Acta Cybernetica **17** (2006) 533–555.

Demonic Fixed Points

Fairouz Tchier*

Abstract

We deal with a relational model for the demonic semantics of programs. The demonic semantics of a while loop is given as a fixed point of a function involving the demonic operators. This motivates us to investigate the fixed points of these functions. We give the expression of the greatest fixed point with respect to the demonic ordering (*demonic inclusion*) of the semantic function. We prove that this greatest fixed coincides with the least fixed point with respect to the usual ordering (*angelic inclusion*) of the same function. This is followed by an example of application.

Keywords: demonic fixed points, demonic functions, while loops, relational demonic semantics.

1 Introduction

We use relations to define the input-output semantics of nondeterministic programs. The relational operators $\cup$ and $;$ have been used for many years to define the so-called *angelic semantics*, which assumes that a program goes right when there is a possibility to go right. On the other hand, the demonic operators $\sqcup$ and $\Box$ (to be introduced below) do the opposite: if there is a possibility to go wrong, a program whose semantics is given by these operators goes wrong; it is the *demonic semantics* of nondeterministic programs.

The concept of fixed points has received increasing attention from researchers in a wide range of scientific areas, especially in computer science. Many important notions in demonic relational semantics are given as fixed points of some monotonic functions involving the demonic operators [4, 16, 25, 39, 44]. In this paper, we concentrate on while loops. We will present some relevant results about the fixed points of the function $f(X) = Q \sqcap P \Box X$ (these operators will be defined later). By considering certain conditions on the domains of P and Q, one gets the demonic semantics we have assigned to while loops in previous papers [22, 47, 48, 49, 50]. Other similar definitions of while loops can be found in [28, 38, 43, 51, 52, 53]. Our results can be applied also to the program verification and construction; while there

*Mathematics department, King Saud University, P.O.Box 22452, Riyadh 11495, Saudi Arabia, E-mail: `ftchier@hotmail.com`

is no systematic way to calculate the relational abstraction of a while loop directly from the definition it is possible to check the correctness of any candidate abstraction by theorem 30. For deterministic programs, Mills has described a checking method [34, 35].

The approach to demonic input-output relation presented here is not the only possible one. In [28, 29, 30], the infinite looping has been treated by adding to the state space a fictitious state $\perp$ to denote nontermination. In [9, 24, 32, 40], the demonic input-output relation is given as a pair (relation,set). The relation describes the input-output behavior of the program, whereas the set component represents the domain of guaranteed termination. In [19, 17, 18, 31], they abstract from relational semantics to the setting of modal Kleene algebras, an extension of Kozen's Kleene algebra with tests.

We note that the preponderant formalism employed until now for the description of demonic input-output relation is the wp-calculus. For more details see [1, 3, 23].

The rest of the paper is organized as follows. In Section 2, we present our mathematical tool, namely relation algebra [13, 42, 45]. First, we recall the basic laws (Subsection 2.1). In Section 3, we present notions related to fixed points followed by a description of our refinement ordering (Section 4) and finally in Section 5, we give our main results. In Section 6, we give an application.

2 Relation algebras

2.1 Definition and basic laws

Our mathematical tool is abstract relation algebra [13, 42, 45], which we now introduce.

Definition 1. *A* (homogeneous) relation algebra *is a structure* $(\mathcal{R}, \cup, \cap, ^-, \breve{}, ;)$ *over a non-empty set* $\mathcal{R}$ *of elements, called* relations*. The following conditions are satisfied.*

- $(\mathcal{R}, \cup, \cap, ^-)$ *is a complete Boolean algebra, with* zero *element* $\emptyset$*,* universal *element* L *and ordering* $\subseteq$*.*
- Composition*, denoted by* $(;)$*, is associative and has an identity element, denoted by* I*.*
- *The Schröder rule is satisfied:* $P;Q \subseteq R \Leftrightarrow \breve{P};\overline{R} \subseteq \overline{Q} \Leftrightarrow \overline{R};\breve{Q} \subseteq \overline{P}$.
- $L;R;L = L \Leftrightarrow R \neq \emptyset$ *(Tarski rule).*

The relation $\breve{R}$ is called the *converse* of R. The standard model of the above axioms is the set $\wp(S \times S)$ of all subsets of $S \times S$. In this model, $\cup, \cap, ^-$ are the usual *union, intersection* and *complement*, respectively; the relation $\emptyset$ is the empty relation, the universal relation is $L = S \times S$ and the identity relation is $I = \{(s, s') \mid s' = s\}$. Converse and composition are defined by

$$\breve{R} = \{(s, s') \mid (s', s) \in R\} \quad \text{and} \quad Q;R = \{(s, s') \mid \exists s'' : (s, s'') \in Q \wedge (s'', s') \in R\}.$$

The precedence of the relational operators from highest to lowest is the following: $\overline{}$ and $\breve{}$ bind equally, followed by $;$, then by $\cap$, and finally by $\cup$. From now on, the composition operator symbol $;$ will be omitted (that is, we write QR for $Q;R$). From Definition 1, the usual rules of the calculus of relations can be derived (see, e.g., [9, 13, 42]). We assume these rules to be known and simply recall a few of them.

Theorem 2. *Let P, Q, R be relations. Then,*

(a) $\overline{Q\cup R} = \overline{Q}\cap\overline{R}$,

(b) $\overline{Q\cap R} = \overline{Q}\cup\overline{R}$,

(c) $Q\cap R\cup\overline{R} = Q\cup\overline{R}$,

(d) $P\cap Q\subseteq R \quad\Leftrightarrow\quad P\subseteq\overline{Q}\cup R$,

(e) $Q\subseteq R \quad\Leftrightarrow\quad \overline{R}\subseteq\overline{Q}$,

(f) $P(Q\cap R)\subseteq PQ\cap PR$,

(g) $(P\cap Q)R\subseteq PR\cap QR$,

(h) $P(Q\cup R) = PQ\cup PR$,

(i) $(P\cup Q)R = PR\cup QR$,

(j) $Q\subseteq R \quad\Rightarrow\quad PQ\subseteq PR$,

(k) $Q\subseteq R \quad\Rightarrow\quad QP\subseteq RP$.

(l) $\overline{RL}L = \overline{RL}$,

(m) $PQ\cap R\subseteq P(Q\cap\breve{P}R)$,

(n) $(P\cap QL)R = PR\cap QL$,

(o) $(\bigcap_{i\in X} R_iL)L = \bigcap_{i\in X} R_iL$.

We now give a definition of various properties of relations.

Definition 3. *A relation R is* functional *iff $\breve{R}R\subseteq I$. A relation v is a vector [42] iff $v = vL$.*

In the standard model, a relation R on a set S is functional iff $(s,s')\in R\wedge(s,s'')\in R\Rightarrow s' = s''$. A vector is a relation of the form $T\times S$, where $T\subseteq S$. A vector can also be viewed as a point set or a predicate. For example, if $S=\{0,1,2\}$ and $T=\{0,1\}$, then $t := T\times S = \{(0,0),(0,1),(0,2),(1,0),(1,1),(1,2)\}$ is a vector that corresponds to the point set T. For any relation R, the relation RL is a vector that characterizes the domain of R. For instance, with $R := \{(0,1),(0,2),(2,1)\}$ (on set $S=\{0,1,2\}$), we obtain $RL = \{(0,0),(0,1),(0,2),(2,0),(2,1),(2,2)\}$; this vector indeed characterizes the domain of R, which is $\{0,2\}$.

2.2 Relative implication

In our work we need to define an operator called *relative implication*. In previous work, we used the monotype and residual operators see [49, 47, 48]

Definition 4. *A binary operator $\triangleleft$, called* relative implication *[50], is defined as follows :*

$$Q\triangleleft R := \overline{Q\overline{R}}.$$

This operator has a dual operator (Definition 10), $\rhd$, given by :

$$Q \rhd R := \overline{\overline{Q}R}.$$

The most interesting case is when the right argument is a vector RL, in other words $Q \lhd RL$. If $x.R$ denotes the set of the images of x by R, then $x \in \mathrm{dom}(Q \lhd RL) \Leftrightarrow x.Q \subseteq \mathrm{dom}(R)$.

