

$\boxed{\Gamma \mid u : A \vdash \bar{e} : C}$ If $\Gamma \vdash u : A$ then $\Gamma \vdash C$.

$$\frac{}{\Gamma \mid u : A \vdash \epsilon : A} \qquad \frac{\Gamma \vdash v : A \qquad \Gamma \mid u\ v : B[v \,/\, x] \vdash \bar{e} : C}{\Gamma \mid u : (x : A) \to B \vdash v\ \bar{e} : C}$$

$$\frac{\Gamma \vdash v : \mathbb{I} \qquad \Gamma \mid u\ @_{u_0,u_1}\ v : B\ v \vdash \bar{e} : C}{\Gamma \mid u : \mathsf{PathP}\ B\ u_0\ u_1 \vdash @_{u_0,u_1}\ v\ \bar{e} : C}$$

$$\frac{\mathsf{projection}\ \mathit{self} : \mathsf{R}\ \Delta \vdash .\pi : A \in \Sigma \qquad \Gamma \mid u\ .\pi : A[\bar{v} \,/\, \Delta, u \,/\, \mathit{self}] \vdash \bar{e} : C}{\Gamma \mid u : \mathsf{R}\ \bar{v} \vdash .\pi\ \bar{e} : C}$$

$$\frac{\Gamma \vdash A = A' \qquad \Gamma \mid u : A' \vdash \bar{e} : C}{\Gamma \mid u : A \vdash \bar{e} : C}$$

Fig. 4. The typing rules for eliminations.

$\boxed{\Gamma \vdash_\ell \Delta}$ Entails $\vdash \Gamma$.

$$\frac{\vdash \Gamma}{\Gamma \vdash_\ell \epsilon} \qquad \frac{\Gamma \vdash A : \mathsf{Set}_{\ell'} \qquad \Gamma(x : A) \vdash_\ell \Delta}{\Gamma \vdash_\ell (x : A)\Delta}\ \ell' \le \ell$$

$\boxed{\Gamma \vdash \bar{u} : \Delta}$ Precondition: $\Gamma \vdash \Delta$.

$$\frac{}{\Gamma \vdash \epsilon : \epsilon} \qquad \frac{\Gamma \vdash u : A \qquad \Gamma \vdash \bar{u} : \Delta[u \,/\, x]}{\Gamma \vdash u\ \bar{u} : (x : A)\Delta}$$

Fig. 5. The typing rules for telescopes and lists of terms.

$\boxed{\Sigma \vdash Z}$ Snippet Z is well-formed in signature Σ.

$$\frac{\Sigma \vdash \Delta}{\Sigma \vdash \mathsf{data}\ \mathsf{D}\ \Delta : \mathsf{Set}_\ell} \qquad \frac{\mathsf{data}\ \mathsf{D}\ \Delta : \mathsf{Set}_\ell \in \Sigma \qquad \Sigma; \Delta \vdash \Delta_c[\bar{i} \mid b] \to \mathsf{D}\ \Delta : \mathsf{Set}_\ell}{\Sigma \vdash \mathsf{constructor}\ \mathsf{c}\ \Delta_c\ [\bar{i} \mid b] : \mathsf{D}\ \Delta}$$

$$\frac{\Sigma \vdash \Delta}{\Sigma \vdash \mathsf{record}\ \mathsf{R}\ \Delta : \mathsf{Set}_\ell} \qquad \frac{\mathsf{record}\ \mathsf{R}\ \Delta : \mathsf{Set}_\ell \in \Sigma \qquad \Sigma; \Delta(x : \mathsf{R}\ \hat{\Delta}) \vdash A : \mathsf{Set}_{\ell'}}{\Sigma \vdash \mathsf{projection}\ x : \mathsf{R}\ \Delta \vdash .\pi : A}\ \ell' \le \ell$$

$$\frac{\Sigma \vdash A}{\Sigma \vdash \mathsf{definition}\ \mathsf{f} : A} \qquad \frac{\mathsf{definition}\ \mathsf{f} : A \in \Sigma \qquad \Sigma \vdash \Delta \qquad \Delta \mid \mathsf{f} : A \vdash \lceil \bar{q} \rceil : B \qquad \Delta \vdash v : B}{\Sigma \vdash \mathsf{clause}\ \Delta \vdash \mathsf{f}\ \bar{q} \hookrightarrow v : B}$$

$\boxed{\Sigma_0 \subseteq \Sigma}$ Signature Σ is a valid extension of Σ_0.

$$\frac{}{\Sigma_0 \subseteq \Sigma_0} \qquad \frac{\Sigma_0 \subseteq \Sigma \qquad \Sigma \vdash Z \qquad \Sigma, Z \ \text{defined}}{\Sigma_0 \subseteq \Sigma, Z}$$

Fig. 6. Rules for well-formed signature snippets and extension.

$\boxed{\vdash \Gamma}$

$$\frac{}{\vdash \epsilon} \qquad \frac{\Gamma \vdash A \qquad x \notin \mathrm{dom}(\Gamma)}{\vdash \Gamma(x : A)}$$

Fig. 7. The typing rules for valid contexts.

$\boxed{\Gamma \vdash u = v : A}$

$$\frac{\Gamma \vdash u : A}{\Gamma \vdash u = u : A} \qquad \frac{\Gamma \vdash u_1 = u_2 : A}{\Gamma \vdash u_2 = u_1 : A} \qquad \frac{\Gamma \vdash u_1 = u_2 : A \qquad \Gamma \vdash u_2 = u_3 : A}{\Gamma \vdash u_1 = u_3 : A}$$

$$\frac{\Gamma \vdash u_1 = u_2 : A_1 \qquad \Gamma \vdash A_1 = A_2}{\Gamma \vdash u_1 = u_2 : A_2} \qquad \frac{\Gamma \vdash A_1 = A_2 : \mathsf{Set}_\ell \qquad \Gamma(x : A_1) \vdash B_1 = B_2 : \mathsf{Set}_{\ell'}}{\Gamma \vdash (x : A_1) \to B_1 = (x : A_2) \to B_2 : \mathsf{Set}_{\mathrm{max}(\ell,\ell')}}$$

$$\frac{\vdash \Gamma \qquad x : A \in \Gamma \qquad \Gamma \mid x : A \vdash \bar{e}_1 = \bar{e}_2 : B}{\Gamma \vdash x\ \bar{e}_1 = x\ \bar{e}_2 : B}$$

$$\frac{\Gamma \vdash A_1 = A_2 : \mathbb{I} \to \mathsf{Set}_\ell \qquad \Gamma \vdash u_1 = u_2 : A_1\ \mathsf{i0} \qquad \Gamma \vdash v_1 = v_2 : A_1\ \mathsf{i1}}{\Gamma \vdash \mathsf{PathP}\ A_1\ u_1\ v_1 = \mathsf{PathP}\ A_2\ u_2\ v_2 : \mathsf{Set}_\ell}$$

$$\frac{\Gamma \vdash A_1 = A_2 : \mathsf{Set}_\ell \qquad \Gamma \vdash u_1 = u_2 : A_1 \qquad \Gamma \vdash v_1 = v_2 : A_1}{\Gamma \vdash \mathsf{Eq}_{A_1}\ u_1\ v_1 = \mathsf{Eq}_{A_2}\ u_2\ v_2 : \mathsf{Set}_\ell}$$

$$\frac{\mathsf{data}\ \mathsf{D}\ \Delta : \mathsf{Set}_\ell \in \Sigma \qquad \Gamma \vdash \bar{u}_1 = \bar{u}_2 : \Delta}{\Gamma \vdash \mathsf{D}\ \bar{u}_1 = \mathsf{D}\ \bar{u}_2 : \mathsf{Set}_\ell} \qquad \frac{\mathsf{record}\ \mathsf{R}\ \Delta : \mathsf{Set}_\ell \in \Sigma \qquad \Gamma \vdash \bar{u}_1 = \bar{u}_2 : \Delta}{\Gamma \vdash \mathsf{R}\ \bar{u}_1 = \mathsf{R}\ \bar{u}_2 : \mathsf{Set}_\ell}$$

$$\frac{\begin{array}{c}\mathsf{constructor}\ \mathsf{c}\ \Delta_\mathsf{c}\ [\bar{j} \mid b] : \mathsf{D}\ \Delta \in \Sigma \qquad \Gamma \vdash \bar{u} : \Delta \\ \Gamma \mid \mathsf{c} : (\Delta_\mathsf{c}[\bar{j} \mid b] \to \mathsf{D}\ \Delta)[\bar{u}\,/\,\Delta] \vdash \bar{e}_1 = \bar{e}_2 : \mathsf{D}\ \bar{u}\end{array}}{\Gamma \vdash \mathsf{c}\ \bar{e}_1 = \mathsf{c}\ \bar{e}_2 : \mathsf{D}\ \bar{u}}$$

$$\frac{\mathsf{definition}\ \mathsf{f} : A \in \Sigma \qquad \Gamma \mid \mathsf{f} : A \vdash \bar{e}_1 = \bar{e}_2 : B}{\Gamma \vdash \mathsf{f}\ \bar{e}_1 = \mathsf{f}\ \bar{e}_2 : B} \qquad \frac{\mathsf{clause}\ \Delta \vdash \mathsf{f}\ \bar{q} \hookrightarrow v : B \in \Sigma \qquad \Gamma \vdash \sigma : \Delta}{\Gamma \vdash \mathsf{f}\ \lceil \bar{q} \rceil \sigma = v\sigma : B\sigma}$$

$$\frac{h ::= x\ \epsilon \mid \mathsf{f}\ \epsilon \mid \mathsf{c}\ \epsilon \qquad b \in \{0, 1\} \qquad \Gamma \vdash h\ \bar{e}_1\ @_{u_0,u_1}\ b\ \bar{e}_2 : B}{\Gamma \vdash h\ \bar{e}_1\ @_{u_0,u_1}\ b\ \bar{e}_2 = u_b \bar{e}_2 : B}$$

Fig. 8. The conversion rules for terms.

