refinement calculus
notes

Peter G. Hancock

1 Oct 2003

Contents

1 syntax

The usual setting for Back and von Wright's refinement calculus is higher order classical logic, with quantification over predicate variables, and a complement operator. Here instead we define certain of these constructions in predicative type theory in which quantification over predicates is not allowed in propositions.

The basic judgements in which we are interested are (primarily) $U \subseteq V$ and (secondarily, the dual judgement) $U \between V$ which says that U and V are compatible. It is necessary to have a separate judgement form such as $U \between V$ in the absence of the complement operator. Here U and V are predicate expressions that may contain variables of various sorts: states, state-predicates, and relations. The variables are implicitly universally quantified. If the predicate expressions depend on a single predicate variable X, we have a (pointwise) relation between predicate transformers. If the predicate expressions depend on a state variable s, we have the inclusion relation between relations. In combination with predicate transformers Φ and iterated form, the basic relations give rise to relations of interest in client-server programming such as the following

$$\begin{array}{l} U \lhd_\Phi V = U \subseteq \Phi^*(V) \\ U \ltimes_\Phi V = U \between \Phi^\infty(V) \end{array}$$

As for predicativity, all reasoning should be essentially 'point-free', or algebraic. That is to say, proofs should be algebraic manipulations in which free subset variables never officially appear.

In the general case we may have beside predicates also relations with various arities, and beside unary transformers of unary predicates also transformers with arity of the form $\langle n_1, .., n_k \rangle \to n$

Two base types (for the two kinds of variables): S (states – s, s', s_1, ...)) and P (predicates – P). We have one binary relation $s \epsilon P$, meaning that state s satisfies predicate P. This gives rise to 3 kinds of statement:

$$\begin{array}{ll} s \epsilon U & \\ s \epsilon Q(s') & Q \text{ a relation} \\ s \epsilon F(U) & F \text{ a predicate transformer} \end{array}$$

The following higher types:

$$\begin{array}{llll} f, g, \dots & : S \to S & \textit{state transformer} & f = g \\ R, Q, \dots & : S \to P & \textit{state relations} & R \subseteq Q \\ F, G, \dots & : P \to P & \textit{predicate transformers} & F \sqsubseteq G \end{array}$$

The first kind of comparison is equality between state expressions that may have free state-variables. Might want apartness.

The second kind of comparison is equivalence and implication between statements that may have free state-variables. Might want overlap.

The third kind of comparison is equivalence and implication between statements that have free occurences of both state-variables and predicate-variables. Again, might want overlap.

2 predicates and families

Predicates over a set form a distributive lattice: closed under sup (empty, binary, set-indexed) and inf (empty, binary, set-indexed). By distributivity I mean that binary sups distribute over binary infs and vice-versa. We also have implication, forms of relative complement etc. Binary infs distribute over arbitrary sups. We also have singleton predicates.

Families on the other-hand are merely a sup-lattice with singletons. .

An obscure point to be explained: the link between families and predicates (or transition structures and relations) involves not just equality (which gets us one way), but also existential quantification over states, which gets us a family from a predicate, in which the function is first projection. There is a question of size here: S may be 'large' compared to the universe of sets we are working in.

3 grammar

$$
\begin{array}{lll}
\textit{states} & s, s' & ::= \; f(s) \\[1em]
\textit{state transformers} & f, g & ::= \; id \mid f \cdot g \\[1em]
\textit{families} & \alpha, \beta & ::= \; \{\, s \,\} \mid \sqcup_i \alpha_i \mid \phi(s) \\[1em]
\textit{predicates} & U, V & ::= \; \alpha \mid \bigcup_i U_i \mid \bigcap_i U_i \mid \Phi(U) \mid R(s) \\[1em]
\textit{transition structures} & \phi, \psi & ::= \; \mathbf{graph}\, f \\
 & & \qquad \mid U \rightharpoonup \phi \\
 & & \qquad \mid \sqcup_i \phi_i \\
 & & \qquad \mid \phi \,;\, \psi \mid \mathrm{id} \\
 & & \qquad \mid \phi^* \mid \phi^+ \mid \phi^? \\[1em]
\textit{relations} & Q, R & ::= \; \phi \\
 & & \qquad \mid U \rightharpoonup Q \\
 & & \qquad \mid \bigcup_i Q_i \mid \bigcap_i Q_i \\
 & & \qquad \mid R \,;\, Q \mid \mathrm{id} \mid Q/R \\
 & & \qquad \mid \phi \,;\, Q \mid Q/\phi \\
 & & \qquad \mid \Phi \cdot Q \\
 & & \qquad \mid Q^* \mid Q^+ \mid Q^? \\
 & & \qquad \mid Q^\sim \\[1em]
\textit{interaction structures} & \Phi, \Psi & ::= \; \mid \langle \phi \rangle \mid [\, \phi \,] \\
 & & \qquad \mid \sqcup_i \Phi_i \mid \sqcap_i \Phi_i \\
 & & \qquad \mid \Phi \,;\, \Psi \mid \mathrm{id} \\
 & & \qquad \mid \mathbf{assign}\, f \\
 & & \qquad \mid \Phi^* \mid \Phi^+ \mid \Phi^? \\
 & & \qquad \mid \Phi^\infty \\
 & & \qquad \mid \Phi^\perp \\[1em]
\textit{predicate transformers} & F, G & ::= \; \Phi \\
 & & \qquad \mid \langle R \rangle \mid [\, R \,] \\
 & & \qquad \mid \sqcup_i F_i \mid \sqcap_i F_i \\
 & & \qquad \mid F \,;\, G \mid \mathrm{id} \\
 & & \qquad \mid \mathbf{assign}\, f \\
 & & \qquad \mid F^* \mid F^+ \mid F^? \\
 & & \qquad \mid F^\infty
\end{array}
$$

4 types and definitions

4.1 State transformers

1. composition of state transformers $\dfrac{f : A \to B \qquad g : B \to C}{g \cdot f : A \to C}$

   $(f.g)(s) \ \triangleq\ f(g(s))$.

