New Generation Computing, 15 (1997) 339-362
OHMSHA, LTD. and Springer-Verlag

Relational Semantics for Locally Nondeterministic Programs

Liangwei XU
Mathematical Systems Institute Inc.,
10F FOUR SEASONS BLDG.,
2-4-3 Shinjuku, Shinjuku-ku, Tokyo, Japan.
Masato TAKEICHI
Department of Mathematical Engineering,
The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan.
Hideya IWASAKI
Faculty of Technology,
Tokyo University of Agriculture and Technology,
2-24-16 Naka-chou, Koganei-shi, Tokyo, Japan.

Received 25 December 1995
Revised manuscript received 5 August 1996

Abstract Introducing nondeterministic operators in a conventional deterministic language gives rise to various semantic difficulties. One of the problems is that there has been no semantic domain that is wholly satisfactory for denoting nondeterministic programs.

In this paper, we propose an approach based on relational algebra. We divide the semantics of a nondeterministic program into two parts. The first part concerns the angelic aspect of programs and the second part concerns the demonic aspect of programs. Because each semantic function used in these parts is monotonic with respect to an ordering on relations, the existence of the fixed points of recursively defined nondeterministic programs is ensured.

Keywords: Nondeterminism (Demonic, Angelic), Semantics, Relation.

§1 Introduction

Nondeterminism, according to their termination properties, can be divided into three different notions: *erratic*, *angelic* and *demonic* ones.

The notion of erratic nondeterminism corresponds to a simple choice

($\diamondsuit$). $e_1 \diamondsuit e_2$ is considered as a competition of the expressions e_1 and e_2 to be chosen. The choice is performed in a totally arbitrary way without taking any particular properties of e_1 and e_2 into account. The term "erratic" is due to M. Broy.

The notion of angelic nondeterminism corresponds to the choice that termination is guaranteed as long as termination is possible. There are two kinds of angelic nondeterminisms: local and global. In locally angelic nondeterminism, a choice $e_1 \diamondsuit e_2$ is defined as

$$\begin{aligned} e_1 \diamondsuit e_2 &= \text{either } e_1 \text{ or } e_2 && \textbf{if } e_1 \neq \bot \wedge e_2 \neq \bot \\ &= e_1 && \textbf{if } e_2 = \bot \\ &= e_2 && \textbf{if } e_1 = \bot \end{aligned}$$

where $\bot$ denotes a nonterminating computation or an undefined value. Such a locally angelic choice is well known as McCarthy's ambiguity operator *amb*. By globally angelic nondeterminism, on the other hand, choices are made so as to obtain termination of the whole computation if possible at all. That is, the choice $e_1 \diamondsuit e_2$ is made according to the surrounding expression.

The notion of demonic nondeterminism corresponds to the choice that if nontermination is possible then it is certain. There are also two kinds of demonic nondeterminisms: local and global. In locally demonic nondeterminism, the evaluation of $e_1 \diamondsuit e_2$ fails to terminate if either of e_1 or e_2 does:

$$\begin{aligned} e_1 \diamondsuit e_2 &= \text{either } e_1 \text{ or } e_2 && \textbf{if } e_1 \neq \bot \wedge e_2 \neq \bot \\ &= \bot && \textbf{otherwise} \end{aligned}$$

In globally demonic nondeterminism, choice will further be made so that a value that makes the entire computation fail. The use of the terms "angelic" and "demonic" is due to C.A.R. Hoare.

There are various approaches to formal specification of nondeterministic programs. It has been known that the erratic, globally angelic and globally demonic nondeterminisms are well connected to the three power domains: Plotkin, Hoare and Smyth.[12] These three nondeterminisms can also be described by relational algebra.[3,2] However, to locally angelic and locally demonic nondeterminisms, there has been no approach that is wholly satisfactory. The main problem is that the locally angelic and locally demonic choices are not monotonic with respect to any of the orderings of the three domains. Therefore, we cannot specify a locally angelic or a locally demonic nondeterministic program directly on any of these domains. This means that some additional manipulations are needed.

In this paper, we propose a relational semantics for both terminating and nonterminating computations of a nondeterministic programming language which includes erratic, locally angelic and locally demonic choices. Our semantic domain is based on relations. Let f be a nondeterministic program. We define the semantics of f with two aspects: angelic aspect (denoted as $\mathcal{A}[\![f]\!]$)

and demonic aspect (denoted as $\mathcal{D}[\![f]\!]$). $\mathcal{A}[\![f]\!]$ is a relation containing all pairs (y,x) where y is a possible result of a terminating computation of $(f\ x)$. $\mathcal{D}[\![f]\!]$ is a subset of $\mathcal{A}[\![f]\!]$ containing all pairs (y,x) such that the computation of $(f\ x)$ never fails to terminate. To show the difference between $\mathcal{A}[\![\_]\!]$ and $\mathcal{D}[\![\_]\!]$, let's consider a simple example $f = \lambda x.x/(1 \diamondsuit 0)$ where $\diamondsuit$ stands for the erratic choice and f is a nondeterministic function on natural numbers. Two semantic aspects on f is defined as follows.

$$\mathcal{A}[\![f]\!] = \{(n, n) \mid n \text{ is a natural number}\}$$
$$\mathcal{D}[\![f]\!] = \{\}$$

$\mathcal{D}[\![f]\!]$ is not identical to $\mathcal{A}[\![f]\!]$ because the possible results of $1 \diamondsuit 0$ includes 0 which is outside the definition domain of $(x/)$. In other words, the evaluation of $(f\ x)$ does not always terminate for any natural number x.

The definition of angelic and demonic aspects can be viewed as a combination of the Hoare and the Plotkin domains. Let A be a base set. For any set $B \subseteq A^{\perp}$ of possible results of a program, B is expressed by a pair of sets $\mathcal{A}_B$ and $\mathcal{D}_B$ where $\mathcal{A}_B = B/\{\perp\}$ and $\mathcal{D}_B =$ **if** $\perp \in B$ **then** emptyset **else** B. $\mathcal{A}_B$ corresponds to the angelic aspect of the program and $\mathcal{D}_B$ corresponds to the demonic aspect of the program. The angelic aspect is based on the well-known idea of using the Hoare domain to express angelic nondeterminism by identifying any set $C \cup \{\perp\}$ with C ($C \neq$ emptyset and $C \subseteq A$) and making emptyset play the the role of $\{\perp\}$. The demonic aspect is defined by identifying $C \cup \{\perp\}$ with the emptyset in the Plotkin domain.

In order to give a precise understanding of our notion, we show a relationship between our $\mathcal{A}[\![\_]\!]$ and $\mathcal{D}[\![\_]\!]$ and Dijkstra's *wlp*/*wp* notions[6] as follows. Let f be a nondeterministic program and Q be a post condition of f. For any input value x, we have

$$(wlp\ f\ Q)x \equiv \forall y\colon (y,x) \in \mathcal{A}[\![f]\!] \Rightarrow (Q\ y)$$
$$(wp\ f\ Q)x \equiv \exists y\colon (y,x) \in \mathcal{D}[\![f]\!] \wedge (wlp\ f\ Q)x$$

The rest of this paper is arranged as follows. After introducing some mathematical preliminaries in the following section, we define a small language with choice operators and specify its semantics in Section 3. Section 4 gives an example to demonstrate our approach. Related work and conclusions are discussed in Section 5. The lemmas and theorems presented in this paper are proved in Appendix.

§2 Relational Algebra

Let $\mathcal{U}$ be a universal set. A *binary relation* on $\mathcal{U}$ is an element of the powerset $\mathcal{R} = \mathcal{P}(\mathcal{U} \times \mathcal{U})$ of the cartesian product $\mathcal{U} \times \mathcal{U}$. All relations (on $\mathcal{U}$) form a complete, completely-distributive lattice with the set inclusion $\subseteq$ as the lattice order. The set union $\cup$ and intersection $\cap$ of two relations are the least upper bound and the greatest lower bound of them. The top element of the

lattice is denoted by $\top$ which is the total relation $\mathcal{U} \times \mathcal{U}$. The bottom element of the lattice is the empty relation $\emptyset$. The identity relation I is defined by $I = \{(x,x) \mid x \in \mathcal{U}\}$. For any relation R, we write aRb as a shorthand for $(a,b) \in R$ where b is regarded as the input and a as the output of the relation R. The *complement* of a relation R is defined by $\tilde{R} = \{(y,x) \mid \neg (yRx)\}$. The *conversion* of a relation is defined by $R^{\circ} = \{(y,x) \mid xRy\}$. There are two sorts of composition of relations: the standard and the demonic ones. Given two relations R and S, the standard composition of them $(R \cdot S)$ is defined by

$$\forall a, c \in \mathcal{U}: \quad a(R \cdot S)c \equiv \exists b \in \mathcal{U}: aRb \wedge bSc.$$

The composition $\cdot$ is associative and distributes universally over $\cup$.

Before defining the demonic composition, we need to define a special sort of relations called *monotype*. A relation A is called a monotype if $A \subseteq I$. For any monotypes A, $B \subseteq I$ and any relation R, the following properties are clearly satisfied.

(1) $A \cap B = A \cdot B$
(2) $A \cdot B = B \cdot A$
(3) $R \cdot A \subseteq R$
(4) $A \cdot A = A$

Thre is a one-to-one correspondence (denoted as ?) between a predicate p over $\mathcal{U}$ and a monotype $p? = \{(x,x) \mid p\ x\}$. The *negation* of a monotype A is also a monotype $\overline{A}$ which is defined by $\overline{A} = \tilde{A} \cap I$. The *domain* and the *codomain* of a relation R are monotypes which is defined by $R\,\square = \{(x,x) \mid \exists y: yRx\}$ and $\square\, R = \{(y,y) \mid \exists x: yRx\}$, respectively. Both the domain and the codomain of a relation are monotypes satisfying:

(5) $R = R \cdot R\,\square = \square\, R \cdot R$
(6) $(R \cdot S)\,\square = (R\,\square \cdot S)\,\square$
(7) $\square(R \cdot S) = \square(R \cdot \square S)$
(8) $\forall A \subseteq I: \quad R\,\square \subseteq A \equiv R \subseteq \top \cdot A$

It follows from property (8) that the domain operator $\square$ distributes universally over $\cup$.