$\boxed{\Gamma \mid u : A \vdash \bar{e}_1 = \bar{e}_2 : B}$

$$\frac{}{\Gamma \mid u : A \vdash \epsilon = \epsilon : A} \qquad \frac{\Gamma \vdash v_1 = v_2 : A \qquad \Gamma \mid u\ v_1 : B[v_1 \,/\, x] \vdash \bar{e}_1 = \bar{e}_2 : C}{\Gamma \mid u : (x : A) \to B \vdash v_1\ \bar{e}_1 = v_2\ \bar{e}_2 : C}$$

$$\frac{\Gamma \vdash r_1 = r_2 : \mathbb{I} \qquad \Gamma \mid u\ @_{u'_0, u'_1}\ r_1 : A\ r_1 \vdash \bar{e}_1 = \bar{e}_2 : C}{\Gamma \mid u : \mathsf{PathP}\ A\ u_0\ u_1 \vdash @_{u'_0, u'_1}, r_1\ \bar{e}_1 = @_{u''_0, u''_1}\ r_2\ \bar{e}_2 : C}$$

$$\frac{\mathsf{projection}\ x : \mathsf{R}\ \Delta \vdash .\pi : A \in \Sigma \qquad \Gamma \mid u\ .\pi : A[\bar{v} \,/\, \Delta, u \,/\, x] \vdash \bar{e}_1 = \bar{e}_2 : C}{\Gamma \mid u : \mathsf{R}\ \bar{v} \vdash .\pi\ \bar{e}_1 = .\pi\ \bar{e}_2 : C}$$

$$\frac{\Gamma \mid u : A \vdash \bar{e}_1 = \bar{e}_2 : B \qquad \Gamma \vdash A = A' \qquad \Gamma \vdash B = B'}{\Gamma \mid u : A' \vdash \bar{e}_1 = \bar{e}_2 : B'}$$

Fig. 9. The conversion rules for eliminations.

$\boxed{\Gamma \vdash \bar{u} = \bar{v} : \Delta}$

$$\frac{\vdash \Gamma}{\Gamma \vdash \epsilon = \epsilon : \epsilon} \qquad \frac{\Gamma \vdash u_1 = u_2 : A \qquad \Gamma \vdash \bar{u}_1 = \bar{u}_2 : \Delta[u_1 \,/\, x]}{\Gamma \vdash u_1\ \bar{u}_1 = u_2\ \bar{u}_2 : (x : A)\Delta}$$

Fig. 10. The conversion rules for lists of terms.

3 TRANSP AND HCOMP FOR RECORD TYPES

Σ fixed, omitted from judgments.

3.1 transp

Given R Δ with fields $\Delta(x : \mathsf{R}\ \hat{\Delta}) \vdash \overline{.\pi : A}$
Let $\Gamma, i : \mathbb{I} \vdash \bar{t} : \Delta$ with $\Gamma, i : \mathbb{I} \vdash \bar{t} = \bar{t}(i/0) : \Delta$ when $r = \mathsf{i1}$.

$$(\mathsf{trans}\ (\lambda\, i.\ \mathsf{R}\ \bar{t})\ r\ u_0).\pi = \mathsf{trans}\ (\lambda\, i.\ A[v\ /\ x])\ r\ (u_0.\pi)$$

where $v = \mathsf{transpFill}\ (\lambda\, i.\ \mathsf{R}\ \bar{t})\ r\ u_0$.

3.2 hcomp

Given R Δ with fields $\Delta(x : \mathsf{R}\ \Delta) \vdash \overline{.\pi : A}$

$(\mathsf{hcomp}_{\mathsf{R}\ \bar{t}}\ (\lambda\, i\, (r = \mathsf{i1}) \to u)\ u_0).\pi = \mathsf{comp}\ (\lambda\, i.\ A[\bar{t}\ /\ \Delta, v\ i\ /\ x])\ (\lambda\, i.\ (r = \mathsf{i1}) \to u.\pi)\ (\mathsf{inS}\ (\mathsf{outS}\ u_0.\pi))$
where $v = \mathsf{hfill}_{\mathsf{R}\ \bar{t}}\ (\lambda\, i\, (r = \mathsf{i1}) \to u)\ u_0$.

4 REDUCTION

$\boxed{\Sigma \vdash u \searrow w}$ (Σ fixed, dropped from rules.)

$$\frac{}{w \searrow w} \qquad \frac{\bar{e} \searrow \bot \qquad \text{clause } \Delta \vdash \mathsf{f}\ \bar{q} \hookrightarrow v : A \in \Sigma \qquad [\bar{e} \,/\, \bar{q}] \searrow \sigma \qquad v\sigma \searrow w}{\mathsf{f}\ \bar{e} \searrow w} \qquad \frac{\bar{e} \searrow w}{h\ \bar{e} \searrow w}$$

Fig. 11. Rules for weak head normalization.

$\boxed{\Sigma \vdash \bar{e} \searrow w_\bot}$ (Σ fixed, dropped from rules.)

$$\frac{u_b\ \bar{e} \searrow w_\bot \ \text{ if} \left\{ \begin{array}{ll} e = @_{u_0,u_1}\ r & \\ r \searrow b & b \in \{0,1\} \end{array} \right\} \qquad \bar{e} \searrow w_\bot \ \text{ otherwise}}{e\ \bar{e} \searrow w_\bot} \qquad \frac{}{\epsilon \searrow \bot}$$

Fig. 12. Rules for weak head normalization through $@_{u_0,u_1}\ i$.

$\boxed{\Sigma \vdash [v \,/\, p] \searrow \sigma_\bot}$ (Σ fixed, dropped from rules.)
Note that c also ranges over refl, hcomp, transpX, i0, and i1.

$$\frac{}{[v \,/\, x] \searrow [v \,/\, x]} \qquad \frac{}{[v \,/\, \lfloor u \rfloor] \searrow [\,]}$$

$$\frac{v \searrow \mathsf{c}\ \bar{e} \qquad [\bar{e} \,/\, \bar{q}] \searrow \sigma_\bot}{[v \,/\, \mathsf{c}\ \bar{q}] \searrow \sigma_\bot} \qquad \frac{v \searrow \mathsf{c}_2\ \bar{e} \qquad [\bar{e} \,/\, \bar{q}] \searrow \sigma_\bot}{[v \,/\, \lfloor \mathsf{c}_1 \rfloor\ \bar{q}] \searrow \sigma_\bot} \qquad \frac{v \searrow \mathsf{c}_2\ \bar{e} \qquad \mathsf{c}_1 \neq \mathsf{c}_2}{[v \,/\, \mathsf{c}_1\ \bar{q}] \searrow \bot}$$

$\boxed{\Sigma \vdash [e \,/\, q] \searrow \sigma_\bot}$

$$\frac{}{[@_{u'_0,u'_1} r \,/\, @_{u_0,u_1} i] \searrow [r \,/\, i]} \qquad \frac{}{[.\pi \,/\, .\pi] \searrow [\,]} \qquad \frac{\pi_1 \neq \pi_2}{[.\pi_2 \,/\, .\pi_1] \searrow \bot}$$

$\boxed{\Sigma \vdash [\bar{e} \,/\, \bar{q}] \searrow \sigma_\bot}$

$$\frac{}{[\epsilon \,/\, \epsilon] \searrow [\,]} \qquad \frac{[e \,/\, q] \searrow \sigma_\bot \qquad [\bar{e} \,/\, \bar{q}] \searrow \tau_\bot}{[e\ \bar{e} \,/\, q\ \bar{q}] \searrow \sigma_\bot \uplus \tau_\bot}$$

Fig. 13. Rules for the pattern matching and mismatching algorithm.

5 CASE TREES

5.1 Typing

$\boxed{\Sigma;\Gamma \vdash \mathsf{f}\ \bar{q} := Q : C \mid \Theta \rightsquigarrow \Sigma'}$ Presupposes: $\Sigma;\Gamma \vdash \mathsf{f}\ \lceil \bar{q} \rceil : C$ and $\mathrm{dom}(\Gamma) = \mathrm{PV}(\bar{q})$ and $\Theta = \alpha_1;\ldots;\alpha_n$ where $\alpha ::= \epsilon \mid (i = \mathsf{i0})\,\alpha \mid (i = \mathsf{i1})\,\alpha$ such that $\Sigma;\Gamma \vdash i : \mathbb{I}$.
Checks case tree Q and outputs an extension Σ' of Σ by the clauses represented by "$\mathsf{f}\ \bar{q} \hookrightarrow Q$".

