Shape-Irrelevant Dependent Types and Unification Constraints
Or:
Why Pattern Inductive Families Need Not Store Their Indices

Andreas Abel, Christoph-Simon Senjak

December 2012

TODO.

1. map and foldr for RVec and IVec
2. Write typing rules

1 Syntax

Grammar.

$$
\begin{array}{llcll}
\mathsf{Ann} & \ni & \star & ::= & : & \text{relevant} \\
 & & & \mid & \hat{:} & \text{shape-irrelevant} \\
 & & & \mid & \div & \text{irrelevant} \\
 & & & & & \\
\mathsf{Exp} & \ni & r, s, t, u, & & & \\
 & & S, T, U & ::= & 1 & \text{unit type} \\
 & & & \mid & () & \text{unit intro} \\
 & & & \mid & r{=}s \mid r\hat{=}s \mid r\overline{=}s & \text{intensional equalities} \\
 & & & \mid & T_1 + T_2 & \text{disjoint sum} \\
 & & & \mid & \mathsf{inl}\ t \mid \mathsf{inr}\ t \mid \mathsf{case}_T(r)\{\ \mathsf{inl}\ \ \Rightarrow s; \mathsf{inr}\ \ \Rightarrow t\ \} & \text{disjoint sum intro/elim} \\
 & & & \mid & (x{\star}U) \to T & \text{dependent function type} \\
 & & & \mid & x \mid \lambda x t \mid r\ s & \text{lambda-calculus} \\
 & & & \mid & (x{\star}U) \times T & \text{dependent pair type} \\
 & & & \mid & (t_1, t_2) \mid \mathsf{unpair}_T(t)\{\ (\cdot,\cdot) \Rightarrow f\ \} & \text{pair intro/elim} \\
 & & & \mid & \mu T & \text{inductive type} \\
 & & & \mid & \mathsf{in}\, t \mid \mathsf{out}\, r \mid \mathsf{rec}_T\ t & \text{inductive type intro/elim} \\
 & & & \mid & \text{ᗺ}_{(\Delta \vdash p \doteq t : T)} U & \text{unification constraint} \\
 & & & \mid & \mathsf{split}\ r & \text{empty type elimination} \\
 & & & \mid & \mathsf{split}\ r\ \mathsf{in}\ \hat{\Delta}, x.\ t & \text{constraint elimination} \\
 & & & \mid & \mathsf{Set} & \text{universe of small types} \\
 & & & & & \\
\mathsf{Cxt} & \ni & \Gamma, \Delta & ::= & \diamond \mid \Gamma.\, x{\star}T & \text{contexts} \\
\mathsf{Pat} & \ni & p, q & ::= & x \mid \mathsf{inl}\ p \mid \mathsf{inr}\ p \mid (p_1, p_2) \mid \mathsf{in}\, p & \text{patterns} \\
\mathsf{UP} & \ni & P, Q & ::= & (\Delta \vdash p \doteq t : T) & \text{unification problem}
\end{array}
$$

We let $\underline{\Delta}$ denote the list of the variables bound in Δ. Thus, $\underline{\diamond} = \diamond$ and $\underline{\Delta.\, x{\star}T} = \underline{\Delta}, x$. The variables $\underline{P}$ bound by a unification problem are $\underline{(\Delta \vdash p \doteq t : T)} = \underline{\Delta}$. Further, let $\mathsf{FV}(t)$ be the list of free variable occurrences in term t.

Substitutions σ are maps from variables to expressions. We require that the domain $\mathsf{dom}(\sigma) = \{x \mid \sigma(x) \neq x\}$ is finite. We write id for the identity substitution and $[u := x]$ for the singleton substitution σ with $\mathsf{dom}(\sigma) = \{x\}$ and $\sigma(x) = u$. Capture avoiding parallel substitution of σ in t is written as juxtaposition $t\sigma$.