The operators $\lhd$ and $\rhd$ bind equally but less than (;) and more than $\cap$ and $\cup$. In the next lemma we give some interesting properties satisfied by the operator $\lhd$. The properties of $\rhd$ can be obtained by dualization of those of the operator $\lhd$ [50].

Lemma 5. *Let P, Q and R be relations.*

(a) $\overline{Q \lhd R} = Q\overline{R}$,

(b) $\overline{Q \rhd R} = \overline{Q}R$,

(c) $P \lhd Q \cap P \lhd R = P \lhd (Q \cap R)$,

(d) $P \lhd R \cap Q \lhd R = (P \cup Q) \lhd R$,

(e) $PQ \lhd R = P \lhd (Q \lhd R)$,

(f) $PQ \cap P \lhd R = P(Q \cup \overline{R}) \cap P \lhd R$,

(g) $P \lhd Q \subseteq PQ \cup \overline{PL}$,

(h) $P \subseteq Q \Rightarrow Q \lhd R \subseteq P \lhd R$,

(i) $P \subseteq Q \Rightarrow R \lhd P \subseteq R \lhd Q$

(j) Q *vector* $\Rightarrow P \lhd Q$ *vector.*

We note that the properties (c) and (e) are similar to those of the logical operators →, $\wedge$ and $\vee$. For example, the property (e) corresponds to $(P \wedge Q \rightarrow R) \leftrightarrow (P \rightarrow (Q \rightarrow R))$.

3 Fixed points

Let f be a monotonic function with respect to $\subseteq$. The least fixed point of f is $\bigcap\{X \mid f(X) = X\}$. Similarly, $\bigcup\{X \mid f(X) = X\}$ is the greatest fixed point of f. Because we assume our relation algebra to be complete (Definition 1), least and greatest fixed points of monotonic functions exist. We will denote the least fixed point of the function $f(X) := E(x)$, where E is some relational expression, by μf or by $\mu(X : E(X))$, when it is desired not to introduce a function name. Similarly,

νf and $\nu(X : E(X))$ denote the greatest fixed point of f. The following properties of fixed points are used below:

$$\begin{array}{lllll} \text{(a)} & \mu f & = \bigcap\{X|f(X) = X\} & = & \bigcap\{X|f(X) \subseteq X\}, \\ \text{(b)} & \nu f & = \bigcup\{X|f(X) = X\} & = & \bigcup\{X|X \subseteq f(X)\}, \\ \text{(c)} & \mu f \subseteq \nu f, & & & \\ \text{(d)} & f(Y) \subseteq Y & \Rightarrow & \mu f \subseteq Y, & \\ \text{(e)} & Y \subseteq f(Y) & \Rightarrow & Y \subseteq \nu f. & \end{array} \qquad (1)$$

We need some auxiliary notions.

Definition 6. *A function f between complete lattices is* strict *if $f(\emptyset) = \emptyset$ and* co-strict *if $f(L) = L$. Further, f is called* continuous *if for every chain $\mathcal{C}$ one has $f(\bigcup \mathcal{C}) = \bigcup f(\mathcal{C})$, and* co-continuous *if for every chain $\mathcal{C}$ one has $f(\bigcap \mathcal{C}) = \bigcup f(\mathcal{C})$.*

Every universally disjunctive function is continuous and strict; every universally conjunctive function is co-continuous and co-strict. Moreover, every continuous or co-continuous function is monotonic.

Theorem 7. *[44] (Knaster-Tarski) Every monotonic endofunction on a complete lattice has a least fixed point μf, which is equal to its least pre-fixed point and a greatest fixed point $\nu(f)$ which is equal to its greatest post-fixed point. If f is continuous, then $\mu(f) = \bigcup\{f^i(\emptyset) : i \in \mathbb{N}\}$. If f is co-continuous, then $\nu(f) = \bigcap\{f^i(L) : i \in \mathbb{N}\}$.*

Here,

$$\begin{array}{rcl} f^0(x) & = & x \ , \\ f^{i+1}(x) & = & f(f^i(x)) \ . \end{array}$$

For more details about these notions see [12, 15]. The comparison of fixed points is sometimes very useful to compare the program semantics. The next proposition will present some results in this direction.

Proposition 8. *Let $(X, \leq)$ be an ordered set f and g endofunctions. Let also the relation $\ll$ on the set of endofunctions on X, defined as follows :*

$$f \ll g \Leftrightarrow (\forall x : x \in X : f(x) \leq g(x)).$$

We have the following properties ($+$ is a monotonic binary operation on X) :

$$\begin{array}{lll} \textit{(a)} & \mu \textit{ monotonic} & f \ll g \ \Rightarrow \ \mu(f) \leq \mu(g), \\ \textit{(b)} & \textit{permutation law} & \mu(f\,;g) = f(\mu(g\,;f)), \\ \textit{(c)} & \textit{diagonal law} & \mu(x \mapsto x + x) = \mu(x \mapsto \mu(y \mapsto x + y)), \\ \textit{(d)} & \mu\textit{-fusion law} & f\,;g = g\,;h \ \Rightarrow \ \mu(f) = g(\mu(h)), \\ & & (g \textit{ is continuous and strict}). \end{array}$$

3.1 Transitive reflexive closure

Another operation that occurs in the definition of the while program semantics is the *reflexive transitive closure*. The *reflexive transitive closure* is a unary operation denoted $*$ and defined for every relation R by :

$$R^* = \mu(X \mapsto I \cup RX). \tag{2}$$

The operation $*$ is defined as a least fixed point ; by the monotonicity of $\cup$ and of (;), and the Knaster-Tarski Theorem (7) this operation is well defined and it satisfies,

$$R^* = I \cup RR^* = I \cup R^*R. \tag{3}$$

The unary operations $*, \breve{}$ and $\bar{}$ bind equally. We can also define a similar operation to $*$, called *transitive closure*, denoted $+$, and defined for every relation R by :

$$R^+ = \mu(X \mapsto R \cup RX). \tag{4}$$

and,

$$\begin{array}{ll} \text{(a)} & R^+ = R^*R = RR^* = R \cup RR^*, \\ \text{(b)} & R^* = I \cup R^+. \end{array} \tag{5}$$

The operations $*$ and $+$ bind equally. The operation $*$ satisfies also

$$R^* = \bigcup_{i \geq 0} R^i, \tag{6}$$

where $R^0 = I$ and $R^{i+1} = RR^i$.

We give some properties of the operation $*$. The properties of the operation $+$ are easily deduced from the equations 5 and of the properties of $*$.

$$R^*Q = \mu(X \mapsto Q \cup RX), \tag{7}$$

Proposition 9. *Let P, Q and R be relations. We have,*

- $PQP = PQ \Rightarrow (PQ)^*P = PQ^*$,
- $QR = \emptyset \Rightarrow (Q \cup R)^* = R^*Q^*$,
- $RQ = \emptyset$ *and* $QR = \emptyset \Rightarrow (R \cup Q)^* = R^* \cup Q^*$,

We need the notion of *dual* function.

Definition 10. *Let f be an endofunction on a Boolean lattice. The* dual *function of f is $f^{\#}(x) := \overline{f(\overline{x})}$.*

The next Lemma states the relationship between the fixed points of a function on a Boolean lattice and those of its dual function.

Lemma 11. *Let f be an endofunction on a Boolean lattice and $f^{\#}$ its dual function. If x is a fixed point of f, then*

(a) $\overline{x}$ is a fixed point of $f^{\#}$,

(b) $\nu(f^{\#}) = \overline{\mu(f)}$,

(c) $\mu(f^{\#}) = \overline{\nu(f)}$.