$$\frac{\Sigma;\Gamma \vdash v : C \qquad \Theta = \alpha_1;\ldots;\alpha_n \qquad (\Sigma;\Gamma_{\alpha_i} \vdash \mathsf{f}\ \bar{q}[\alpha_i] = v[\alpha_i] : C[\alpha_i])_{i=1\ldots n}}{\Sigma;\Gamma \vdash \mathsf{f}\ \bar{q} := v : C \mid \Theta \rightsquigarrow \Sigma, (\mathsf{clause}\ \Gamma \vdash \mathsf{f}\ \bar{q} \hookrightarrow v : C)}\ \textsc{CtDone}$$

$$\frac{\Sigma;\Gamma \vdash C = (x : A) \to B : \mathsf{Set}_\ell \qquad \Sigma;\Gamma(x : A) \vdash \mathsf{f}\ \bar{q}\ x := Q : B \mid \Theta \rightsquigarrow \Sigma'}{\Sigma;\Gamma \vdash \mathsf{f}\ \bar{q} := \lambda x.\ Q : C \mid \Theta \rightsquigarrow \Sigma'}\ \textsc{CtIntro}$$

$$\frac{\Sigma;\Gamma \vdash C = \mathsf{PathP}\ B\ u_0\ u_1 : \mathsf{Set}_n \qquad \Theta' = (i = 0);(i = 1);\Theta \qquad \Sigma;\Gamma(i : \mathbb{I}) \vdash \mathsf{f}\ \bar{q}\ @_{u_0,u_1}\ i := Q : B\ i \mid \Theta' \rightsquigarrow \Sigma'}{\Sigma;\Gamma \vdash \mathsf{f}\ \bar{q} := \lambda i.\ Q : C \mid \Theta \rightsquigarrow \Sigma'}\ \textsc{CtIntroPath}$$

$$\frac{\Sigma_0;\Gamma \vdash C = \mathsf{PartialP}\ r\ A : \mathsf{Set}_\omega \qquad \Sigma_0;\Gamma \vdash r = \bigvee_i \bigwedge \alpha_i : \mathbb{I} \qquad (\Sigma_{i-1};\Gamma_{\alpha_i} \vdash (\mathsf{f}\ \bar{q}[\alpha_i])\ \lfloor \mathsf{1{=}1} \rfloor) := Q_i : (A\ \mathsf{1{=}1}) \mid (\alpha_1;\ldots;\alpha_{i-1};\Theta)[\alpha_i] \rightsquigarrow \Sigma_i)_{i=1\ldots n}}{\Sigma_0;\Gamma \vdash \mathsf{f}\ \bar{q} := \mathsf{split}\{\alpha_1 \mapsto Q_1;\ldots;\alpha_n \mapsto Q_n\} : C \mid \Theta \rightsquigarrow \Sigma_n}\ \textsc{CtSplitPartial}$$

$$\frac{\Sigma_0;\Gamma \vdash C = \mathsf{R}\ \bar{v} : \mathsf{Set}_n \qquad \mathsf{record}\ \mathit{self} : \mathsf{R}\ \Delta : \mathsf{Set}_n\ \mathsf{where}\ \overline{\pi_i : A_i} \in \Sigma_0 \qquad \sigma = [\bar{v} \,/\, \Delta, \mathsf{f}\ \lceil \bar{q} \rceil \,/\, \mathit{self}] \qquad (\Sigma_{i-1};\Gamma \vdash \mathsf{f}\ \bar{q}\ .\pi_i := Q_i : A_i\sigma \mid \Theta \rightsquigarrow \Sigma_i)_{i=1\ldots n}}{\Sigma_0;\Gamma \vdash \mathsf{f}\ \bar{q} := \mathsf{record}\{\pi_1 \mapsto Q_1;\ldots;\pi_n \mapsto Q_n\} : C \mid \Theta \rightsquigarrow \Sigma_n}\ \textsc{CtCosplit}$$

$$\frac{\Sigma_0;\Gamma_1 \vdash A = \mathsf{D}\ \bar{v} : \mathsf{Set}_n \qquad \mathsf{data}\ \mathsf{D}\ \Delta : \mathsf{Set}_n\ \mathsf{where}\ \overline{\mathsf{c_i}\ \Delta_i\ []} \in \Sigma_0 \qquad (\Delta_i' = \Delta_i[\bar{v} \,/\, \Delta])_{i=1\ldots n} \qquad (\rho_i = \mathbb{1}_{\Gamma_1} \uplus [\mathsf{c_i}\ \hat{\Delta}_i' \,/\, x] \quad \rho_i' = \rho_i \uplus \mathbb{1}_{\Gamma_2})_{i=1\ldots n} \qquad (\Sigma_{i-1};\Gamma_1\Delta_i'(\Gamma_2\rho_i) \vdash \mathsf{f}\ \bar{q}\rho_i' := Q_i : C\rho_i' \mid \Theta \rightsquigarrow \Sigma_i)_{i=1\ldots n}}{\Sigma_0;\Gamma_1(x : A)\Gamma_2 \vdash \mathsf{f}\ \bar{q} := \mathsf{case}_x\{\mathsf{c_1}\ \hat{\Delta}_1' \mapsto Q_1;\ldots;\mathsf{c_n}\ \hat{\Delta}_n' \mapsto Q_n\} : C \mid \Theta \rightsquigarrow \Sigma_n}\ \textsc{CtSplitCon}$$

$$\frac{\Sigma_0;\Gamma_1 \vdash A = \mathsf{D}\ \bar{v} : \mathsf{Set}_n \qquad \Gamma = \Gamma_1(x : A)\Gamma_2 \qquad \mathsf{data}\ \mathsf{D}\ \Delta : \mathsf{Set}_n\ \mathsf{where}\ \overline{\mathsf{c_i}\ \Delta_i\ [\bar{j}_i \mid b_i]} \in \Sigma_0 \qquad \exists k.\ [\bar{j}_k \mid b_k] \neq [] \qquad \left(\begin{array}{c} \Delta_i' = \Delta_i(\bar{j}_i : \mathbb{I})[\bar{v} \,/\, \Delta] \qquad \bar{q}_i = \hat{\Delta}_i[\bar{j}_i \mid b_i][\bar{v} \,/\, \Delta] \\ \rho_i = \mathbb{1}_{\Gamma_1} \uplus [\mathsf{c_i}\ \bar{q}_i \,/\, x] \qquad \rho_i' = \rho_i \uplus \mathbb{1}_{\Gamma_2} \\ \Theta_i = \textsc{boundary}(\bar{j}_i);\Theta \\ \Sigma_{i-1};\Gamma_1\Delta_i'(\Gamma_2\rho_i) \vdash \mathsf{f}\ \bar{q}\rho_i' := Q_i : C\rho_i' \mid \Theta_i \rightsquigarrow \Sigma_i \end{array}\right)_{i=1\ldots n}}{\Sigma_n;\Gamma_1(x : \mathsf{D}\ \bar{v})\Gamma_2 \vdash \mathsf{f}\ \bar{q} := \mathsf{case}_x\{\mathsf{hcomp}\ r\ u\ u_0 \mapsto Q_{\mathsf{hc}}\} : C \mid \Theta \rightsquigarrow \Sigma_{n+1}}\ \textsc{CtSplitConHIT}$$

$$\Sigma_0;\Gamma \vdash \mathsf{f}\ \bar{q} := \mathsf{case}_x\left\{\begin{array}{l} \mathsf{c_1}\ \bar{q}_1 \mapsto Q_1;\ldots;\mathsf{c_n}\ \bar{q}_n \mapsto Q_n \\ \mathsf{hcomp}\ r\ u\ u_0 \mapsto Q_{\mathsf{hc}} \end{array}\right\} : C \mid \Theta \rightsquigarrow \Sigma_{n+1}$$

Fig. 14. Typing rules for case trees (excluding Eq).

$\boxed{\Sigma;\Gamma \vdash \mathsf{f}\ \bar{q} := \mathsf{case}_x\{\mathsf{hcomp}\ r\ u\ u_0 \mapsto Q\} : C \mid \Theta \rightsquigarrow \Sigma'}$ Presupposes: $(x : A) \in \Gamma$ where A is a type supporting hcomp, and the presuppositions made by the typing of case trees judgment (Figure 14).
Checks case tree Q can be used for the hcomp case of a split on x.

$$\frac{\begin{array}{c}\Delta_{\mathsf{hc}} = (r : \mathbb{I})(u : \mathbb{I} \to \mathsf{Partial}\ r\ A)(u_0 : A\ [\ r \mapsto u\ \mathsf{i0}\ ]) \\ \rho_{\mathsf{hc}} = \mathbb{1}_{\Gamma_1} \uplus [\mathsf{hcomp}\ r\ u\ u_0 \,/\, x] \qquad \rho'_{\mathsf{hc}} = \rho_{\mathsf{hc}} \uplus \mathbb{1}_{\Gamma_2} \\ \Sigma;\Gamma_1\Delta_{\mathsf{hc}}(\Gamma_2\rho_{\mathsf{hc}}) \vdash \mathsf{f}\ \bar{q}\rho'_{\mathsf{hc}} := Q : C\rho'_{\mathsf{hc}} \mid (r = \mathsf{i1}); \Theta \rightsquigarrow \Sigma'\end{array}}{\Sigma;\Gamma_1(x : A)\Gamma_2 \vdash \mathsf{f}\ \bar{q} := \mathsf{case}_x\{\mathsf{hcomp}\ r\ u\ u_0 \mapsto Q\} : C \mid \Theta \rightsquigarrow \Sigma'}$$

Fig. 15. Typing a match against hcomp.