Operations on relevance annotations and contexts Let the linear order *more irrelevant* $\leq$ on relevance annotations be given by $: \leq \hat{:} \leq \div$. Relevance composition is given by:

$$\star_1 \star_2 = \max(\star_1, \star_2)$$

Relevance "exponentiation".

$$\begin{array}{lcll} \div^{\oplus} & = & \hat{:} & \\ \star^{\oplus} & = & \star & \text{for } \star \neq \div \end{array} \qquad \begin{array}{lcll} \hat{:}^{\ominus} & = & \div & \\ \star^{\ominus} & = & \star & \text{for } \star \neq \hat{:} \end{array} \qquad {\star_1}^{\star_2} = \begin{cases} : & \text{if } \star_1 \leq \star_2 \\ \star_1 & \text{otherwise} \end{cases}$$

These operations are idempotent. $-^{\oplus}$ and ${\star_0}^{-}$ are antitone, $-^{\ominus}$ and $-^{\star}$ are monotone. The operation $-^{\oplus}$ means *going to the type world (and shape equality)* and the inverse operation $-^{\ominus}$ means *going to the term world (and exact equality)*. Further laws:

1. $({\star_1}^{\star_2})^{\star_3} = {\star_1}^{(\star_2 \star_3)}$. (Proof: If $\star_1 = :$ or $\div \in \{\star_2, \star_3\}$ then the result is $:$ on both sides. Relevance level $: \in \{\star_2, \star_3\}$ is ignored in composition and is neutral as exponent. In the remaining case, $\star_2 = \star_3 = \hat{:}$, and we conclude by idempotency.)

2. ${\star_1}^{\oplus} \leq \star_2$ iff $\star_1 \leq {\star_2}^{\ominus}$ (Galois connection) iff $\star_1 \leq \star_2$ or $\star_1 = \div$ and $\star_2 = \hat{:}$.

3. $(\star^{\ominus})^{\oplus} = \star^{\oplus}$ and $(\star^{\oplus})^{\ominus} = \star^{\ominus}$.

4. $({\star^{\oplus}}_1)^{\star_2} = {\star_1}^{{\star_2}^{\ominus}}$, in particular, $(\star^{\oplus})^{\hat{:}} = \star^{\hat{:}^{\ominus}} = \star^{\div} = :$.

These operations are lifted to contexts Γ by applying them to the relevance operations of all bindings of Γ, the relevant $(x\!:\!U)$, shape-irrelevant $(x \mathrel{\hat{:}} U)$, and irrelevant $(x \div U)$ ones.

Resurrection $\Gamma^{\oplus}$ turns all irrelevant bindings $x \div T$ into shape-irrelevant $x \mathrel{\hat{:}} T$ ones. Operation $\Gamma^{:}$ returns just Γ, $\Gamma^{\hat{:}}$ makes all shape-irrelevant bindings relevant, and $\Gamma^{\div}$ makes *all* bindings relevant.

We can extend relevance composition to include the symbols $\oplus$ and $\ominus$; let $o ::= \star \mid \oplus \mid \ominus$. Relevance $:$ remains the neutral element of composition, and $\div$ the dominant element. Altogether, we get a non-commutative operation with the following laws:

1. $:o = o: = o$ and $\div o = o \div = \div$.

2. $\oplus\ominus = \ominus$ and $\ominus\oplus = \oplus$.

3. $\ominus\hat{:} = \ominus$ and $\hat{:}\ominus = \hat{:}$.

4. $\oplus\hat{:} = \div$, but $\Gamma^{\hat{:}\oplus}$ is a new context operation, a kind of "decrement", making shape-irrelevant assumptions relevant, and irrelevant ones shape-irrelevant.

We extend the relevance ordering by $\div \geq \oplus \geq : \geq \ominus$. Note that $\oplus$ and $\hat{:}$ are incomparable. We have $\forall\star(\star^{o} \leq \star^{o'})$ iff $o \geq o'$. It does not make sense to extend exponentiation to $o^{o'}$, since while the exponent may be an arbitrary context operation, the base must be a relevance level.

2 Declarative Presentation

Judgements.

$\Gamma \vdash T$	T is a wellformed type in context Γ
$\Gamma \vdash t : T$	t has type T in context Γ
$\Gamma \vdash T = T'$	T and T' are equal types in context Γ
$\Gamma \vdash t = t' : T$	t and t' are equal terms of type T in context Γ

Derived Judgements.