3.2 The initial part of a relation

In the sequel, we describe notions that are useful for the description of the set of initial states of a program for which termination is guaranteed. These notions are the *initial part* of a relation and *progressive finiteness.* The *initial part* of a relation R, denoted $\mathcal{L}(R)$, is the vector characterizing the set of points s_0 such that there is no infinite chain $s_0, s_1, s_2, \ldots$, with $(s_i, s_{i+1}) \in R$, for all $i \geq 0$. The algebraic definition is

Definition 12. *[42] The* initial part *of a relation R, denoted $\mathcal{L}(R)$, is given by :*

$$\mathcal{L}(R) := \bigcap\{x \mid R \triangleleft x = x\},$$

where x takes its value in the set of the vectors (by Theorem 2(o), $\mathcal{L}(R)$ is a vector). (see [41, 42]); in other words, $\mathcal{L}(R)$ is the least fixed point of the $\subseteq$-monotonic function $g(x) := R \triangleleft x$, where x is a vector (the least fixed point of g exists since the set of vectors is also a complete lattice [42]). A relation R is said to be progressively finite iff $\mathcal{L}(R) = L$, in other words if there is no infinite path by R. Progressive finiteness of a relation R is the same as well-foundedness of $\breve{R}$. (Mnemonics : $\mathcal{L}$ for loop because, in the program semantics, $\mathcal{L}(R)$ represents the set of states from which no infinite loop is possible.)

We find in [50] an equivalent definition of $\mathcal{L}(R)$ on the set of relations instead of vectors. This definition is given in the next proposition

Proposition 13. *Let R be a relation.*

$$\begin{array}{lll}(a) & \mathcal{L}(R) & = \bigcap\{X \mid R \triangleleft X = X\} \\ & & = \bigcap\{X \mid R \triangleleft X \subseteq X\} \\ & & = \bigcap\{X \mid R\overline{X} = \overline{X}\} \\ & & = \mu(X \mapsto R \triangleleft X)\end{array}$$

and

$$\begin{array}{lll}(b) & \overline{\mathcal{L}(R)} & = \bigcup\{X \mid X = RX\} \\ & & = \bigcup\{X \mid X \subseteq RX\} \\ & & = \nu(X \mapsto RX).\end{array}$$

Proof. These results can be easily deduced from the Equations 1, 13(a), of the definition of $\triangleleft$ and certain Boolean laws. □

Let us give the formal definition of a progressively finite relation.

Definition 14. *A relation R is called* progressively finite *[42] iff $\mathcal{L}(R) = L$. In an omega algebra [14] the complement of the initial part is known as the* infinite iteration.

By using the results of Proposition 13, we have :

$$R \text{ is progressively finite } \;\Leftrightarrow\; \overline{\mathcal{L}(R)} = \emptyset \;\Leftrightarrow\; (\forall X : X \subseteq RX \;\Rightarrow\; X = \emptyset). \quad (8)$$

The next proposition presents some properties of the initial part of a relation [50]. The properties (a), (b) and (c) can be found also in [42].

Proposition 15. *Let Q and R be relations.*

(a) $\mathcal{L}(R) \subseteq R^*\overline{RL}$,

(b) $\bigcup_{i\geq 0} \overline{R^i L} \subseteq \mathcal{L}(R)$,

(c) R *deterministic* $\Rightarrow \mathcal{L}(R) = R^*\overline{RL}$,

(d) $Q \subseteq R \Rightarrow \mathcal{L}(R) \subseteq \mathcal{L}(Q)$,

(e) $R \lhd \mathcal{L}(R) = \mathcal{L}(R)$ *(equivalent to* $R\overline{\mathcal{L}(R)} = \overline{\mathcal{L}(R)}$*)*,

(f) $\mathcal{L}(R) = \mathcal{L}(R^+)$,

(g) $R^* \lhd \mathcal{L}(R) = R^+ \lhd \mathcal{L}(R) = \mathcal{L}(R)$ *(equivalent to* $R^*\overline{\mathcal{L}(R)} = R^+\overline{\mathcal{L}(R)} = \overline{\mathcal{L}(R)}$*)*,

(h) Q *progressively finite* $\Rightarrow Q \cap R$ *progressively finite,*

(i) $R \cap \mathcal{L}(R)$ *is progressively finite.*

Proof. See [50].

4 A demonic refinement ordering

We now define the refinement ordering we will be using in the sequel. This ordering induces a complete join semilattice, called a *demonic semilattice*. The associated operations are demonic join ($\sqcup$), demonic meet ($\sqcap$) and demonic composition ($\Box$). We give the definitions and needed properties of these operations. For more details on relational demonic semantics and demonic operators, see [6, 7, 8, 9, 21, 22, 50].

Definition 16. *We say that a relation Q refines a relation R [33], denoted by $Q \sqsubseteq R$, iff*

$$Q \cap RL \subseteq R \;\;\wedge\;\; RL \subseteq QL.$$

Thus, for instance,

$$\begin{pmatrix}1 & 0 & 1\\ 1 & 1 & 0\\ 1 & 0 & 0\end{pmatrix} \sqsubseteq \begin{pmatrix}1 & 0 & 1\\ 0 & 0 & 0\\ 1 & 0 & 0\end{pmatrix}, \text{ but } \begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix} \not\sqsubseteq \begin{pmatrix}1 & 1\\ 1 & 0\end{pmatrix}$$

(these Boolean matrices represent relations over sets by the well-known correspondence).

Proposition 17. *Let Q and R be relations. We have,*

(a) The greatest lower bound (wrt $\sqsubseteq$) of relations Q and R is

$$Q \sqcup R = (Q \cup R) \cap QL \cap RL.$$

And here is an example of this operation:

$$\begin{pmatrix}1 & 1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} \sqcup \begin{pmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 1 & 1\end{pmatrix} = \begin{pmatrix}0 & 0 & 0\\ 1 & 1 & 0\\ 0 & 0 & 0\end{pmatrix}.$$

This operation corresponds to a demonic non-deterministic choice, since the possibility of failure (row 3 of the first matrix or row 1 of the second) is reflected in the result. For the middle row, failure is not possible, and the set of allowed results is the union of the results of the two operands.

(b) If Q and R satisfy the condition $QL \cap RL = (Q \cap R)L$, their least upper bound is

$$Q \sqcap R = (Q \cap R) \cup \overline{QL} \cap R \cup Q \cap \overline{RL},$$

otherwise, the least upper bound does not exist see [11, 22].

The existence condition simply means that on the intersection of their domains, Q and R have to agree for at least one value.

Figure 1 shows the general structure of $(\mathcal{A}_{\cup R}, \sqsubseteq)$, for a given R. It is shown in [22] that it is a complete join semilattice. Let f be a monotonic function (wrt $\sqsubseteq$) having at least one fixed point. Because $(\mathcal{A}_{\cup R}, \sqsubseteq)$ is a complete join semilattice, the following properties of fixed points can be transferred from Equations 1.

$$\begin{array}{lllll} \text{(a)} & \nu f & = \bigsqcup\{X \mid f(X) = X\} & = & \bigsqcup\{X \mid X \sqsubseteq f(X)\}, \\ \text{(b)} & Y \sqsubseteq f(Y) & \Rightarrow \; Y \sqsubseteq \nu f. & & \end{array} \tag{9}$$

In the sequel we will introduce some operation, related to the usual relational composition, the so-called *demonic composition.* Its definition is

Definition 18. $Q \Box R := QR \cap Q \lhd RL.$

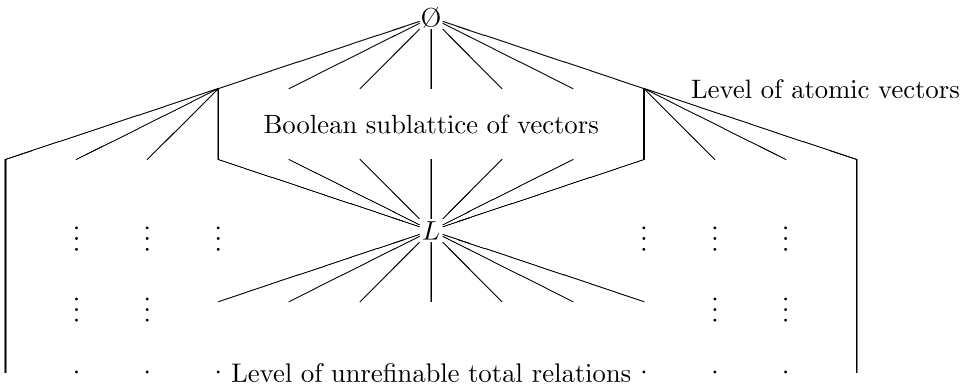
$\emptyset$
Level of atomic vectors
Boolean sublattice of vectors
L
Level of unrefinable total relations

Figure 1: General structure of a semilattice ordered by $\sqsubseteq$.