2. identity of state transformers: $\quad \mathrm{id} : A \to A$

   $\mathrm{id}(a) \ \triangleq\ a$.

3. base state transformers

   Often, the state space is a product $S = \prod_{v:V} S_v$, where V is a set of variable-names (with decidable equality) and S_v is a factor of the state space corresponding to variable v: the type of v. Then a state is a function which assigns to a variable-name v a value of the type S_v appropriate to v. This can also be thought of as a record.

   If e is an expression built up from variable-names using function constants, we define by recursion on its build up the value $|e|_s$ of e in the start state s – in the obvious way (see below). We then define an update function $v := e$ on state records:

   $$\begin{array}{l}(v := e) : S \to S \\ (v := e)(s) \ \triangleq\ (\lambda\, v' : V)\ \mathbf{if}\ v' = v\ \mathbf{then}\ |e|_s\ \mathbf{else}\ s(v)\end{array}$$

   Thus an *assignment* statement can be interpreted as a state transformer.

   We may obviously extend the definition to *simultaneous* assignment, where we have a finite vector $\vec{v}$ of (distinct) variables, and a vector $\vec{e}$ of expressions of the same length.

   Example : $x, y := y, x$ – atomically swap contents of variables x and y.

   Tangentially, if we are interested in (statically) *typed* variables, then one can represent the type system in the form of an interaction structure – a set of sorts or types, and for each sort σ a family of families of sorts:

   $$\{\, \{\, \sigma[c/r] \mid r : R(\sigma, c) \,\} \mid c : C(\sigma) \,\}$$

   Here C describes the constructors that can be used to form an expression of sort σ, and for each such constructor c an element r of $R(\sigma, c)$ selects the location of an immediate subexpression. The sort of the subexpression must equal $\sigma[c/r]$.

   An expression of sort σ is now a well-founded structure, in which there are 'exit points' named after state variables. (If there are none, the expression is closed.)

   We can define the value of an expression under an assignment of appropriately typed values to variable names, by wellfounded recursion on the structure of those proofs.

4.2 Category of relations and simulations

By passing from predicates to relations, and from families to transition structures (allowing a state-parameter), we add to the lattice operations a 'sequencing' monoid with unit the graph of the identity function.

Survey: Closure properties. Other notes.

- graphs of functions are transition structures.
- transition structures are closed under restriction by predicates (guarding).
- transition structures are closed under sup.
- transition structures are closed under composition, eq etc.
- Each transition structure determines two relation transformers (ϕ ;) and ($/\phi$). The definition of these does not use equality. These are closely related to the predicate transformers $\langle\phi\rangle$ and $[\,\phi\,]$.
- in the homogeneous case, we have the usual closure operators (reflexive, transitive, etc).
- transition functions are *not* closed under converse, nor intersection, nor division. (That is, without specific use of the equality relation.)

Transition structures are 'regular' – form a sup-lattice (with set-indexed sups), and have an iteration star operation. (We get back the infs with interaction structures, and two notions of iteration.)

Transition structures form a Kleene (regular) algebra in the following sense. It has an associative and commutative binary sup $\cup$ with unit 0 (the empty transition structure); associative binary sequencing with unit id, distributing over sup. 0 is absorbing. An iteration operator star, satisfying ϕ^* is a solution of the equations $\mathrm{id} \cup (\phi \ ; \ x) \subseteq x$ and $\mathrm{id} \cup (x \ ; \ \phi) \subseteq x$; and if $\phi \ ; \ x \subseteq x$ or $x \ ; \ \phi \subseteq x$, then $\phi^* \ ; \ x \subseteq x$ or $x \ ; \ \phi^* \subseteq x$ respectively. Transition structures also include 'tests' somewhat in Kozen's sense, except that they needn't form a Boolean algebra (but a Heyting algebra).

4.2.1 Relations and transition structures

1. *st*'s as *rel*'s $\dfrac{f : A \to B}{\mathbf{graph}\, f : A \to \mathbb{P}B}$

$$b \mathrel{\epsilon} (\mathbf{graph}\, f)a \ \triangleq \ b = f(a) \tag{1}$$

2. *st*'s as *ts*'s $\dfrac{f : A \to B}{\mathbf{graph}\, f : A \to \mathbb{F}B}$

$$T(a) \ \triangleq \ N_1 \quad ; \quad s[\_] \ \triangleq \ f(s) \tag{2}$$

3. predicates as *rel*'s $\dfrac{U : \mathbb{P}A}{\mathbf{test}\, U : A \to \mathbb{P}A}$

$$b \,\epsilon\, (\mathbf{test}\, U)a \;\triangleq\; a \,\epsilon\, U \wedge b = a \tag{3}$$

4. predicates as *ts*'s $\dfrac{U : \mathbb{P}A}{\mathbf{test}\, U : A \to \mathbb{F}A}$

$$T \;\triangleq\; U \quad ; \quad s[\_] \;\triangleq\; s \tag{4}$$

5. domain restriction of *rel*'s $\dfrac{Q : A \to \mathbb{P}B \qquad U : \mathbb{P}A}{U \rightharpoonup Q : A \to \mathbb{P}B}$

$$b \,\epsilon\, (U \rightharpoonup Q)a \;\triangleq\; a \,\epsilon\, U \wedge b \,\epsilon\, Q(a) \tag{5}$$

Note: some redundancy. $U \rightharpoonup R = \mathbf{test}\, U \,;\, R$.

