Tones and Types

Michael Arntzenius, daekharel@gmail.com

16 May 2018

Abstract. Certain properties of maps between preorders (e.g. preserving equivalence) reduce to monotonicity with respect to an altered domain ordering. I dub such alterations "tones", and explore their theory. I sketch a typed λ-calculus of monotone functions, using tones to allow selective non-monotonicity.

1 Preorders

A preorder is a relation $a \leqslant b$ satisfying:

1. **Reflexivity:** $a \leqslant a$.
2. **Transitivity:** If $a \leqslant b$ and $b \leqslant c$ then $a \leqslant c$.

Preorders generalize partial orders by not requiring antisymmetry. Let $a \equiv b$ iff $a \leqslant b$ and $b \leqslant a$. Antisymmetry means $a \equiv b$ implies $a = b$. A good example preorder is "lists under containment", where $a \leqslant b$ iff every element of a is also in b. Note that $[0, 1] \equiv [1, 0]$, but $[0, 1] \neq [1, 0]$.

To a category theorist, a preorder is a "thin" category: between any two objects there is at most one morphism. I suspect much of the "tone theory" in this document, ostensibly about maps between preorders, extends to functors between categories.

2 Tones

Tones are ways a function f may respect a preorder. I will consider four tones, id, op, iso, and path:

Tone	*Name*	*Property of* f
id	Monotone	$x \leqslant y \implies f(x) \leqslant f(y)$
op	Antitone	$x \geqslant y \implies f(x) \leqslant f(y)$
iso	Invariant	$x \leqslant y \wedge y \leqslant x \implies f(x) \leqslant f(y)$
path	Bivariant	$x \leqslant y \vee y \leqslant x \implies f(x) \leqslant f(y)$

Informally,

1. id is monotone (order-preserving).
2. op is antitone (order-inverting).
3. iso is invariant, preserving only equivalence.
4. path is bivariant: both monotone and antitone.

2.1 Tones transform orders

Fix preorders A, B. Let A^{op} be A, ordered oppositely. Now, observe that

$$f : A \to B \text{ is antitone}$$
iff
$$f : A^{op} \to B \text{ is monotone}$$

So "antitone" is a special case of "monotone"! This observation generalizes: every tone is really monotonicity with a transformation applied to the domain's ordering. So **tones transform orders**. I write A^s for the preorder A transformed by the tone s, defined:

Tone	*Meaning*	*Transformation on A*		
id	same ordering	$a \leqslant b : A$	$\iff$	$a \leqslant b : A^{id}$
op	opposite ordering	$a \geqslant b : A$	$\iff$	$a \leqslant b : A^{op}$
iso	induced equivalence	$a \leqslant b \wedge b \leqslant a : A$	$\iff$	$a \leqslant b : A^{iso}$
path	equivalence closure	$a \leqslant b \vee b \leqslant a : A$	$\implies$	$a \leqslant b : A^{path}$

With this, we can state the theorem generalizing our observation:

Theorem 1 (Tones transform orders)**.**

$$f : A \to B \text{ has tone } s$$
iff
$$f : A^s \to B \text{ is monotone}$$

From this point on, when I write $f : A \to B$, I mean implicitly that f is monotone; and therefore $f : A^s \to B$ means that f has tone s.

Here are a few more useful properties of tones, which I invite you to verify:

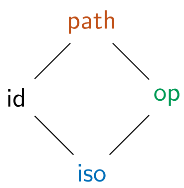
Theorem 2 (Functoriality of tones)**.** If $f : A \to B$ then $f : A^s \to B^s$.

Theorem 3 (Tones distribute over $\times$ and $+$)**.**

$$(A + B)^s = A^s + B^s$$
$$(A \times B)^s = A^s \times B^s$$

2.2 Understanding iso and path

An intuition about iso and path: iso keeps only the *strongly connected components*, while path turns *weakly* connected components into *strong* ones. I should use an example, with diagrams. I should also mention that iso and path turn preorders into equivalence relations, and the corresponding category-and-functor diagram.