A pair (s, t) belongs to $Q \mathbin{\square} R$ if and only if it belongs to QR and there is no possibility of reaching, from s, by Q, an element u that does not belong to the domain of R. For example, if $Q = \{(0,0), (0,1), (1,2)\}$ and $R = \{(0,0), (2,3)\}$, one finds that $Q \mathbin{\square} R = \{(1,3)\}$; the pair $(0,0)$, which belongs to QR, does not belong to $Q \mathbin{\square} R$, since $(0,1) \in Q$ and 1 is not in the domain of R. Note that we assign to $\square$ the same binding power as that of $;$.

Proposition 19.

(a) *Q deterministic* $\Rightarrow Q \mathbin{\square} R = QR$,

(b) *P deterministic* $\Rightarrow P \mathbin{\square} (Q \sqcap R) = PQ \sqcap PR$,

(c) *R total* $\Rightarrow Q \mathbin{\square} R = QR$,

(d) $PL \cap QL = \emptyset \Rightarrow P \sqcap Q = P \cup Q$,

(e) $PL \cap QL = \emptyset \Rightarrow P \sqcup Q = \emptyset$,

(f) $PL \cap QL = \emptyset \Rightarrow (P \cup Q) \mathbin{\square} R = P \mathbin{\square} R \cup Q \mathbin{\square} R$,

(g) $P \sqsubseteq Q \quad \wedge \quad R \sqsubseteq S \ \wedge$
$(P \cap RL) \cup (R \cap \mathcal{S}L) \subseteq (Q \cap PL) \cup (\mathcal{S} \cap RL) \Rightarrow P \cup R \sqsubseteq Q \cup \mathcal{S}$.

The next proposition demonstrates a number of additional properties. Of particular interest is item (c), which shows that demonic composition distributes on the right over intersection when one of the intersected entities is a vector.

Lemma 20. *Let Q, R be relations and u, v vectors. We have,*

(a) $(v \cap Q) \mathbin{\square} R = v \cap Q \mathbin{\square} R$,

(b) $R \mathbin{\square} v = RL \cap R \lhd v$, *(c)* $Q \mathbin{\Box} (v \cap R) = Q \mathbin{\Box} v \cap Q \mathbin{\Box} R$, *[26, 43]*

(d) $Q \sqsubseteq R \Leftrightarrow v \cap Q \sqsubseteq v \cap R \wedge \overline{v} \cap Q \sqsubseteq \overline{v} \cap R$,

(e) $u \sqsubseteq v \;\Rightarrow\; P \triangleleft u \cap Q \sqsubseteq P \triangleleft v \cap Q$,

(f) $R \sqsubseteq v \cap R$,

(g) $v \cap Q \sqsubseteq R \;\Rightarrow\; Q \sqsubseteq R$.

5 Demonic Fixed points

During the execution of a program in an input state, by considering a demonic point of view (if there is a possibility for the program not to terminate normally then it will not terminate normally), three cases may happen: normal termination, abnormal termination and infinite loops. In this section, we will give formally the input-output relation of a program and show that is equal to the greatest fixed point of the the semantic function of the while loop **do** $\mathtt{P} \rightarrow \mathtt{Q}$ **od**. We will give a few facts about fixed points.

The next theorem highlights the importance of progressive finiteness in the simplification of fixed point-related properties.

Theorem 21. *[5] Let $f(X) := Q \cup PX$ be a function. If P is progressively finite, the function f has a unique fixed point which means that $\nu(f) = \mu(f) = P^*Q$.*

Because YL is a vector characterizing the domain of relation Y, the following theorem qualifies the range of domains of fixed points of f. We note that in the case when the relation P is progressively finite, we find the results of Theorem 21.

Theorem 22. *Every fixed point Y of $f(X) := Q \cup PX$ satifies*

$$P^*Q \subseteq Y \subseteq P^*Q \cup \overline{\mathcal{L}(P)}$$

and P^*Q and $P^*Q \cup \overline{\mathcal{L}(P)}$ are respectively the least and the greatest fixed point of the function f.

For proof see [5]. The next corollary is about the fixed points of the function $g(X) := Q \cap P \triangleleft X$.

Corollary 23. *Every fixed point Y of $g(X) := Q \cap P \triangleleft X$ satisfies*

$$P^* \triangleleft Q \cap \mathcal{L}(P) \subseteq Y \subseteq P^* \triangleleft Q$$

$P^* \triangleleft Q \cap \mathcal{L}(P)$ and $P^* \triangleleft Q$ are respectively the least and the greatest fixed points of the function g.

Proof. It is easy to verify that g is the dual function (Definition 10) of $f(X) := \overline{Q} \cup PX$. By Lemma 11, $\overline{Y}$ is a fixed point of f. By Theorem 22, $\overline{Y}$ verifies

$$P^*\overline{Q} \subseteq \overline{Y} \subseteq P^*\overline{Q} \cup \overline{\mathcal{L}(P)}.$$

By applying DeMorgan Laws, this is equivalent to

$$\overline{P^*\overline{Q}} \cap \mathcal{L}(P) \subseteq Y \subseteq \overline{P^*\overline{Q}}.$$

Finally, by Definition 4,

$$P^* \triangleleft Q \cap \mathcal{L}(P) \subseteq Y \subseteq P^* \triangleleft Q.$$

We note that, if the relation P is progressively finite, the function g has a unique fixed point which is $P^* \triangleleft Q$. □

We introduce the next Abbreviations:

Abbreviation 24. Let P and Q be relations. The Abbreviations d, d_L and $\mathcal{A}(P,Q)$ are defined as follows (x is a vector) :

$$\begin{aligned}
&d(X) := (Q \cup PX) \cap P \triangleleft XL,\\
&d_L(x) := (PL \cup QL) \cap P \triangleleft x,\\
&\mathcal{A}(P,Q) := P^* \triangleleft (PL \cup QL)\\
&\mathcal{S} := P^*Q \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P).
\end{aligned}$$

(Mnemonics : the subscript L refers to the fact that d_L is obtained from d by composition with L; $\mathcal{A}$ stands for *a*bnormal, since it represents states from which abnormal termination is not possible; finally, $\mathcal{S}$ stands for *s*emantics, since it represents states from which no infinite loop is possible.

5.1 Intuition

- The function d can be considered as a generalization of the semantic function of the while loop **do** $\mathtt{P} \rightarrow \mathtt{Q}$ **od**.
- In a nondeterministic loop **while** P **do** Q, P is iteratively applied to a state s until Q holds. As, P is nondeterministic s can have many outputs. If among these outputs there is a state which can lead outside the domain of P or of Q, so this state is excluded (abnormal termination of the loop). So, $\mathcal{A}(P,Q)$ represents the states from which no abnormal termination is possible.
- We note that $\mathcal{S}$ is an intersection of three terms; P^*Q, $\mathcal{A}(P,Q)$ and $\mathcal{L}(P)$. By taking in consideration the intuition behind these terms, it is easy to see that the relation $\mathcal{S}$ represents the set of states from which the termination is guaranteed because all the states from which there is a possibility of nontermination (abortion or infinite loop) are excluded by the terms $\mathcal{A}(P,Q)$ and $\mathcal{L}(P)$.

The following lemma presents the relationship between the fixed points of the functions d and d_L (Abbreviation 24).

Lemma 25. *If Y is a fixed point of d then YL is a fixed point of d_L.*

Proof. Suppose that $d(Y) = Y$.

$$\begin{array}{ll} & d_L(YL) \\ = & \qquad \langle \text{ Definition of } d_L \text{ (24). } \rangle \\ & (PL \cup QL) \cap P \triangleleft YL \\ = & \qquad \langle\ L = YL \cup \overline{YL}.\ \rangle \\ & (QL \cup P(YL \cup \overline{YL}) \cap P \triangleleft YL \\ = & \qquad \langle \text{ Boolean laws, Theorem 2(i) and Lemma 5(f) . } \rangle \\ & (Q \cup PY)L \cap P \triangleleft YL \\ = & \qquad \langle \text{ Lemma 5(j) and Theorem 2(n). } \rangle \\ & ((Q \cup PY) \cap P \triangleleft YL)L \\ = & \qquad \langle \text{ Definition of } d \text{ (24) and } d(Y) = Y.\ \rangle \\ & YL \end{array}$$

□

In the sequel we give the bounds of the fixed points of d_L and show that, these bounds are also fixed points of d_L.