$\Delta \vdash_\Gamma p : T$	p is a wellformed pattern of type T with pattern variables Δ
$\Gamma \vdash P$	P is a well-formed unification problem in context Γ
$\Gamma \vdash P\lightning$	unification problem P is unsolvable
$\Gamma \vdash P \searrow \Gamma' \vdash \sigma; \tau$	unification problem P has most general solution σ, τ

Pattern typing.

$$\Delta \vdash_\Gamma p : T \;:\Longleftrightarrow\; \Gamma \vdash T \text{ and } \Gamma.\, \Delta \vdash p : T \text{ and } \mathsf{FV}(p) = \underline{\underline{\Delta}}$$

Well-formed unification problems.

$$\frac{\Gamma \vdash T \qquad \Gamma^{\hat{\vdots}} \vdash t : T \qquad \Gamma \vdash \Delta \qquad \Delta^{\hat{\vdots}} \vdash_{\Gamma^{\hat{\vdots}}} p : T}{\Gamma \vdash (\Delta \vdash p \doteq t : T)}$$

Typing. Invariant: $\Gamma \vdash t : T$ implies $\Gamma^{\oplus} \vdash T$.

Basic types.

$$\frac{}{\Gamma \vdash 1} \qquad \frac{}{\Gamma \vdash () : 1} \qquad \frac{\Gamma \vdash t * 1}{\Gamma \vdash t = () * T}$$

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A + B} \qquad \frac{\Gamma \vdash x : A \qquad \Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash (\mathsf{inl}\ x) : (A + B)} \qquad \frac{\Gamma \vdash x : A \qquad \Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash (\mathsf{inr}\ x) : (B + A)}$$

$$\frac{\Gamma \vdash r : (A + B) \qquad \Gamma \vdash s : A \to C \qquad \Gamma \vdash t : B \to C}{\Gamma \vdash (\mathsf{case}_C(r)\{\ \mathsf{inl}\ \Rightarrow s; \mathsf{inr}\ \Rightarrow t\ \}) : C}$$

$$\frac{\Gamma.x * A \vdash B}{\Gamma \vdash (x * A) \times B} \qquad \frac{\Gamma.x * A \vdash y : B}{\Gamma \vdash (x, y) : ((x * A) \times B)}$$

$$\frac{\Gamma \vdash t : (x * A) \times B \qquad \Gamma \vdash f : (z * A) \to B[x := z] \to C}{\Gamma \vdash (\mathsf{unpair}_C(t)\{\ (\cdot, \cdot) \Rightarrow f\ \}) : C}$$

Term equality.

$$\frac{\Gamma \vdash t : T}{\Gamma \vdash t = t : T} \qquad \frac{\Gamma \vdash p[x := t] : Q[x := t] \qquad \Gamma \vdash t = s : T}{\Gamma \vdash p[x := s] : Q[x := s]}\ s, t \text{ free for } x \text{ in } p, Q$$

$$\frac{\Gamma \vdash s = t : T}{\Gamma \vdash t = s : T} \qquad \frac{\Gamma \vdash t : T}{\Gamma \vdash s = t : T}\ s =_{\beta\eta} t$$

Type equality.

$$\frac{\Gamma \vdash T}{\Gamma \vdash T = T} \qquad \frac{\Gamma \vdash T[x := t] \qquad \Gamma \vdash t = u : V}{\Gamma \vdash T[x := t] = T[x := u]}\ t, u \text{ free for } x \text{ in } T$$

$$\frac{\Gamma \vdash x : T \qquad \Gamma \vdash T = U}{\Gamma \vdash x : U}$$

Functions.

$$\frac{\Gamma \vdash U \qquad \Gamma .\, x{\star}^{\oplus} U \vdash T}{\Gamma \vdash (x{\star}U) \to T}$$

instances: $$\frac{\Gamma \vdash U \qquad \Gamma .\, x{:}U \vdash T}{\Gamma \vdash (x{:}U) \to T} \qquad \frac{\Gamma \vdash U \qquad \Gamma .\, x \mathrel{\hat{:}} U \vdash T}{\Gamma \vdash (x \mathrel{\hat{:}} U) \to T} \qquad \frac{\Gamma \vdash U \qquad \Gamma .\, x \mathrel{\hat{:}} U \vdash T}{\Gamma \vdash (x{\div}U) \to T}$$

$$\frac{(x \star U) \in \Gamma}{\Gamma \vdash x : U}\ \star = : \qquad \frac{\Gamma .\, x{\star}U \vdash t : T}{\Gamma \vdash \lambda x t : (x{\star}U) \to T} \qquad \frac{\Gamma \vdash r : (x{\star}U) \to T \qquad \Gamma^{\star} \vdash u : U}{\Gamma \vdash r\, u : T[u := x]}$$