The *demonic composition* of relations R and S $(R \odot S)$ is defined by

$$R \odot S = R \cdot S \cdot (R \rhd S)$$
$$R \rhd S = \cup\{A \subseteq I \mid \square(S \cdot A) \subseteq R\,\square\}$$

We call $\rhd$ *demonic limitation*, which denotes a monotype such that

$$\forall A \subseteq I: \quad A \subseteq R \rhd S \equiv \square(S \cdot A) \subseteq R\,\square.$$

holds.* The demonic composition has the following properties.

* This definition is due to Backhouse and others.[2] Its form is convenient for calculational proofs.

(9) $(R \odot S)\Box = S\Box \cdot (R \rhd S)$
(10) $R \odot S = R \cdot S \cdot (R \odot S)\Box$
(11) $(R \cdot S = R \odot S) \Leftarrow (R\Box \supseteq \Box S)$
(12) $R \cdot P = R \odot P$ (provided P is simple: $P \cdot P^{\circ} \subseteq I$)
(13) $R = I \odot R = R \odot I$
(14) $(R \odot S) \odot T = R \odot (S \odot T)$

The proof of (1) ~ (13) are straightforward from definitions and the proof of (14) is referred to Backhouse.[2] The conditional expression $R \rightarrow S,T$ is defined by

$$R \rightarrow S,T = S \cdot ((true =)? \cdot R)\Box \cup T \cdot ((false =)? \cdot R)\Box$$

A relation R is said to be a *function* if it is *entire* and *simple*. Because a function f is simple, function application $(f\ x)$ is uniquely defined.

For ensuring the existence of the fixed points of a recursively defined program, we need an ordering among relations on which the demonic aspects of all operators in our language are monotonic.

Definition 1
Let $R,S \in \mathcal{R}$ be relations, we define a relation $\sqsubseteq$ between R and S as follows.

$$R \sqsubseteq S \equiv R \subseteq S \wedge S \cdot R\Box \subseteq R.$$

Note that for any relations R and S, $R \sqsubseteq S$ is related to the Plotkin ordering ($\sqsubseteq_P$). If we identify any set containing bottom with the empty set.

$$(\forall x\colon R.x \sqsubseteq_P S.x) \equiv R \sqsubseteq S$$

holds where $R.x = \{y \mid (y,x) \in R\}$ and $S.x = \{y \mid (y,x) \in S\}$.

The relation $\sqsubseteq$ is an ordering on $\mathcal{R}$ according to the following theorem.

Theorem 1
$(\mathcal{R}, \sqsubseteq)$ is a complete partial ordered set (CPO).

Our basic reasoning tool is the following Tarski theorem.

Theorem 2 (*Tarski Theorem*)
For any CPO $(A,\leq)$ and monotonic function $\Phi \in A \leftarrow A$, there exists a least fixed point $(\mu X.\Phi(X))$ such that

1. $\Phi(\mu X.\Phi(X)) = \mu X.\Phi(X)$
2. $\forall X \in A\colon\ (\mu X.\Phi(X) \leq X \Leftarrow \Phi(X) \leq X)$

The Tarski theorem has the following corollary.

Corollary 1
For any CPO$(A,\leq)$ and binary function $\Psi \in A \leftarrow (A \times A)$ where Ψ is monotonic w.r.t. $\leq$ on both of its arguments, we have $\mu X.\Psi(X,Y)$ is monotonic on Y and $\mu Y.\Psi(X,Y)$ is monotonic on X:

1. $\mu X.\Psi(X, Y_1) \leq \mu X.\Psi(X, Y_2) \Leftarrow Y_1 \leq Y_2$
2. $\mu Y.\Psi(X_1, Y) \leq \mu Y.\Psi(X_2, Y) \Leftarrow X_1 \leq X_2$

§3 A Small Language

In order to explain our approach precisely, we define a small language and give its semantics. Our lnguage consists of a set of nondeterministic functional expressions in which the choices (erratic, locally angelic and locally demonic) are lifted to those between functions:

$f \,]\![\, g = \lambda x.\ (f\ x \diamondsuit g\ x)$	(where $\diamondsuit$ is the erratic choice)
$f \,[\!]\, g = \lambda x.\ (f\ x \diamondsuit g\ x)$	(where $\diamondsuit$ is the locally angelic choice)
$f \,\blacksquare\, g = \lambda x.\ (f\ x \diamondsuit g\ x)$	(where $\diamondsuit$ is the locally demonic choice)

3.1 The Syntax

Let D denotes all functional expressions that do not include any nondeterministic choices and F be a primitive syntax class of binding identifiers. The syntax is defined as follows.

$E ::= D$	(deterministic functional expression)
$\mid F$	(binding identifier)
$\mid E \circ E$	(sequential composition)
$\mid E \rightarrow E,E$	(conditional expression)
$\mid E \,]\![\, E$	(erratic choice)
$\mid E \,[\!]\, E$	(locally angelic choice)
$\mid E \,\blacksquare\, E$	(locally demonic choice)
$\mid \mu F.E$	(recursion)

A conditional expression $((e_1 \rightarrow e_2,e_3)\ x)$ is interpreted as follows. Firstly, check the result of $(e_1\ x)$, when it is true, executes $(e_2\ x)$; otherwise, executes $(e_3$ x).

In functional expressions D, there are two special functions *id* and ε. *id* denotes the identity function and ε denotes the empty function which maps any input to no results.

Recursions are denoted by $\mu f.e$ in which f is its binding identifier and e is its body. We denote the set of all free occurrence of identifiers in e as *free*(e). An expression e is called *closed* if *free*(e) is empty. The outermost expression of our language must be closed.

In operation, we assume that a recursion is treated by replacing the binding occurrence with the body of the recursion whenever it is needed.

3.2 Denotational Semantics

We define two kinds of semantic functons: angelic and demonic functions. The angelic semantic function is denoted as $\mathcal{A}[\![\_]\!] \in \mathcal{R} \leftarrow Env \leftarrow E$.

Similarly, the demonic semantic function is denoted as $\mathcal{D}[\![\_]\!] \in \mathcal{R} \leftarrow Env \leftarrow Env \leftarrow E$. Here, the type E is the set of all expressions and the type $Env = \mathcal{R} \leftarrow F$ denotes the set of environments. An environment $\sigma \in Env$ is a function which associates bound identifiers to relations. Adding a mapping $\langle f \mapsto X \rangle$ to an environment σ is denoted as $(\sigma + \langle f \mapsto X \rangle)$. We call σ to be an environment of an expression if σ associates every free binding identifier in the expression to a relation. To recursions, the angelic and the demonic semantic functions are defined as the least fixed point of mappings on relations with respect to the ordering $\subseteq$ and the ordering $\sqsubseteq$, respectively.

In addition, we denote $\mathcal{A}[\![e]\!]\sigma$ as $\mathcal{A}_0[\![e]\!]$ and $\mathcal{D}[\![e]\!]\sigma\rho$ as $\mathcal{D}_0[\![e]\!]$ when σ and ρ are empty relations. We also use $\mu_{\subseteq}X.H$ and $\mu_{\sqsubseteq}X.H$ as the least fixed points of the function $H \in \mathcal{R} \leftarrow \mathcal{R}$ based on orderings $\subseteq$ and $\sqsubseteq$, respectively.

We assume that there is a semantic function $\mathcal{I}[\![\_]\!]$ which maps "built-in" functional expressions to simple relations. For example, the identity function *id* is mapped to the identity relation I ($\mathcal{I}[\![id]\!] = I$) and the empty function ε is mapped to empty relation $\emptyset$ ($\mathcal{I}[\![\varepsilon]\!] = \emptyset$).

The semantic functions are defined by induction on the structure of the syntax of expressions. In the following definitions, relations $\mathcal{A}[\![e_1]\!]\sigma$, $\mathcal{A}[\![e_2]\!]\sigma$ and $\mathcal{A}[\![e_3]\!]\sigma$ are expressed as R_1, R_2 and R_3; relations $\mathcal{D}[\![e_1]\!]\sigma\rho$, $\mathcal{D}[\![e_2]\!]\sigma\rho$ and $\mathcal{D}[\![e_3]\!]\sigma\rho$ are expressed as S_1, S_2 and S_3, respectively.