$$\Sigma;\Gamma_1 \vdash A = \mathsf{Eq}_B\ u\ v : \mathsf{Set}_\ell \qquad \Sigma;\Gamma_1 \vdash^r_p u =^? v : B \Rightarrow \textsc{yes}(\Gamma'_1, \rho, \tau, \_)$$

$$\begin{array}{c}\rho' = (\rho \uplus [\mathsf{refl} \,/\, x]) \uplus \mathbb{1}_{\Gamma_2} \qquad \tau' = \tau \uplus \mathbb{1}_{\Gamma_2} \\ \Sigma;\Gamma'_1(\Gamma_2(\rho \uplus [\mathsf{refl} \,/\, x])) \vdash \mathsf{f}\ \bar{q}\rho' := Q_{\mathsf{refl}} : C\rho' \rightsquigarrow \Sigma_1\end{array}$$

$$\Sigma_1;\Gamma_1(x : \mathsf{Eq}_B\ u\ v)\Gamma_2 \vdash \mathsf{f}\ \bar{q} := \mathsf{case}_x\{\mathsf{hcomp}\ r\ t\ t_0 \mapsto Q_{\mathsf{hc}}\} : C \rightsquigarrow \Sigma_2$$

$$\frac{\begin{array}{c}\Delta_{\mathsf{tX}} = (b : B)(r : \mathbb{I})(p : \mathsf{Path}^r\ B\ b\ v)(t_0 : \mathsf{Eq}_B\ u\ b) \\ \rho_{\mathsf{tX}} = \mathbb{1}_{\Gamma_1} \uplus [\mathsf{transpX}\ p\ r\ t_0 \,/\, x] \qquad \rho'_{\mathsf{tX}} = \rho_{\mathsf{tX}} \uplus \mathbb{1}_{\Gamma_2} \\ \Sigma_2;\Gamma_1\Delta_{\mathsf{tX}}(\Gamma_2\rho_{\mathsf{tX}}) \vdash \mathsf{f}\ \bar{q}\rho'_{\mathsf{tX}} := Q_{\mathsf{tX}} : C\rho'_{\mathsf{tX}} \mid (r = \mathsf{i1}); \Theta \rightsquigarrow \Sigma'\end{array}}{\Sigma;\Gamma_1(x : A)\Gamma_2 \vdash \mathsf{f}\ \bar{q} := \mathsf{case}_x\left\{\begin{array}{l}\mathsf{refl} \mapsto^{\tau'} Q_{\mathsf{refl}} \\ \mathsf{hcomp}\ r\ t\ t_0 \mapsto Q_{\mathsf{hc}} \\ \mathsf{transpX}_b\ p\ r\ t_0 \mapsto Q_{\mathsf{tX}}\end{array}\right\} : C \rightsquigarrow \Sigma'}\ \textsc{CtSplitEq}$$

$$\frac{\Sigma;\Gamma_1 \vdash A = \mathsf{Eq}_B\ u\ v : \mathsf{Set}_\ell \qquad \Sigma;\Gamma_1 \vdash^r_p u =^? v : B \Rightarrow \textsc{no}}{\Sigma;\Gamma_1(x : A)\Gamma_2 \vdash \mathsf{f}\ \bar{q} := \mathsf{case}_x\{\} : C \rightsquigarrow \Sigma}\ \textsc{CtSplitAbsurdEq}$$

Fig. 16. Typing rules for case trees involving Eq.

5.2 Evaluation

$\boxed{\Sigma \vdash \alpha\ \sigma \longrightarrow ans}$ where $ans ::= \top \mid \bot$
(Σ dropped from rules, used only for weak head normalization $\Sigma \vdash t \searrow w$.)

$$\frac{}{\epsilon\ \sigma \longrightarrow \top} \qquad \frac{i\sigma \searrow b \qquad \alpha\ \sigma \longrightarrow ans}{(i = b)\ \alpha\ \sigma \longrightarrow ans} \qquad \frac{i\sigma \searrow b' \qquad b' \neq b}{(i = b)\ \alpha\ \sigma \longrightarrow \bot}$$

Fig. 17. Evaluation of α.

In the case for partial elements split, evaluating the boundary assignments α as stated would be quite inefficient, but one can easily cache the reductions of the interval vars. Another option is to further refine such a split into a series of splits that consider a single variable at a time.z

$\boxed{\Sigma \vdash Q\sigma\ \bar{e} \longrightarrow v}$ (Σ dropped from rules, used only for weak head normalization $\Sigma \vdash t \searrow w$.)

$$\frac{}{v\sigma\ \bar{e} \longrightarrow v\sigma\ \bar{e}}$$

$$\frac{Q(\sigma \uplus [u\,/\,x])\ \bar{e} \longrightarrow v}{(\lambda x.\ Q)\sigma\ u\ \bar{e} \longrightarrow v} \qquad \frac{Q_i\sigma\ \bar{e} \longrightarrow v}{(\mathsf{record}\{\pi_1 \mapsto Q_1; \ldots; \pi_n \mapsto Q_n\})\sigma\ .\pi_i\ \bar{e} \longrightarrow v}$$

$$\frac{x\sigma \searrow \mathsf{c}_i\ \bar{u} \qquad \mathsf{c}_i \neq \mathsf{transpX} \qquad Q_i(\sigma \backslash x \uplus [\bar{u}\,/\,\hat{\Delta}_i])\ \bar{e} \longrightarrow v}{(\mathsf{case}_x\{\mathsf{c}_1\ \hat{\Delta}_1 \mapsto Q_1; \ldots; \mathsf{c}_n\ \hat{\Delta}_n \mapsto Q_n\})\sigma\ \bar{e} \longrightarrow v}$$

$$\frac{x\sigma \searrow \mathsf{transpX}\ q\ s\ u \qquad Q(\sigma \backslash x \uplus [q\ \mathsf{i}1, q, s, u\,/\,b, p, r, t])\ \bar{e} \longrightarrow v}{(\mathsf{case}_x\{\ldots; \mathsf{transpX}_b\ p\ r\ t \mapsto Q; \ldots\})\sigma\ \bar{e} \longrightarrow v}$$

$$\frac{x\sigma \searrow \mathsf{refl} \qquad Q(\tau; \sigma)\ \bar{e} \longrightarrow v}{(\mathsf{case}_x\{\mathsf{refl} \mapsto^{\tau} Q; \mathsf{c}_1\ \hat{\Delta}_1 \mapsto Q_1; \ldots; \mathsf{c}_n\ \hat{\Delta}_n \mapsto Q_n\})\sigma\ \bar{e} \longrightarrow v}$$

$$\frac{\alpha_1\ \sigma \longrightarrow \bot \ldots \alpha_{i-1}\ \sigma \longrightarrow \bot \qquad \alpha_i\ \sigma \longrightarrow \top \qquad Q_i\sigma\ \bar{e} \longrightarrow v}{(\mathsf{split}\{\alpha_1 \mapsto Q_1; \ldots; \alpha_n \mapsto Q_n\})\sigma\ u\ \bar{e} \longrightarrow v}$$

Fig. 18. Evaluation of case trees.

6 ELABORATION

$\boxed{\Sigma \vdash \mathit{decl} \rightsquigarrow \Sigma'}$ Presupposes: $\vdash \Sigma$. Entails: $\vdash \Sigma'$.

$$\frac{\Sigma \vdash \Delta \qquad (\Sigma, \mathsf{data}\ \mathsf{D}\ \Delta : \mathsf{Set}_\ell) \mid \mathsf{D}\ \Delta : \mathsf{Set}_\ell \vdash \overline{\mathit{con}} \rightsquigarrow \Sigma'}{\Sigma \vdash (\mathsf{data}\ \mathsf{D}\ \Delta : \mathsf{Set}_\ell\ \mathsf{where}\ \overline{\mathit{con}}) \rightsquigarrow \Sigma'}$$

$$\frac{\Sigma \vdash \Delta \qquad (\Sigma, \mathsf{record}\ \mathsf{R}\ \Delta : \mathsf{Set}_\ell) \mid \mathit{self} : \mathsf{R}\ \Delta : \mathsf{Set}_\ell \vdash \overline{\mathit{fld}} \rightsquigarrow \Sigma'}{\Sigma \vdash (\mathsf{record}\ \mathit{self} : \mathsf{R}\ \Delta : \mathsf{Set}_\ell\ \mathsf{where}\ \overline{\mathit{fld}}) \rightsquigarrow \Sigma'}$$

$$\frac{\Sigma \vdash A \qquad (\Sigma, \mathsf{definition}\ \mathsf{f} : A) \vdash P \mid \mathsf{f} := Q : A \rightsquigarrow \Sigma'}{\Sigma \vdash (\mathsf{definition}\ \mathsf{f} : A\ \mathsf{where}\ P) \rightsquigarrow \Sigma'}$$

$\boxed{\Sigma \mid \mathsf{D}\ \Delta : \mathsf{Set}_\ell \vdash \overline{\mathit{con}} \rightsquigarrow \Sigma'}$ Presupposes: $\vdash \Sigma$ and $\Sigma \vdash \mathsf{D} : \Delta \to \mathsf{Set}_\ell$. Entails: $\vdash \Sigma'$.

$$\frac{}{\Sigma \mid \mathsf{D}\ \Delta : \mathsf{Set}_\ell \vdash \epsilon \rightsquigarrow \Sigma} \qquad \frac{\begin{array}{c}\Sigma; \Delta \vdash \Delta_c[\bar{j} \mid b] \to \mathsf{D}\ \Delta : \mathsf{Set}_\ell \\ (\Sigma, \mathsf{constructor}\ \mathsf{c}\ \Delta_c\ [\bar{j} \mid b] : \mathsf{D}\ \Delta) \mid \mathsf{D}\ \Delta : \mathsf{Set}_\ell \vdash \overline{\mathit{con}} \rightsquigarrow \Sigma'\end{array}}{\Sigma \mid \mathsf{D}\ \Delta : \mathsf{Set}_\ell \vdash \mathsf{c}\ \Delta_c\ [\bar{j} \mid b], \overline{\mathit{con}} \rightsquigarrow \Sigma'}$$

