1 Semantics

1.1 Grammar and definitions

Value expressions

$$\begin{aligned} V &\rightarrow C^{\gamma} \text{ Typed contants} \\ &\mid () \text{ Unit constructor} \\ &\mid \lambda x : \tau . M \text{ term lambda expressions} \\ &\mid x \text{ Variables} \\ &\mid \Lambda \phi . V \text{ Effect lambda expressions} \\ &\mid V \epsilon \text{ Effect specialisation} \end{aligned} \tag{1}$$

Computation expressions

$$\begin{aligned} M &\rightarrow V\ V \text{Application} \\ &\mid do\ x\ \leftarrow\ M\ in\ M \text{ Sequencing} \\ &\mid return\ V \text{ Unit computation} \\ &\mid Cons^{\tau_1 \rightarrow T_{\xi}\tau_2}(V) \text{ Computation Constructors} \\ &\mid Op^{T_{\xi}\tau_1 \rightarrow T_{\xi}\tau_2}(M) \text{ Computation Combinators} \end{aligned} \tag{2}$$

Value types

$$\begin{aligned} \tau &\rightarrow \gamma \text{ Ground Types} \\ &\mid \tau \rightarrow \Theta \text{ Functions are kleisli arrows} \\ &\mid \forall \phi . \tau \text{ Effect quantification} \end{aligned} \tag{3}$$

Computation (Monad) types

$$\Theta \rightarrow T_{\epsilon}\tau \text{ Monad type constructor} \tag{4}$$

Effects

$$\epsilon \rightarrow \phi \mid \xi .. \text{ Effect variables or ground effects}$$

1.2 typing

Value typing rules

$$(\text{Const})\frac{\Gamma\ OK}{\Gamma \vdash C^{\gamma} \colon \gamma}$$

$$(\text{Unit})\frac{\Gamma\ OK}{\Gamma \vdash () \colon unit}$$

$$(\text{Eff-Gen})\frac{\Gamma \vdash V \colon A}{\Gamma \vdash \Lambda \phi . V \colon \forall \phi . A} \phi \notin fev(\Gamma)$$

$$(\text{Eff-Gen})\frac{\Gamma \vdash V \colon \forall \phi . A}{\Gamma \vdash V \epsilon \colon A} \epsilon \phi\,]$$

The rules for types of Single queries are similar:

$$\text{(Find)}\frac{\Sigma(f) \downarrow (A)}{\Gamma \vdash \mathit{Find}(f)\colon A}$$

$$\text{(From)}\frac{\Gamma \vdash P\colon (A, B) \qquad \Gamma \vdash S\colon A}{\Gamma \vdash \mathit{From}(S, P)\colon B}$$

$$\text{(AndS)}\frac{\Gamma \vdash S\colon A \qquad \Gamma \vdash S'\colon A}{\Gamma \vdash \mathit{AndS}(S, S')\colon A}$$

$$\text{(OrS)}\frac{\Gamma \vdash S\colon A \qquad \Gamma \vdash S'\colon A}{\Gamma \vdash \mathit{OrS}(S, S')\colon A}$$

1.3 Operational Semantics

Now we shall define a set of rules for determining if a pair of objects is a valid result of a query. We're interested in forming a relation $a \lhd p$ to mean"a is a valid result of query Q". This is dependent on the current view $v : \mathit{View}_\Sigma$, and the type of the expression. Hence we define $(a, b) \lhd_{(A,B),v} P$ for pair queries P and $a \lhd_{A,v} S$ for single queries S.

$$\text{(Rel)}\frac{(a,b) \in v(r)}{(a,b) \lhd_{(A,B),v} \mathit{Rel}(r)}$$

$$\text{(Rev)}\frac{(b,a) \in v(r)}{(a,b) \lhd_{(A,B),v} \mathit{RevRel}(r)}$$

$$\text{(Id)}\frac{a \in v(A)}{(a,a) \lhd_{(A,A),v} \mathit{Id}_A}$$

$$\text{(Distinct)}\frac{(a,b) \lhd_{(A,B),v} P \qquad a \neq b}{(a,b) \lhd_{(A,B),v} \mathit{Distinct}(P)}$$

$$\text{(And)}\frac{(a,b) \lhd_{(A,B),v} P \qquad (a,b) \lhd_{(A,B),v} Q}{(a,b) \lhd_{(A,B),v} \mathit{And}(P,Q)}$$