6. domain restriction of *ts*'s $\dfrac{\phi : A \to \mathbb{F}B \qquad U : \mathbb{P}A}{U \rightharpoonup \phi : A \to \mathbb{F}B}$

$$T \;\triangleq\; U \cap T_\phi \quad ; \quad s[\langle \_, t\rangle] \;\triangleq\; s[t]_\phi \tag{6}$$

7. mapping *rel*'s by a st $\dfrac{R : A \to \mathbb{P}B \qquad f : C \to B}{\mathbb{P}(f) \cdot R : A \to \mathbb{P}C}$

$$c \,\epsilon\, (\mathbb{P}f \cdot R)a \;\triangleq\; f(c) \,\epsilon\, R(a) \tag{7}$$

8. mapping *ts*'s by a st $\dfrac{\phi : A \to \mathbb{F}B \qquad f : B \to C}{\mathbb{F}(f) \cdot \phi : A \to \mathbb{F}C}$

$$T(a) \;\triangleq\; T_\phi(a) \quad ; \quad a[t] \;\triangleq\; f(a[t]_\phi) \tag{8}$$

Note: redundant. $\mathbb{F}f \cdot \phi = \phi \,;\, \mathbf{graph}\, f$.

9. union, *and* intersection of *rel*'s

$$\frac{Q_i : A \to \mathbb{P}B}{(\cup_i Q_i), (\cap_i Q_i) : A \to \mathbb{P}B}$$

$$(\cap_i Q_i)(a) \;\triangleq\; \cap_i (Q_i(a)) \tag{9}$$
$$(\cup_i Q_i)(a) \;\triangleq\; \cup_i (Q_i(a)) \tag{10}$$

Note, no counterpart to intersection on *ts*'s.

10. union of ts's $\dfrac{\phi_i : A \to \mathbb{F}B}{(\sqcup_i \phi_i) : A \to \mathbb{F}B}$

$$T(a) \;\triangleq\; \sum_i T_{\phi_i}(a) \quad ; \quad a[\langle i, t\rangle] \;\triangleq\; a[t]_i \tag{11}$$

Note, no counterpart of intersection.

11. argument swapping

$$\frac{Q : A \to \mathbb{P}B}{Q^{\sim} : B \to \mathbb{P}A}$$

$$a \mathrel{\epsilon} Q^{\sim}(b) \;\triangleq\; b \mathrel{\epsilon} Q(a) \tag{12}$$

Notes

- no counterpart operation on ts's
- prime example of a relation transformer. Contravariant functor on the category of sets and relations. An involution. Called converse, inverse, reverse, interchange, twist, flip, swap, and so on.
- determines a notion of $(^{\sim})$-duality for relation transformers. The $(^{\sim})$-dual of a relation transformer Φ is $(^{\sim}) \cdot \Phi \cdot (^{\sim})$. For example, the division $(Q\backslash)$ is $(^{\sim})$-dual to $(/(Q^{\sim}))$.

12. *rel*'s closed under sequential composition

$$\frac{R : A \to \mathbb{P}B \qquad Q : B \to \mathbb{P}C}{(R\,;Q) : A \to \mathbb{P}C}$$

$$c \,\epsilon\, (R\,;Q)(a) \;\triangleq\; R(a) \between Q^{\smile}(c) \tag{13}$$

13. identity *rel*: $\quad \mathrm{id} = \mathbf{graph}\,\mathrm{id} : A \to \mathbb{P}A$

$$a \,\epsilon\, \mathrm{id}(a') \;\triangleq\; a = a' \tag{14}$$

14. *ts*'s closed under sequential composition

$$\frac{\phi : A \to \mathbb{F}B \qquad \psi : B \to \mathbb{F}C}{(\phi\,;\psi) : A \to \mathbb{F}C}$$

$$T(a) \;\triangleq\; \sum_{t_1 : T_\phi(a)} T_\psi(a[t_1]_\phi) \quad ; \quad a[\langle t_1, t_2\rangle] \;\triangleq\; (a[t_1]_\phi)[t_2]_\psi \tag{15}$$

15. identity *ts*: $\quad \mathrm{id} = \mathbf{graph}\,\mathrm{id} : A \to \mathbb{F}A$

$$T(\_) \;\triangleq\; N_1 \quad ; \quad a[\_] \;\triangleq\; a \tag{16}$$

16. closures of *rel*'s $\dfrac{R : A \to \mathbb{P}A}{R^?, R^+, R^* : A \to \mathbb{P}A}$

$$\begin{array}{rcl} R^* & \triangleq & \cap\{\, T : A \to \mathbb{P}A \mid (\mathrm{id} \cup (R \mathbin{;} T)) \subseteq T \,\} \\ & = & \cup\{\, (R \mathbin{;})^n \mathrm{id} \mid n = 0, 1, \ldots \} \end{array} \tag{17}$$