$\boxed{\Sigma \mid \mathit{self} : \mathsf{R}\ \Delta : \mathsf{Set}_\ell \vdash \overline{\mathit{fld}} \rightsquigarrow \Sigma'}$ Presupposes: $\vdash \Sigma$ and $\Sigma \vdash \mathsf{R} : \Delta \to \mathsf{Set}_\ell$. Entails: $\vdash \Sigma'$.

$$\frac{}{\Sigma \mid \mathit{self} : \mathsf{R}\ \Delta : \mathsf{Set}_\ell \vdash \epsilon \rightsquigarrow \Sigma} \qquad \frac{\begin{array}{c}\Sigma; \Delta(\mathit{self} : \mathsf{R}\ \hat{\Delta}) \vdash A : \mathsf{Set}_{\ell'} \qquad \ell' \leq \ell \\ (\Sigma, \mathsf{projection}\ \mathit{self} : \mathsf{R}\ \Delta \vdash .\pi : A) \mid \mathit{self} : \mathsf{R}\ \Delta : \mathsf{Set}_\ell \vdash \overline{\mathit{fld}} \rightsquigarrow \Sigma'\end{array}}{\Sigma \mid \mathit{self} : \mathsf{R}\ \Delta : \mathsf{Set}_\ell \vdash \pi : A, \overline{\mathit{fld}} \rightsquigarrow \Sigma'}$$

Fig. 19. Rules for checking declarations of data types, record types, and defined symbols.

6.1 Inferring hcomp Clauses

$\boxed{\Sigma;\Gamma(x : \mathsf{D}\ \bar{v})\Delta \vdash \mathsf{f}\ \bar{q} : C \mid \Theta \Rightarrow^{x} \textsc{hc-rhs}(rhs)}$

$$\Theta = \alpha_1;\ldots;\alpha_n \quad \Delta_{\mathsf{hc}} = (r : \mathbb{I})(u : \mathbb{I} \to \mathsf{Partial}\ r\ (\mathsf{D}\ \bar{v}))(u_0 : \mathsf{D}\ \bar{v}\ [\, r \mapsto u\ \mathsf{i0}\,])$$
$$\Delta^i = \Delta[\mathsf{hfill}\ u\ u_0\ i\ /\ x]$$
$$\delta^i : \Delta^i = \textsc{transp-tel}\ (j.\ \Delta^{\sim\, j \vee i})\ i\ \hat{\Delta}^{\mathsf{i1}}$$
$$sys = \lambda i.\ \left\{ \begin{array}{ll} (r = \mathsf{i1}) & \to \mathsf{f}\ \bar{q}[u\ i\ \mathsf{1{=}1}\ /\ x, \delta^i\ /\ \Delta^i] \\ \alpha_1 & \to \mathsf{f}\ \bar{q}[\mathsf{hfill}\ u\ u_0\ i\ /\ x, \delta^i\ /\ \Delta^i][\alpha_1] \\ \vdots & \\ \alpha_n & \to \mathsf{f}\ \bar{q}[\mathsf{hfill}\ u\ u_0\ i\ /\ x, \delta^i\ /\ \Delta^i][\alpha_n] \end{array} \right\}$$
$$rhs = \mathsf{comp}\ (\lambda i.\ C[\mathsf{hfill}\ u\ u_0\ i\ /\ x, \delta^i\ /\ \Delta^i])\ sys\ (\mathsf{inS}\ (\mathsf{f}\ \bar{q}[\mathsf{outS}\ u_0\ /\ x, \delta^{\mathsf{i0}}\ /\ \Delta]))$$
$$\text{Derivable typing:} \quad \Gamma\Delta_{\mathsf{hc}}\Delta[\mathsf{hcomp}\ u\ u_0\ /\ x] \vdash rhs : C[\mathsf{hcomp}\ u\ u_0\ /\ x]$$

$$\Sigma;\Gamma(x : \mathsf{D}\ \bar{v})\Delta \vdash \mathsf{f}\ \bar{q} : C \mid \Theta \Rightarrow^{x} \textsc{hc-rhs}(rhs)$$

Fig. 20. Computing the right hand side of a hcomp match.

$\boxed{\Sigma;\Gamma(x : \mathsf{Eq}_A\ u\ v)\Delta \mid (\Gamma', \rho, \tau, \eta) \vdash \mathsf{f}\ \bar{q} : C \mid \Theta \Rightarrow \textsc{trxrefl}(rhs)}$

$$\Theta = \alpha_1;\ldots;\alpha_n \quad \Delta_{\mathsf{refl}} = (r : \mathbb{I})(p : \mathsf{Path}^r\ B\ u\ v)$$
$$\Delta^i = \Delta[\mathsf{transpX}\ p\ r\ \mathsf{refl}\ /\ x][\eta]$$
$$\delta^i : \Delta^i = \textsc{transp-tel}\ (j.\ \Delta^{\sim\, j \vee i})\ (i \vee r)\ \hat{\Delta}^{\mathsf{i1}}$$
$$sys = \lambda i.\ \left\{ \begin{array}{ll} (r = \mathsf{i1}) & \to \mathsf{f}\ \bar{q}[(\rho \uplus \mathsf{refl}\ /\ x)[\tau] \uplus \delta^i\ /\ \Delta] \\ \alpha_1 & \to \mathsf{f}\ \bar{q}[\mathsf{transpX}\ p\ r\ \mathsf{refl}\ /\ x][\eta \uplus \delta^i\ /\ \Delta^i][\alpha_1] \\ \vdots & \\ \alpha_n & \to \mathsf{f}\ \bar{q}[\mathsf{transpX}\ p\ r\ \mathsf{refl}\ /\ x][\eta \uplus \delta^i\ /\ \Delta^i][\alpha_n] \end{array} \right\}$$
$$base = \mathsf{inS}\ (\mathsf{f}\ \bar{q}[(\rho \uplus \mathsf{refl}\ /\ x)[\tau] \uplus \delta^{\mathsf{i0}}\ /\ \Delta])$$
$$rhs = \mathsf{comp}\ (\lambda i.\ C[\mathsf{transpX}\ p\ r\ \mathsf{refl}\ /\ x][\eta \uplus \delta^i\ /\ \Delta^i])\ sys\ base$$
$$\text{Derivable typing:} \quad \Gamma\Delta_{\mathsf{refl}}\Delta[\mathsf{transpX}\ p\ r\ \mathsf{refl}\ /\ x] \vdash rhs : C[\mathsf{transpX}\ p\ r\ \mathsf{refl}\ /\ x]$$

$$\Sigma;\Gamma(x : \mathsf{Eq}_A\ u\ v)\Delta \mid (\Gamma', \rho, \tau, \eta) \vdash \mathsf{f}\ \bar{q} : C \mid \Theta \Rightarrow \textsc{trxrefl}(rhs)$$

Fig. 21. Computing the right hand side of a transpX p r refl match.

$\boxed{\Sigma;\Gamma(x : \mathsf{Eq}_A\ u\ v)\Delta \vdash \mathsf{f}\ \bar{q} : C \mid \Theta \Rightarrow \textsc{trxtrx}(rhs)}$

$$\Theta = \alpha_1;\ldots;\alpha_n \quad \Delta_{\mathsf{tX}} = (b : B)(r : \mathbb{I})(p : \mathsf{Path}^r\ B\ b\ v)$$
$$(b' : B)(s : \mathbb{I})(q : \mathsf{Path}^s\ B\ b'\ b)(t : \mathsf{Eq}_B\ u\ b')$$
$$eq_0 = \mathsf{transpX}\ (q\ {}^{\cdot s,r}\ p)\ (r \wedge s)\ t \quad eq_1 = \mathsf{transpX}\ p\ r\ (\mathsf{transpX}\ q\ s\ t)$$
Let $eq : eq_0 \equiv eq_1$ and $eq_{s,r} : \mathsf{PartialP}\ (s \vee r)\ (\lambda(s \vee r = \mathsf{i1}) \to eq \equiv \mathsf{refl})$ both constant when $r \wedge s = \mathsf{i1}$.

$$\Delta' = \Delta[eq_1\ /\ x]$$
$$\delta(r', eq', j) = \textsc{transp-tel}\ (i.\ \Delta[eq'(\sim i \vee j)\ /\ x])\ ((s \wedge r) \vee r' \vee j)\ \hat{\Delta}'$$
$$f(r', eq', j) = \mathsf{f}\ \bar{q}[eq'(j)\ /\ x, \delta(r', eq', j)\ /\ \Delta]$$
$$C'(r', eq') = \lambda j \to C[eq'(j)\ /\ x, \delta(r', eq', j)\ /\ \Delta]$$
$$sys(r', eq') = \lambda j \to \left\{\begin{array}{ll} \alpha_1 & \to f(r', eq', j)[\alpha_1] \\ \vdots & \\ \alpha_n & \to f(r', eq', j)[\alpha_n] \\ (r \wedge s = \mathsf{i1}) & \to f(r', eq', j) \\ (r' = \mathsf{i1}) & \to f(r', eq', j) \end{array}\right\}$$
$$c(r', eq') = \mathsf{comp}\ C'(r', eq')\ sys(r', eq')\ (\mathsf{inS}\ (f(r', eq', \mathsf{i0})))$$
$$sys' = \lambda i \to \left\{\begin{array}{ll} \alpha_1 & \to \mathsf{f}\ \bar{q}[eq_0\ /\ x][\alpha_1] \\ \vdots & \\ \alpha_n & \to \mathsf{f}\ \bar{q}[eq_0\ /\ x][\alpha_1] \\ (r = \mathsf{i1}) & \to c(i, eq_{s,r}\ \mathsf{1{=}1}\ i) \\ (s = \mathsf{i1}) & \to c(i, eq_{s,r}\ \mathsf{1{=}1}\ i) \end{array}\right\}$$
$$rhs = \mathsf{hcomp}\ sys'\ c(\mathsf{i0}, eq)$$
Derivable typing: $\Gamma\Delta_{\mathsf{tX}}\Delta' \vdash rhs : C[eq_1\ /\ x]$

$$\Sigma;\Gamma(x : \mathsf{Eq}_A\ u\ v)\Delta \vdash \mathsf{f}\ \bar{q} : C \mid \Theta \Rightarrow \textsc{trxtrx}(rhs)$$