$s \wedge t$	id	path	op	iso
id	id	id	iso	iso
path	id	path	op	iso
op	iso	op	op	iso
iso	iso	iso	iso	iso

	$s \cdot t$	t			
		id	op	iso	path
s	id	id	op	iso	path
	op	op	id	iso	path
	iso	iso	iso	iso	iso
	path	path	path	path	path

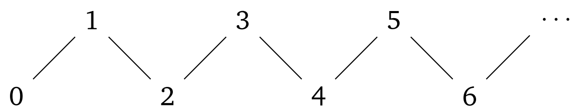
Figure 1: Tone lattice, meet, and composition

2.3 Defining path

A^{path} is the smallest preorder satisfying

$$a \leqslant b \vee b \leqslant a : A \implies a \leqslant b : A^{\text{path}}$$

Unfortunately A^{path} is not so easy to define directly (without language like "the smallest preorder satisfying") because the reverse implication, from $a \leqslant b : A^{\text{path}}$ to $a \leqslant b \vee b \leqslant a : A$, does not always hold. A good counterexample is the fencepost preorder on $\mathbb{N}$, where $a < a + 1$ for even a, and $a > a + 1$ for odd a:



Note that $a \leqslant b \vee b \leqslant a : \text{fencepost} \iff |a - b| \leqslant 1$, which isn't transitive! Instead, $\text{fencepost}^{\text{path}}$ is the *transitive closure* of $|a - b| \leqslant 1$, which makes everything equivalent:

$$0 \equiv 1 \equiv 2 \equiv 3 \equiv 4 \equiv \ldots$$

This suggests a direct definition: let a *path* from a_0 to a_n be a "zig-zagging" list $a_0, \ldots, a_n$ such that $a_i \leqslant a_{i+1} \vee a_i \geqslant a_{i+1} : A$. Then $a \leqslant b : A^{\text{path}}$ iff there is a path from a to b.

2.4 Tone operators

Figure 1 defines two operators on tones:

1. Meet $s \wedge t$ is greatest lower bound in the lattice ordered $\text{iso} < \{\text{id}, \text{op}\} < \text{path}$. This gives the general tone of the pairing $\langle f, g \rangle : A^{s \wedge t} \to B \times C$ of two functions $f : A^s \to B$ and $g : A^t \to C$.
2. Composition $s \cdot t$ gives the tone of a composed function $g \circ f : A^{s \cdot t} \to C$ when $f : A^s \to B$ and $g : B^t \to C$. Equivalently, $A^{s \cdot t} = (A^s)^t$ for any preorder A.

Properties of $\wedge$		*Properties of* $\cdot$	
Associativity	$(s \wedge t) \wedge u = s \wedge (t \wedge u)$	Associativity	$(s \cdot t) \cdot u = s \cdot (t \cdot u)$
Commutativity	$s \wedge t = t \wedge s$	Identity	$\mathsf{id} \cdot s = s = s \cdot \mathsf{id}$
Idempotence	$s \wedge s = s$	path left-absorbs	$\mathsf{path} \cdot s = \mathsf{path}$
path is identity	$\mathsf{path} \wedge s = s$	iso left-absorbs	$\mathsf{iso} \cdot s = \mathsf{iso}$
iso absorbs	$\mathsf{iso} \wedge s = \mathsf{iso}$	op self-inverts	$\mathsf{op} \cdot \mathsf{op} = \mathsf{id}$

Left-distribution	$s(t \wedge u) = st \wedge su$
Right-distribution	$(s \wedge t)u = su \wedge tu$

Figure 2: Properties of tone operators

By convention, composition binds tighter than meet, and $s \cdot t$ may be written simply as st. Thus $(s \cdot t) \wedge u = s \cdot t \wedge u = st \wedge u$. Together, $\wedge$ and $\cdot$ form a semiring whose properties are given in figure 2. TODO: Reference "I Got Plenty o' Nuttin'" — another use of variable annotations drawn from a semiring, and with similar (the same?) behavior of those annotations in the inference rules.

TODO: check the distribution laws hold!

2.5 Monotonicity Types

Monotonicity Types by Clancy, Miller, and Meiklejohn has similar tables for their composition $\circ$ and "contraction" $+$ operators! However, instead of bivariance they have constancy, a stricter condition. Constancy corresponds to respecting the *indiscrete ordering* (which sets $a \leqslant b$ for all a, b).[1] Interestingly, because constancy is so much stricter than bivariance, their composition operator is commutative.

They write $\uparrow$ for id, $\downarrow$ for op, ? for iso, and $\sim$ for constancy. They also add $=$ for the "tone" that *only the identity function* has. This doesn't fit my framework; it cannot be phrased as a transformation on orderings. However, it seems related to subtyping.

3 Tones, categorically

This section gives a categorical semantics of tones. You may safely skip it.

Let's change perspective. Section 2 defines tones as function properties, then gives corresponding preorder transformations. Now, let's define tones as preorder transformations, and derive corresponding function properties.

[1] Indiscreteness and its dual, discreteness, are also tones — that is, functorial transformations on the ordering component of a preorder — but they complicate things, so I omit them.