$R^? \triangleq R \cup \mathrm{id}, \qquad R^+ \triangleq R \mathbin{;} R^*.$

17. closures of *ts*'s $\dfrac{\phi : A \to \mathbb{F}A}{\phi^*, \phi^?, \phi^+ : A \to \mathbb{F}A}$

ϕ^*:

$$\begin{array}{rcl} T & \triangleq & (\mu\, X : A \to \mathrm{Set}) \\ & & (\forall\, a : A) \\ & & \quad \{\, \mathrm{nil} \,\} \\ & & \ \cup\{\, \mathrm{cons}(t_0, t') \mid t_0 : T_\phi(a), t' : X(a[t_0]_\phi) \,\} \\ & & \quad \subseteq X(a) \end{array} \tag{18}$$

$$\begin{array}{rcl} a[\mathrm{nil}] & \triangleq & a \\ a[\mathrm{cons}(t_0, t')] & \triangleq & (a[t_0]_\phi)[t'] \end{array}$$

$\phi^? \triangleq \phi \sqcup \mathrm{id}, \qquad \phi^+ \triangleq \phi \mathbin{;} \phi^*.$

18. *rel*'s closed under post-division

$$\frac{Q : A \to \mathbb{P}B \qquad R : C \to \mathbb{P}B}{(Q/R) : A \to \mathbb{P}C}$$

$$c \mathrel{\epsilon} (Q/R)(a) \;\triangleq\; R(c) \subseteq Q(a) \tag{19}$$

19. pre-composition of *ts*'s to *rel*'s

$$\frac{\phi : A \to \mathbb{F}B \qquad Q : B \to \mathbb{P}C}{(\phi \,;\, Q) : A \to \mathbb{P}C}$$

$$c \mathrel{\epsilon} (\phi \,;\, Q)(a) \;\triangleq\; \phi(a) \between Q^{\sim}(c) \tag{20}$$

$(\phi \,;\, Q)(a) = \cup\{\, Q(a[t]_\phi) \mid t : T_\phi(a) \,\}$
Note that this lets us lift a *ts* ϕ to the *rel* $(\phi \,;\, \mathrm{id})$.

20. post-division of *rel*'s by *ts*'s

$$\frac{Q : A \to \mathbb{P}B \qquad \psi : C \to \mathbb{F}B}{(Q/\psi) : A \to \mathbb{P}C}$$

$$c \mathrel{\epsilon} (Q/\psi)(a) \;\triangleq\; \psi(c) \subseteq Q(a) \tag{21}$$

Note that this gives us a 'reciprocal' lift of *ts* ϕ to *rel* (id/ϕ). The reciprocal of a relation is completely different from its converse. For what relations are the converse and reciprocal the same?

21. *pt*'s as operations on *rel*'s $\dfrac{\Phi : \mathbb{P}A \to \mathbb{P}B \qquad Q : C \to \mathbb{P}A}{(\Phi \cdot Q) : C \to \mathbb{P}B}$

$$b \mathrel{\epsilon} (\Phi \cdot R)(c) \;\triangleq\; b \mathrel{\epsilon} \Phi(R(c)) \tag{22}$$

4.2.2 Morphisms between relations and transition structures

Consider the Kleisli category for the monadic functor $\mathbb{F}_\_$. The arrows in this category (diagrams of shape $A \to \mathbb{F}B$, which we picture as vertical arrows) are called transition structures. A morphism between two such arrows $\phi : A \to \mathbb{F}B$ and $\psi : C \to \mathbb{F}D$ is a pair of horizontal relations $Q_1 : A \to \mathbb{P}C$ and $Q_2 : B \to \mathbb{P}D$ which form a "sub-commuting" square:

$$Q_1{}^{\sim} \,;\, \phi \subseteq \psi ; Q_2{}^{\sim}$$

This is equivalent to

$$Q_1{}^{\sim} \subseteq (\psi; Q_2{}^{\sim})/\phi$$

or even

$$\begin{array}{rl} Q_1 \subseteq & (\sim) \cdot (/\phi) \cdot (\psi;\,) \cdot (\sim)\ Q_2 \\ = & (\sim) \cdot ([\,\phi\,]\cdot) \cdot ((\sim) \cdot (\langle\psi\rangle\cdot) \cdot (\sim)) \cdot (\sim)\ Q_2 \\ = & (\sim) \cdot ([\,\phi\,]\cdot) \cdot (\sim) \cdot (\langle\psi\rangle\cdot)\ Q_2 \end{array}$$

One composes morphisms between arrows "horizontally", as relations. (Pairs of relations are compared pointwise, for inclusion and equality.)

If instead of arrows we restrict ourselves to cycles (*i.e.* homogeneous transition structures, *i.e.* endomorphisms in the Kleisli category, the appropriate notion of morphism is a simulation.

4.3 Category of predicate transformers and simulations

Mumble mumble.