$$\text{(Or1)}\frac{(a,b) \lhd_{(A,B),v} P}{(a,b) \lhd_{(A,B),v} \mathit{Or}(P,Q)}$$

$$\text{(Or2)}\frac{(a,b) \lhd_{(A,B),v} Q}{(a,b) \lhd_{(A,B),v} \mathit{Or}(P,Q)}$$

$$\text{(Chain)}\frac{(a,b) \lhd_{(A,B),v} P \qquad (b,c) \lhd_{(B,C),v} Q}{(a,c) \lhd_{(A,C),v} \mathit{Chain}(P,Q)}$$

$$\text{(AndLeft)}\frac{(a,b) \lhd_{(A,B),v} P \qquad a \lhd_{A,v} S}{(a,b) \lhd_{(A,B),v} \mathit{AndLeft}(P,S)}$$

$$\text{(AndRight)}\frac{(a,b) \lhd_{(A,B),v} P \qquad b \lhd_{B,v} S}{(a,b) \lhd_{(A,B),v} \mathit{AndRight}(P,S)}$$

$$\text{(Exactly˙0)}\frac{(a,b) \lhd_{(A,A),v} \mathit{Id}_A}{(a,b) \lhd_{(A,A),v} \mathit{Exactly}(0,P)}$$

$$\text{(Exactly˙n+1)}\frac{(a,b) \lhd_{(A,A),v} P \qquad (b,c) \lhd_{(A,A),v} \mathit{Exactly}(n,P)}{(a,c) \lhd_{(A,A),v} \mathit{Exactly}(n+1,P)}$$

$$\text{(Upto˙0)}\frac{(a,b) \lhd_{(A,A),v} \mathit{Id}_A}{(a,b) \lhd_{(A,A),v} \mathit{Upto}(0,P)}$$

$$\text{(Upto˙n)}\frac{(a,b) \lhd_{(A,A),v} \mathit{Upto}(n,P)}{(a,b) \lhd_{(A,A),v} \mathit{Upto}(n+1,P)}$$

$$\text{(Upto˙n+1)}\frac{(a,b) \lhd_{(A,A),v} P \qquad (b,c) \lhd_{(A,A),v} \mathit{Upto}(n,P)}{(a,c) \lhd_{(A,A),v} \mathit{Upto}(n+1,P)}$$

$$\text{(fix1)}\frac{(a,b) \lhd_{(A,A),v} \mathit{Id}_A}{(a,b) \lhd_{(A,A),v} \mathit{FixedPoint}(P)}$$

$$\text{(fix2)}\frac{(a,b) \lhd_{(A,A),v} P \qquad (b,c) \lhd_{(A,A),v} \mathit{FixedPoint}(P)}{(a,b) \lhd_{(A,A),v} \mathit{FixedPoint}(P)}$$

And the FindSingle rules

$$\text{(Find)}\frac{a \in v(A) \qquad f(a) \downarrow True}{a \lhd_{A,v} Find(f)}$$

$$\text{(From)}\frac{a \lhd_{A,v} S \qquad (a,b) \lhd_{)(A,B),v} P}{b \lhd_{A,v} From(S,P)}$$

$$\text{(AndS)}\frac{a \lhd_{A,v} S \qquad a \lhd_{A,v} S'}{a \lhd_{A,v} And(S,S')}$$

$$\text{(OrS1)}\frac{a \lhd_{A,v} S}{a \lhd_{A,v} Or(S,S')}$$

$$\text{(OrS1)}\frac{a \lhd_{A,v} S'}{a \lhd_{A,v} Or(S,S')}$$

1.4 Denotational Semantics

The operational semantics clearly demonstrate membership of a query, but don't give a means to efficiently generate the results of query. To this end, we introduce denotations $[\![P]\!]$ and $[\![S]\!]$ such that

$$\Sigma \vdash P\colon (A,B) \Rightarrow [\![P]\!]\colon View_\Sigma \to \wp(A \times B)$$

and

$$\Sigma \vdash S\colon A \Rightarrow [\![S]\!]\colon View_\Sigma \to \wp(A)$$

Such denotations should be compositional and syntax directed, whilst still corresponding to the operational semantics.