Note also that discreteness and iso coincide on posets; many intuitions transfer from one to the other.

Definition 4. $|{-}| : \textsc{preord} \to \textsc{set}$ is the functor taking a preorder to its set of elements.

Definition 5 (Tones)**.** A tone s is a functor $-^s : \textsc{preord} \to \textsc{preord}$ such that $|{-}^s| = |{-}|$.[2]

Corollary. For any preorder A and monotone map f,

1. $|A^s| = |A|$: tone functors alter a preorder's *ordering*, not its elements.
2. $|f^s| = |f|$: tone functors do not alter functions' behavior.

Definition 6. A function f from A to B has tone s iff $f : A^s \to B$ is monotone.

3.1 Tone composition

Tone composition $s \cdot t$ corresponds to composition of tone functors:

Theorem 7. The composition $(-^s)^t$ of two tones is itself a tone.

Proof. Applying definition 5, we have $|(-^s)^t| = |{-}^s| = |{-}|$. ☐

3.2 The TONE category

Preorders have a natural partial order, letting $A \leqslant B$ iff A is a *subpreorder* of B — that is, if $\lambda x.\ x : A \to B$.[3] This lifts pointwise to a partial order on tones: let $s \leqslant t$ iff $A^s \leqslant A^t$ for all A. As functors, tones also form a category, TONE, with natural transformations as morphisms. However, TONE is but a façade over this partial order:

Theorem 8.

$$s \leqslant t \iff \exists \eta : \textsc{tone}(s,t) \iff \exists! \eta : \textsc{tone}(s,t)$$

Proof. Expanding definitions, $s \leqslant t$ means $\lambda x.\ x : A^s \to A^t$ for all $A : \textsc{preord}$. By lemma 9, any $\eta : \textsc{tone}(s,t)$ is of the form $\eta_A = \lambda x.\ x : A^s \to A^t$. ☐

The crux here is that natural transformations between tones are *boring*:

Lemma 9. For any natural transformation $\eta : -^s \to -^t$, we have $\eta_A = \lambda x.\ x$.

Proof. Let $\mathbf{1}$ be the singleton preorder $\{\star\}$. Fix some $x : A$. Let $f : \mathbf{1} \to A = \lambda \star.\ x$. Then by naturality of η, this square commutes:

$$\begin{array}{ccc} \mathbf{1} & \xrightarrow{\eta_1} & \mathbf{1} \\ \downarrow{\scriptstyle f^s} & & \downarrow{\scriptstyle f^t} \\ A^s & \xrightarrow{\eta_A} & A^t \end{array}$$

[2] A more categorical approach might require only a natural isomorphism $\iota : |{-}^s| \simeq |{-}|$. I'm not yet comfortable generalizing that far.

[3] In other words, if $A \subseteq B$ and $x \leqslant y : A \implies x \leqslant y : B$.

From definition 5, $f^s = f = f^t$; and since **1** is a singleton, $\eta_1 = id$, thus:

$$\begin{aligned} \eta_A \circ f^s &= f^t \circ \eta_1 \\ \implies \eta_A \circ f &= f \\ \implies \eta_A(x) &= x \end{aligned}$$

□

3.3 The TONE lattice

Conjecture 10. TONE is a distributive lattice.

Proof. TODO □

Isn't $\wedge$ greatest lower bound / product? Don't we have $\vee$ / least upper bound / coproducts? Old stuff:

Tone meet $\wedge$ and composition $\cdot$ also have semantic versions. $s \wedge t$ corresponds to intersection of relations (noting that the intersection of two reflexive, transitive relations is itself reflexive and transitive), and $s \cdot t$ corresponds to functor composition.

$$\begin{aligned} a \leqslant b : A^{s \wedge t} &\iff a \leqslant b : A^s \wedge a \leqslant b : A^t \\ a \leqslant b : A^{st} &\iff a \leqslant b : (A^s)^t \end{aligned}$$

Conjecture 11. The intersection of two tones is a tone.

Proof. TODO □

Conjecture 12. The definitions of $\wedge$ and $\cdot$ in figure 1 coincide with their semantic definitions, applied to the semantic interpretation of id, op, path, and iso.