4.3.1 Predicate transformers (and interaction structures)

1. Relational update, lifting *rel*'s to *pt*'s.

$$\frac{R : A \to \mathbb{P}B}{\langle R\rangle, [\,R\,] : \mathbb{P}B \to \mathbb{P}A}$$

$$a \mathrel{\epsilon} \langle R\rangle(U) \quad \triangleq \quad R(a) \between U \tag{23}$$
$$a \mathrel{\epsilon} [\,R\,](U) \quad \triangleq \quad R(a) \subseteq U \tag{24}$$

2. angelic and demonic lifting of a *ts* to an *is*

$$\frac{\phi : A \to \mathbb{F}B}{\langle \phi\rangle, [\,\phi\,] : A \to \mathbb{F}(\mathbb{F}B)}$$

angel $\langle\phi\rangle$

$$\begin{aligned} C &\triangleq T_\phi \\ R(\_,\_) &\triangleq N_1 \\ a[t/\_] &\triangleq a[t]_\phi \end{aligned} \tag{25}$$

demon $[\,\phi\,]$

$$\begin{aligned} C(\_) &\triangleq N_1 \\ R(a,\_) &\triangleq T_\phi(a) \\ a[\_/t] &\triangleq a[t]_\phi \end{aligned} \tag{26}$$

3. *is*'s as *pt*'s $\dfrac{\Phi : A \to \mathbb{F}(\mathbb{F}B)}{\Phi : \mathbb{P}B \to \mathbb{P}A}$

$$a \mathrel{\epsilon} \Phi(U) \quad \triangleq \quad (\exists\, c : C_\Phi(a))\ \{\, a[c/r]_\Phi \mid r : R_\Phi(a,c)\,\} \subseteq U \tag{27}$$

Note: $\Phi : A \to \mathbb{F}(\mathbb{F}B)$ can always be written $\langle\phi\rangle$; $[\,\psi\,]$ for certain $\phi : A \to A'$, $\psi : A' \to \mathbb{F}B$, as follows. Write Φ_{pre},Φ_{post} for ϕ,ψ .

$$\begin{aligned} A' &= \{\, \mathrm{pending}(a,c) \mid a : A, c : C_\Phi(a)\,\} \\ \phi(a) &= \{\, \mathrm{pending}(a,c) \mid c : C_\Phi(a)\,\} \\ \psi(\mathrm{pending}(a,c)) &= \{\, a[c/r]_\Phi \mid r : R_\Phi(a,c)\,\} \end{aligned}$$

4. infima, suprema of pt's $\dfrac{F_i : \mathbb{P}A \to \mathbb{P}B}{(\sqcap_i F_i), (\sqcup_i F_i) : \mathbb{P}A \to \mathbb{P}B}$

$$
\begin{aligned}
(\sqcap_i F_i)(U) &\triangleq \cap_i (F_i(U)) && (28)\\
(\sqcup_i F_i)(U) &\triangleq \cup_i (F_i(U)) && (29)
\end{aligned}
$$

5. infima, suprema of is's $\dfrac{\Phi_i : A \to \mathbb{F}(\mathbb{F}B)}{(\sqcap_i \Phi_i), (\sqcup_i \Phi_i) : A \to \mathbb{F}(\mathbb{F}B)}$

   angelic $\sqcup$:

$$
\begin{aligned}
C &\triangleq \cup_i C_i && (30)\\
R(s, \langle i, c\rangle) &\triangleq R_i(s, c)\\
s[\langle i, c\rangle / r] &\triangleq s[c/r]_i
\end{aligned}
$$

   demonic $\sqcap$:

$$
\begin{aligned}
C &\triangleq \cap_i C_i && (31)\\
R(s, f) &\triangleq (\exists\, i)\ R_i(s, f(i))\\
s[f/\langle i, r\rangle] &\triangleq s[f(i)/r]_i
\end{aligned}
$$

6. sequential composition of pt's

$$\frac{F : \mathbb{P}A \to \mathbb{P}B \qquad G : \mathbb{P}B \to \mathbb{P}C}{(F\ ;\ G) : \mathbb{P}A \to \mathbb{P}C}$$

$$(F\ ;\ G)(U) \triangleq F(G(U)) \tag{32}$$

7. sequential composition of is's

$$\frac{\Phi : A \to \mathbb{F}(\mathbb{F}B) \qquad \Psi : B \to \mathbb{F}(\mathbb{F}C)}{(\Phi\ ;\ \Psi) : A \to \mathbb{F}(\mathbb{F}C)}$$

$$
\begin{aligned}
C &\triangleq \Phi(C_\Psi) && (33)\\
&= \{\, a : A \mid (\exists\, c : C_\Phi(s))(\forall\, r : R_\Phi(a, c))\ C_\Psi(a[c/r]_\Phi)\,\}\\
R(a, \langle c, f\rangle) &\triangleq (\exists\, r : R_\Phi(a, c))\ R_\Psi(a[c/r]_\Phi, f(r))\\
a[\langle c, f\rangle / \langle r_0, r'\rangle] &\triangleq (s[c/r_0]_\Phi)[f(r_0)/r']_\Psi
\end{aligned}
$$

8. identity *pt*: $\mathrm{id} = \langle \mathrm{id} \rangle = [\,\mathrm{id}\,] : \mathbb{P}A \to \mathbb{P}A$

$$b \mathrel{\epsilon} \mathrm{id}(U) \quad \triangleq \quad b \mathrel{\epsilon} U \tag{34}$$

9. identity *is*: $\mathrm{id} = \langle \mathrm{id} \rangle = [\,\mathrm{id}\,] : A \to \mathbb{F}(\mathbb{F}A)$

$$\begin{aligned} C(\_) &\triangleq N_1 \\ R(\_,\_) &\triangleq N_1 \\ a[\_/\_] &\triangleq a \end{aligned} \tag{35}$$

10. dual of an *is*

$$\frac{\Phi : A \to \mathbb{F}(\mathbb{F}B)}{\Phi^{\perp} : A \to \mathbb{F}(\mathbb{F}B)}$$

$$\begin{aligned} C(a) &\triangleq (\forall\, c : C_\Phi(a)) R_\Phi(a, c) \\ R(a, \_) &\triangleq C_\Phi(a) \\ a[f/c] &\triangleq a[c/f(c)]_\Phi \end{aligned} \tag{36}$$

Note: this doesn't have very good properties constructively. One can however calculate duals formally, by changing suprema to infima, angels by demons, *etc.*.