Fig. 22. Computing the right hand side of a transpX p r (transpX q s t) match.

$$\Sigma;\Gamma_1(b:B) \vdash_{\_} u =^? b : B \Rightarrow \textsc{yes}(\Gamma_1,[u\,/\,b],\tau,\_) \quad \tau' = \tau \uplus \mathbb{1}_{\Gamma_2}$$
$$\Delta_{\mathsf{refl}} = (r:\mathbb{I})(p:\mathsf{Path}^r\ B\ u\ v)$$
$$\rho_{\mathsf{refl}} = \mathbb{1}_{\Gamma_1} \uplus [\mathsf{transpX}\ r\ p\ \mathsf{refl}\,/\,x] \qquad \rho'_{\mathsf{refl}} = \rho_{\mathsf{refl}} \uplus \mathbb{1}_{\Gamma_2}$$
$$\Sigma;\Gamma_1(x:\mathsf{Eq}_B\ u\ v)\Gamma_2 \mid (\Gamma'_1,\rho,\tau,\eta) \vdash \mathsf{f}\ \bar{q} : C \mid \Theta \Rightarrow \textsc{trxrefl}(rhs_{\mathsf{refl}}) \qquad P_{\mathsf{refl}} = \{[]\ \epsilon \hookrightarrow rhs_{\mathsf{refl}}\}$$
$$\Sigma;\Gamma_1\Delta_{\mathsf{refl}}(\Gamma_2\rho_{\mathsf{refl}}) \vdash P_{\mathsf{refl}} \mid \mathsf{f}\ \bar{q}\rho'_{\mathsf{refl}} := Q_{\mathsf{refl}} : C\rho'_{\mathsf{refl}} \mid (r = \mathsf{i1});\Theta \rightsquigarrow \Sigma_1$$

$$\Delta = (b:B)(r:\mathbb{I})(p:\mathsf{Path}^r\ B\ b\ v)$$

$$\Delta_{\mathsf{tX}} = \Delta(b':B)(s:\mathbb{I})(q:\mathsf{Path}^s\ B\ b'\ b)(t_1:\mathsf{Eq}_B\ u\ b')$$
$$\rho_{\mathsf{tX}} = \mathbb{1}_{\Gamma_1} \uplus [\mathsf{transpX}\ p\ r\ (\mathsf{transpX}\ q\ s\ t_1)\,/\,x] \qquad \rho'_{\mathsf{tX}} = \rho_{\mathsf{tX}} \uplus \mathbb{1}_{\Gamma_2}$$
$$\Sigma_2;\Gamma_1(x:\mathsf{Eq}_B\ u\ v)\Gamma_2 \vdash \mathsf{f}\ \bar{q} : C \mid \Theta \Rightarrow \textsc{trxtrx}(rhs_{\mathsf{tX}}) \qquad P_{\mathsf{tX}} = \{[]\ \epsilon \hookrightarrow rhs_{\mathsf{tX}}\}$$
$$\Sigma_1;\Gamma_1\Delta_{\mathsf{tX}}(\Gamma_2\rho_{\mathsf{tX}}) \vdash P_{\mathsf{tX}} \mid \mathsf{f}\ \bar{q}\rho'_{\mathsf{tX}} := Q_{\mathsf{tX}} : C\rho'_{\mathsf{tX}} \mid (s = \mathsf{i1});(r = \mathsf{i1});\Theta \rightsquigarrow \Sigma_2$$

$$\Delta_{\mathsf{hc}} = \Delta(s:\mathbb{I})(w:\mathbb{I} \to \mathsf{Partial}\ r\ (\mathsf{Eq}_B\ u\ b))(w_0:\mathsf{Eq}_B\ u\ b\ [\,r \mapsto w\ \mathsf{i0}\,])$$
$$\rho_0 = [\mathsf{transpX}\ p\ r\ x\,/\,x] \qquad \rho'_0 = \rho_0 \uplus \mathbb{1}_{\Gamma_2}$$
$$\rho_{\mathsf{hc}} = \mathbb{1}_{\Gamma_1} \uplus [\mathsf{transpX}\ r\ p\ (\mathsf{hcomp}\ r\ w\ w_0)\,/\,x] \qquad \rho'_{\mathsf{hc}} = \rho_{\mathsf{hc}} \uplus \mathbb{1}_{\Gamma_2}$$
$$\Sigma_2;\Gamma_1\Delta(x:\mathsf{Eq}_B\ u\ b)\Gamma_2\rho_0 \vdash \mathsf{f}\ \bar{q}\rho'_0 : C\rho'_0 \mid (r = \mathsf{i1});\Theta \Rightarrow \qquad \textsc{hc-rhs}(rhs_{\mathsf{hc}})P_{\mathsf{hc}} = \{[]\ \epsilon \hookrightarrow rhs_{\mathsf{hc}}\}$$
$$\Sigma_2;\Gamma_1\Delta_{\mathsf{hc}}(\Gamma_2\rho_{\mathsf{hc}}) \vdash P_{\mathsf{hc}} \mid \mathsf{f}\ \bar{q}\rho'_{\mathsf{hc}} := Q_{\mathsf{hc}} : C\rho'_{\mathsf{hc}} \mid (s = \mathsf{i1});(r = \mathsf{i1});\Theta \rightsquigarrow \Sigma'$$

$$Q = \mathsf{case}_{t_0}\left\{\begin{array}{l} \mathsf{refl} \mapsto^{\tau'} Q_{\mathsf{refl}} \\ \mathsf{transpX}_{b'}\ q\ s\ t_1 \mapsto Q_{\mathsf{tX}} \\ \mathsf{hcomp}\ s\ w\ w_0 \mapsto Q_{\mathsf{hc}} \end{array}\right\}$$

$$\Sigma;\Gamma_1(x:\mathsf{Eq}_B\ u\ v)\Gamma_2 \vdash (\Gamma'_1,\rho,\tau,\eta) \mid \mathsf{f}\ \bar{q} := \mathsf{case}_x\{\mathsf{transpX}_b\ p\ r\ t_0 \mapsto Q\} : C \mid \Theta \rightsquigarrow \Sigma'$$

Fig. 23. Computing case tree for a transpX match.

7 ELABORATION OF A DEFINED FUNCTION INTO A CASE TREE

$\boxed{\Sigma;\Gamma \vdash P \mid \mathsf{f}\ \bar{q} := Q : C \mid \Theta \rightsquigarrow \Sigma'}$ In all rules $P = \{[E_i]\bar{q}_i \hookrightarrow rhs_i \mid i = 1 \ldots m\}$.

Presupposes: $\Sigma;\Gamma \vdash \mathsf{f}\ \lceil\bar{q}\rceil : C$ and $\mathrm{dom}(\Gamma) = \mathrm{PV}(\bar{q})$. Entails: $\Sigma;\Gamma \vdash \mathsf{f}\ \bar{q} := Q : C \mid \Theta \rightsquigarrow \Sigma'$.

$$\frac{\begin{array}{c}\bar{q}_1 = \epsilon \qquad \Sigma;\Gamma \vdash E_1 \Rightarrow \textsc{solved}(\sigma) \qquad rhs_1 = v \qquad \Sigma;\Gamma \vdash v\sigma : C \\ \Theta = \alpha_1;\ldots;\alpha_n \qquad (\Sigma;\Gamma_{\alpha_i} \vdash \mathsf{f}\ \bar{q}[\alpha_i] = v[\alpha_i] : C[\alpha_i])_{i=1\ldots n}\end{array}}{\Sigma;\Gamma \vdash P \mid \mathsf{f}\ \bar{q} := v\sigma : C \mid \Theta \rightsquigarrow \Sigma, \mathsf{clause}\ \Gamma \vdash \mathsf{f}\ \bar{q} \hookrightarrow v\sigma : C}\ \textsc{Done}$$

$$\frac{\bar{q}_1 = p\ \bar{q}_1' \qquad \Sigma \vdash C \searrow (x : A) \to B \qquad \Sigma;\Gamma(x : A) \vdash P\ (x : A) \mid \mathsf{f}\ \bar{q}\ x := Q : B \mid \Theta \rightsquigarrow \Sigma'}{\Sigma;\Gamma \vdash P \mid \mathsf{f}\ \bar{q} := \lambda x.\ Q : C \mid \Theta \rightsquigarrow \Sigma'}\ \textsc{Intro}$$

$$\frac{\begin{array}{c}\Sigma \vdash C \searrow \mathsf{PathP}\ B\ u_0\ u_1 \qquad \Theta' = (i = 0);(i = 1);\Theta \\ \Sigma;\Gamma(i : \mathbb{I}) \vdash P\ (i : \mathbb{I}) \mid \mathsf{f}\ \bar{q}\ @_{u_0,u_1}\ i := Q : B\ i \mid \Theta' \rightsquigarrow \Sigma'\end{array}}{\Sigma;\Gamma \vdash P \mid \mathsf{f}\ \bar{q} := \lambda i.\ Q : C \rightsquigarrow \Sigma'}\ \textsc{IntroPath}$$