Theorem 26. *If Y is a fixed point of d, then*

(a) $\mathcal{A}(P,Q) \cap \mathcal{L}(P) \subseteq YL \subseteq \mathcal{A}(P,Q)$,

(b) $\mathcal{A}(P,Q) \cap \mathcal{L}(P)$ *and* $\mathcal{A}(P,Q)$ *are fixed points of* d_L.

Proof.

1. By Lemma 25, YL is a fixed point of d_L. By taking $Q := PL \cup QL$ in Corollary 23, we find,

$$P^* \triangleleft (PL \cup QL) \cap \mathcal{L}(P) \subseteq YL \subseteq P^* \triangleleft (PL \cup QL);$$

by using Abbreviation 24, we find

$$\mathcal{A}(P,Q) \cap \mathcal{L}(P) \subseteq YL \subseteq \mathcal{A}(P,Q).$$

2. By Corollary 23, we deduce that $\mathcal{A}(P,Q) \cap \mathcal{L}(P)$ and $\mathcal{A}(P,Q)$ are fixed points of d_L.

□

The next theorem characterizes the domain of $\mathcal{S}$ (24). This domain is the set of points for which normal termination is guaranteed (no possibility of abnormal termination or infinite loop).

Theorem 27. *Let $\mathcal{S}$ given by the Abbreviation 24. We have*

$$\mathcal{S}L = \mathcal{A}(P, Q) \cap \mathcal{L}(P).$$

Proof. $\mathcal{S}L$

$=$ ⟨ Abbreviation 24. ⟩

$(P^*Q \cap \mathcal{A}(P, Q) \cap \mathcal{L}(P))L$

$=$ ⟨ Theorem 2(n). ⟩

$P^*QL \cap \mathcal{A}(P, Q) \cap \mathcal{L}(P)$

$=$ ⟨ Abbreviation 24 and Lemma 5(f). ⟩

$(P^*QL \cup P^*\overline{PL \cup QL}) \cap \mathcal{A}(P, Q) \cap \mathcal{L}(P)$

$=$ ⟨ Theorem 2(b,h). ⟩

$P^*(QL \cup \overline{PL} \cap \overline{QL}) \cap \mathcal{A}(P, Q) \cap \mathcal{L}(P)$

$=$ ⟨ Theorem 2(c). ⟩

$P^*(QL \cup \overline{PL}) \cap \mathcal{A}(P, Q) \cap \mathcal{L}(P)$

$=$ ⟨ Theorem 2(h). ⟩

$P^*QL \cap \mathcal{A}(P, Q) \cap \mathcal{L}(P) \cup P^*\overline{PL} \cap \mathcal{A}(P, Q) \cap \mathcal{L}(P)$

$=$ ⟨ Proposition 15(a) and Boolean Law. ⟩

$\mathcal{A}(P, Q) \cap \mathcal{L}(P)$

□

In what follows, we will present some interesting properties satisfied by $\mathcal{S}$ and relations that have the same domain as $\mathcal{S}$.

Lemma 28. *Let R be a relation and $\mathcal{S}$ given by Abbreviation 24. Then*

$$R \mathbin{\Box} \mathcal{S} = RP^*Q \cap R \mathbin{\lhd} (\mathcal{A}(P, Q) \cap \mathcal{L}(P)),$$

Proof. $R \mathbin{\Box} \mathcal{S}$

$=$ ⟨ Definition 18 and Theorem 27. ⟩

$R\mathcal{S} \cap R \mathbin{\lhd} (\mathcal{A}(P, Q) \cap \mathcal{L}(P))$

$=$ ⟨ Abbreviation 24, Theorem 2(c,h), Lemma 5(f) and Boolean law. ⟩

$R(P^*Q \cap \mathcal{A}(P, Q) \cap \mathcal{L}(P) \cup \overline{\mathcal{A}(P, Q) \cap \mathcal{L}(P)}) \cap R \mathbin{\lhd} (\mathcal{A}(P, Q) \cap \mathcal{L}(P))$

$=$ ⟨ Theorem 2(c,h) and Lemma 5(f). ⟩

$RP^*Q \cap R \mathbin{\lhd} (\mathcal{A}(P, Q) \cap \mathcal{L}(P))$

□

In the following theorem, we will show that $\mathcal{S}$ is a fixed point of d (Abbreviation 24 and demonic composition 18). The function d can be also given by,

$$d(X) = Q \cap P \mathbin{\lhd} XL \cup P \mathbin{\Box} X \tag{10}$$

Theorem 29. *$\mathcal{S}$ (Abbreviation 24) is a fixed point of d.*

Proof. $d(\mathcal{S})$

$=$ $\langle$ Abbreviation 24 and Theorem 27. $\rangle$

$(Q \cup P\mathcal{S}) \cap (P \lhd (\mathcal{A}(P,Q) \cap \mathcal{L}(P)))$

$=$ $\langle$ Theorem 2(h). $\rangle$

$Q \cap (P \lhd (\mathcal{A}(P,Q) \cap \mathcal{L}(P))) \cup P\mathcal{S} \cap (P \lhd (\mathcal{A}(P,Q) \cap \mathcal{L}(P)))$

$=$ $\langle$ $Q \subseteq QL \cup PL$ and Theorem 27. $\rangle$

$Q \cap (PL \cup QL) \cap (P \lhd (\mathcal{A}(P,Q) \cap \mathcal{L}(P))) \cup P\mathcal{S} \cap P \lhd \mathcal{S}L$

$=$ $\langle$ Definition 18, Abbreviation 24 and Theorem 26(b). $\rangle$

$Q \cap (\mathcal{A}(P,Q) \cap \mathcal{L}(P)) \cup P \mathbin{\square} \mathcal{S}$

$=$ $\langle$ Lemma 28. $\rangle$

$Q \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P) \cup PP^{*}Q \cap P \lhd \mathcal{A}(P,Q) \cap \mathcal{L}(P)$

$=$ $\langle$ $PP^{*}Q \subseteq PL \cup QL$, Theorem 26(b) and Boolean law. $\rangle$

$(Q \cup PP^{*}Q) \cap (\mathcal{A}(P,Q) \cap \mathcal{L}(P))$

$=$ $\langle$ Theorem 2(i) and Equation 3. $\rangle$

$P^{*}Q \cap (\mathcal{A}(P,Q) \cap \mathcal{L}(P))$

$=$ $\langle$ Abbreviation 24. $\rangle$

$\mathcal{S}$

□

The following theorem is a generalization to a nondeterministic context of the *while statement verification rule* of Mills [34, 35]. It shows that the least fixed point W of d (Abbreviation 24) is uniquely characterized by conditions (a) and (b), that is, by the fact that W is a fixed point of d and by the fact that no infinite loop is possible when the execution is started in a state that belongs to the domain of W. Half of this theorem (the $\Leftarrow$ direction) is also proved by Sekerinski (the *main iteration theorem* [43]) in a predicative programming setup.

Theorem 30. *W is the least fixed point wrt $\subseteq$ of d (Abbreviation 24) ($W = \mu_{\subseteq}(d)$) iff*

$$\begin{array}{ll} (a) & W = d(W), \\ (b) & WL \subseteq \mathcal{L}(P). \end{array}$$

Proof. ($\Rightarrow$) **:** As W is the least fixed point of d then, (a) is evident. Since $W = \mu(d) \subseteq \mathcal{S}$, then $WL = \mu(d)L \subseteq \mathcal{S}L$, by using Theorem 27, we have $WL \subseteq \mathcal{L}(P)$.

($\Leftarrow$) **:** By Hypothesis (a), W is a fixed point of d. Then, by Theorem 26, $\mathcal{A}(P,Q) \cap \mathcal{L}(P) \subseteq WL \subseteq \mathcal{A}(P,Q)$. But, by using Hypothesis (b), $WL \subseteq \mathcal{L}(P)$, then $WL = \mathcal{A}(P,Q) \cap \mathcal{L}(P)$.