$$[\![Rel(r)]\!](v) = v(r)$$

$$[\![RevRel(r)]\!](v) = swap(v(r))$$

$$[\![Id_A]\!](v) = dup(v(A))$$

$$[\![Chain(P, Q)]\!](v) = join([\![P]\!](v), [\![Q]\!](v))$$

$$[\![And(P, Q)]\!](v) = [\![P]\!](v) \cap [\![Q]\!](v)$$

$$[\![Or(P, Q)]\!](v) = [\![P]\!](v) \cup [\![Q]\!](v)$$

$$[\![AndLeft(P, S)]\!](v) = filterLeft([\![P]\!](v), [\![S]\!](v))$$

$$[\![AndRight(P, S)]\!](v) = filterRight([\![P]\!](v), [\![S]\!](v))$$

$$[\![Distinct(P)]\!](v) = distinct([\![P]\!](v))$$

$$[\![Exactly(n, P)]\!](v) = (\lambda pairs.join([\![P]\!](v), pairs))^n [\![Id_A]\!](v)$$

$$[\![Upto(n, P)]\!](v) = (\lambda pairs.join([\![P]\!](v), pairs) \cup pairs)^n [\![Id_A]\!](v)$$

$$[\![FixedPoint(P)]\!](v) = fix(\lambda pairs.join([\![P]\!](v), pairs) \cup pairs) \text{ in the domain } closure(A, v)$$

And similarly with single queries

$$[\![Find(f)]\!](v) = \{a \in v(A) \mid f(a) \downarrow True\} \text{ for } \Sigma(f) = A$$

$$[\![From(S, P)]\!](v) = \{b \mid (a, b) \in [\![P]\!](v) \wedge a \in [\![S]\!](v)\}$$

$$[\![AndS(S, S')]\!](v) = [\![S]\!](v) \cap [\![S']\!](v)$$

$$[\![OrS(S, S')]\!](v) = [\![S]\!](v) \cup [\![S']\!](v)$$

with the following definitions:

$$swap(s) = \{(b, a) \mid (a, b) \in s\}$$
$$dup(s) = \{(a, a) \mid a \in s\}$$
$$join(p, q) = \{(a, c) \mid \exists b.(a, b) \in p \wedge (b, c) \in q\}$$
$$distinct(s) = \{(a, b) \in s \mid a \neq b\}$$
$$filterLeft(p, s) = \{(a, b) \in p \mid a \in s\}$$
$$filterRight(p, s) = \{(a, b) \in p \mid b \in s\}$$

1.5 The domain closure(A, v)

For the subsequent proofs it is necessary to define the scott domain $closure(A, v)$ for some object type A and $View_\Sigma\ v$. This domain is the set of subsets x such that $[\![Id_A]\!](v) \subseteq x \subseteq A \times A$ with bottom element $\bot = [\![Id_A]\!](v)$ and partial order $x \sqsubseteq y \Leftrightarrow x \subseteq y$. From this point on, I shall use $x \subseteq y$ to mean $y \sqsubseteq x$.

Theorem: ***closure*(A, v) is a domain** Firstly, by definition,

$$\forall x. x \in closure(A, v) \Rightarrow x \supseteq [\![Id_A]\!](v)$$

hence $[\![Id_A]\!](v)$ is the bottom element.
Secondly for any chain $x_1 \subseteq x_2 \subseteq x_3 \subseteq ...,\ x_i \in closure(A, v)$, there exists a value $\bigsqcup_n x_n \in closure(A, v)$ such that $\forall i. \bigsqcup_n x_n \supseteq x_i$ and $\forall y.(\forall i. x_i \subseteq y) \Rightarrow y \supseteq \bigsqcup_n x_n$
proof: Take $\bigsqcup_n x_n = \bigcup_n x_n$. This is in $closure(A, v)$, since both $\bigcup_n x_n \supseteq [\![Id_A]\!](v)$ due to $\forall i. x_i \supseteq [\![Id_A]\!](v)$ by definition and $\bigcup_n x_n \subseteq A \times A$, by

$$\forall a.(a \in \bigcup_n x_n \wedge \neg(a \in A \times A)) \Rightarrow (\exists i. a \in x_i \wedge \neg a \in A \times A) \Rightarrow (\exists i. \neg x_i \subseteq A \times A)$$

yielding a contradiction if $\bigcup_n x_n \subseteq A \times A$ does not hold.
We know $\forall i. \bigcup_n x_n \supseteq x_i$ by definition, so it is an upper bound.
To prove it is a least upper bound, consider y such that $(\forall i.) y \supseteq x_i$

$$\begin{gathered} \forall a.(\exists i. a \in x_i) \Rightarrow a \in y \\ \forall a. \bigvee_n (a \in x_n) \Rightarrow a \in y \\ \forall a.(a \in \bigcup_n x_n) \Rightarrow a \in y \\ \therefore \bigcup_n x_n \subseteq y \\ \square \end{gathered} \tag{5}$$