$$\frac{\begin{array}{c}(i\ /^?\ b : \mathbb{I}) \in E_1,\ b \in \{\mathsf{i0}, \mathsf{i1}\} \qquad \Sigma_0 \vdash C \searrow \mathsf{PartialP}\ r\ A \qquad \Sigma_0;\Gamma \vdash r = \bigvee_i \bigwedge \alpha_i : \mathbb{I} \\ \left(\begin{array}{c}\Sigma_{i-1} \vdash P[\alpha_i] \Rightarrow P_i \\ \Sigma_{i-1};\Gamma_{\alpha_i} \vdash P_i \mid (\mathsf{f}\ \bar{q}[\alpha_i]\ \lfloor \mathsf{1{=}1} \rfloor) := Q_i : (A\ \mathsf{1{=}1}) \mid (\alpha_1;\ldots;\alpha_{i-1};\Theta)[\alpha_i] \rightsquigarrow \Sigma_i\end{array}\right)_{i=1\ldots n}\end{array}}{\Sigma_0;\Gamma \vdash P \mid \mathsf{f}\ \bar{q} := \mathsf{split}\{\alpha_1 \mapsto Q_1;\ldots;\alpha_n \mapsto Q_n\} : C \mid \Theta \rightsquigarrow \Sigma_n}\ \textsc{SplitPartial}$$

$$\frac{\begin{array}{c}\bar{q}_1 = .\pi_i\ \bar{q}_1' \qquad \Sigma \vdash C \searrow \mathsf{R}\ \bar{v} \qquad \mathsf{record}\ self : \mathsf{R}\ \Delta : \mathsf{Set}_\ell\ \mathsf{where}\ \overline{\pi_i : A_i} \in \Sigma_0 \\ (\Sigma_{i-1};\Gamma \vdash P\ .\pi_i \mid \mathsf{f}\ \bar{q}\ .\pi_i := Q_i : A_i[\bar{v}\,/\,\Delta, \mathsf{f}\ \lceil\bar{q}\rceil\,/\,self] \mid \Theta \rightsquigarrow \Sigma_i)_{i=1\ldots n}\end{array}}{\Sigma;\Gamma \vdash P \mid \mathsf{f}\ \bar{q} := \mathsf{record}\{\pi_1 \mapsto Q_1;\ldots;\pi_n \mapsto Q_n\} : C \mid \Theta \rightsquigarrow \Sigma_n}\ \textsc{Cosplit}$$

$$\frac{\begin{array}{c}\bar{q}_1 = \emptyset \qquad m = 1 \qquad \Sigma \vdash C \searrow \mathsf{R}\ \bar{v} \qquad \mathsf{record}\ \_ : \mathsf{R}\ \Delta : \mathsf{Set}_\ell\ \mathsf{where}\ \epsilon \in \Sigma \\ rhs_1 = \mathsf{impossible}\end{array}}{\Sigma;\Gamma \vdash P \mid \mathsf{f}\ \bar{q} := \mathsf{record}\{\} : C \mid \Theta \rightsquigarrow \Sigma}\ \textsc{CosplitEmpty}$$

$$\frac{\begin{array}{c}(x\ /^?\ \mathsf{c_j}\ \bar{p} : A) \in E_1 \qquad \Sigma \vdash A \searrow \mathsf{D}\ \bar{v} \qquad \Gamma = \Gamma_1(x : A)\Gamma_2 \\ \mathsf{data}\ \mathsf{D}\ \Delta : \mathsf{Set}_\ell\ \mathsf{where}\ \overline{\mathsf{c_i}\ \Delta_i\ []} \in \Sigma_0 \\ \left(\begin{array}{c}\Delta_i' = \Delta_i[\bar{v}\,/\,\Delta] \qquad \rho_i = \mathbb{1}_{\Gamma_1} \uplus [\mathsf{c_i}\ \hat{\Delta}_i'\,/\,x] \qquad \rho_i' = \rho_i \uplus \mathbb{1}_{\Gamma_2} \\ \Sigma_{i-1} \vdash P\rho_i' \Rightarrow P_i \qquad (\Sigma_{i-1};\Gamma_1\Delta_i'(\Gamma_2\rho_i) \vdash P_i \mid \mathsf{f}\ \bar{q}\rho_i' := Q_i : C\rho_i' \mid \Theta \rightsquigarrow \Sigma_i\end{array}\right)_{i=1\ldots n}\end{array}}{\Sigma_0;\Gamma \vdash P \mid \mathsf{f}\ \bar{q} := \mathsf{case}_x\{\mathsf{c_1}\ \hat{\Delta}_1' \mapsto Q_1;\ldots;\mathsf{c_n}\ \hat{\Delta}_n' \mapsto Q_n\} : C \mid \Theta \rightsquigarrow \Sigma_n}\ \textsc{SplitCon}$$

$$\frac{(x\ /^?\ \mathsf{c_j}\ \bar{q} : A) \in E_1 \qquad \Sigma \vdash A \searrow \mathsf{D}\ \bar{v} \qquad \Gamma = \Gamma_1(x : A)\Gamma_2 \qquad \mathsf{data\ D}\ \Delta : \mathsf{Set}_n\ \mathsf{where}\ \mathsf{c_i}\ \Delta_i\ [\bar{j_i} \mid b_i] \in \Sigma_0 \qquad \exists k.\ [\bar{j_k} \mid b_k] \neq []}{\left(\begin{array}{c} \Delta_i' = \Delta_i(\bar{j_i} : \mathbb{I})[\bar{v} \,/\, \Delta] \qquad \bar{q}_i = \hat{\Delta}_i[\bar{j_i} \mid b_i][\bar{v} \,/\, \Delta] \\ \rho_i = \mathbb{1}_{\Gamma_1} \uplus [\mathsf{c_i}\ \bar{q}_i \,/\, x] \qquad \rho_i' = \rho_i \uplus \mathbb{1}_{\Gamma_2} \\ \Sigma_{i-1} \vdash P\rho_i' \Rightarrow P_i \qquad \Theta_i = \textsc{boundary}(\bar{j_i}); \Theta \\ \Sigma_{i-1}; \Gamma_1\Delta_i'(\Gamma_2\rho_i) \vdash P_i \mid \mathsf{f}\ \bar{q}\rho_i' := Q_i : C\rho_i' \mid \Theta_i \rightsquigarrow \Sigma_i \end{array}\right)_{i=1\ldots n}}$$

$$\frac{\begin{array}{c} \Delta_{\mathsf{hc}} = (r : \mathbb{I})(u : \mathbb{I} \to \mathsf{Partial}\ r\ (\mathsf{D}\ \bar{v}))(u_0 : \mathsf{D}\ \bar{v}\ [\, r \mapsto u\ \mathsf{i0}\,]) \\ \rho_{\mathsf{hc}} = \mathbb{1}_{\Gamma_1} \uplus [\mathsf{hcomp}\ r\ u\ u_0 \,/\, x] \qquad \rho_{\mathsf{hc}}' = \rho_{\mathsf{hc}} \uplus \mathbb{1}_{\Gamma_2} \\ \Sigma_n; \Gamma_1(x : \mathsf{D}\ \bar{v})\Gamma_2 \vdash \mathsf{f}\ \bar{q} : C \mid \Theta \Rightarrow \qquad \textsc{hc-rhs}(rhs)P_{\mathsf{hc}} = \{[]\ \epsilon \hookrightarrow rhs\} \\ \Sigma_n; \Gamma_1\Delta_{\mathsf{hc}}(\Gamma_2\rho_{\mathsf{hc}}) \vdash P_{\mathsf{hc}} \mid \mathsf{f}\ \bar{q}\rho_{\mathsf{hc}}' := Q_{\mathsf{hc}} : C\rho_{\mathsf{hc}}' \mid (r = \mathsf{i1}); \Theta \rightsquigarrow \Sigma_{n+1} \end{array}}{\Sigma_0; \Gamma \vdash P \mid \mathsf{f}\ \bar{q} := \mathsf{case}_x \left\{ \begin{array}{l} \mathsf{c_1}\ \bar{q}_1 \mapsto Q_1; \ldots; \mathsf{c_n}\ \bar{q}_n \mapsto Q_n \\ \mathsf{hcomp}\ r\ u\ u_0 \mapsto Q_{\mathsf{hc}} \end{array} \right\} : C \mid \Theta \rightsquigarrow \Sigma_{n+1}}\ \textsc{SplitConHIT}$$