(a1) $\mathcal{A}[\![d]\!]\sigma = \mathcal{I}[\![d]\!] \quad (d \in D)$
(a2) $\mathcal{A}[\![f]\!]\sigma = \sigma[\![f]\!] \quad (f \in F)$
(a3) $\mathcal{A}[\![e_1 \circ e_2]\!]\sigma = R_1 \cdot R_2$
(a4) $\mathcal{A}[\![e_1 \rightarrow e_2, e_3]\!]\sigma = R_1 \rightarrow R_2, R_3$
(a5) $\mathcal{A}[\![e_1 \,][\, e_2]\!]\sigma = R_1 \cup R_2$
$\mathcal{A}[\![e_1 \,[]\, e_2]\!]\sigma = R_1 \cup R_2$
$\mathcal{A}[\![e_1 \blacksquare e_2]\!]\sigma = (R_1 \cup R_2) \cdot R_1\,\square \cdot R_2\,\square$
(a6) $\mathcal{A}[\![\boldsymbol{\mu} f.e]\!]\sigma = \mu_{\subseteq}X.\mathcal{A}[\![e]\!](\sigma + \langle f \mapsto X \rangle)$

(d1) $\mathcal{D}[\![d]\!]\sigma\rho = \mathcal{I}[\![d]\!] \quad (d \in D)$
(d2) $\mathcal{D}[\![f]\!]\sigma\rho = \rho[\![f]\!] \quad (f \in F)$
(d3) $\mathcal{D}[\![e_1 \circ e_2]\!]\sigma\rho = S_1 \odot S_2$
(d4) $\mathcal{D}[\![e_1 \rightarrow e_2, e_3]\!]\sigma\rho = (\mathcal{A}[\![e_1 \rightarrow e_2, e_3]\!]\sigma) \cdot (S_1 \rightarrow S_2\,\square, e_3\,\square)$
(d5) $\mathcal{D}[\![e_1 \,][\, e_2]\!]\sigma\rho = (S_1 \cup S_2) \cdot S_1\,\square \cdot S_2\,\square$
$\mathcal{D}[\![e_1 \,[]\, e_2]\!]\sigma\rho = (\mathcal{A}[\![e_1 \,[]\, e_2]\!]\sigma) \cdot (S_1\,\square \cup S_2\,\square)$
$\mathcal{D}[\![e_1 \blacksquare e_2]\!]\sigma\rho = (S_1 \cup S_2) \cdot S_1\,\square \cdot S_2\,\square$
(d6) $\mathcal{D}[\![\boldsymbol{\mu} f.e]\!]\sigma\rho = \mu_{\sqsubseteq}Y.\mathcal{D}[\![e]\!](\sigma + \langle f \mapsto \mathcal{A}[\![\boldsymbol{\mu} f.e]\!]\sigma \rangle)(\rho + \langle f \mapsto Y \rangle)$

In contrast to the definition of $\mathcal{A}[\![\_]\!]$, $\mathcal{D}[\![\_]\!]$ needs two environments σ and ρ where the environment σ is used for associating the angelic aspect of the corresponding expression.

The definition of $\mathcal{D}_0[\![e_1 \,[]\, e_2]\!]$ in (d5) implies that if the execution of

either $(e_1\ x)$ or $(e_2\ x)$ always terminates, then the execution of $(e_1 \,[\!]\, e_2)x$ also terminates. Our definition of $e_1 \,[\!]\, e_2$ is identical to that of Broy and Nelson[5]'s dovetail when e_1 and e_2 are total. A program (f) is said to be total if it satisfies

$$wp\ f\ False = False.$$

Totality leads to an important relationship between $\mathcal{A}_0[\![\_]\!]$ and $\mathcal{D}_0[\![\_]\!]$ (see Theorem 5 and Corollary 4).

As an example of our definitions, consider the semantics of the following expressions *ee*, *ea* and *ed*.

$$\begin{aligned} ee&: \ ((0 \neq) \rightarrow \varepsilon, succ)\ ]\![\ id \\ ea&: \ ((0 \neq) \rightarrow \varepsilon, succ) \,[\!]\, id \\ ed&: \ ((0 \neq) \rightarrow \varepsilon, succ) \blacksquare id \end{aligned}$$

The semantic functions of them are derived from the above definitions.

$$\begin{aligned} \mathcal{A}_0[\![ee]\!] &= \{(1,0)\} \cup I & \mathcal{D}_0[\![ee]\!] &= \{(1,0),(0,0)\} \\ \mathcal{A}_0[\![ea]\!] &= \{(1,0)\} \cup I & \mathcal{D}_0[\![ea]\!] &= \{(1,0)\} \cup I \\ \mathcal{A}_0[\![ed]\!] &= \{(1,0),(0,0)\} & \mathcal{D}_0[\![ed]\!] &= \{(1,0),(0,0)\} \end{aligned}$$

3.3 Well-Definedness

As we have stated before, two kinds of orderings ($\subseteq$ and $\sqsubseteq$) are introduced in $\mathcal{R}$. $(\mathcal{R},\subseteq)$ and $(\mathcal{R},\sqsubseteq)$ are complete lattice and CPO, respectively. The following lemmas say that the operators we used are monotonic w.r.t. to either $\subseteq$ or $\sqsubseteq$. In the following discussion, a function which is monotonic with respect to $\subseteq$ ($\sqsubseteq$) is called $\subseteq$-monotonic ($\sqsubseteq$-monotonic).

Lemma 1
The relational operators $\cdot$, $\cup$ and the domain operator $\Box$ are $\subseteq$-monotonic.

From the above lemma and the **Corollary 1**, we can prove that $\mathcal{A}[\![\_]\!]$ is well defined to any expression in our language.

Theorem 3
$\mathcal{A}[\![e]\!]\sigma$ is well defined for any expression e and any nonempty environment σ (say $\sigma = \sigma' + \langle f \mapsto X \rangle$) of e. In addition, the function

$$X \mapsto \mathcal{A}[\![e]\!](\sigma' + \langle f \mapsto X \rangle)$$

is $\subseteq$-monotonic.

This theorem has the following corollary.

Corollary 2
$\mathcal{A}_0[\![e]\!]$ is well defined for any closed expression e.

Lemma 2
The demonic composition $\odot$ is $\sqsubseteq$-monotonic. Also the following functions:

(1) $X \mapsto (X \cup R) \cdot X\,\square \cdot R\,\square$
(2) $X \mapsto R \cdot (A \cdot (B \cdot X)\,\square \cup C)$

are $\sqsubseteq$-monotonic for any relation R and monotypes A, B and C.

Similar to the case of $\mathcal{A}[\![\_]\!]$, $\mathcal{D}[\![\_]\!]$ is well defined for any expression.

Theorem 4
The semantic function $\mathcal{D}[\![e]\!]\sigma\rho$ is well defined for any expression e and any nonempty environments σ and ρ (say $\rho = \rho' + \langle f \mapsto X \rangle$) of e. In addition, the function

$$X \mapsto \mathcal{D}[\![e]\!]\sigma(\rho' + \langle f \mapsto X \rangle)$$

is $\sqsubseteq$-monotonic.

This theorem has the following corollary.

Corollary 3
The semantic function $\mathcal{D}_0[\![e]\!]$ is well defined for any closed expression e.

3.4 The Relationship Between Angelic and Demonic Semantics

So far, we have only informally stated that the demonic semantics of an expression is a subset of its angelic semantics. In this section, a formal description of the relationship between them is given.

Theorem 5
For any expression e and its environments σ and ρ where σ and ρ satisfy

$$\forall f \in \mathit{free}(e): \quad \rho[\![f]\!] \sqsubseteq \sigma[\![f]\!]$$

we have $\mathcal{D}[\![e]\!]\sigma\rho \sqsubseteq \mathcal{A}[\![e]\!]\sigma$.

The following corollary is straightforward from the above theorem.

Corollary 4
For any closed expression e, we have $\mathcal{D}_0[\![e]\!] \sqsubseteq \mathcal{A}_0[\![e]\!]$.

From the relationship between the $\mathcal{A}_0[\![\_]\!]$ / $\mathcal{D}_0[\![\_]\!]$ and the *wp* / *wlp* notions, it is clear that the above Corollary shows that programs of our language are total.

Using the $\mathcal{A}_0[\![\_]\!]$ / $\mathcal{D}_0[\![\_]\!]$ interpretations, we can reason about termination properties of nondeterministic programs. For example, the termination properties of the following three simple programs from natural numbers ($\mathcal{N}$) to natural numbers

ne: $\mu f.id \;]\,[\; (f \circ succ)$
na: $\mu f.id \;[]\; (f \circ succ)$
nd: $\mu f.id \;\blacksquare\; (f \circ succ)$

are known from their angelic and demonic aspects. By the interpretations of (a1) ~ (a6) and (d1) ~ (d6), we have:

$$\mathcal{A}_0[\![ne]\!] = \mu_{\subseteq} X.I \cup X \cdot succ = \{(m,n) \mid m,n \in \mathcal{N} \wedge (m \geq n)\}$$
$$\mathcal{D}_0[\![ne]\!] = \mu_{\sqsubseteq} X.(I \cup X \odot succ) \cdot X \sqsubset = \emptyset$$
$$\mathcal{A}_0[\![na]\!] = \mu_{\subseteq} X.I \cup X \cdot succ = \{(m,n) \mid m,n \in \mathcal{N} \wedge (m \geq n)\}$$
$$\mathcal{D}_0[\![na]\!] = \mathcal{A}[\![na]\!] = \{(m,n) \mid m,n \in \mathcal{N} \wedge (m \geq n)\}$$
$$\mathcal{A}_0[\![nd]\!] = \mu_{\subseteq} X.(I \cup X \cdot succ) \cdot X \sqsubset = \emptyset$$
$$\mathcal{D}_0[\![nd]\!] = \mu_{\sqsubseteq} X.(I \cup X \odot succ) \cdot X \sqsubset = \emptyset$$

According to the interpretations of $\mathcal{A}_0[\![e]\!]$ and $\mathcal{D}_0[\![e]\!]$, for any functional expression e and input value x:

- $(e\ x)$ never terminates when $(x,x) \notin \mathcal{A}_0[\![e]\!] \sqsubset$.
- $(e\ x)$ may either terminate or nonterminate if $(x,x) \in \mathcal{A}_0[\![e]\!] \sqsubset$ and $(x, x) \notin \mathcal{D}_0[\![e]\!] \sqsubset$.
- $(e\ x)$ always terminates when $(x,x) \in \mathcal{D}_0[\![e]\!] \sqsubset$.