11. closures of pt's:

$$\frac{F : \mathbb{P}A \to \mathbb{P}A}{F^?, F^+, F^*, F^\infty : \mathbb{P}A \to \mathbb{P}A}$$

$$\begin{aligned}
F^* &\triangleq \cap\{\, T : \mathbb{P}A \to \mathbb{P}A \,|\, (\mathrm{id} \cup (F \,;\, T)) \subseteq T \,\} && (37)\\
&= U \mapsto \{\, a : A \,|\, (\forall\, V : \mathbb{P}A)\ ((U \cup F(V)) \subseteq V) \to V(a) \,\} \\
&= \cup_\alpha (F\,;)^\alpha(\mathrm{id}) \\
F^\infty &\triangleq \cup\{\, T : \mathbb{P}A \to \mathbb{P}A \,|\, T \subseteq (\mathrm{id} \cap (F^\perp \,;\, T)) \,\} && (38)\\
&= U \mapsto \{\, a : A \,|\, (\exists\, V : \mathbb{P}A)\ V \subseteq (U \cap (F^\perp(V)) \wedge V(a) \,\} \\
&= \cap_n \{\, F_n \,|\, F_0 = \mathrm{id}; F_{n+1} = F_n \cap (F^\perp \,;\, F_n) \,\}
\end{aligned}$$

12. closures of is's $\dfrac{\Phi : A \to \mathbb{F}(\mathbb{F}A)}{\Phi^*, \Phi^?, \Phi^+, \Phi^\infty : A \to \mathbb{F}(\mathbb{F}A)}$

Φ^*

$$\begin{aligned}
C \triangleq\ & (\mu\ X : A \to \mathrm{Set}) && (39)\\
& (\forall\ a : A) \\
& \quad \{\, \mathrm{exit} \,\} \\
& \cup\{\, \mathrm{call}(c, f) \,|\, c : C_\Phi(a), f : (\forall\ r : R(a, c))\ X(a[c/r]_\Phi) \,\} \\
& \quad \subseteq X(a)
\end{aligned}$$

$$\begin{aligned}
R(a, \mathrm{exit}) &\triangleq \{\, \mathrm{nil} \,\} \\
R(a, \mathrm{call}(c, f)) &\triangleq \{\, \mathrm{cons}(r_0, r') \,|\, r_0 : R_\Phi(a, c), r' : R(a[c/r_0], f(r_0)) \,\} \\
a[\mathrm{exit}/\mathrm{nil}] &\triangleq a \\
a[\mathrm{call}(c, f)/\mathrm{cons}(r_0, r')] &\triangleq (a[c/r_0]_\Phi)[f(r_0)/r']
\end{aligned}$$

$\Phi^? \triangleq \Phi \sqcup \mathrm{id}$, $\qquad \Phi^+ \triangleq \Phi \,;\, \Phi^*$.

Φ^∞

$$\begin{aligned}
C \triangleq\ & (\nu\ X : A \to \mathrm{Set}) && (40)\\
& (\forall\ a : A) \\
& \ X(a) \subseteq \quad \{\, \mathrm{srv}(f, g) \,|\ \ f : (\forall\ c : C_\Phi(a)) R_\Phi(a, c), \\
& \qquad\qquad\qquad\qquad\quad g : (\forall\ c : C_\Phi(a)) X(a[c/f(c)]_\Phi) \,\}
\end{aligned}$$

$$\begin{aligned}
R(a, \mathrm{srv}(f, g)) &\triangleq \quad \{\, \mathrm{nil} \,\} \\
&\quad\ \cup\{\, \mathrm{cons}(c_0, c') \,|\, c_0 : C_\Phi(a), c' : R(a[c_0/f(c_0)], g(c_0)) \,\} \\
a[\mathrm{srv}(f, g)/\mathrm{nil}] &\triangleq a \\
a[\mathrm{srv}(f, g)/\mathrm{cons}(c_0, c')] &\triangleq (a[c_0/f(c_0)]_\Phi)[g(c_0)/c']
\end{aligned}$$

13. functional assignment as a *pt*

$$\frac{f : A \to B}{\mathbf{assign}\, f : \mathbb{P}B \to \mathbb{P}A}$$

$$s \mathrel{\epsilon} (\mathbf{assign}\, f)(P) \;\triangleq\; f(s) \mathrel{\epsilon} P$$

Note: $\mathbf{assign}\, f = (\cdot f) = f^{-1} = \mathbb{P}f$.

Note: $\mathbf{assign}\, f \mathbin{;} \mathbf{assign}\, g = \mathbf{assign}\, g \cdot f$.

Redundant.

$$\mathbf{assign}\, f = \langle \mathbf{graph}\, f \rangle = [\, \mathbf{graph}\, f\,]$$

14. functional assignment as an *is*

$$\frac{f : A \to B}{\mathbf{assign}\, f : A \to \mathbb{F}(\mathbb{F}B)}$$

$$\begin{aligned} C(\_) &\triangleq N_1 \\ R(\_,\_) &\triangleq N_1 \\ a[\_/\_] &\triangleq f(a) \end{aligned}$$

4.3.2 Morphisms between predicate transformers

In analogy with transition structures, we could define a morphism between interaction structures to be a pair of predicate transformers that satisfy the appropriate sub-commutativity property. However, in this case we insist that the predicate transformer is *angelic*, that is commutes with all disjunctions, that is is determined by its value at singletons, that is is an angelic relational update.