4 A tonal sequent calculus

$$(A_1^{t_1}, \ldots, A_n^{t_n})^u = (A_1^{t_1 u}, \ldots, A_n^{t_n u})$$

$$\textsc{Hypothesis}\ \frac{t \leqslant \mathsf{id}}{\Gamma, A^t \vdash A} \qquad \textsc{T-Right}\ \frac{\Gamma \vdash A}{\Gamma^t \vdash t\,A} \qquad \textsc{T-Left}\ \frac{\Gamma, A^{t \cdot u} \vdash C}{\Gamma, (t\,A)^u \vdash C} \qquad \textsc{Weakening}\ \frac{\Gamma, A^t \vdash C \quad u \leqslant t}{\Gamma, A^u \vdash C}$$

$$\textsc{Contraction}\ \frac{\Gamma, A^t, A^u \vdash C}{\Gamma, A^{t \wedge u} \vdash C} \qquad \textsc{Cut}\ \frac{\Gamma \vdash A \quad \Delta, A^t \vdash C}{\Gamma^t, \Delta \vdash C}$$

Jason Reed sent me these rules for a sequent calculus, where $t\,A$ is the type internalizing the tone functor $-^t$ applied to the type A. I aim to adapt this into a natural-deduction-style system with proof terms, i.e. a simply typed λ-calculus.

5 A bidirectional λ-calculus with tone inference

variables	x		
base types	P		
tones	s, t, u	$::=$	$\mathsf{id} \mid \mathsf{op} \mid \mathsf{path} \mid \mathsf{iso}$
cartesian operators	$\circledast$	$::=$	$+ \mid \times$
types	A, B, C	$::=$	$P \mid s\,A \mid A \to B \mid A \circledast B$
inferring terms	e	$::=$	$x \mid e\ m \mid \pi_i\ e \mid m : A$
checking terms	m, n	$::=$	$e \mid \lambda x.\ m \mid (m, n) \mid \mathsf{in}_i\ m$
		$\mid$	$\textbf{let}\ x = e\ \textbf{in}\ m$
		$\mid$	$\textbf{case}\ e\ \textbf{of}\ \mathsf{in}_1\ x \to m;\ \mathsf{in}_2\ y \to n$
contexts	Γ	$::=$	$\varepsilon \mid \Gamma, x : A^s$
judgments	J	$::=$	$m \Leftarrow \Gamma \vdash A \mid e \Rightarrow \Gamma \vdash A$
		$\mid$	$s \dashv t \mid s \leqslant t \mid A^s \leqslant B \mid A^s \prec B$

TODO: explain my various abuses of notation, e.g. Γ^t and $\Gamma_1 \wedge \Gamma_2$

5.1 Typing rules

Inferring forms

$$\frac{m \Leftarrow \Gamma \vdash A}{m : A \Rightarrow \Gamma \vdash A} \qquad \frac{}{x \Rightarrow x : A^{\mathsf{id}} \vdash A} \qquad \frac{e \Rightarrow \Gamma \vdash A \qquad A^s \prec B_1 \times B_2}{\pi_i\ e \Rightarrow \Gamma^s \vdash B_i}$$

$$\frac{e \Rightarrow \Gamma_1 \vdash A \qquad A^u \prec B \to C \qquad m \Leftarrow \Gamma_2 \vdash B}{e\ m \Rightarrow \Gamma_1^u \wedge \Gamma_2 \vdash C}$$

Checking forms

$$\frac{e \Rightarrow \Gamma \vdash A \qquad A^t \leqslant B}{e \Leftarrow \Gamma^t \vdash B} \qquad \frac{m \Leftarrow \Gamma \vdash A}{m \Leftarrow \Gamma^s \vdash s\,A} \qquad \frac{e \Rightarrow \Gamma_1 \vdash A \qquad m \Leftarrow \Gamma_2, x : A^s \vdash C}{\textbf{let}\ x = e\ \textbf{in}\ m \Leftarrow \Gamma_1^s \wedge \Gamma_2 \vdash C}$$

$$\frac{m \Leftarrow \Gamma, x : A^s \vdash B \qquad \mathsf{id} \leqslant s}{\lambda x.\ m \Leftarrow \Gamma \vdash A \to B} \qquad \frac{m \Leftarrow \Gamma_1 \vdash A_1 \qquad n \Leftarrow \Gamma_2 \vdash A_2}{(m, n) \Leftarrow \Gamma_1 \wedge \Gamma_2 \vdash A_1 \times A_2}$$

$$\frac{m \Leftarrow \Gamma \vdash A_i}{\mathsf{in}_i\ m \Leftarrow \Gamma \vdash A_1 + A_2}$$

$$\frac{e \Rightarrow \Gamma \vdash A \qquad A^s \prec B_1 + B_2 \qquad m \Leftarrow \Gamma_1, x : B_1^{t_1} \vdash C \qquad n \Leftarrow \Gamma_2, y : B_2^{t_2} \vdash C}{\textbf{case}\ e\ \textbf{of}\ \mathsf{in}_1\ x \to m;\ \mathsf{in}_2\ y \to n \Leftarrow \Gamma^{s \cdot (t_1 \wedge t_2)} \wedge \Gamma_1 \wedge \Gamma_2 \vdash C}$$