W

$$\begin{aligned}
= \quad & \langle \text{ Hypothesis } \rangle \\
& d(W) \\
= \quad & \langle \text{ Abbreviation24. } \rangle \\
& (Q \cup PW) \cap P \triangleleft WL \\
= \quad & \langle\ (Q \cup PW) \subseteq (PL \cup QL) \text{ and } WL = \mathcal{A}(P,Q) \cap \mathcal{L}(P)\ \rangle \\
& (Q \cup PW) \cap (PL \cup QL) \cap P \triangleleft (\mathcal{A}(P,Q) \cap \mathcal{L}(P,Q)) \\
= \quad & \langle \text{ Lemma 28(b) } \rangle \\
& (Q \cup PW) \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P) \\
= \quad & \langle \text{ Theorem 2(n)and Boolean law. } \rangle \\
& Q \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P) \cup (P \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P))W
\end{aligned}$$

So W is a fixed point of the function $g(X) := Q \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P) \cup (P \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P))X$. Since, by Proposition 15(i,h), $P \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P)$ is progressively finite, invoking Theorem 21 shows that g has a unique fixed point which is the least fixed point $\mu(d)$. We conclude that $W = \mu(d)$. □

The next theorem shows that $\mathcal{S}$ is the least fixed point of d wrt $\subseteq$ $(\mathcal{S} = \mu_{\subseteq}(d))$.

Theorem 31. *$\mathcal{S}$ (Abbreviation24) is the least fixed point of d wrt $\subseteq$ $(\mathcal{S} = \mu_{\subseteq}(d))$.*

Proof. It suffices to prove that $\mathcal{S}$ satisfies the conditions of the Theorem 30. By Theorem 29, $\mathcal{S}$ is a fixed point of d. So, the condition (a) is satisfied. By Theorem 27, $\mathcal{S}L = \mathcal{A}(P,Q) \cap \mathcal{L}(P)$, so that condition (b) holds too. □

In the sequel, we show that $\mathcal{S}$ is the greatest fixed point with respect to $\sqsubseteq$ of d (Equation 10). In other words, we will show that the least fixed point of d wrt $\subseteq$ is equal to the greatest fixed point of the same function d wrt $\sqsubseteq$. But before we have to prove that d is monotonic wrt $\sqsubseteq$.

Lemma 32. *The function d (Abbreviation 24) is monotonic wrt $\sqsubseteq$.*

Proof. Let X and Y be relations such that $X \sqsubseteq Y$.

As $\square$ is monotonic, we have $P \square X \sqsubseteq P \square Y$ and $XL \sqsubseteq YL$. By Proposition 20(e), we have $Q \cap P \triangleleft XL \sqsubseteq Q \cap P \triangleleft YL$.

By taking in Proposition 19(g) $P := P \square X \wedge Q := P \square Y \wedge R := Q \cap P \triangleleft XL \wedge \mathcal{S} := R := Q \cap P \triangleleft YL$, it is easy to see that the conditions are satisfied, whence we conclude that $d(X) \sqsubseteq d(Y)$. Thus, d is monotonic wrt $\sqsubseteq$ □

As the function d is monotonic (wrt $\sqsubseteq$)(Lemma 32), by Theorem 7 and Equation 9(a) the greatest fixed point $\mathcal{S}$ of d exists and is given by

$$\mathcal{S} = \bigsqcup\{X | X = Q \cap P \triangleleft XL \cup P \square X\}; \tag{11}$$

(Since $\mathcal{S}$ is a fixed point of d (Theorem 29) and thus $\bigsqcup\{X | X = Q \cap P \triangleleft XL \cup P \square X\}$ is well defined, by completness of the $\bigsqcup$-semilattice; for more information about the properties of the $\bigsqcup$-semilattice see [22]).

In the next theorem, we show that $\mathcal{S}$ is the greatest fixed point with respect to $\sqsubseteq$ of d (Equation 10).

Theorem 33. *$\mathcal{S}$ is the greatest fixed point with respect to $\sqsubseteq$ of d ($\mathcal{S} = \nu_{\sqsubseteq}(d)$).*

Proof. Let $W = \nu_{\sqsubseteq}(d)$. Let's show that $W = \mu_{\subseteq}(d)$. Then $\mathcal{S} \sqsubseteq W$, by Definition 16 and Theorem 26, $WL \subseteq \mathcal{S}L \subseteq \mathcal{L}(P)$. Also, we have $W = d(W)$. By Theorem 30, $W = \mu_{\subseteq}(d)$. Hence $W = \mathcal{S}$. □

The following example is an application of our results. It is rather contrived, but it is simple and fully illustrate the various cases that may happen.

6 Application

In [7, 8], Berghammer and Schmidt propose abstract relation algebra as a practical means for the specification of data types and programs. Often, in these specifications, a relation is characterized as a fixed point of some function. Can demonic operators be used in the definition of such a function? Let us now show with a simple example that the concepts presented in this paper give useful insights for answering this question.

In [7, 8], it is shown that the natural numbers can be characterized by the relations z and S (*zero* and *successor*) and the laws

$$\begin{array}{lll}
\text{(a)} & \emptyset \neq z = zL \;\;\wedge\;\; z\breve{z} \subseteq I & (z \text{ is a point}), \\
\text{(b)} & S\breve{S} = I \;\;\wedge\;\; \breve{S}S \subseteq I & (S \text{ is a one to one function.}), \\
\text{(c)} & Sz = \emptyset & (z \text{ has no predecessor}), \\
\text{(d)} & L = \bigcap\{x | z \cup \breve{S}x = x\} & (\text{generation principle}).
\end{array} \tag{12}$$

By Proposition 19(a,c) these equations imply,

$$\begin{array}{lll}
\text{(a)} & \emptyset \neq z = z \,\Box\, L \;\;\wedge\;\; z \,\Box\, \breve{z} \subseteq I & (z \text{ is a } \textit{demonic} \text{ point}), \\
\text{(b)} & S \,\Box\, \breve{S} = I \;\;\wedge\;\; \breve{S} \,\Box\, S \subseteq I & (S \text{ is a one to one function.}), \\
\text{(c)} & S \,\Box\, z = \emptyset & (z \text{ has no } \textit{demonic} \text{ predecessor}), \\
\text{(d)} & L = \bigcap\{x | z \cup \breve{S} \,\Box\, x = x\} & (\textit{demonic} \text{ generation principle}).
\end{array} \tag{13}$$

By Proposition 19(a), Theorem 22 and the last axiom, the function

$$g(X) = z \cup \breve{S} \,\Box\, X \tag{14}$$

obviously has a unique solution for X, namely, $X = L$ and this solution is expressed by

$$L = \breve{S}^{*}z \tag{15}$$

However, it is easy to show that $z \sqcup \breve{S} \,\Box\, X \subseteq X$, obtained by replacing the join in Equation 14 by its demonic counterpart, has infinitely many solutions. Indeed, from $S \,\Box\, z = \emptyset$, Proposition 19(c) and the Schröder rule, it follows that $z \cap \breve{S} \,\Box\, L = \emptyset$, so by Proposition 19(e), we have $z \sqcup \breve{S} \,\Box\, X = \emptyset$.

Hence, any relation R is a solution to $z \sqcup \breve{S} \Box X \subseteq X$. Looking at Figure 1, one immediately sees why it is impossible to reach L by joining anything to z (which is a point and hence is an immediate $\sqsubseteq$-predecessor of $\emptyset$), since this can only lead to z or to $\emptyset$.

Let us now go 'fully demonic' and ask what is a solution to $z \sqcup \breve{S} \Box X \sqsubseteq X^1$. By the discussion above, this is equivalent to $\emptyset \sqsubseteq X$, which has a unique solution, $X = \emptyset$. This raises the question whether it is possible to find some fully demonic function similar to (14), whose fixed point is $X = L$. Because L is in the middle of the demonic semilattice (Figure 1), there are in fact two possibilities: either approach L from above or from below.