Therefore, the above semantics of *ne*, *na*, and *nd* show three different termination properties. The result of $\mathcal{A}_0[\![ne]\!]$ says that the possible results of terminating evaluation of $(ne\ x)$ are natural numbers which are greater than x. Also, the result $\mathcal{D}_0[\![ne]\!] = \emptyset$ indicates that $(ne\ x)$ may not terminate because for any natural number x, (x,x) is not an element of $\mathcal{D}_0[\![ne]\!] \sqsubset$ (the empty set). In contrast to *ne*, the possible results of terminating evaluation of $(na\ x)$ are the same as those of $(ne\ x)$, but $(na\ x)$ will never fail to terminate because (x,x) belongs to $\mathcal{D}_0[\![na]\!] \sqsubset = \{(n,n) \mid n \in \mathcal{N}\}$ for any natural number x. Finally, $(nd\ x)$ will never terminate in any result because $\mathcal{A}_0[\![nd]\!] \sqsubset$ is empty. Obviously, among the above three nondeterministic functions, *na* has the only one that is unbounded nondeterminism. Unbounded nondeterminism refers to the case where even computations that are known to terminate may have a (countable) infinite number of possible results.

§4 An Example

In this section, we give an application example of the demonic aspect. The example is expressed as a game played between two players (p_1 and p_2). Two players alternatively remove one or two matches from a pile. The player p_1 is assumed to be the first to move, and the player who removes the last match losses. The problem is to find a winning strategy (if exists) of the player p_1 when the game starts on an initial state (the finite number of matches of a pile).

4.1 Formal Definition

We express p_1 and p_2 as nondeterministic functions from integers to integers. The game on an initial state x that p_1 wins is expressed as the following nondeterministic function g such that $g\ x$ terminates:

$$\begin{aligned} g &= (emp \rightarrow id, g) \circ p_2 \circ (emp \rightarrow \varepsilon, id) \circ p_1 \\ emp &= (0 \geq) \end{aligned}$$

The above definition shows that if p_2 is the last one to move, g will terminate; if p_1 is the last one to move, g will never terminate (expressed as ε).

Let's consider the definition of p_1 and p_2. From the rule of the game, the permitted moves by p_1 and p_2 are (-1) or (-2). The strategy taken by p_1 is expressed as a condition st:

$$p_1 = st \rightarrow (-1), (-2)$$

In addition, the player p_1 has a winning strategy on an initial state x implies that for any choice made by p_2, $g\ x$ always terminates. In other words, we need to design (if exists) st such that x exists in the domain of the demonic aspect of g where

$$p_2 = (-1) \sqcap (-2)$$

Since the game may start from any one finite number, the problem becomes of designing a winning strategy (v) such that for any strategy (st) and any initial state x, we have:

$$(x,x) \in \mathcal{D}_0[\![g_{st}]\!]\ \square \Rightarrow (x,x) \in \mathcal{D}_0[\![g_v]\!]\ \square$$

Put all above together, we are to design a predicate v such that $\mathcal{D}_0[\![g_v]\!]\ \square$ is the largest monotype (among $\mathcal{D}_0[\![g_{st}]\!]\ \square$) in the sense of set inclusion where

$$\begin{aligned} g_{st} &= (emp \rightarrow id, g_{st}) \circ p_2 \circ (emp \rightarrow \varepsilon, id) \circ p_1 \\ emp &= (0 \geq) \\ p_1 &= st \rightarrow (-1), (-2) \\ p_2 &= (-1) \sqcap (-2) \end{aligned}$$

4.2 Derivation

According to the semantic definitions, we have:

$$\begin{aligned} \mathcal{D}_0[\![g_{st}]\!] &= \mu_{\sqsubseteq} X.\ (emp? \cup \mathcal{A}_0[\![g_{st}]\!] \cdot X\ \square \cdot \overline{emp?}) \odot R_{st} \\ R_{st} &= \mathcal{D}_0[\![p_2 \circ (over \rightarrow \varepsilon, id) \circ p_1]\!] \end{aligned}$$

Through a simple calculation, we get:

$$(15)\quad R_{st} = M_{23} \cdot st? \cdot (2 \leq)? \cup M_{34} \cdot \overline{st?} \cdot (3 \leq)?$$

where $M_{23} = (-2) \cup (-3)$ and $M_{34} = (-3) \cup (-4)$ are relations. Because M_{23} and M_{34} are total, the domain of R_{st} can be expressed as:

(16) $R_{st}\,\square = st? \cdot (2 \leq)? \cup \overline{st?} \cdot (3 \leq)?$

Our derivation is along that following steps. Firstly, we find a necessary condition for a monotype A such that $\forall st\colon \mathcal{D}_0[\![g_{st}]\!]\,\square \subseteq A$ (the following property (20)). We then find the least one (K) among these monotypes. Finally, we define a monotype $v?$ according to K and prove that $\mathcal{D}_0[\![g_v]\!]\,\square$ satisfies the necessary condition in (20).

Two functions J and H are defined as follows.

$$\begin{aligned} J(X) &= emp? \cup X \cdot \overline{emp?} \\ H(X,A,B) &= A \cdot (2 \leq)? \cdot J(X) \rhd M_{23} \cup B \cdot (3 \leq)? \cdot J(X) \rhd M_{34} \end{aligned}$$

J and H have properties (17) ~ (21):

(17) H is $\subseteq$-monotonic with respect to its three arguments

(18) $(J(X) \odot R_{st})\,\square = H(X,st?,\overline{st?})$

(19) $\mathcal{D}_0[\![g_{st}]\!]\,\square = H(\mathcal{D}_0[\![g_{st}]\!]\,\square,st?,\overline{st?})$

(20) $(\forall st\colon \mathcal{D}_0[\![g_{st}]\!]\,\square \subset A) \Leftarrow H(A,I,I) \subseteq A$

(21) Denoting $K = \bigcup_{n\geq 0} H^n(\emptyset,I,I)$, K is the least fixed point of $X \mapsto H(X,I,I)$

where $H^0(\emptyset,I,I) = \emptyset$ and $H^{n+1}(\emptyset,I,I) = H(H^n(\emptyset,I,I),I,I)$. Proofs of these properties are described in Appendix.

The property (20) shows that any fixed point of $X \mapsto H(X,I,I)$ are larger than $\mathcal{D}_0[\![g_{st}]\!]\,\square$ for any st. Since $H(X,I,I)$ is $\subseteq$-monotonic on X, the existence of such a fixed point is ensured by the Tarski theorem. The property (21) describes its least fixed point (K). Through a simple proof by induction on n, we know that

$$\begin{aligned} K &= \bigcup_{n\geq 0} H^n(\emptyset,I,I) = P \cup Q \\ P &= \bigcup_{n\geq 0} P_n = \{(n,n) \mid n \geq 2 \wedge n \bmod 3 = 2\} \\ Q &= \bigcup_{n\geq 0} Q_n = \{(n,n) \mid n \geq 3 \wedge n \bmod 3 = 0\} \end{aligned}$$

where $P_0 = \emptyset$; $P_1 = (2 =)?$; $P_{n+2} = P_{n+1} \rhd M_{23}$; $Q_0 = \emptyset$; $Q_1 = (3 =)?$ and $Q_{n+2} = Q_{n+1} \rhd M_{34}$. It is also known that $J(K) \rhd M_{23} = J(P \cup Q) \rhd M_{23} \subseteq P$ and $J(K) \rhd M_{34} = J(P \cup Q) \rhd M_{34} \subseteq Q$ hold. This hints us that if we define $v? = P$, we have

$$\begin{aligned} & H(\mathcal{D}_0[\![g_v]\!]\,\square,I,I) \\ &= \quad \{\text{Def. of } H\} \\ &\quad (2 \leq)? \cdot J(\mathcal{D}_0[\![g_v]\!]\,\square) \rhd M_{23} \cup (3 \leq)? \cdot J(\mathcal{D}_0[\![g_v]\!]\,\square) \rhd M_{34} \\ &= \quad \{\mathcal{D}_0[\![g_v]\!]\,\square \subseteq K\} \\ &\quad \{J(\mathcal{D}_0[\![g_v]\!]\,\square) \rhd M_{23} \subseteq J(K) \rhd M_{23} \subseteq P\} \\ &\quad \{J(\mathcal{D}_0[\![g_v]\!]\,\square) \rhd M_{34} \subseteq J(K) \rhd M_{34} \subseteq Q\} \end{aligned}$$

$$
\begin{aligned}
& P \cdot (2 \leq)? \cdot J(\mathcal{D}_0[\![g_v]\!] \sqcap) \rhd M_{23} \cup Q \cdot (3 \leq)? \cdot J(\mathcal{D}_0[\![g_v]\!] \sqcap) \\
& \rhd M_{34} \\
\subseteq\ & \{v? = P;\ Q \subseteq \overline{P} = \overline{v?}\} \\
& v? \cdot (2 \leq)? \cdot J(\mathcal{D}_0[\![g_v]\!] \sqcap) \rhd M_{23} \cup \overline{v?} \cdot (3 \leq)? \cdot J(\mathcal{D}_0[\![g_v]\!] \\
& \sqcap) \rhd M_{34} \\
=\ & \{\text{Def. of } H\} \\
& H(\mathcal{D}_0[\![g_v]\!] \sqcap, v?, \overline{v?}) \\
=\ & \{\text{Prop. (19)}\} \\
& \mathcal{D}_0[\![g_v]\!] \sqcap
\end{aligned}
$$

and therefore, $K \subseteq \mathcal{D}_0[\![g_v]\!] \sqcap$ holds.