5.2 Tone judgments

TODO: Explain judgment $s \leqslant t$, for tone ordering, and $s \dashv t$, for tone adjunction.

$$\mathsf{id} \dashv \mathsf{id} \qquad \mathsf{op} \dashv \mathsf{op} \qquad \mathsf{path} \dashv \mathsf{iso} \qquad s \leqslant s \qquad \mathsf{iso} \leqslant s \qquad s \leqslant \mathsf{path}$$

5.3 Subtyping

TODO: Explain why we use tone-annotated subtyping.

In $A^s \leqslant B$, the types A and B are inputs, and the tone s is output.

$$\textsc{Refl}\ \frac{}{A^{\mathsf{id}} \leqslant A} \qquad \textsc{Map}\ \frac{A^s \leqslant B}{A^{st} \leqslant t\,B} \qquad \textsc{Adjoint}\ \frac{t \dashv s \qquad A^u \leqslant B}{(s\,A)^{tu} \leqslant B} \qquad \textsc{Distribute}\ \frac{A_1^s \leqslant A_2 \qquad B_1^t \leqslant B_2}{(A_1 \circledast B_1)^{s \wedge t} \leqslant A_2 \circledast B_2}$$

The semantic justification for ADJOINT is as follows. From $-^t \dashv -^s$ we have $A^{st} \to A \simeq A^s \to A^s$, thus $A^{st} \leqslant A$, thus $(s\,A)^t \leqslant A$, and so finally $(s\,A)^{tu} \leqslant A^u \leqslant B$. TODO: Bridge the gap between $\simeq$ and $=$; I think "tones don't alter the elements" suffices? Clean up this explanation. Explain that we use adjunction rather than $st \leqslant \mathsf{id}$ directly because adjunction gives us the *most informative* result; $st \leqslant \mathsf{id}$ is declarative, $t \dashv s$ is algorithmic. Give explanations for each other rule as well.

Function subtyping, $(A_1 \to B_1)^s \leqslant A_2 \to B_2$, has four rules — one for each tone s produced by $\leqslant sB_1B_2$:

$$\frac{A_1^s \leqslant A_2 \qquad B_1^{\mathsf{id}} \leqslant B_2 \qquad \mathsf{id} \leqslant s}{(A_1 \to B_1)^{\mathsf{id}} \leqslant A_2 \to B_2} \qquad \frac{A_1^s \leqslant A_2 \qquad B_1^{\mathsf{op}} \leqslant B_2 \qquad \mathsf{op} \leqslant s}{(A_1 \to B_1)^{\mathsf{op}} \leqslant A_2 \to B_2}$$

$$\frac{A_1^s \leqslant A_2 \qquad B_1^{\mathsf{path}} \leqslant B_2 \qquad \mathsf{iso} < s}{(A_1 \to B_1)^{\mathsf{path}} \leqslant A_2 \to B_2} \qquad \frac{A_1^{\mathsf{path}} \leqslant A_2 \qquad B_1^{\mathsf{iso}} \leqslant B_2}{(A_1 \to B_1)^{\mathsf{iso}} \leqslant A_2 \to B_2}$$

Can these $t \leqslant s$ constraints be turned into "composing with u is $\geqslant \mathsf{id}$", for some choice of u depending on t?

Subtyping at base types will depend on the base types you choose. Frequently, some base types' preorders will be symmetric (or even discrete, $x \leqslant y \iff x = y$), and therefore equivalence relations. Let "P equiv" hold if P's order is symmetric. Then the following rule is useful:

$$\frac{P \text{ equiv}}{P^{\mathsf{iso}} \leqslant P}$$

5.4 Mode stripping

$A^s \prec B$ is a specialization of $A^s \leqslant B$ which strips off modal operators on A, turning them into transformations on s. As in subtyping, A is an input and s an output;

however, B is now an output.

$$\frac{(\forall s, B)\ \ A \neq s\,B}{A^{\mathsf{id}} \prec A} \qquad \frac{t \dashv s \qquad A^{u} \prec B}{(s\,A)^{tu} \prec B}$$

TODO: note that we cannot strip the mode path. path is basically a pariah; we cannot eliminate it through mode stripping, and we don't have an explicit elimination rule.