1.6 Correspondence of operational and denotational semantics

In order to use the denotational semantics to construct an interpreter or compiler we need to prove they are equivalent to the operational semantics. Namely:

For any pair query P, schema Σ and $View_\Sigma\ v$:

$$\Gamma \vdash P : (A, B) \Rightarrow (a, b \lhd_{(A,B),v} P \Leftrightarrow (a, b) \in [\![P]\!](v))$$

And for any single query S, schema Σ and $View_\Sigma\ v$:

$$\Gamma \vdash S : A \Rightarrow (a \lhd_{A,v} S \Leftrightarrow a \in [\![S]\!](v))$$

In order to prove these two propositions, we define two induction hypotheses

$$\Phi(\Sigma, P, A, B) \Leftrightarrow (\Gamma \vdash P : (A, B) \Rightarrow \forall v \in View_\Sigma, (a, b) \in A \times B.((a, b) \lhd_{(A,B),v} P) \Leftrightarrow (a, b) \in [\![P]\!](v))))$$

$$\Psi(\Sigma, S, A) \Leftrightarrow (\Gamma \vdash S\colon A \Rightarrow \forall v \in View_\Sigma, a \in A.(a \lhd_{A,v} S) \Leftrightarrow (a \in [\![S]\!](v)))$$

Now we shall induct over the structures of P and S starting with the SingleQuery cases

Case $S = AndS(S', S'')$

$$\begin{aligned} a \in [\![S]\!](v) &\Leftrightarrow a \in ([\![S']\!](v) \cap [\![S'']\!](v)) \\ &\Leftrightarrow (a \in [\![S']\!](v) \wedge a \in [\![S'']\!](v)) \\ &\Leftrightarrow (a \lhd_{A,v} S' \wedge a \lhd_{A,v} S'') \text{ by } \Psi(\Sigma, S', A),\ \Psi(\Sigma, S'', A) \\ &\Leftrightarrow (a \lhd_{A,v} AndS(S', S'') \text{ by inversion of (AndS)} \end{aligned} \tag{6}$$

Case $S = OrS(S', S'')$

$$\begin{aligned} a \in [\![S]\!](v) &\Leftrightarrow a \in ([\![S']\!](v) \cup [\![S'']\!](v)) \\ &\Leftrightarrow (a \in [\![S']\!](v) \vee a \in [\![S'']\!](v)) \\ &\Leftrightarrow (a \lhd_{A,v} S' \vee a \lhd_{A,v} S'') \text{ by } \Psi(\Sigma, S', A),\ \Psi(\Sigma, S'', A) \\ &\Leftrightarrow (a \lhd_{A,v} OrS(S', S'') \text{ by inversion of (OrS)} \end{aligned} \tag{7}$$

Case $S = From(S', P)$

$$\begin{aligned} b \in [\![S]\!](v) &\Leftrightarrow \exists a \in A. \quad (a \in [\![S']\!](v) \wedge (a, b) \in [\![[\![P]\!](v)]\!](v)) \\ &\Leftrightarrow \exists a \in A. \quad (a \lhd_{A,v} S' \wedge (a, b) \lhd_{(A,B),v} P) \text{ by } \Psi(\Sigma, S', A),\ \Phi(\Sigma, P, A, B) \\ &\Leftrightarrow b \lhd_{B,v} From(S', P) \qquad \text{by inversion of (OrS)} \end{aligned} \tag{8}$$

Case $S = Find(f)$

$$\begin{aligned} a \in [\![S]\!](v) &\Leftrightarrow a \in v(A) \wedge f(a) \downarrow True \qquad \text{by inversion of the type rule (Find)} \\ &\Leftrightarrow a \lhd_{A,v} Find(f) \text{ by definition} \end{aligned} \tag{9}$$