Put all above together, we have

$$\forall st:\quad \mathcal{D}_0[\![g_{st}]\!] \sqcap \subseteq K = \mathcal{D}_0[\![g_v]\!] \sqcap$$

where $v? = P = \{(n,n) \mid n \geq 2 \wedge n \ mod\ 3 = 2\}$. We conclude that v is the winning strategy of p_1 and $K = \{(n,n) \mid n \geq 2 \wedge n \ mod\ 3 \neq 1\}$ is the set of initial states in which p_1 having a winning strategy.

§5 Related Work and Conclusions

Broy[4] proposed an approach to define a locally nondeterministic functional language but it seems a little complicated and not so much suitable for reasoning about programs. In imperative languages, the angelic choice can be described by Dijkstra's *wlp*/*wp* notions. Broy[5] showed that the angelic choice (which is called dovetail) between partial commands can be defined by a combination of *wlp*/*wp*. There are other approaches[10] which sidestep the problem by describing programs only for terminating computations. The drawback of such approaches is that they cannot distinguish the value 1 from that of $1/(1 \diamondsuit 0)$.

In relational algebra, several papers[3,7] defined a nondeterministic program with a pair of a relation and a vector (monotype), where the relation express the "breadth" and the vector express the "definedness" of the program. In our notion, such a relation-vector pair can be expressed as $(\mathcal{A}_0[\![f]\!], \mathcal{D}_0[\![f]\!] \sqcap)$ for any nondeterministic program f.

Although we restricted ourselves to the semantics of nondeterministic functional language, we believe that the approach is also applicable to the other program languages. We did not use the explicit element $\perp$ in our semantic domain to express undefined values since it will seriously destroy the advantage of the relational calculus (see Refs. 3) and 2)).

Our approach has its limitations. Because we lifted the nondeterministic choices between values into those between nondeterministic functions, the theory can only be applied to evaluators in singular form not in plural form. An evaluator is in singular nondeterministic form means that $\lambda x.(x + x)(1 \diamondsuit 2)$ is evaluated to $(\lambda x.(x + x)1) \diamondsuit (\lambda x.(x + x)2)$ which results in 2 or 4. In contrast,

an evaluator in plural form means that the above expression is evaluated to $(1 \diamondsuit 2) + (1 \diamondsuit 2)$ which results in 2, 3 or 4. Developing semantics for such an evaluator is a subject for future research.

Acknowledgements

We would like to thank the anonymous referees for their suggestions, which greatly improved the presentation of this paper.

References

1) Backhouse, R. and Hoogendijk, P., "Elements of Relational Theory of Datatypes," *Lecture Notes in Computer Science*, *755*, 1993.
2) Backhouse, R. and Jaap van der Woude, "Demonic Operators and Monotype Factors," *Mathematical Structures in Computer Science*, *3*, *4*, pp. 417-433, 1993.
3) Berghammer, R. and Zierer, H., "Relational Algebraic Semantics of Deterministic and Nondeterministic Programs," *Theoretical Computer Science*, *43*, pp. 123-147, 1986.
4) Broy, M., "A Theory for Nondeterminism, Parallelism, Communication and Concurrency," *Theoretical Computer Science*, *45*, pp. 1-61, 1986.
5) Broy, M. and Nelson, G., "Adding Fair Choice to Dijkstra's Calculus," *Trans. Program. Lang. and Syst. ACM*, *16*, *3*, pp. 924-938, 1994.
6) Dijkstra, E.W. and Scholten, C.S., *Predicate Calculus and Program Semantics*, Springer-Verlag, Berlin, 1990.
7) Henk Doornbos, "A Relational Model of Programs without the Restriction to Egli-Milner-Monotone Constructs," in *Programming Concepts, Methods and Calculi* (E.-R. Olderog), North-Holland, pp. 363-384, 1994.
8) Hesselink, Wim H., "Angelic Termination in Dijkstra's Calculus," *Lecture Notes in Computer Science*, *947*, 1995.
9) Hoare, C.A.R. and Jifeng He, "The Weakest Prespecification," *Fundamenta Informaticae*, *9*, pp. 51-84 and pp. 217-252, 1986.
10) Hughes, J. and O'Donnel, J., "Expressing and Reasoning about Non-Deterministic Functional Programs," in *Proceeding of Glasgow Workshop on Functional Programming*, Fraserburgh, Scotland, Springer-Verlag, August 1989.
11) McCarthy, J., "Towards a Mathematical Science of Computation," in *Computer Programming and Formal Systems* (P. Braffort and D. Hirschberg, eds.), North-Holland, Amsterdam, pp. 33-70, 1963.
12) Sondergaard, H. and Sestoft, P., "Non-Determinism in Functional Languages," *The Computer Journal*, *35*, *5*, pp. 514-523, October 1992.
13) Tarski, A., "On the Calculus of Relations," *Journal of Symbolic Logic*, *6*, *3*, pp. 73-89, 1941.

Appendix: Proofs

In this section, we prove the theorems of this paper.

Proof of Theorem 1:

From the properties of $\subseteq$, we know that $\sqsubseteq$ is obviously a partial order. The empty relation $\emptyset$ is the least element of $(\mathcal{R}, \sqsubseteq)$. Let $\mathcal{C}$ be an arbitrary nonempty chain

of $\mathcal{R}$. That is, $\mathcal{C}$ is a subset of $\mathcal{R}$ such that $R \sqsubseteq S \vee S \sqsubseteq R$ holds for all $R, S \in \mathcal{C}$. We show that $\bigcup \mathcal{C}$ is the least upper bound of the chain by proving

(1) $\forall S \in \mathcal{C}: S \sqsubseteq \bigcup \mathcal{C}$
(2) $(\forall S \in \mathcal{C}: S \sqsubseteq T) \Rightarrow \bigcup \mathcal{C} \sqsubseteq T$

hold. (1) is proved as follows. For any relation $S \in \mathcal{C}$, we argue:

$$
\begin{array}{ll}
& S \sqsubseteq \bigcup \mathcal{C} \\
\equiv & \{\text{Def. of } \sqsubseteq\} \\
& S \subseteq \bigcup \mathcal{C} \wedge (\bigcup \mathcal{C}) \cdot S^{\square} \subseteq S \\
\Leftarrow & \{S \in \mathcal{C};\ \text{Prop. of } \bigcup;\ \text{distributivity of } \cdot \text{ over } \bigcup\} \\
& \forall R \in \mathcal{C}: R \cdot S^{\square} \subseteq S \\
\Leftarrow & \{\text{Def. of } \sqsubseteq;\} \\
& \{S \sqsubseteq R \Rightarrow R \cdot S^{\square} \subseteq S\} \\
& \{R \sqsubseteq S \Rightarrow R \subseteq S \Rightarrow R \cdot S^{\square} \subseteq S\} \\
& \forall R \in \mathcal{C}: R \sqsubseteq S \vee S \sqsubseteq R \\
\equiv & \{S \in \mathcal{C};\ \mathcal{C} \text{ is a chain}\} \\
& \mathit{truc}
\end{array}
$$

For any relation $T \in \mathcal{R}$, we prove (2) as follows.

$$
\begin{array}{ll}
& \bigcup \mathcal{C} \sqsubseteq T \\
\equiv & \{\text{Def. of } \sqsubseteq\} \\
& \bigcup \mathcal{C} \subseteq T \wedge T \cdot (\bigcup \mathcal{C})^{\square} \subseteq \bigcup \mathcal{C} \\
\Leftarrow & \{\text{Domain operator and distributivity}\} \\
& \{\forall R \in \mathcal{C}: T \cdot R^{\square} \subseteq R \Rightarrow T \cdot R^{\square} \subseteq \bigcup \mathcal{C}\} \\
& \forall S \in \mathcal{C}: S \subseteq T \wedge T \cdot S^{\square} \subseteq S \\
\equiv & \{\text{Def. of } \sqsubseteq\} \\
& \forall S \in \mathcal{C}: S \sqsubseteq T
\end{array}
$$

Proof of Lemma 2:

The monotonicity of $\odot$:

$$R \odot S \sqsubseteq R' \odot S' \Leftarrow R \sqsubseteq R' \wedge S \sqsubseteq S'$$

is proved as follows.

$$
\begin{array}{ll}
& R \odot S \subseteq R' \odot S' \\
\Leftarrow & \{U \odot V = U \cdot V \cdot (U \odot V)^{\square}\} \\
& R \cdot S \subseteq R' \cdot S' \wedge (R \odot S)^{\square} \subseteq (R' \odot S')^{\square} \\
\Leftarrow & \{\text{Monotonicity of } \cdot \text{ w.r.t. } \subseteq\} \\
& \{\text{Def. of } \odot\text{: } (R \odot S)^{\square} = S^{\square} \cdot (R \rhd S)\} \\
& R \subseteq R' \wedge S \subseteq S' \wedge S^{\square} \cdot (R \rhd S) \subseteq R' \rhd S' \\
\Leftarrow & \\
& R \subseteq R' \wedge S \subseteq S' \wedge S^{\square} \cdot (R \rhd S) \subseteq R \rhd S' \\
\equiv & \{\text{Def. of } \rhd\} \\
& R \subseteq R' \wedge S \subseteq S' \wedge {}^{\square}(S' \cdot S^{\square} \cdot (R \rhd S)) \subseteq R^{\square} \\
\Leftarrow & \{{}^{\square}(S \cdot (R \rhd S)) \subseteq R^{\square}\} \\
& R \subseteq R' \wedge S \subseteq S' \wedge S' \cdot S^{\square} \subseteq S \\
\Leftarrow & \{\text{Def. of } \sqsubseteq\} \\
& R \sqsubseteq R' \wedge S \sqsubseteq S'
\end{array}
$$

and

$$
\begin{array}{l}
(R' \odot S') \cdot (R \odot S)\,\square \subseteq R \odot S \\
\quad \Leftarrow \quad \{U \odot V = U \cdot V \cdot (U \rhd V)\} \\
\quad\quad R' \cdot S' \cdot S\,\square \cdot (R \rhd S) \subseteq R \cdot S \cdot (R \rhd S) \\
\quad \Leftarrow \quad \{(R \rhd S) \text{ is a monotype}\} \\
\quad\quad R' \cdot S' \cdot S\,\square \cdot (R \rhd S) \subseteq R \cdot S \\
\quad \Leftarrow \\
\quad\quad S' \cdot S\,\square \subseteq S \wedge R' \cdot S \cdot (R \rhd S) \subseteq R \cdot S \\
\quad = \quad \{\square(S \cdot (R \rhd S)) \subseteq R\,\square\} \\
\quad\quad S' \cdot S\,\square \subseteq S \wedge R' \cdot R\,\square \cdot S \cdot (R \rhd S) \subseteq R \cdot S \\
\quad \Leftarrow \\
\quad\quad S' \cdot S\,\square \subseteq S \wedge R' \cdot R\,\square \subseteq R \wedge R \cdot S \cdot (R \rhd S) \subseteq R \cdot S \\
\quad \Leftarrow \quad \{\text{Def. of } \sqsubseteq \text{ and } R \cdot S \cdot (R \rhd S) \subseteq R \cdot S\} \\
\quad\quad R \sqsubseteq R' \wedge S \sqsubseteq S'
\end{array}
$$