Give definition.

Give it for interaction structures. Note really between an is and a pt.

5 laws

Laws for (inferring inclusion and equality between) relations.

1. Predicate transformers determine relation transformers. However, the effect of certain predicate transformers can sometimes be expressed merely from sequential composition, and division. The following are equations between relation transformers.

   $$\begin{array}{rclcrcl} (Q\,;) & = & ({}^{\sim})\cdot(\langle Q\rangle\cdot)\cdot({}^{\sim}) & \qquad & (/Q) & = & ([\,Q\,]\cdot) \\ (;\,Q) & = & (\langle Q^{\sim}\rangle\cdot) & & (\backslash Q) & = & ({}^{\sim})\cdot([\,Q^{\sim}\,]\cdot)\cdot({}^{\sim}) \end{array} \tag{41}$$

   To a certain extent, we are interested in representing relations with transition structures – we may represent a relation as a transition structure, or as the converse of a transition structure, or as the reciprocal of a transition structure (and so on and on).

2. I might have introduced binary operators for relative complement and implication. Then the adjunctions for relations are:

   $$(R_1\,;\,R_2)\subseteq Q \iff R_1\subseteq(Q/R_2) \tag{42}$$
   $$(R_1-R_2)\subseteq Q \iff R_1\subseteq(R_2\cup Q) \tag{43}$$
   $$(R_1\cap R_2)\subseteq Q \iff R_1\subseteq(R_2\Rightarrow Q) \tag{44}$$

   Some laws (need to check):

   $$Q-(R_1\cup R_2) \;=\; (Q-R_1)\cap(Q-R_2)=((Q-R_1)-R_2) \tag{45}$$
   $$Q-(R_1\cap R_2) \;\supseteq\; (Q-R_1)\cup(Q-R_2) \tag{46}$$
   $$(R_1\cup R_2)\Rightarrow Q \;=\; (R_1\Rightarrow Q)\cap(R_2\Rightarrow Q) \tag{47}$$
   $$(R_1\cap R_2)\Rightarrow Q \;=\; R_1\Rightarrow(R_2\Rightarrow Q) \tag{48}$$

   Non-binary sups and infs?

3. laws for sups and infs.

   $$(\cup R_i)\subseteq Q \iff (\forall\, i)\ R_i\subseteq Q \tag{49}$$
   $$Q\subseteq(\cap R_i) \iff (\forall\, i)\ Q\subseteq R_i \tag{50}$$

   For relations, sequential composition commutes with union (on both sides).

4. The laws for relational converse ${}^{\sim}$. This commutes with infima and suprema (and closures), is monotone, and

   $$Q^{\sim\sim}=Q \tag{51}$$
   $$(\mathbf{graph}\,f)^{\sim}\,;\,(\mathbf{graph}\,f)\subseteq \mathrm{id};\quad \mathrm{id}\subseteq(\mathbf{graph}\,f)\,;\,(\mathbf{graph}\,f)^{\sim} \tag{52}$$
   $$\mathrm{id}^{\sim}=\mathrm{id} \tag{53}$$
   $$(Q\,;\,R)^{\sim}=R^{\sim}\,;\,Q^{\sim} \tag{54}$$

5. What are right laws for domain restriction?

$$\textbf{test}\,U = U \rightharpoonup \text{id} \tag{55}$$
$$U \rightharpoonup R = \textbf{test}\,U \; ; \, R \tag{56}$$
$$\textbf{test}\,U \; ; \textbf{test}\,V = \textbf{test}\,(U \cap V) \tag{57}$$
$$R_1 \subseteq R_2 \Rightarrow (U \rightharpoonup R_1) \subseteq (U \rightharpoonup R_2) \tag{58}$$
$$U \subseteq V \Rightarrow (U \rightharpoonup R) \subseteq (V \rightharpoonup R) \tag{59}$$

6. Dedekind's law (modular law). This is what governs sequential composition, converse and intersection.

$$(Q \; ; R) \cap S \subseteq Q \; ; (R \cap (Q^{\smile} \; ; S)) \tag{60}$$

This law is used to prove that if Q is a partial function ('deterministic' relations) then $Q \; ; (R_1 \cap R_2) = (Q \; ; R_1) \cap (Q \; ; R_2)$. In other words $(Q \; ;)$ is conjunctive (it is always disjunctive).