Now, looking at the FindPair queries

Case $P = Rel(r)$

$$\begin{aligned} (a, b) \in [\![P]\!](v) &\Leftrightarrow (a, b) \in v(r) \\ &\Leftrightarrow (a, b) \lhd_{(A,B),v} Rel(r) \text{ by definition} \end{aligned} \tag{10}$$

Case $P = RevRel(r)$

$$\begin{aligned} (a, b) \in [\![P]\!](v) &\Leftrightarrow (b, a) \in v(r) \\ &\Leftrightarrow (a, b) \lhd_{(A,B),v} RevRel(r) \text{ by definition} \end{aligned} \tag{11}$$

Case $P = Id_A$

$$\begin{aligned} (a, b) \in [\![P]\!](v) &\Leftrightarrow a \in v(A) \wedge a = b \\ &\Leftrightarrow (a, b) \lhd_{(A,B),v} Id_A \text{ by definition} \end{aligned} \tag{12}$$

Case $P = Chain(P', Q)$

We have by inversion of the (Chain) type rule $\Gamma \vdash P\colon (A, C) \Leftrightarrow \exists B. \quad \Gamma \vdash P'\colon (A, B) \wedge \Gamma \vdash Q\colon (B, C)$

$$\begin{aligned}
(a,c) \in [\![P]\!](v) &\Leftrightarrow (a,c) \in join([\![P']\!](v), [\![Q]\!](v)) \\
&\Leftrightarrow \exists b \in B. \quad (a,b) \in [\![P']\!](v) \wedge (b,c) \in [\![Q]\!](v) \\
&\Leftrightarrow \exists b \in B. \quad (a,b) \triangleleft_{(A,B),v} P' \wedge (b,c) \triangleleft_{(B,C),v} Q \quad \text{ by } \Phi(\Sigma, P', A, B),\ \Phi(\Sigma, Q, B, C) \\
&\Leftrightarrow (a,c) \triangleleft_{(A,C),v} Chain(P', Q) \text{ by definition}
\end{aligned} \tag{13}$$

Case $P = And(P', Q)$

$$\begin{aligned}
(a,b) \in [\![P]\!](v) &\Leftrightarrow (a,b) \in ([\![P']\!](v) \cap [\![Q]\!](v)) \\
&\Leftrightarrow (a,b) \in [\![P']\!](v) \wedge (a,b) \in [\![Q]\!](v) \\
&\Leftrightarrow (a,b) \triangleleft_{(A,B),v} P' \wedge (a,b) \triangleleft_{(A,B),v} Q \quad \text{ by } \Phi(\Sigma, P', A, B),\ \Phi(\Sigma, Q, A, B) \\
&\Leftrightarrow (a,b) \triangleleft_{(A,B),v} And(P', Q) \text{ by inversion of (And)}
\end{aligned} \tag{14}$$

Case $P = Or(P', Q)$

$$\begin{aligned}
(a,b) \in [\![P]\!](v) &\Leftrightarrow (a,b) \in ([\![P']\!](v) \cup [\![Q]\!](v)) \\
&\Leftrightarrow (a,b) \in [\![P']\!](v) \vee (a,b) \in [\![Q]\!](v) \\
&\Leftrightarrow (a,b) \triangleleft_{(A,B),v} P' \vee (a,b) \triangleleft_{(A,B),v} Q \quad \text{ by } \Phi(\Sigma, P', A, B),\ \Phi(\Sigma, Q, A, B) \\
&\Leftrightarrow (a,b) \triangleleft_{(A,B),v} Or(P', Q) \text{ by inversion of (Or1), (Or2)}
\end{aligned} \tag{15}$$

Case $P = AndLeft(P', S)$

$$\begin{aligned}
(a,b) \in [\![P]\!](v) &\Leftrightarrow (a,b) \in [\![P']\!](v) \wedge a \in [\![S]\!](v) \\
&\Leftrightarrow (a,b) \triangleleft_{(A,B),v} P' \wedge a \triangleleft_{A,v} S \quad \text{ by } \Phi(\Sigma, P', A, B),\ \Psi(\Sigma, S, A) \\
&\Leftrightarrow (a,b) \triangleleft_{(A,B),v} AndLeft(P', S) \text{ by inversion of (AndLeft)}
\end{aligned} \tag{16}$$