Next, we prove that for any relation R, the function

$$X \mapsto (X \cup R) \cdot X\,\square \cdot R\,\square$$

is $\sqsubseteq$-monotonic:

$$(U \cup R) \cdot U\,\square \cdot R\,\square \sqsubseteq (V \cup R) \cdot V\,\square \cdot R\,\square \Leftarrow U \sqsubseteq V$$

From the monotonicity of $\cdot$, $\cup$ and $\square$ w.r.t. $\subseteq$, we have that

$$(U \cup R) \cdot U\,\square \cdot R\,\square \subseteq (V \cup R) \cdot V\,\square \cdot R\,\square \Leftarrow U \subseteq V \Leftarrow U \sqsubseteq V$$

holds. Assume that $U \sqsubseteq V$, we check that

$$
\begin{array}{l}
(V \cup R) \cdot V\,\square \cdot R\,\square \cdot ((U \cup R) \cdot U\,\square \cdot R\,\square)\,\square \\
\quad \subseteq (U \cup R) \cdot U\,\square \cdot R\,\square
\end{array}
$$

as follows.

$$
\begin{array}{l}
(V \cup R) \cdot V\,\square \cdot R\,\square \cdot ((U \cup R) \cdot U\,\square \cdot R\,\square)\,\square \\
\quad = \quad \{V\,\square \cdot R\,\square \text{ is a monotype}\} \\
\quad\quad\quad \{(T \cdot X)\,\square = T\,\square \cdot X \text{ provided } X \text{ is a monotype}\} \\
\quad\quad\quad \{X \cdot Y = Y \cdot X \text{ provided } X \text{ and } Y \text{ are monotypes}\} \\
\quad\quad (V \cup R) \cdot (U \cup R)\,\square \cdot V\,\square \cdot R\,\square \cdot U\,\square \cdot R\,\square \\
\quad = \quad \{X \cdot Y = X \Leftarrow X \subseteq Y \subseteq I\} \\
\quad\quad (V \cup R) \cdot U\,\square \cdot R\,\square \\
\quad = \quad \{\text{Distributivity}\} \\
\quad\quad (V \cdot U\,\square \cup R) \cdot U\,\square \cdot R\,\square \\
\quad \subseteq \quad \{V \cdot U\,\square \cup R \subseteq U \cup R \Leftarrow V \cdot U\,\square \subseteq U \Leftarrow U \sqsubseteq V\} \\
\quad\quad (U \cup R) \cdot U\,\square \cdot R\,\square
\end{array}
$$

Finally, we prove that for any relation R and monotypes A, B and C, the function

$$X \mapsto R \cdot (A \cdot (B \cdot X)\,\square \cup C)$$

is $\sqsubseteq$-monotonic:

$$R \cdot (A \cdot (B \cdot U)\,\square \cup C) \sqsubseteq R \cdot (A \cdot (B \cdot V)\,\square \cup C) \Leftarrow U \sqsubseteq V$$

From the monotonicity of $\cdot$, $\cup$ and $\square$ w.r.t. $\subseteq$, we have that

$$R \cdot (A \cdot (B \cdot U)^{\square} \cup C) \subseteq R \cdot (A \cdot (B \cdot V)^{\square} \cup C) \Leftarrow U \subseteq V$$
$$\Leftarrow U \sqsubseteq V$$

holds. Assume that $U \sqsubseteq V$, we check that

$$R \cdot (A \cdot (B \cdot V)^{\square} \cup C) \cdot (R \cdot (A \cdot (B \cdot U)^{\square} \cup C))^{\square}$$
$$\subseteq R \cdot (A \cdot (B \cdot U)^{\square} \cup C)$$

as follows.

$$\begin{aligned}
& R \cdot (A \cdot (B \cdot V)^{\square} \cup C) \cdot (R \cdot (A \cdot (B \cdot U)^{\square} \cup C))^{\square} \\
= \quad & \{(A \cdot (B \cdot U)^{\square} \cup C) \text{ is a monotype}\} \\
& \{(T \cdot X)^{\square} = T^{\square} \cdot X \text{ provided } X \text{ is a monotype}\} \\
& \{X \cdot Y = Y \cdot X \text{ provided } X \text{ and } Y \text{ are monotypes}\} \\
& \{R \cdot R^{\square} = R\} \\
& R \cdot (A \cdot (B \cdot V)^{\square} \cup C) \cdot (A \cdot (B \cdot U)^{\square} \cup C) \\
= \quad & \{\text{Distributivity}\} \\
& \{X \cdot Y \subseteq Y \text{ provided } X \text{ and } Y \text{ are monotypes}\} \\
& \{(B \cdot U)^{\square} \subseteq (B \cdot V)^{\square} \Leftarrow U \subseteq V\} \\
& \{X \cdot Y = X \Leftarrow X \subseteq Y \subseteq I\} \\
& R \cdot (A \cdot (B \cdot U)^{\square} \cup C)
\end{aligned}$$

Proof of Theorem 3:

We prove it by induction on the structure of e as follows.

1. $e = d$: (functional expression)
   The well-definedness is obvious:

$$\begin{aligned}
& \mathcal{A}[\![d]\!](\sigma' + \langle f \mapsto X \rangle) \text{ is well defined} \\
\equiv \quad & \{\text{Def. (a1)}\} \\
& \mathcal{I}[\![d]\!] \text{ is well defined} \\
\Leftarrow \quad & \{\text{Our assumption for } \mathcal{I}[\![_]\!]\} \\
& \mathit{true}
\end{aligned}$$

   The monotonicity follows from that $X \mapsto \mathcal{A}[\![d]\!](\sigma' + \langle f \mapsto X \rangle)$ is defined to be a constant function on X.
2. $e = f$:
   The well-definedness is as follows.

$$\begin{aligned}
& \mathcal{A}[\![f]\!](\sigma' + \langle f \mapsto X \rangle) \text{ is well defined} \\
\equiv \quad & \{\text{Def. (a2)}\} \\
& X \text{ is well defined} \\
\Leftarrow \quad & \{\text{In environments, } X \text{ is assumed to be a relation}\} \\
& \mathit{true}
\end{aligned}$$

   The monotonicity follows from that $X \mapsto \mathcal{A}[\![f]\!](\sigma' + \langle f \mapsto X \rangle)$ is defined to be the identity function on X.
3. $e = g(\neq f)$: (binding identifier)
   The well-definedness is as follows.

$$\begin{aligned}
& \mathcal{A}[\![g]\!](\sigma' + \langle f \mapsto X \rangle) \text{ is well defined} \\
= \quad & \{\text{Def. (a1)}\} \\
& \sigma'[\![g]\!] \text{ is well defined}
\end{aligned}$$

$\Leftarrow$ {In environments, $\sigma[\![g]\!]$ is assumed to be a relation}
true

The monotonicity follows from that $X \mapsto \mathcal{A}[\![g]\!](\sigma' + \langle f \mapsto X\rangle)$ is defined to be the constant function on X.

4. $e = e_1 \circ e_2$: Follows from the well-definedness and monotonicity of $\cdot$.
5. $e = e_1 \rightarrow e_2, e_3$: Follows from the well-definedness and monotonicity of $\cdot$, $\cup$ and $\sqcup$.
6. $e = e_1 \,][\, e_2$, $e = e_1 \sqcap e_2$, $e = e_1 \blacksquare e_2$: Follows from the well-definedness and monotonicity of $\cdot$, $\cup$ and $\sqcup$.
7. $e = \mu g.e_1$:
The well-definedness is proved as follows.

$\mathcal{A}[\![\mu g.e_1]\!](\sigma' + \langle f \mapsto X\rangle)$ is well defined
$\equiv$ {Def. (a6)}
$\mu_{\subseteq} Y.\mathcal{A}[\![e_1]\!](\sigma' + \langle f \mapsto X\rangle + \langle g \mapsto Y\rangle)$ is well defined
$\Leftarrow$ {Tarski Theorem}
$Y \mapsto \mathcal{A}[\![e_1]\!](\sigma' + \langle f \mapsto X\rangle + \langle g \mapsto Y\rangle)$ is $\subseteq$-monotonic
$\Leftarrow$ {Induction Hypothesis on e_1}
true

The monotonicity is proved as follows.