Unification constraints.

$$\frac{\Gamma \vdash (\Delta \vdash p \doteq t : T) \qquad \Gamma .\, \Delta \vdash U}{\Gamma \vdash \reflectbox{B}_{(\Delta \vdash p \doteq t : T)} U}$$

$$\frac{\Gamma \vdash P \searrow \Gamma' \vdash \sigma ; \tau \qquad \Gamma' \vdash u[\tau] : U[\tau, \sigma]}{\Gamma \vdash u : \reflectbox{B}_P U}$$

$$\frac{\Gamma \vdash r : \reflectbox{B}_P S \qquad \Gamma^{\oplus} \vdash P\lightning}{\Gamma \vdash \mathsf{split}\ r : U}$$

$$\frac{\Gamma \vdash r : \reflectbox{B}_P S \qquad \Gamma^{\oplus} \vdash P \searrow \Gamma' \vdash \sigma ; \tau \qquad \Gamma', x{:}S[\tau, \sigma] \vdash u[\tau, \sigma] : U[\tau]}{\Gamma \vdash \mathsf{split}\ r\ \mathsf{in}\ \underline{P}, x.\ u : U}$$

Recursion.

$$\frac{\Gamma \vdash t : F\,(\mu F)\,\vec{t}}{\Gamma \vdash \mathsf{in}\, t : \mu F\, \vec{t}} \qquad \frac{\Gamma \vdash r : \mu F\, \vec{t}}{\Gamma \vdash \mathsf{out}\, r : F\,(\mu F)\,\vec{t}}$$

$$\frac{\begin{array}{l}\Gamma .\, \Delta .\, x{:}\mu F \underline{\Delta} \vdash C[\underline{\Delta}, x] \\ \Gamma \vdash f : (\vec{y} : \Delta .\ y : F\,((\Delta .\, x{:}\mu F \underline{\Delta} \vec{y}) \times C[\underline{\Delta}, x])) \to C[\vec{y}, \mathsf{in}\, y]\end{array}}{\Gamma \vdash \mathsf{rec}_{\underline{\Delta}, x. C[\underline{\Delta}, x]}\ f : (\Delta .\, \mathsf{x}{:}\mu F \underline{\Delta}) \to C[\underline{\Delta}, x]}$$

Relevance-relative typing. $\boxed{\Gamma \vdash t \star T}$ iff $\Gamma^{\star} \vdash t : T$.

Substitution typing. $\boxed{\Gamma \vdash \sigma : \Delta}$ iff $\Gamma \vdash \sigma(x) \star T$ for all $(x \star T) \in \Delta$.

Lemma 1 (Substitution) *If $\Gamma \vdash \sigma : \Delta$ and $\Delta \vdash t : T$ then $\Gamma \vdash t\sigma : T\sigma$.*

Lemma 2 (Substitution promotion)

1. *$\oplus \star_0^{\oplus} \geq \star_0$ and $\star \star_0^{\star} \geq \star_0$.*
2. *If $\Gamma \vdash \sigma : \Delta$ then $\Gamma^{\oplus} \vdash \sigma : \Delta^{\oplus}$ and $\Gamma^{\star} \vdash \sigma : \Delta^{\star}$.*

Example 1 (Booleans)

Bool	:	Set
Bool	:=	$1 + 1$
false, true	:	Bool
false	:=	inl ()
true	:=	inr ()