Case $P = AndRight(P', S)$

$$\begin{aligned}
(a,b) \in [\![P]\!](v) &\Leftrightarrow (a,b) \in [\![P']\!](v) \wedge b \in [\![S]\!](v) \\
&\Leftrightarrow (a,b) \triangleleft_{(A,B),v} P' \wedge b \triangleleft_{B,v} S \quad \text{ by } \Phi(\Sigma, P', A, B),\ \Psi(\Sigma, S, B) \\
&\Leftrightarrow (a,b) \triangleleft_{(A,B),v} AndRight(P', S) \text{ by inversion of (AndRight)}
\end{aligned} \tag{17}$$

Case $P = Distinct(P')$

$$\begin{aligned}
(a,b) \in [\![P]\!](v) &\Leftrightarrow (a,b) \in [\![P']\!](v) \wedge a \neq b \\
&\Leftrightarrow (a,b) \triangleleft_{(A,B),v} P' \wedge a \neq b \quad \text{ by } \Phi(\Sigma, P', A, B) \\
&\Leftrightarrow (a,b) \triangleleft_{(A,B),v} Distinct(P') \text{ by inversion of (AndRight)}
\end{aligned} \tag{18}$$

Case $P = \mathit{Exactly}(n, P')$

We have by inversion of the (Exactly) type rule $\Gamma \vdash P\colon (A, A) \wedge \Gamma \vdash P'\colon (A, A)$

$$\text{let } f = (\lambda pairs.join([\![P']\!](v), pairs)) \tag{19}$$

then

$$(a, b) \in [\![P]\!](v) \Leftrightarrow (a, b) \in f^n[\![Id_A]\!](v) \tag{20}$$

hence it suffices to prove

$$(a, b) \in f^n[\![Id_A]\!](v) \Leftrightarrow (a, b) \lhd_{(A,A),v} \mathit{Exactly}(n, P') \tag{21}$$

Case $\mathit{Exactly}(0, P')$:

$$f^0[\![Id_A]\!](v) = [\![Id_A]\!](v)$$

so by $\Phi(\Sigma, (a, b), A, A), Id_A)$

$$\begin{aligned} (a, b) \in f^0[\![Id_A]\!](v) &\Leftrightarrow (a, b) \in [\![Id_A]\!](v) \\ &\Leftrightarrow (a, b) \lhd_{(A,A),v} Id_A \\ &\Leftrightarrow (a, b) \lhd_{(A,A),v} \mathit{Exactly}(0, P') \end{aligned} \tag{22}$$

Case $\mathit{Exactly}(n + 1, P')$**, assuming** $\Phi(\Sigma, \mathit{Exactly}(n, P'), A, A)$**:**

$$f^{n+1}[\![Id_A]\!](v) = f(f^n([\![Id_A]\!](v)))$$

so by $\Phi(\Sigma, \mathit{Exactly}(n, P'), A, A)$

$$\begin{aligned} (a, b) \in f^{n+1}[\![Id_A]\!](v) &\Leftrightarrow \exists a'.(a, a') \in [\![P']\!](v) \wedge (a', b) \in f^n([\![Id_A]\!](v)) \quad \text{by definition of } join \text{ and } f \\ &\Leftrightarrow (a, a') \lhd_{(A,A),v} P \wedge (a', b) \lhd_{(A,A),v} \mathit{Exactly}(n, p) \\ &\Leftrightarrow (a, b) \lhd_{(A,A),v} \mathit{Exactly}(n + 1, P') \text{ by (Exactly n+1)} \end{aligned} \tag{23}$$

Case $P = \mathit{Upto}(n, P')$

We have by inversion of the (Upto) type rule $\Gamma \vdash P\colon (A, A) \wedge \Gamma \vdash P'\colon (A, A)$

$$\text{let } f = (\lambda pairs.join([\![P']\!](v), pairs) \cup pairs) \tag{24}$$

then

$$[\![P]\!](v) = f^n[\![Id_A]\!](v) \text{We now case split on n} \tag{25}$$

Case $\mathit{Upto}(0, P')$

$$\begin{aligned} (a, b) \in [\![\mathit{Upto}(0, P')]\!](v) &\Leftrightarrow (a, b) \in f^0[\![Id_A]\!](v) \\ &\Leftrightarrow (a, b) \in [\![Id_A]\!](v) \\ &\Leftrightarrow (a, b) \lhd_{(A,A),v} Id_A \text{ by } \Phi(\Sigma, Id_A, A, A) \\ &\Leftrightarrow (a, b) \lhd_{(A,A),v} \mathit{Upto}(0, P') \text{ by (Upto0)} \end{aligned} \tag{26}$$