$\mathcal{A}[\![\mu g.e_1]\!](\sigma' + \langle f \mapsto X\rangle)$ is monotonic on X
$\equiv$ {Def. (a6)}
$\mu_{\subseteq} Y.\mathcal{A}[\![e_1]\!](\sigma' + \langle f \mapsto X\rangle + \langle g \mapsto Y\rangle)$ is monotonic on X
$\Leftarrow$ {Induction Hypothesis on e_1:}
{$\mathcal{A}[\![e_1]\!](\sigma' + \langle f \mapsto X\rangle + \langle g \mapsto Y\rangle)$ is monotonic on both X and Y}
{**Corollary 1**}
true

Proof of Theorem 4:
We prove it by induction on the structure of e as follows.

1. $e = d$: (functional expression)
The well-definedness is obvious:

$\mathcal{D}[\![d]\!]\sigma(\rho' + \langle f \mapsto X\rangle)$ is well defined
$\equiv$ {Def. (d1)}
$\mathcal{I}[\![d]\!]$ is well defined
$\Leftarrow$ {Our assumption for $\mathcal{I}[\![\_]\!]$}
true

The monotonicity follows from that $X \mapsto \mathcal{D}[\![d]\!]\sigma(\rho' + \langle f \mapsto X\rangle)$ is defined to be a constant function on X.

2. $e = f$:
The well-definedness is as follows.

$\mathcal{D}[\![f]\!]\sigma(\rho' + \langle f \mapsto X\rangle)$ is well defined
$\equiv$ {Def. (d2)}
X is well defined

$\Leftarrow$ {In environments, X is assumed to be a relation}
true

The monotonicity follows from that $X \mapsto \mathcal{D}[\![f]\!]\sigma(\rho' + \langle f \mapsto X\rangle)$ is defined to be the identity function on X.

3. $e = g(\neq f)$: (binding identifier)
The well-definedness is as follows.

$\mathcal{D}[\![g]\!]\sigma(\rho' + \langle f \mapsto X\rangle)$ is well defined
$=$ {Def. (d2)}
$\rho'[\![g]\!]$ is well defined
$\Leftarrow$ {In environments, $\rho[\![g]\!]$ is assumed to be a relation}
true

The monotonicity follows from that $X \mapsto \mathcal{D}[\![g]\!]\sigma(\rho' + \langle f \mapsto X\rangle)$ is defined to be the constant function on X.

4. $e = e_1 \circ e_2$: Follows from the well-definedness and monotonicity of $\odot$ (**Lemma 2**).
5. $e = e_1 \rightarrow e_2, e_3$: Follows from the well-definedness of $\cdot$ and $\square$ and **Lemma 2**.
6. $e = e_1][e_2$, $e = e_1 [] e_2$, $e = e_1 \blacksquare e_2$: Follows from the well-definedness of $\cdot$ and $\cup$ and the angelic aspect (**Theorem 3**) and **Lemma 2**.
7. $e = \mu g.e_1$:
The well-definedness is proved as follows.

$\mathcal{D}[\![\mu g.e_1]\!]\sigma(\rho' + \langle f \mapsto X\rangle)$ is well defined
$\equiv$ {Def. (d5)}
$\mu_{\sqsubseteq} Y.\mathcal{D}[\![e_1]\!]\sigma(\rho' + \langle f \mapsto X\rangle + \langle g \mapsto Y\rangle)$ is well defined
$\Leftarrow$ {Tarski Theorem}
$Y \mapsto \mathcal{D}[\![e_1]\!]\sigma(\rho' + \langle f \mapsto X\rangle + \langle g \mapsto Y\rangle)$ is $\sqsubseteq$-monotonic
$\Leftarrow$ {Induction Hypothesis on e_1}
true

The monotonicity is proved as follows.

$\mathcal{D}[\![\mu g.e_1]\!]\sigma(\rho' + \langle f \mapsto X\rangle)$ is monotonic on X
$\equiv$ {Def. (d6)}
$\mu_{\sqsubseteq} Y.\mathcal{D}[\![e_1]\!]\sigma(\rho' + \langle f \mapsto X\rangle + \langle g \mapsto Y\rangle)$ is monotonic on X
$\Leftarrow$ {Induction Hypothesis on e_1:}
{$\mathcal{D}[\![e_1]\!]\sigma(\rho' + \langle f \mapsto X\rangle + \langle g \mapsto Y\rangle)$ is monotonic on both X and Y}
{**Corollary 1**}
true

Proof of Theorem 5:
We prove it by induction on the structure of e as follows.

1. $e = d$: (functional expression)

$\mathcal{D}[\![d]\!]\sigma\rho$
$=$ {Def. (d1)}
$\mathcal{I}[d]$
$=$ {Def. (a1)}

$\mathcal{A}[\![d]\!]\sigma$

2. $e = f$: (binding identifier)

$$
\begin{array}{ll}
& \mathcal{D}[\![f]\!]\sigma\rho \\
= & \quad \{\text{Def. (d2)}\} \\
& \rho[\![f]\!] \\
\sqsubseteq & \quad \{\text{Hypothesis}\} \\
& \sigma[\![f]\!] \\
= & \quad \{\text{(Def. a2)}\} \\
& \mathcal{A}[\![f]\!]\sigma
\end{array}
$$

3. $e = e_1\,][\,e_2$:

$$
\begin{array}{ll}
& \mathcal{D}[\![e_1\,][\,e_2]\!]\sigma\rho \\
= & \quad \{\text{Def. } S_1 = \mathcal{D}[\![e_1]\!]\sigma\rho\} \\
& \quad \{\text{Def. } S_2 = \mathcal{D}[\![e_2]\!]\sigma\rho\} \\
& \quad \{\text{Def. (d5)}\} \\
& (S_1 \cup S_2)\cdot S_1\square \cdot S_2\square \\
\sqsubseteq & \quad \{\text{Def. } R_1 = \mathcal{A}[\![e_1]\!]\sigma\} \\
& \quad \{\text{Def. } R_2 = \mathcal{A}[\![e_2]\!]\sigma\} \\
& \quad \{\text{Induction Hypothesis: } S_1 \sqsubseteq R_1 \text{ and } S_2 \sqsubseteq R_2\} \\
& \quad \{\textbf{Lemma 2}\} \\
& (R_1 \cup R_2)\cdot R_1\square \cdot R_2\square \\
\sqsubseteq & \quad \{R_1\square \cdot R_2\square \text{ is a monotype}\} \\
& R_1 \cup R_2 \\
= & \quad \{\text{Def. (a5)}\} \\
& \mathcal{A}[\![e_1\,][\,e_2]\!]\sigma
\end{array}
$$

4. $e = e_1\,[]\,e_2$:

$$
\begin{array}{ll}
& \mathcal{D}[\![e_1\,[]\,e_2]\!]\sigma\rho \\
= & \quad \{\text{Def. } S_1 = \mathcal{D}[\![e_1]\!]\sigma\rho\} \\
& \quad \{\text{Def. } S_2 = \mathcal{D}[\![e_2]\!]\sigma\rho\} \\
& \quad \{\text{Def. (d5)}\} \\
& (\mathcal{A}[\![e_1\,[]\,e_2]\!]\sigma)\cdot(S_1\square \cup S_2\square) \\
\sqsubseteq & \quad \{S_1\square \cup S_2\square \text{ is a monotype}\} \\
& \mathcal{A}[\![e_1\,][\,e_2]\!]\sigma
\end{array}
$$

5. $e = e_1 \blacksquare e_2$:

$$
\begin{array}{ll}
& \mathcal{D}[\![e_1 \blacksquare e_2]\!]\sigma\rho \\
= & \quad \{\text{Def. } S_1 = \mathcal{D}[\![e_1]\!]\sigma\rho\} \\
& \quad \{\text{Def. } S_2 = \mathcal{D}[\![e_2]\!]\sigma\rho\} \\
& \quad \{\text{Def. (d5)}\} \\
& (S_1 \cup S_2)\cdot S_1\square \cdot S_2\square \\
\sqsubseteq & \quad \{\text{Def. } R_1 = \mathcal{A}[\![e_1]\!]\sigma\} \\
& \quad \{\text{Def. } R_2 = \mathcal{A}[\![e_2]\!]\sigma\} \\
& \quad \{\text{Induction Hypothesis: } S_1 \sqsubseteq R_1 \text{ and } S_2 \sqsubseteq R_2\} \\
& \quad \{\textbf{Lemma 2}\} \\
& (R_1 \cup R_2)\cdot R_1\square \cdot R_2\square \\
= & \quad \{\text{Def. (a5)}\}
\end{array}
$$

$\mathcal{A}[\![e_1 \blacksquare e_2]\!]\sigma$

6. $e = e_1 \circ e_2$: Similar to the above proof.
7. $e = e_1 \rightarrow e_2, e_3$: Similar to the above proof.
8. $e = \mu g.e_1$:

$$
\begin{aligned}
&\mathcal{D}[\![\mu g.e_1]\!]\sigma\rho \sqsubseteq \mathcal{A}[\![\mu g.e_1]\!]\sigma \\
&\quad\equiv \quad \{\text{Def. (d6)}\} \\
&\quad\quad \mu_{\sqsubseteq} X.\mathcal{D}[\![e_1]\!](\sigma + \langle g \mapsto \mathcal{A}[\![\mu g.e_1]\!]\sigma\rangle)(\rho + \langle g \mapsto X\rangle) \sqsubseteq \mathcal{A}[\![\mu g.e_1]\!]\sigma \\
&\quad\Leftarrow \quad \{\text{Tarski Theorem}\} \\
&\quad\quad \mathcal{D}[\![e_1]\!](\sigma + \langle g \mapsto \mathcal{A}[\![\mu g.e_1]\!]\sigma\rangle)(\rho + \langle g \mapsto \mathcal{A}[\![\mu g.e_1]\!]\sigma\rangle) \\
&\quad\quad \sqsubseteq \mathcal{A}[\![\mu g.e_1]\!]\sigma \\
&\quad\Leftarrow \quad \{\mathcal{A}[\![\mu g.e_1]\!]\sigma \sqsubseteq \mathcal{A}[\![\mu g.e_1]\!]\sigma\} \\
&\quad\quad \{\text{Induction Hypothesis on } e_1\} \\
&\quad\quad \mathcal{A}[\![e_1]\!](\sigma + \langle g \mapsto \mathcal{A}[\![\mu g.e_1]\!]\sigma\rangle) \sqsubseteq \mathcal{A}[\![\mu g.e_1]\!]\sigma \\
&\quad\equiv \quad \{\mathcal{A}[\![\mu g.e_1]\!]\sigma \text{ is a fixed point of } X \mapsto \mathcal{A}[\![e_1]\!](\sigma + \langle g \mapsto X\rangle)\} \\
&\quad\quad \mathit{true}
\end{aligned}
$$

Proofs of property (17)~(21):

The property (17) is straightforward from the fact that $\cdot$, $\cup$ and $X \mapsto X \rhd U$ are $\subseteq$-monotonic.