For the approach from above, consider the function

$$h(X) := z \sqcap \breve{S} \Box X. \tag{16}$$

In other words, we have to find the post-fixed point of this function (wrt $\sqsubseteq$). By Proposition 19(d), we have, $z \sqcap \breve{S}X = z \cup \breve{S}X$. This means that 16 reduces to

$$h(X) := z \cup \breve{S}X. \tag{17}$$

By Equation 15 and Theorem 33, L is the greatest fixed point of k (wrt $\sqsubseteq$); so $L = \bigsqcup\{X | X \sqsubseteq z \sqcap \breve{S} \Box X\}$.

Now consider $\bigsqcap_{n \geq 0} \breve{S}^{\underline{n}} \Box z$, where $\breve{S}^{\underline{n}}$ is an n-fold demonic composition defined by $\breve{S}^{\underline{0}} = I$ and $\breve{S}^{\underline{n+1}} = \breve{S} \Box \breve{S}^{\underline{n}}$. By axiom 12(b), $\breve{S}$ is deterministic, so that, by 19(a) and associativity of demonic composition, $\breve{S}^{\underline{n}} \Box z = \breve{S}^n z$. In the sequel, we will prove that $\bigsqcap_{n \geq 0} \breve{S}^{\underline{n}} \Box z$ is defined and is equal to L.

First,

$$
\begin{array}{ll}
 & \bigsqcap_{n \geq 0} \breve{S}^{\underline{n}} \Box z \text{ defined} \\
\Leftrightarrow & \bigsqcap_{n \geq 0} \breve{S}^n z \text{ defined} \\
\Leftrightarrow & \qquad \langle \text{ Proposition 17(a). } \rangle \\
 & L \subseteq (\bigcap_{n \geq 0} \breve{S}^n z \cup \overline{\breve{S}^n zL})L \\
\Leftrightarrow & \qquad \langle \text{ Axiom 12(a), } zL = z \text{ and thus } \breve{S}^n z \cup \overline{\breve{S}^n zL} = L. \ \rangle \\
 & \text{true.}
\end{array}
$$

Second,

$$
\begin{array}{ll}
 & \bigsqcap_{n \geq 0} \breve{S}^{\underline{n}} \Box z \\
= & \bigsqcap_{n \geq 0} \breve{S}^n z \\
= & \qquad \langle \text{ Proposition 17(b). } \rangle \\
 & (\bigcap_{n \geq 0} \breve{S}^n \cup \overline{\breve{S}^n zL}) \cap (\bigcup_{n \geq 0} \breve{S}^n zL)
\end{array}
$$

[1] This inequation contains the conversion operator, which is not a demonic operator. However, it could be suppressed by using relation $P := \breve{S}$ as a basic constant in specification 12, rather than using S.

$=$ $\quad\langle$ Axiom 12(a), $zL = z$ and thus $\breve{S}^n z \cup \overline{\breve{S}^n zL} = L.$ $\rangle$

$\bigcup_{n\geq 0} \breve{S}^n z$

$=$ $\quad\langle$ 15. $\rangle$

$L.$

□

So, $\bigsqcap_{\,n\geq 0} \breve{S}^{\underline{n}} \mathbin{\square} z = \bigsqcup\{X | X \sqsubseteq z \sqcap \breve{S} \mathbin{\square} X\}$.

It is easy to show that for any $n \geq 0$, $\breve{S}^n z$ is a point (it is the n-th successor of zero) and that $m \neq n \Rightarrow \breve{S}^m z \neq \breve{S}^n z$. Hence, in $(\mathcal{R}_L, \sqsubseteq)$, $\{\breve{S}^n z | n \geq 0\}$ (i.e. $\{\breve{S}^{\underline{n}} \mathbin{\square} z | n \geq 0\}$) is the set of immediate predecessors of $\emptyset$; looking at Figure 1 shows how the universal relation L arises as the greatest lower bound $\bigsqcap_{\,n\geq 0} \breve{S}^{\underline{n}} \mathbin{\square} z$ of this set of points.

Note that, whereas there is a unique solution to 14, there are infinitely many solutions to 16 (equivalently, to 17), for example $\bigsqcup_{n\geq h} S^{\underline{n}}$ $\;(= \bigcup_{n\geq k} S^n)$, for any k.

For the upward approach, consider

$$k(X) := \breve{z} \sqcup X \mathbin{\square} S. \tag{18}$$

We will find the pre-fixed point of k (wrt $\sqsubseteq$) Here again there are infinitely many solutions to this case; in particular, any vector v, including $\emptyset$ and L, is a solution to 18. Because $(\mathcal{R}, \sqsubseteq)$ is only a join semilattice, it is not at all obvious that the least fixed point of $k(X) := \breve{z} \sqcup X \mathbin{\square} S$ exists. It does, however, since the following derivation shows that $\bigsqcup_{n\geq 0} \breve{z} \mathbin{\square} S^{\underline{n}}$ $\;(= \bigsqcup_{n\geq 0} k^n(\breve{z})$, where $k^0(\breve{z}) = \breve{z})$ is a fixed point of k and hence is obviously the least solution of 18:

$\breve{z} \sqcup (\bigsqcup_{n\geq 0} \breve{z} \mathbin{\square} S^{\underline{n}}) \mathbin{\square} S$

$=$ $\quad\langle$ Associativity of $\square$, and $S \mathbin{\square} I = SI = S.$ $\rangle$

$\breve{z} \mathbin{\square} I \sqcup (\bigsqcup_{n\geq 0} \breve{z} \mathbin{\square} S^{\underline{n+1}})$

$=$ $\quad\langle$ $S^{\underline{0}} = I.$ $\rangle$

$\breve{z} \mathbin{\square} S^{\underline{0}} \sqcup (\bigsqcup_{n\geq 1} \breve{z} \mathbin{\square} S^{\underline{n}})$

$=$ $\bigsqcup_{n\geq 0} \breve{z} \mathbin{\square} S^{\underline{n}}.$

Because $\breve{z}$ and S are functions, Proposition 19(a) implies that $\breve{z} \mathbin{\square} S^{\underline{n}} = \breve{z}S^n$, for any $n \geq 0$. But $\breve{z}S^n$ is also a function (it is the inverse of the point $\breve{S}^n z$) and hence is total, from which, by Proposition 17(a), law 2(l) and equation 15,

$$\bigsqcup_{n\geq 0} \breve{z} \mathbin{\square} S^{\underline{n}} = \bigsqcup_{n\geq 0} \breve{z}S^n = \bigcup_{n\geq 0} \breve{z}S^n = (\bigcup_{n\geq 0} \breve{S}^n z)^{\smile} = \breve{L} = L.$$

This means that L is the least upper bound of the set of mappings $\{\breve{z} \mathbin{\square} S^{\underline{n}} | n \geq 0\}$. Again, a look at Figure 1 gives some intuition to understand this result, after recalling that mappings are minimal elements in $(\mathcal{R}_L, \sqsubseteq)$ (though not all mappings have the form $\breve{z} \mathbin{\square} S^{\underline{n}}$).

Thus, building L from below using the set of mappings $\{\breve{z} \mathbin{\square} S^{\underline{n}} | n \geq 0\}$ is symmetric to building it from above using the set of points $\{\breve{S}^{\underline{n}} \mathbin{\square} z | n \geq 0\}$. □

Acknowledgement

We thank the anonymous referee for his or her comments.

References

[1] R. J. R. Back. On the correctness of refinement in program development. Thesis, Department of Computer Science, University of Helsinki, 1978.

[2] R. J. R. Back. On correct refinement of programs. *J. Comput. System Sci. 23*, 1, 1981, 49–68.

[3] R. J. R. Back and J. von Wright. Combining angels, demons and miracles in program specifications. *Theoret. Comput. Sci. 100*, 1992, 365–383.

[4] R. C. Backhouse et al. Fixed points calculus. *Inform. Pro. Letters,53*, 1995, 131–136.

[5] R. C. Backhouse and H. Doornbos. Mathematical induction made calculational. Computing Science Note 94/16, Dept. of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands, 1994.

[6] R. C. Backhouse and J. van der Woude. Demonic operators and monotype factors. *Mathematical Structures in Comput. Sci. 3*, 4, 417–433, 1993.

[7] R. Berghammer. Relational specification of data types and programs. Technical report 9109, Fakultät für Informatik, Universität der Bundeswehr München, Germany, 1991.