Example 2 (Empty type)

$$
\begin{array}{lll}
0 & : & \mathsf{Set} \\
0 & := & \reflectbox{B}_{(\diamond \vdash \mathsf{true} \doteq \mathsf{false} : \mathsf{Bool})}\, 1 \\
\mathsf{abort}_{A:\mathsf{Set}} & : & 0 \to A \\
\mathsf{abort}_{A}\, r & := & \mathsf{split}\ r
\end{array}
$$

Example 3 (Option type)

$$
\begin{array}{lll}
\mathsf{Maybe} & : & \mathsf{Set} \to \mathsf{Set} \\
\mathsf{Maybe} & : & \lambda A.\, 1 + A \\
\\
\mathsf{nothing}_{A:\mathsf{Set}} & : & \mathsf{Maybe}\ A \\
\mathsf{nothing}_{A} & := & \mathsf{inl}\ () \\
\\
\mathsf{just}_{A:\mathsf{Set}} & : & A \to \mathsf{Maybe}\, A \\
\mathsf{just}_{A}\, x & := & \mathsf{inr}\ x \\
\\
\mathsf{maybe}_{A:\mathsf{Set}, C[x:\mathsf{Maybe}\, A]} & : & C[\mathsf{nothing}] \to ((y\!:\!A) \to C[\mathsf{just}\, y]) \to (x\!:\!\mathsf{Maybe}\, A) \to C[x] \\
\mathsf{maybe}_{A, C[x]} & := & \lambda n.\, \lambda j.\, \lambda x.\, \mathsf{case}(x)\{\ \mathsf{inl}\ \Rightarrow \lambda\_.\, n; \mathsf{inr}\ \Rightarrow j\ \}
\end{array}
$$

Example 4 (Natural numbers)

$$
\begin{array}{lll}
\mathsf{Nat} & : & \mathsf{Set} \\
\mathsf{Nat} & := & \mu\, \mathsf{Maybe} \\
\\
\mathsf{zero} & : & \mathsf{Nat} \\
\mathsf{zero} & := & \mathsf{in}\, \mathsf{nothing} \\
\\
\mathsf{suc} & : & \mathsf{Nat} \to \mathsf{Nat} \\
\mathsf{suc}\, x & := & \mathsf{in}\, (\mathsf{just}\, x)
\end{array}
$$

Specialized recursor for natural numbers:

$$
\begin{array}{lll}
\mathsf{natrec}_{C[x:\mathsf{Nat}]} & : & C[\mathsf{zero}] \to ((x\!:\!\mathsf{Nat}) \to C[x] \to C[\mathsf{suc}\, x]) \to (x\!:\!\mathsf{Nat}) \to C[x] \\
\mathsf{natrec}_{C} & := & \lambda z.\, \lambda s.\, \mathsf{rec}_{C}\ \ldots
\end{array}
$$

Example 5 (Recursive vectors)

$$
\begin{array}{lll}
\mathsf{RVec} & : & \mathsf{Set} \to \mathsf{Nat} \to \mathsf{Set} \\
\mathsf{RVec} & := & \lambda A.\, \mathsf{natrec}_{\_.\mathsf{Set}}\, 1\, (\lambda\_.\, \lambda X.\, A \times X) \\
\\
\mathsf{rnil}_{A:\mathsf{Set}} & : & \mathsf{RVec}\, A\, \mathsf{zero} \\
\mathsf{rnil}_{A} & := & () \\
\\
\mathsf{rcons}_{A:\mathsf{Set}} & : & (n\!:\!\mathsf{Nat}) \to A \to \mathsf{RVec}\, A\, n \to \mathsf{RVec}\, A\, (\mathsf{suc}\, n) \\
\mathsf{rcons}_{A}\, n\, x\, xs & := & (x, xs)
\end{array}
$$