The property (18) is proved as follows.

$$
\begin{aligned}
&\text{lhs} \\
&= \quad \{\text{Def. of } \odot\} \\
&\quad R_{st}\,\Box \cdot J(X) \rhd R_{st} \\
&= \quad \{\text{Prop. (15, 16); } U \rhd (V \cup W) = (U \rhd V) \cdot U \rhd W\} \\
&\quad (st? \cdot (2 \leq)? \cup \overline{st?} \cdot (3 \geq)?) \cdot \\
&\quad (J(X) \rhd (M_{23} \cdot st? \cdot (2 \leq)?)) \cdot (J(X) \rhd (M_{34} \cdot \overline{st?} \cdot (3 \leq)?)) \\
&= \quad \{B \subseteq I \Rightarrow B \cdot B \rhd (V \cdot B) = B \cdot B \rhd V;\} \\
&\quad\quad \{\text{Distribution; } st? \cdot \overline{st?} = \emptyset;\ U \rhd \emptyset = I\} \\
&\quad st? \cdot (2 \leq)? \cdot J(X) \rhd M_{23} \cup \overline{st?} \cdot (3 \leq)? \cdot J(X) \rhd M_{34} \\
&= \quad \{\text{Def. of } H\} \\
&\quad \text{rhs}
\end{aligned}
$$

The proof of (19) is as follows.

$$
\begin{aligned}
&\text{lhs} \\
&= \quad \{\text{Def. of } \mathcal{D}_0[\![g_{st}]\!]\} \\
&\quad ((emp? \cup \mathcal{A}_0[\![g_{st}]\!] \cdot \mathcal{D}_0[\![g_{st}]\!]\,\Box \cdot \overline{emp?}) \odot R_{st})\,\Box \\
&= \quad \{true \equiv \mathcal{D}_0[\![g_{st}]\!] \sqsubseteq \mathcal{A}_0[\![g_{st}]\!] \Rightarrow \mathcal{A}_0[\![g_{st}]\!] \cdot \mathcal{D}_0[\![g_{st}]\!]\,\Box = \mathcal{D}_0[\![g_{st}]\!]\} \\
&\quad ((emp? \cup \mathcal{D}_0[\![g_{st}]\!] \cdot \overline{emp?}) \odot R_{st})\,\Box \\
&= \quad \{\text{Def. of } J\} \\
&\quad (J(\mathcal{D}_0[\![g_{st}]\!]) \odot R_{st})\,\Box \\
&= \quad \{\text{Prop. (18)}\} \\
&\quad \text{rhs}
\end{aligned}
$$

For any predicate st, we prove the property (20):

$$
\begin{aligned}
&\mathcal{D}_0[\![g_{st}]\!]\,\Box \subseteq A \\
&\quad\equiv \quad \{B \subseteq I\colon\ U\,\Box \subseteq B \equiv U \subseteq U \cdot B \equiv U \sqsubseteq U \cdot B\}
\end{aligned}
$$

$$
\begin{array}{ll}
& \mathcal{D}_0[\![g_{st}]\!] \sqsubseteq \mathcal{D}_0[\![g_{st}]\!] \cdot A \\
\equiv & \{\text{Def. of } \mathcal{D}_0[\![g_{st}]\!]\} \\
& \mu_{\sqsubseteq} X.(emp? \cup \mathcal{A}_0[\![g_{st}]\!] \cdot X \,\square\, \cdot \overline{emp?}) \odot R_{st} \subseteq \mathcal{D}_0[\![g_{st}]\!] \cdot A \\
\Leftarrow & \{\text{Tarski Theorem}\} \\
& (emp? \cup \mathcal{A}_0[\![g_{st}]\!] \cdot (\mathcal{D}_0[\![g_{st}]\!] \cdot A)\,\square\, \cdot \overline{emp?}) \odot R_{st} \sqsubseteq \mathcal{D}_0[\![g_{st}]\!] \cdot A \\
\equiv & \{B \subseteq I\colon (U \cdot B)\,\square = U\,\square \cdot B;\ \text{Distribution}\} \\
& ((emp? \cup \mathcal{A}_0[\![g_{st}]\!] \cdot \mathcal{D}_0[\![g_{st}]\!]\,\square\, \cdot \overline{emp?}) \cdot (emp? \cup A \cdot \overline{emp?})) \odot R_{st} \\
& \sqsubseteq \mathcal{D}_0[\![g_{st}]\!] \cdot A \\
\Leftarrow & \{B \subseteq I\colon U \cdot B = U \odot B;\ \text{Associativity of } \odot;\ \text{Def. of } J\} \\
& (emp? \cup \mathcal{A}_0[\![g_{st}]\!] \cdot \mathcal{D}_0[\![g_{st}]\!]\,\square\, \cdot \overline{emp?}) \odot (J(A) \odot R_{st}) \\
& \sqsubseteq \mathcal{D}_0[\![g_{st}]\!] \cdot A \\
\Leftarrow & \{\odot \text{ is } \sqsubseteq\text{-monotonic; Associativity of } \odot;\ \text{Def. of } \mathcal{D}_0[\![g_{st}]\!]\} \\
& J(A) \odot R_{st} \sqsubseteq R_{st} \odot A \\
\equiv & \{B \subseteq I\colon B \odot U = U \cdot (B \rhd U),\ U \odot B = U \cdot B\} \\
& R_{st} \cdot J(A) \rhd R_{st} \sqsubseteq R_{st} \cdot A \\
\Leftarrow & \{B, C \subseteq I\colon U \cdot B \sqsubseteq U \cdot C \Leftarrow B \subseteq C;\ \text{Def. of } \odot\} \\
& (J(A) \odot R_{st})\,\square \subseteq A \\
\equiv & \{\text{Prop. (18)}\} \\
& H(A, st?, \overline{st?}) \subseteq A \\
\Leftarrow & \{st?, \overline{st?} \subseteq I;\ \text{Monotonicity of } H\} \\
& H(A, I, I) \subseteq A
\end{array}
$$

The property (21) can be proved by checking that K is a fixed point of $X \mapsto H(X, I, I)$ and it is the least one.

$$
\begin{array}{ll}
& H(K, I, I) \\
= & \{\text{Def. of } H\} \\
& (2 \leq)? \cdot J(K) \rhd M_{23} \cup (3 \leq)? \cdot J(K) \rhd M_{34} \\
\subseteq & \{J(K) \rhd M_{23} \subseteq P\} \\
& \{J(K) \rhd M_{34} \subseteq Q\} \\
& P \cup Q \\
= & \{\text{Prop. of } K\} \\
& K
\end{array}
$$

Finally, we prove that K is the least fixed point. Because for any n,

$$H^0(\emptyset, I, I) = \emptyset \subseteq \mu_{\subseteq} X.H(X, I, I)$$

and

$$
\begin{array}{ll}
& H^{n+1}(\emptyset, I, I) \\
= & \{\text{Def. of } H^n\} \\
& H(H^n(\emptyset, I, I), I, I) \\
\subseteq & \{\text{Induction Hypothesis, Monotonicity of } H\} \\
& H(\mu_{\subseteq} X.H(X, I, I), I, I) \\
\subseteq & \{\text{Fixed Point}\} \\
& \mu_{\subseteq} X.H(X, I, I)
\end{array}
$$

We know that $\mu_{\subseteq} X.H(X, I, I)$ is an upper bound of the chain $\{H^n(\emptyset, I, I)\}$. Therefore, $\mu_{\subseteq} X.H(X, I, I)$ is larger than the least upper bound of the chain $\{H^n(\emptyset, I, I)\}$:

$$K = \bigcup_{n \geq 0} H^n(\emptyset, I, I) \subseteq \mu_{\subseteq} X.H(X, I, I)$$

of the complete lattice $(\mathcal{R}, \subseteq)$. Combined with the fact that K is a fixed point, we have the result.

Liangwei Xu: His research interests are computational model, program transformation and derivation methodology. He received the B. E. degree from Shanghai JiaoTong University in 1982 and the M.E. degree from University of Tokyo in 1992. He currently joins Mathematical Systems Institute Inc.

Masato Takeichi, Dr. Eng.: He is a Professor of Department of Mathematical Engineering. Graduate School of Engineering, University of Tokyo. His research interests are functional programming, language implementation and constructive algorithmics.

Hideya Iwasaki, Dr. Eng.: He is an Associate Professor of Faculty of Technology, Tokyo University of Agriculture and Technology. He received the M.E. degree in 1985, the Dr. Eng. degree in 1988 from University of Tokyo. His research interests are list processing languages, functional languages, parallel processing, and constructive algorithmics.