Case $Upto(n+1, P')$

$$\begin{aligned}
(a,b) \in [\![Upto(m+1,P')]\!](v) &\Leftrightarrow (a,b) \in f^{m+1}[\![Id_A]\!](v) \\
&\Leftrightarrow (a,b) \in (join([\![P']\!](v), f^m[\![Id_A]\!](v)) \cup f^m[\![Id_A]\!](v)) \\
&\Leftrightarrow (a,b) \in join([\![P']\!](v), [\![Upto(m,P)]\!](v)) \vee (a,b) \in [\![Upto(m,P')]\!](v) \\
&\Leftrightarrow (\exists a'.(a,a') \in [\![P']\!](v) \wedge (a',b) \in [\![Upto(m,P')]\!](v)) \vee (a,b) \lhd_{(A,A),v} Upto(m,P') \\
&\Leftrightarrow (\exists a'.(a,a') \lhd_{(A,A),v} P' \wedge (a',b) \lhd_{(A,A),v} Upto(m,P')) \vee (a,b) \lhd_{(A,A),v} Upto(m,P' \\
&\Leftrightarrow (a,b) \lhd_{(A,A),v} Upto(m+1,P') \text{ by (Upto n+1), (Upto n)}
\end{aligned} \tag{27}$$

case $P = FixedPoint(P')$

We have by inversion of the (FixedPoint) type rule $\Gamma \vdash P\colon (A,A) \wedge \Gamma \vdash P'\colon (A,A)$

$$\begin{gathered}
\text{let } f = (\lambda pairs.join([\![P']\!](v), pairs) \cup pairs) \\
\text{then} \\
[\![P]\!](v) = fix(f) \text{ In the domain } closure(A,v)
\end{gathered} \tag{28}$$

Lemma: f is continuous in the domain $closure(A,v)$

Firstly, f is monotonous.
Let $x \subseteq y$

$$\begin{aligned}
(a,b) \in f(x) &\Rightarrow (a,b) \in x \vee (\exists a'.(a,a') \in [\![P']\!](v) \wedge (a',b) \in x) \\
&\Rightarrow (a,b) \in y \vee (\exists a'.(a,a') \in [\![P']\!](v) \wedge (a',b) \in y) \\
&\Rightarrow (a,b) \in f(y) \\
&\therefore f(x) \subseteq f(y)
\end{aligned} \tag{29}$$

Secondly, f preserves the lubs of chains.

Consider a chain $x1 \subseteq x2 \subseteq ...$ in $closure(A,v)$ Since $closure(A,v)$ is a domain, the lub, $\bigcup_n x_n$ is also in $closure(A,v)$

$$\begin{gathered}
\forall m.x_m \subseteq \bigcup_n x_n \\
\forall m.f(x_n) \subseteq f(\bigcup_n(x_n)) \\
\therefore \bigcup_n f(x_n) \subseteq f(\bigcup_n(x_n))
\end{gathered} \tag{30}$$

To get the inverse relation,

$$(a,b) \in f(\bigcup_n(x_n)) \Rightarrow ((\exists n.(a,b) \in x_n) \vee (\exists m, a'.(a,a') \in [\![P']\!](v) \wedge (a',b) \in x_m) \tag{31}$$

let $n' = max(n, m)$ so $x_n \subseteq x_{n'} \wedge x_m \subseteq x_{n'}$

$$\begin{aligned}
&\exists n'.((a,b) \in x_{n'}) \vee (\exists a'.(a,a') \in [\![P']\!](v) \wedge (a',b) \in x_{n'}) \\
&\therefore \exists n'.(a,b) \in f(x_{n'}) \\
&\therefore \exists n'.f(\bigcup_n x_n) \subseteq f(x_{n'} \\
&\therefore f(\bigcup_n x_n) \subseteq \bigcup_n f(x_{n'})
\end{aligned} \tag{32}$$

So f is Scott-continuous.