5.5 Tones and the λ rule

Here are two more general variations on the λ rule I've considered:

$$\textsc{Fn-1}\ \frac{m \Leftarrow \Gamma, x : A^{s} \vdash B \qquad A \leqslant A^{s}}{\lambda x.\ m \Leftarrow \Gamma \vdash A \to B} \qquad \textsc{Fn-2}\ \frac{m \Leftarrow \Gamma, x : A^{t} \vdash B \qquad A^{s} \leqslant A \qquad \mathsf{id} \leqslant s \cdot t}{\lambda x.\ m \Leftarrow \Gamma \vdash A \to B}$$

Fn-1 requires a new judgment, $A \leqslant B^{s}$, where A, B, s are all inputs; this doesn't seem difficult to define, but it's Yet Another Subtyping Judgment. Fn-2 avoids this, but is much less easy to explain.

However, it's not clear to me I need to generalize the λ rule. The reason I thought I did was to justify something like the following:

$$\frac{\dfrac{\vdots}{\Gamma, x : A^{\mathsf{iso}} \vdash m : B}}{\Gamma \vdash \lambda x.\ m : \Box A \to B}$$

But this *could* check as follows:

$$\frac{\dfrac{\vdots}{m \Leftarrow \Gamma, x : \Box A^{s} \vdash B} \qquad \mathsf{id} \leqslant s}{\lambda x.\ m \Leftarrow \Gamma \vdash \Box A \to B}$$

So the crucial question is: can we always substitute $x : \Box A^{\mathsf{id}}$ for $x : A^{\mathsf{iso}}$? It would suffice to prove the subtyping and substitution principles given in § 8.1. Can we prove these with our original, subtyping-less λ rule?

6 Pattern matching

$$\begin{array}{rccl} \text{patterns} & p, q & ::= & x \mid (p, q) \mid \mathsf{in}_i\ p \\ \text{checking terms} & m, n & ::= & \textbf{case}\ e\ \textbf{of}\ \overline{p_i \to m_i}^i \\ \text{toneless contexts} & \phi, \psi & ::= & \varepsilon \mid \phi, x : A \\ \text{judgments} & J & ::= & A \equiv B \circledast C \\ & & \mid & p : A \vdash \phi \\ & & \mid & p \to m \Leftarrow A^s ; \Gamma \vdash C \end{array}$$

The types in toneless contexts ϕ aren't annotated with tones. TODO: Explain why and when we use toneless contexts. Explain $\phi^{\vec{t}}$ notation for a context split into its types and its tones.

6.1 Distributing modes

The pattern (x, y) matches values of type $A \times B$. But how shall we match values of type $\Box(A \times B)$? We might add a pattern, **box** p, matching values of type $\Box A$. Then **box** (x, y) would match values of type $\Box(A \times B)$. But do we *need* this **box** annotation? Theorem 3 says $\Box(A \times B)$ and $\Box A \times \Box B$ mean the same thing. Why not let (x, y) match $\Box(A \times B)$ *directly*, yielding $x : \Box A$ and $y : \Box B$?

To this end, we'll need a judgment $A \equiv B \circledast C$ for distributing modes over a cartesian operator $\circledast$ (either $\times$ or $+$). Here A is an input and B, C are outputs.

$$\frac{}{A \circledast B \equiv A \circledast B} \qquad \frac{A \equiv B \circledast C}{s\,A \equiv s\,B \circledast s\,C}$$

6.2 Typing patterns

The judgment $p : A \vdash \phi$ corresponds to a PREORD-morphism $A \to 1 + \phi$. It means that the pattern p, when it matches a value of type A, produces values for ϕ's variables.

$$\frac{}{x : A \vdash x : A} \qquad \frac{A \equiv A_1 + A_2 \qquad p : A_i \vdash \phi}{\mathsf{in}_i\ p : A \vdash \phi}$$

$$\frac{A \equiv A_1 \times A_2 \qquad (\forall i)\ \ p_i : A_i \vdash \phi_i \qquad \phi_1, \phi_2 \text{ disjoint}}{(p_1, p_2) : A \vdash \phi_1, \phi_2}$$

6.3 Typing case-expressions

Typing **case** as a single rule is complicated:

$$\frac{e \Rightarrow \Gamma \vdash A \qquad (\forall i)\ \ p_i : A \vdash \phi_i \qquad (\forall i)\ \ m_i \Leftarrow \Gamma_i, \phi_i^{\vec{s}_i} \vdash C}{\textbf{case}\ e\ \textbf{of}\ \overline{p_i \to m_i}^{i} \Leftarrow \Gamma^{\bigwedge_i \bigwedge \vec{s}_i} \wedge \bigwedge_i \Gamma_i \vdash A}$$