Example 6 (Inductive vectors, equality encoding)

$$
\begin{array}{lll}
\mathsf{IVec} & : & \mathsf{Set} \to \mathsf{Nat}\ \mathrel{\hat{\to}} \mathsf{Set} \\
\mathsf{IVec} & := & \lambda A.\, \mu\, \lambda X\ :\ \mathsf{Nat}\ \mathrel{\hat{\to}} \mathsf{Set}.\, \lambda n \mathrel{\hat{:}} \mathsf{Nat}.\, (n \equiv \mathsf{zero}) + (m \mathrel{\hat{:}} \mathsf{Nat}) \times (n \equiv \mathsf{suc}\, m) \times A \times X\, m \\
\\
\mathsf{inil}_{A:\mathsf{Set}} & : & \mathsf{IVec}\, A\ \mathsf{zero} \\
\mathsf{inil}_{A} & := & \mathsf{in}\, (\mathsf{inl}\ \mathsf{refl}) \\
\\
\mathsf{icons}_{A:\mathsf{Set}} & : & (n \mathrel{\hat{:}} \mathsf{Nat}) \to A \to \mathsf{IVec}\, A\, n \to \mathsf{IVec}\, A\, (\mathsf{suc}\, n) \\
\mathsf{icons}_{A}\, n\, x\, xs & := & \mathsf{in}\, (\mathsf{inr}\ (m, (\mathsf{refl}, (x, xs))))
\end{array}
$$

Example 7 (Inductive vectors, constraint encoding)

$$
\begin{array}{lll}
\mathsf{IVec} & : & \mathsf{Set} \to \mathsf{Nat} \mathrel{\hat{\to}} \mathsf{Set} \\
\mathsf{IVec} & := & \lambda A.\, \mu\, \lambda X : \mathsf{Nat} \mathrel{\hat{\to}} \mathsf{Set}.\, \lambda n \mathrel{\hat{:}} \mathsf{Nat}. \\
 & & (ᗺ_{(\diamond\, \vdash \mathsf{zero} \doteq n : \mathsf{Nat})} 1) + (ᗺ_{(m \hat{:} \mathsf{Nat} \vdash \mathsf{suc}\, m \doteq n : \mathsf{Nat})} A \times X\, m) \\
\mathsf{inil}_{A:\mathsf{Set}} & : & \mathsf{IVec}\, A\, \mathsf{zero} \\
\mathsf{inil}_{A} & := & \mathsf{in}\, (\mathsf{inl}\, ()) \\
\mathsf{icons}_{A:\mathsf{Set}} & : & (n \div \mathsf{Nat}) \to A \to \mathsf{IVec}\, A\, n \to \mathsf{IVec}\, A\, (\mathsf{suc}\, n) \\
\mathsf{icons}_{A}\, n\, x\, xs & := & \mathsf{in}\, (\mathsf{inr}\, (x, xs))
\end{array}
$$

Excerpts of typing derivations: Case IVec. *Let* $\Gamma = A : \mathsf{Set}.\, X : \mathsf{Nat} \mathrel{\hat{\to}} \mathsf{Set}.\, n \mathrel{\hat{:}} \mathsf{Nat}$.

$$
\dfrac{\dfrac{m : \mathsf{Nat} \vdash_{\Gamma^{\hat{:}}} \mathsf{suc}\, m : \mathsf{Nat} \qquad \Gamma^{\hat{:}} \vdash n : \mathsf{Nat}}{\Gamma \vdash (m \mathrel{\hat{:}} \mathsf{Nat} \vdash \mathsf{suc}\, m \doteq n : \mathsf{Nat})} \qquad \dfrac{\dfrac{\Gamma^{\hat{:}}.\, m : \mathsf{Nat} \vdash m : \mathsf{Nat}}{\Gamma.\, m \mathrel{\hat{:}} \mathsf{Nat} \vdash X\, m}}{\Gamma.\, m \mathrel{\hat{:}} \mathsf{Nat} \vdash A \times X\, m}}{\Gamma \vdash ᗺ_{(m \hat{:} \mathsf{Nat} \vdash \mathsf{suc}\, m \doteq n : \mathsf{Nat})} A \times X\, m}
$$

Case icons. *Let* $\Gamma = A : \mathsf{Set}.\, n \div \mathsf{Nat}.\, x : A.\, xs : \mathsf{IVec}\, A\, n$.