7. Intersection of relations.

$$\begin{aligned}
\textbf{test}\,U \cap \textbf{test}\,V &= \textbf{test}\,U \cap V && (61)\\
(Q_1 \cap Q_2) \cap Q_3 &= Q_1 \cap (Q_2 \cap Q_3) && (62)\\
Q \cap \text{void} &= \text{void} && (63)\\
Q \cap \text{chaos} &= Q && (64)\\
Q_1 \cap Q_2 &= Q_2 \cap Q_1 && (65)\\
Q \cap Q &= Q && (66)\\
Q \cap (Q \cup R) &= Q && (67)\\
Q \cap (\cup_i R_i) &= \cup_i (Q \cap R_i) && (68)\\
Q \cap (\cap_i R_i) &= \cap_i (Q \cap R_i) && (69)\\
Q \cap (R_1 \; ; R_2) &\subseteq R_1 \; ; (R_2 \cap (R_1{}^{\smile} \; ; Q)) && (70)\\
Q \cap R^{\smile} &= (Q^{\smile} \cap R)^{\smile} && (71)
\end{aligned}$$

8. Union of relations.

$$\begin{aligned}
\textbf{test}\,U \cup \textbf{test}\,V &= \textbf{test}\,U \cup V && (72)\\
(Q_1 \cup Q_2) \cup Q_3 &= Q_1 \cup (Q_2 \cup Q_3) && (73)\\
Q \cup \text{void} &= Q && (74)\\
Q \cup \text{chaos} &= \text{chaos} && (75)\\
Q_1 \cap Q_2 &= Q_2 \cap Q_1 && (76)\\
Q \cup Q &= Q && (77)\\
Q \cup (Q \cap R) &= Q && (78)\\
Q \cup (\cup_i R_i) &= \cup_i (Q \cup R_i) && (79)\\
Q \cup (\cap_i R_i) &\subseteq \cap_i (Q \cup R_i) && (80)\\
Q \cup (R_1 \cap R_2) &= (Q \cup R_1) \cap (Q \cup R_2) && (81)\\
Q \cup (R_1 \; ; R_2) &\;\; \ldots && (82)
\end{aligned}$$

9. Sequential composition of relations.

$$\begin{array}{rcll}
(Q_1 \mathbin{;} Q_2) \mathbin{;} Q_3 &=& Q_1 \mathbin{;} (Q_2 \mathbin{;} Q_3) & (83)\\
\mathrm{id} \mathbin{;} Q &=& Q & (84)\\
Q \mathbin{;} \mathrm{id} &=& Q & (85)\\
Q \mathbin{;} (\cap_i R_i) &\subseteq& \cap_i (Q \mathbin{;} R_i) & (86)\\
(\cap_i Q_i) \mathbin{;} R &\subseteq& \cap_i (Q_i \mathbin{;} R) & (87)\\
Q \mathbin{;} (\cup_i R_i) &=& \cup_i (Q \mathbin{;} R_i) & (88)\\
(\cup_i Q_i) \mathbin{;} R &=& \cup_i (Q_i \mathbin{;} R) & (89)\\
Q \mathbin{;} \mathrm{void} &=& \mathrm{void} & (90)\\
\mathrm{void} \mathbin{;} Q &=& \mathrm{void} & (91)\\
\mathbf{test}\, U \mathbin{;} \mathbf{test}\, V &=& \mathbf{test}\, (U \cap V) & (92)\\
\mathrm{id} &=& \mathbf{test}\, \mathrm{chaos} & (93)\\
\mathbf{graph}\, f \mathbin{;} \mathbf{graph}\, g &=& \mathbf{graph}\, (g \cdot f) & (94)\\
\mathrm{id} &=& \mathbf{graph}\, \mathrm{id} & (95)
\end{array}$$

Laws for (inferring inclusion and equality between) predicate transformers.

1. For predicate transformers, sequential composition commutes with sups and infs in its left-hand argument.(Unlike the case of relations, where we don't commute with inf's, but commute with sups on both sides.)

$$\begin{array}{rcll}
(F_1 \mathbin{;} F_2) \mathbin{;} F_3 &=& F_1 \mathbin{;} (F_2 \mathbin{;} F_3) & (96)\\
\mathrm{id} \mathbin{;} F &=& F &\\
F \mathbin{;} \mathrm{id} &=& F &\\
(\cup_i F_i) \mathbin{;} G &=& \cup_i (F_i \mathbin{;} G) &\\
(\cap_i F_i) \mathbin{;} G &=& \cap_i (F_i \mathbin{;} G) &\\
F \mathbin{;} (\cup_i G_i) &\supseteq& \cup_i (F \mathbin{;} G_i) &\\
F \mathbin{;} (\cap_i G_i) &\subseteq& \cap_i (F \mathbin{;} G_i) &
\end{array}$$

The last 2 semi-equations are just by monotonicity.

They can be strengthened to equality for certain F.

$$\begin{array}{rcl}
\langle \phi \rangle \mathbin{;} (\cup_i G_i) &=& \cup_i (\langle \phi \rangle \mathbin{;} G_i)\\
[\,\phi\,] \mathbin{;} (\cap_i G_i) &=& \cap_i ([\,\phi\,] \mathbin{;} G_i)
\end{array}$$

2. Special cases of sequential composition. Should be associative with unit id.

$$\begin{array}{rcll}
\langle Q \rangle \mathbin{;} \langle R \rangle &=& \langle Q \mathbin{;} R \rangle & (97)\\
[\,Q\,] \mathbin{;} [\,R\,] &=& [\,Q \mathbin{;} R\,] &\\
\mathrm{id} &=& \mathbf{assign}\, \mathrm{id} &\\
\mathbf{assign}\, f &=& \langle \mathbf{graph}\, f \rangle = [\,\mathbf{graph}\, f\,] &
\end{array}$$

3. Are these true?

$$\begin{array}{rcll}
F; \langle Q \rangle &=& \langle F \cdot Q \rangle & (98)\\
F; [\,Q\,] &=& [\,F \cdot Q\,] & (99)
\end{array}$$

An interaction structure is something of the form $\langle\phi\rangle\ ;\ [\,\psi\,]$.