We can split this up using a helper judgment, $p \to m \Leftarrow A^s; \Gamma \vdash C$, corresponding to a morphism $A^s \times \Gamma \to 1 + C$. This says that the arm $p \to m$ matches a scrutinee of type A that it uses at tone s, along with variables in Γ, to produce (if it matches) a result of type C. Then we have:

$$\frac{p : A \vdash \phi \qquad m \Leftarrow \Gamma, \phi^{\vec{t}} \vdash C}{p \to m \Leftarrow A^{\bigwedge \vec{t}}; \Gamma \vdash C} \qquad \frac{e \Rightarrow \Gamma \vdash A \qquad (\forall i)\ \ p_i \to m_i \Leftarrow A^{s_i}; \Gamma_i \vdash C}{\textbf{case}\ e\ \textbf{of}\ \overline{p_i \to m_i}^{i} \Leftarrow \Gamma^{\bigwedge_i s_i} \wedge \bigwedge_i \Gamma_i \vdash C}$$

6.4 Why do we need both stripping and distribution?

Can we also use modal distribution instead of modal stripping in our typing rules for expressions? Not quite. We can rewrite the tuple-projection rule:

$$\frac{e \Rightarrow \Gamma \vdash A \qquad A \equiv A_1 \times A_2}{\pi_i\ e \Rightarrow \Gamma \vdash A_i}$$

However, we cannot rewrite function application (shown below) this way; in general, $\square(A \to B) \not\equiv \square A \to \square B$. So it seems there is no choice but to use subtyping.

$$\frac{e \Rightarrow \Gamma_1 \vdash A \qquad A^u \prec B \to C \qquad m \Leftarrow \Gamma_2 \vdash B}{e\ m \Rightarrow \Gamma_1^u \wedge \Gamma_2 \vdash C}$$

TODO: explain why using modal stripping for pattern matching doesn't work, with the $(x, (y, z))$ versus $A \times \square(B \times C)$ example.

TODO: explain why using modal stripping rather than distribution for the tuple projection rule is fine, because of the adjunction between path and iso.

6.5 Case analysis with guarded arms

checking expressions	m	$::=$	$\textbf{case}\ e\ \textbf{of}\ \overline{p_i\ \textbf{if}\ m_i \to n_i}^{i}$
judgments	J	$::=$	$p\ \textbf{if}\ m \to n : \Gamma \times A^s \rightharpoonup C$

$$\frac{p : A \vdash \phi \qquad m \Leftarrow \Gamma_1,\ \phi^{\vec{s}} \vdash \Box 2 \qquad n \Leftarrow \Gamma_2,\ \phi^{\vec{t}} \vdash C}{p \ \mathbf{if}\ m \to n : \Gamma \times A^{\bigwedge \vec{s} \wedge \bigwedge \vec{t}} \rightharpoonup C}$$

$$\frac{e \Rightarrow \Gamma \vdash A \qquad (\forall i)\ \ p_i \ \mathbf{if}\ m_i \to n_i : \Gamma_i \times A^{s_i} \rightharpoonup C}{\mathbf{case}\ e\ \mathbf{of}\ \overline{p_i \ \mathbf{if}\ m_i \to n_i}^{\,i} \Leftarrow \Gamma^{\bigwedge_i s_i} \wedge \bigwedge_i \Gamma_i \vdash C}$$

6.6 Patterns with embedded guards

$$\begin{array}{rcll} \text{patterns} & p, q & ::= & p \ \mathbf{if}\ m \\ \text{judgments} & J & ::= & p : A^s \times \Gamma \rightharpoonup \phi \end{array}$$

Now that patterns can contain expressions, our pattern typing judgment takes an input context Γ, becoming $p : A^s \times \Gamma \rightharpoonup \phi$. This corresponds to a morphism $A^s \times \Gamma \to 1 + \phi$.

$$\frac{x \notin \phi}{x : A^{\mathsf{id}} \times \Gamma \rightharpoonup x : A} \qquad \frac{A \equiv A_1 + A_2 \qquad p : A_i^s \times \Gamma \rightharpoonup \phi}{\mathsf{in}_i\ p : A^s \times \Gamma \rightharpoonup \phi}$$

$$\frac{A \equiv B \times C \qquad p : B^s \times \Gamma_1 \rightharpoonup \phi \qquad q : C^t \times \Gamma_2,\ \phi^{\vec{u}} \rightharpoonup \psi}{(p, q) : A^{s \cdot (\mathsf{id} \wedge \bigwedge \vec{u}) \wedge t} \times {\Gamma_1}^{\mathsf{id} \wedge \bigwedge \vec{u}} \wedge \Gamma_2 \rightharpoonup \phi,\ \psi}$$