$$\frac{\begin{array}{c} (x\ /^?\ \mathsf{refl} : A) \in E_1 \qquad \Sigma \vdash A \searrow \mathsf{Eq}_B\ u\ v \qquad \Gamma = \Gamma_1(x : A)\Gamma_2 \\ \Sigma; \Gamma_1 \vdash_p^r u =^? v : B \Rightarrow \textsc{yes}(\Gamma_1', \rho, \tau, \eta) \qquad \rho' = (\rho \uplus [\mathsf{refl} \,/\, x]) \uplus \mathbb{1}_{\Gamma_2} \qquad \tau' = \tau \uplus \mathbb{1}_{\Gamma_2} \\ \Sigma \vdash P\rho' \Rightarrow P' \qquad \Sigma; \Gamma_1'(\Gamma_2(\rho \uplus [\mathsf{refl} \,/\, x])) \vdash P' \mid \mathsf{f}\ \bar{q}\rho' := Q_{\mathsf{refl}} : C\rho' \mid \Theta \rightsquigarrow \Sigma_1 \end{array}}{\begin{array}{c} \Delta_{\mathsf{hc}} = (r : \mathbb{I})(t : \mathbb{I} \to \mathsf{Partial}\ r\ (\mathsf{Eq}_B\ u\ v))(t_0 : \mathsf{Eq}_B\ u\ v\ [\, r \mapsto t\ \mathsf{i0}\,]) \\ \rho_{\mathsf{hc}} = \mathbb{1}_{\Gamma_1} \uplus [\mathsf{hcomp}\ r\ t\ t_0 \,/\, x] \qquad \rho_{\mathsf{hc}}' = \rho_{\mathsf{hc}} \uplus \mathbb{1}_{\Gamma_2} \\ \Sigma_1; \Gamma_1(x : \mathsf{Eq}_B\ u\ v)\Gamma_2 \vdash \mathsf{f}\ \bar{q} : C \mid \Theta \Rightarrow \qquad \textsc{hc-rhs}(rhs)P_{\mathsf{hc}} = \{[]\ \epsilon \hookrightarrow rhs\} \\ \Sigma_1; \Gamma_1\Delta_{\mathsf{hc}}(\Gamma_2\rho_{\mathsf{hc}}) \vdash P_{\mathsf{hc}} \mid \mathsf{f}\ \bar{q}\rho_{\mathsf{hc}}' := Q_{\mathsf{hc}} : C\rho_{\mathsf{hc}}' \mid (r = \mathsf{i1}); \Theta \rightsquigarrow \Sigma_2 \end{array}}{\Sigma_2; \Gamma_1(x : \mathsf{Eq}_B\ u\ v)\Gamma_2 \vdash (\Gamma_1', \rho, \tau, \eta) \mid \mathsf{f}\ \bar{q} := \mathsf{case}_x\{\mathsf{transpX}_b\ p\ r\ t_0 \mapsto Q_{\mathsf{tX}}\} : C \mid \Theta \rightsquigarrow \Sigma'}\ \textsc{SplitEq}$$

$$\Sigma; \Gamma \vdash P \mid \mathsf{f}\ \bar{q} := \mathsf{case}_x \left\{ \begin{array}{l} \mathsf{refl} \mapsto^{\tau'} Q_{\mathsf{refl}} \\ \mathsf{hcomp}\ r\ t\ t_0 \mapsto Q_{\mathsf{hc}} \\ \mathsf{transpX}_b\ p\ r\ t_0 \mapsto Q_{\mathsf{tX}} \end{array} \right\} : C \mid \Theta \rightsquigarrow \Sigma'$$

$$\frac{(x\ /^?\ \emptyset : A) \in E_1 \qquad \Sigma; \Gamma \vdash \emptyset : A \qquad rhs_1 = \mathsf{impossible}}{\Sigma; \Gamma \vdash P \mid \mathsf{f}\ \bar{q} := \mathsf{case}_x\{\} : C \mid \Theta \rightsquigarrow \Sigma}\ \textsc{SplitEmpty}$$

Fig. 23. Rules for checking a list of clauses and elaborating them to a well-typed case tree.

$$\boxed{\Sigma; \Gamma \vdash E \Rightarrow \textsc{solved}(\sigma)}$$

$$\frac{(\Sigma \vdash [w_k \,/\, p_k] \searrow \sigma_k)_{k=1\ldots n} \qquad \sigma = \biguplus_k \sigma_k \qquad (\Sigma; \Gamma \vdash \lceil p_k \rceil \sigma = w_k : A_k)_{k=1\ldots n}}{\Sigma; \Gamma \vdash \{w_k\ /^?\ p_k : A_k \mid k = 1 \ldots n\} \Rightarrow \textsc{solved}(\sigma)}$$

Fig. 24. Rule for constructing the final substitution and checking all constraints when splitting is done.

$\boxed{P\ (x : A)}$ Replace the first (path) application pattern p in each clause by the constraint $x \mathrel{/^?} p : A$.

$$\begin{array}{rcl} \epsilon\ (x : A) & = & \epsilon \\ ([E]p\ \bar{q} \hookrightarrow rhs,\ P)\ (x : A) & = & ([E \cup \{x \mathrel{/^?} p : A\}]\bar{q} \hookrightarrow rhs),\ P\ (x : A) \\ ([E]@\_\ \_\ j\ \bar{q} \hookrightarrow rhs,\ P)\ (i : \mathbb{I}) & = & ([E \cup \{i \mathrel{/^?} j : A\}]\bar{q} \hookrightarrow rhs),\ P\ (i : A) \end{array}$$

Fig. 25. Partially decomposed clauses after introducing a new variable (partial function).

$\boxed{P\ .\pi}$ Keep only clauses with copattern $.\pi$, with this copattern removed.

$$\begin{array}{rcll} \epsilon\ .\pi & = & \epsilon & \\ ([E].\pi\ \bar{q} \hookrightarrow rhs,\ P)\ .\pi & = & ([E]\bar{q} \hookrightarrow rhs),\ P\ .\pi & \\ ([E].\pi'\ \bar{q} \hookrightarrow rhs,\ P)\ .\pi & = & P\ .\pi & \text{if } \pi \neq \pi' \end{array}$$

Fig. 26. Partially decomposed clauses after a copattern split (partial function).

$\boxed{\Sigma \vdash P\sigma \Rightarrow P'}$ (Σ fixed, dropped from rules.)

$$\frac{}{\epsilon\sigma \Rightarrow \epsilon} \qquad \frac{(v \mathrel{/^?} p : A) \in E \qquad v\sigma \mathrel{/^?} p : A\sigma \Rightarrow \bot \qquad P\sigma \Rightarrow P'}{([E]\bar{q} \hookrightarrow rhs, P)\sigma \Rightarrow P'}$$

$$\frac{E = \{w_k \mathrel{/^?} p_k : A_k \mid k = 1 \ldots n\} \qquad (w_k\sigma \mathrel{/^?} p_k : A_k\sigma \Rightarrow E_i)_{k=1\ldots n} \qquad P\sigma \Rightarrow P'}{([E]\bar{q} \hookrightarrow rhs, P)\sigma \Rightarrow ([\bigcup_i E_i]\bar{q} \hookrightarrow rhs), P'}$$

Fig. 27. Rules for transforming partially decomposed clauses after refining the pattern with a case split.

$\boxed{\Sigma \vdash v \mathrel{/^?} p : A \Rightarrow E_\bot}$ and $\boxed{\Sigma \vdash \bar{v} \mathrel{/^?} \bar{p} : \Delta \Rightarrow E_\bot}$ (Σ fixed, dropped from rules)
For the purpose of these rules, c also ranges over hcomp, i0, and i1.

$$\frac{v \searrow \mathsf{c}\ \bar{e} \qquad A \searrow \mathsf{D}\ \bar{u} \qquad \text{constructor c } \Delta_\mathsf{c}\ [\bar{j} \mid b] : \mathsf{D}\ \Delta \in \Sigma \qquad \bar{e} \mathrel{/^?} \bar{q} : \Delta_\mathsf{c}(\bar{j} : \mathbb{I})[\bar{u} \,/\, \Delta] \Rightarrow E_\bot}{v \mathrel{/^?} \mathsf{c}\ \bar{q} : A \Rightarrow E_\bot}$$

$$\frac{v \searrow \mathsf{c}'\ \bar{v} \qquad \mathsf{c} \neq \mathsf{c}'}{v \mathrel{/^?} \mathsf{c}\ \bar{e} : A \Rightarrow \bot} \qquad \frac{}{v \mathrel{/^?} p : A \Rightarrow \{v \mathrel{/^?} p : A\}}$$

$$\frac{}{\epsilon \mathrel{/^?} \epsilon : \epsilon \Rightarrow \{\}} \qquad \frac{v \mathrel{/^?} p : A \Rightarrow E_\bot \qquad \bar{v} \mathrel{/^?} \bar{p} : \Delta[v \,/\, x] \Rightarrow E'_\bot}{v\ \bar{v} \mathrel{/^?} p\ \bar{p} : (x : A)\Delta \Rightarrow E_\bot \cup E'_\bot}$$

Fig. 28. Rules for simplifying the constraints of a partially decomposed clause.

$\boxed{\Sigma; \Gamma \vdash \emptyset : A}$ (Σ fixed, dropped from rules)

$$\frac{A \searrow \mathsf{D}\ \bar{v} \qquad \mathsf{data}\ \mathsf{D}\ \Delta : \mathsf{Set}_\ell\ \mathsf{where}\ \epsilon \in \Sigma}{\Gamma \vdash \emptyset : A} \qquad \frac{A \searrow \mathsf{Eq}_B\ u\ v \qquad \Gamma \vdash^r_p u =^? v : B \Rightarrow \mathrm{NO}}{\Gamma \vdash \emptyset : A}$$

Fig. 29. Rules for caseless types.

REFERENCES

Jesper Cockx and Andreas Abel. 2018. Elaborating Dependent (Co)Pattern Matching. *Proc. ACM Program. Lang.* 2, ICFP, Article 75 (July 2018), 30 pages. https://doi.org/10.1145/3236770

Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. 2018. Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom. In *Types for Proofs and Programs (TYPES 2015) (LIPIcs)*, Vol. 69. 5:1–5:34.

Simon Huber. 2017. A Cubical Type Theory for Higher Inductive Types. (2017). http://www.cse.chalmers.se/~simonhu/misc/hcomp.pdf