Now, by Tarski's fixed point theorem

$$[\![FixedPoint(P')]\!](v) = fix(f) = \bigsqcup_n f^n(\bot)$$

Lemma $(a,b) \lhd_{(A,A),v} FixedPoint(P') \Leftrightarrow \exists n.(a,b) \in f^n(\bot)$ Firstly, in the forwards direction, $(a,b) \lhd_{(A,A),v} FixedPoint(P') \Rightarrow \exists n.(a,b) \in f^n(\bot)$

By inversion of the operational rules (FixedPoint0), (FixedPoint n) and $(a,b) \lhd_{(A,A),v} FixedPoint(P')$, we get two cases.

Case $(a,b) \lhd_{A,A,v} Id_A$**:**
by $\Phi(\Sigma, Id_A, A, A), (a,b) \in [\![Id_A]\!](v) = \bot$
so $n = 0$

Case $(a,b) \lhd_{(A,A),v} P' \wedge (b,c) \lhd_{(A,A),v} FixedPoint(P')$
(hence $(a,c) \lhd_{(A,A),v} FixedPoint(P')$)
by $\Phi(\Sigma, P', A, A), and (b,c) \lhd_{(A,A),v} FixedPoint(P')$

$$(a,b) \in [\![P']\!](v) \wedge \exists n.(b,c) \in f^n(\bot)$$

Instantiating with $m = n$ gives

$$(a,b) \in [\![P']\!](v) \wedge (b,c) \in f^m(\bot)$$

so $(a,c) \in f(f^m(\bot)) = f^{m+1}(\bot)$
Hence $(a,b) \lhd_{(A,A),v} FixedPoint(P') \Rightarrow \exists n.(a,b) \in f^n(\bot)$

To go the other way, we need to prove $(a,b) \in \bigsqcup_n f^n(\bot) \Rightarrow (a,b) \lhd_{(A,A),v} FixedPoint(P')$
$(a,b) \in \bigsqcup_n f^n(\bot)$ Means that either:

Case $(a,b) \in \bot$

$$\begin{aligned}
&\therefore (a,b) \in [\![Id_A]\!](v) \\
&\therefore (a,b) \lhd_{(A,A),v} Id_A \text{ By } \Phi(\Sigma, Id_a, A, A) \\
&\therefore (a,b) \in [\![Id_A]\!](v) \text{ By (Fix1)}
\end{aligned} \tag{33}$$

Case $\exists n \geq 0.(a,b) \in f^{n+1}(\bot) \wedge \neg((a,b) \in f^n(\bot))$

$$\begin{aligned}
&(\exists a'.(a,a') \in [\![P']\!](v) \wedge (a',b) \in f^n(\bot)) \vee (a,b) \in f^n(\bot) \wedge \neg((a,b) \in f^n(\bot)) \\
&\qquad \therefore \exists a'.(a,a') \in [\![P']\!](v) \wedge (a',b) \in f^n(\bot) \\
&\qquad \therefore (a,a') \lhd_{(A,A),v} P' \wedge (a',b) \lhd_{(A,A),v} FixedPoint(P') \\
&\qquad \therefore (a,b) \lhd_{(A,A),v} FixedPoint(P')
\end{aligned} \tag{34}$$

so we have $(a,b)\lhd_{(A,A),v} FixedPoint(P') \Leftrightarrow \exists n.(a,b) \in f^n(\bot) \Leftrightarrow (a,b) \in [\![FixedPoin(P')]\!](v)$

□

1.7 Write Semantics

We have fairly simple write semantics, we define the type of the *write* function as mapping a view and a set of pairs related by a relation to a new view.

$$write\colon View_\Sigma \to \wp(A \times R \times B) \rightharpoonup View_\Sigma \text{For } A, B \in \tau \tag{35}$$

We define *write* as so:
let $rs = \{(a_i, r_i, b_i) \in (A \times R \times B) \mid 0 < i \leqslant n\}$ for some n being the size of the set.

$$\begin{aligned}
write(v)(rs) =& v\,[A \mapsto v_{table}(A) \cup \{a_i \mid 0 < i \leqslant n\}] \\
& [B \mapsto v_{table}(B) \cup \{b_i \mid 0 < i \leqslant n\}] \\
& [r \mapsto v_{rel}(R) \cup \{(a_i, b_i) \mid 0 < i \leqslant n\}]
\end{aligned} \tag{36}$$