$$\frac{p : A^s \times \Gamma_1 \rightharpoonup \phi \qquad m \Leftarrow \Gamma_2,\ \phi^{\vec{t}} \vdash \Box 2}{p \ \mathbf{if}\ m : A^{s \cdot (\mathsf{id} \wedge \bigwedge \vec{t} \cdot \mathsf{path})} \times \Gamma_1 \wedge (\Gamma_1^{\bigwedge \vec{t}} \wedge \Gamma_2)^{\mathsf{path}} \rightharpoonup \phi}$$

These rules are getting so complicated I don't trust them without a proof.

Now we update the rules for $p \to m \Leftarrow A^s; \Gamma \vdash C$ to pass through Γ to the pattern:

$$\frac{p : \Gamma_1^A \times s \rightharpoonup \phi \qquad m \Leftarrow \Gamma_2, \phi^{\vec{t}} \vdash C}{p \to m \Leftarrow A^s; {\Gamma_1}^{\bigwedge \vec{t}} \wedge \Gamma_2 \vdash C}$$

7 Declarative rules

This is where I'm stashing important inference rules, stated in a way that makes them obviously valid, but leaves non-obvious how to algorithmically check them.

7.1 Subtyping and type equivalence

Type equivalence $A^s \equiv B^t$ is a synonym for $A^s \leqslant B^t \wedge B^t \leqslant A^s$.

TODO: declarative subtyping rules for sum types

$$\frac{s \leqslant t}{A^s \leqslant A^t} \qquad \frac{A^s \leqslant B^t \qquad B^t \leqslant C^u}{A^s \leqslant C^u} \qquad \frac{A^s \leqslant B^t}{A^{s \cdot u} \leqslant B^{s \cdot u}} \qquad \frac{A_1^s \leqslant B_1^t \qquad A_2^s \leqslant B_2^t}{(A_1 \times A_2)^s \leqslant (B_1 \times B_2)^t}$$

$$\frac{A_2^{\mathsf{id}} \leqslant A_1^{\mathsf{id}} \qquad B_1^{\mathsf{id}} \leqslant B_2^{\mathsf{id}}}{(A_1 \to B_1)^{\mathsf{id}} \leqslant (A_2 \to B_2)^{\mathsf{id}}} \qquad \Box A^{\mathsf{id}} \equiv A^{\mathsf{iso}} \qquad (\mathsf{op}\ A)^{\mathsf{id}} \equiv A^{\mathsf{op}}$$

$$(A \to B)^{\mathsf{op}} \equiv (\mathsf{op}\ A \to \mathsf{op}\ B)^{\mathsf{id}} \qquad (A \to \Box B)^{\mathsf{id}} \equiv (A \to \Box B)^{\mathsf{iso}}$$

$$(A \to B)^{\mathsf{iso}} \leqslant (\Box A \to \Box B)^{\mathsf{id}}$$

TODO: check we can derive the algorithmic function subtyping rules. $(A \to B)^{\mathsf{iso}} \leqslant (\Box A \to \Box B)^{\mathsf{id}}$ handles one of the cases; $(A \to B)^{\mathsf{op}} \equiv (\mathsf{op}\ A \to \mathsf{op}\ B)^{\mathsf{id}}$ handles another; what about the last one?

8 Metatheory

8.1 Weakening, subtyping, and substitution

We wish to prove admissible the following rules:

$$\textsc{Tone Weakening}\ \frac{m \Leftarrow \Gamma \vdash A}{m \Leftarrow \Gamma \wedge \Gamma' \vdash A} \qquad \textsc{Subtyping Left}\ \frac{m \Leftarrow \Gamma, x : A^s \vdash C \qquad A^s \leqslant B^t}{m \Leftarrow \Gamma, x : B^t \vdash C} \qquad \textsc{Subtyping Right}\ \frac{m \Leftarrow \Gamma \vdash A \qquad A^s \leqslant B^{\mathsf{id}}}{m \Leftarrow \Gamma^s \vdash B}$$

$$\textsc{Substitution}\ \frac{e \Rightarrow \Gamma_1 \vdash A \qquad A^s \leqslant B^t \qquad m \Leftarrow \Gamma_2, x : B^t \vdash C}{m[e/x] \Leftarrow \Gamma_1^s \wedge \Gamma_2 \vdash C}$$

TODO: Doesn't TONE WEAKENING follow from SUBTYPING LEFT?