[8] R. Berghammer and G. Schmidt. Relational specifications. In C. Rauszer, editor, *Algebraic Logic*, volume 28 of *Banach Center Publications*. Polish Academy of Sciences, 1993.

[9] R. Berghammer and H. Zierer. Relational Algebraic semantics of deterministic and nondeterministic programs. *Theoret. Comput. Sci. 43*, 123–147, 1986.

[10] R. Bird and O. de Moor. *Algebra of programming*. Prentice Hall, London, 1997.

[11] N. Boudriga, F. Elloumi and A. Mili. On the lattice of specifications: Applications to a specification methodology. *Formal Aspects of Computing 4*, 1992, 544–571.

[12] C. Brink, W. Kahl, and G. Schmidt, editors. *Relational Methods in Computer Science*. Springer, 1997.

[13] L. H. Chin and A. Tarski. Distributive and modular laws in the arithmetic of relation algebras. *University of California Publications 1*, 1951, 341–384.

[14] E. Cohen. Separation and reduction. In R. Backhouse and J. N. Oliveira, editors, *Proc. of Mathematics of Program Construction, 5th International Conference,MPC 2000,*volume 1837 of *LNCS*, pages 45-59. Springer, 2000.

[15] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order.* Cambridge Mathematical Textbooks, Cambridge University Press, Cambridge, 1990.

[16] J. Desharnais and B. Möller. Least Reflexive Points of Relations. To appear in *Higher Order and Symbolic Computation.*

[17] J. Desharnais, B. Möller, and G Struth. Termination in modal Kleene algebra. In J.-J. Levy, E Mayr, and J. Mitchell, editors, *Proc. IFIP TCS 2004*, pages 653-666. Kluwer, 2004.

[18] J. Desharnais, B. Möller, and G Struth. Applications of modal Kleene algebra. *Journal on Relational Methods in Computer Science 1:93–131 (2004).* http://www.cosc.brocku.ca/Faculty/Winter/JoRMiCS/

[19] J. Desharnais, B. Möller, and G Struth. Kleene algebra with domain. Technical Report 2003-07, Universität Augsburg, Institut für Informatik, June 2003. Revised version to appear in ACM Transactions on Computational Logic. http://www.informatik.uni-augsburg.de/forschung/techBerichte/reports/2003-7.pdf

[20] J. Desharnais and B. Möller. Characterizing determinacy in Kleene algebras. *Informations Sciences,* 139(3-4):253-273, December 2001.

[21] J. Desharnais, B. Möller, and F. Tchier. Kleene under a demonic star. *8th International Conference on Algebraic Methodology And Software Technology (AMAST 2000)*, May 2000, Iowa City, Iowa, USA, *Lecture Notes in Computer Science*, Vol. 1816, pages 355–370, Springer-Verlag, 2000.

[22] J. Desharnais, N. Belkhiter, S. B. M. Sghaier, F. Tchier, A. Jaoua, A. Mili and N. Zaguia. Embedding a demonic semilattice in a relation algebra. *Theoretical Computer Science*, 149(2):333–360, 1995.

[23] E. W. Dijkstra. *A Discipline of Programming.* Prentice-Hall, Englewood Cliffs, N.J., 1976.

[24] H. Doornbos. A relational model of programs without the restriction to Egli-Milner monotone constructs. *IFIP Transactions*, A-56:363–382. North-Holland, 1994.

[25] M. Frappier, J. Desharnais, A. Mili and F. Mili. Verifying objects against axiomatic specifications: A fixed point approach. *5*th Conf. Maghrebian Conference on Software Engeneering and Artificial Intelligence, (MCSEAI‘98). Pages 259-274, Tunis, Tunisie, December 1998.

[26] M. Frappier. A relational basis for program construction by parts. Dept. of Computer Science, University of Ottawa, 1994.

[27] E. Hehner. Predicative programming, Parts I and II. *Commun. ACM, 27*, February 1984, 134–151.

[28] C. A. R. Hoare and J. He. The weakest prespecification. *Fundamenta Informaticae IX*, 1986, Part I: 51–84, 1986.

[29] C. A. R. Hoare and J. He. The weakest prespecification. *Fundamenta Informaticae IX*, 1986, Part II: 217–252, 1986.

[30] C. A. R. Hoare and al. Laws of programming. *Communications of the ACM*, 30:672–686, 1986.

[31] Kleene algebras with tests. *ACM Transactions on Programming Languages and Systems*, 19, 427-443, 1997.

[32] R. D. Maddux. Relation-algebraic semantics. *Theoretical Computer Science*, 160:1–85, 1996.

[33] A. Mili, J. Desharnais and F. Mili. Relational heuristics for the design of deterministic programs. *Acta Inform. 24*, 3, 1987, 239–276.

[34] H. D. Mills. The new math of computer programming. *Commun. ACM 18*, 1, January 1975, 43–48.

[35] H. D. Mills, V. R. Basili, J. D. Gannon and R. G. Hamlet. *Principles of Computer Programming. A Mathematical Approach.* Allyn and Bacon, Inc., 1987.

[36] C. Morgan and K. Robinson. Specification statements and refinement. *IBM J. Res. Dev. 31*, 5, 1987. Reprinted in: C. Morgan and T. Vickers (eds). *On the refinement calculus.* Springer-Verlag, 1994, 23–46.

[37] J. M. Morris. A theoretical basis for stepwise refinement and the programming calculus. *Sci. Comput. Prog. 9*, 1987, 287–306.

[38] T. T. Nguyen. A relational model of demonic nondeterministic programs. *Int. J. Found. Comput. Sci. 2*, 2, 1991, 101–131.

[39] J. Cai and R. Paige. Program Derivation by Fixed point Computation. *Science of Computer Programming,*, 11,1989, 197–261.

[40] D. L. Parnas. A Generalized Control Structure and its Formal Definition. *Communications of the ACM*, 26:572–581, 1983

[41] G. Schmidt. Programs as partial graphs I: Flow equivalence and correctness. *Theoret. Comput. Sci. 15*, 1981, 1–25.

[42] G. Schmidt and T. Ströhlein. *Relations and Graphs.* EATCS Monographs in Computer Science, Springer-Verlag, Berlin, 1993.

[43] E. Sekerinski. A calculus for predicative programming. *Second International Conf. on the Mathematics of Program Construction.* R. S. Bird, C. C. Morgan and J. C. P. Woodcock (eds), Oxford, June 1992, *Lect. Notes in Comput. Sci. 669*, Springer-Verlag, 1993.

[44] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5, 1955, 285-309.

[45] A. Tarski. On the calculus of relations. *J. Symb. Log. 6*, 3, 1941, 73–89.

[46] F. Tchier. While loop demonic relational semantics monotype/residual style. *2003 International Conference on Software Engineering Research and Practice (SERP03)*, Las Vegas, Nevada, USA, 23-26, June 2003.

[47] F. Tchier. Demonic relational semantics monotype/residual style. *International Journal of Mathematics and Mathematical sciences*, 2003.

[48] F. Tchier. Demonic semantics by monotypes. *International Arab conference on Information Technology* (Acit2002), University of Qatar, Qatar, 16-19 December 2002.

[49] F. Tchier. Demonic relational semantics of compound diagrams. In: Jules Desharnais, Marc Frappier and Wendy MacCaull, editors. Relational Methods in computer Science: The Québec seminar, pages 117-140, Methods Publishers 2002.

[50] F. Tchier. Sémantiques relationnelles démoniaques et vérification de boucles non déterministes. Thése de doctorat, Département de Mathématiques et de statistique, Université Laval, Canada, 1996. http://auguste.ift.ulaval.ca/∼ desharn/Theses/index.html

[51] M. Walicki and S. Meldal. Algebraic approches to nondeterminism: An overview. *ACM computing Surveys,29(1)*, 1997, 30-81.

[52] N. T. E. Ward. A refinement Calculus for Nondeterministic Expressions. Ph D thesis, University of Queensland, Australia, 1994.

[53] L.Xu, M. Takeichi and H. Iwasaki. Relational semantics for locally nondeterministic programs. *New Generation Computing 15*, 1997, 339-362.

Received March, 2004