$$
\dfrac{\dfrac{m \mathrel{\hat{:}} \mathsf{Nat} \vdash^{\mathsf{a}}_{\Gamma} m \doteq n : \mathsf{Nat} \searrow \Gamma \vdash \mathsf{id}; [n/m]}{m \mathrel{\hat{:}} \mathsf{Nat} \vdash^{\mathsf{a}}_{\Gamma} \mathsf{suc}\, m \doteq \mathsf{suc}\, n : \mathsf{Nat} \searrow \Gamma \vdash \mathsf{id}; [n/m]} \qquad \dfrac{\Gamma \vdash xs : \mathsf{IVec}\, A\, n}{\Gamma \vdash (x, xs) : A \times \mathsf{IVec}\, A\, n}}{\Gamma \vdash (x, xs) : (ᗺ_{(m \hat{:} \mathsf{Nat} \vdash \mathsf{suc}\, m \doteq \mathsf{suc}\, n : \mathsf{Nat})} A \times \mathsf{IVec}\, A\, m)}
$$

3 Algorithmic Presentation

Judgements.

$\Gamma \vdash^{\mathsf{a}} T$	T is a wellformed type in context Γ
$\Gamma \vdash^{\mathsf{a}} t \leftleftarrows T$	t checks against type T in context Γ
$\Gamma \vdash^{\mathsf{a}} r \rightrightarrows T$	r has inferred type T in context Γ
$\Gamma \vdash^{\mathsf{a}} P$	P is a well-formed unification problem in context Γ
$\Delta \vdash^{\mathsf{a}}_{\Gamma} p \lightning t : T$	unification of expression t and pattern p fails
$\Delta \vdash^{\mathsf{a}}_{\Gamma} p \doteq t : T \searrow \Gamma' \vdash \sigma; \tau$	expression t unifies with pattern p under m.g.u. σ, τ

Unification. $\boxed{\Delta \vdash^{\mathsf{a}}_{\Gamma} p \doteq t : T \searrow \Gamma' \vdash \tau; \sigma}$

Algorithm for unification. Concatenation of pairs of substitutions are meant component-wise, concatenation with $\lightning$ yields $\lightning$. z is always a new variable which has not been used yet, q and r are placeholders for either $(\cdot, \cdot)$, or for in inl inr ignoring the second argument.

$$
\begin{array}{rcl}
x \overset{?}{\searrow} y & = & ([x := z], [y := z]) \\
x \overset{?}{\searrow} q(t, u) & = & ([x := q(t, u)[x := z]], [x := z]) \\
q(t, u) \overset{?}{\searrow} x & = & \begin{cases} (\tau, \sigma) & \text{for } x \overset{?}{\searrow} q(t, u) = (\sigma, \tau) \\ \lightning & \text{for } x \overset{?}{\searrow} q(t, u) = \lightning \end{cases} \\
q(t, u) \overset{?}{\searrow} r(v, w) & = & \begin{cases} (t \overset{?}{\searrow} v) \circ (u \overset{?}{\searrow} w) & \text{for } q = r \\ \lightning & \text{otherwise} \end{cases}
\end{array}
$$

Invariants: If $\Gamma^{\oplus} \vdash T$ and $\Gamma^{\oplus} \vdash \Delta$ and $\Gamma^{\oplus\hat{\vdots}} \vdash t : T$ and $\Delta^{\hat{\vdots}} \vdash_{\Gamma^{\oplus\hat{\vdots}}} p : T$ then $\Gamma' \vdash \tau : \Gamma$ and $\Gamma'^{\oplus} \vdash \sigma : \Delta\tau$ and $\Gamma'^{\oplus\hat{\vdots}} \vdash p\sigma = t\tau : T\tau$. Remember that $\Gamma^{\oplus\hat{\vdots}} = \Gamma^{\div}$.

$$\frac{}{\Delta \vdash^{\mathsf{a}}_{\Gamma} x \doteq t : T \searrow \Gamma \vdash \mathsf{id}; \langle t \rangle x} \qquad \frac{}{\Delta \vdash^{\mathsf{a}}_{\Gamma.\, y \star T, \Gamma'} p \doteq y : T \searrow \Gamma.\, \Delta.\, \Gamma' \vdash \langle p \rangle y; \mathsf{id}}$$

References

[AS12] Andreas Abel and Gabriel Scherer. On irrelevance and algorithmic equality in predicative type theory. *Logical Meth. in Comput. Sci.*, 8(1:29):1–36, 2012. TYPES'10 special issue.