ISSN 2347-1921

Nondeterministic Relational Semantics of a while Program

Fairouz Tchier
Mathematics department, King Saud University, Riyadh
Saudi Arabia
ftchier@ksu.edu.sa

Abtract

A relational semantics is a mapping of programs to relations. We consider that the input-output semantics of a program is given by a relation on its set of states; in a nondeterministic context, this relation is calculated by considering the worst behavior of the program (demonic relational semantics). In this paper, we concentrate on while loops. Calculating the relational abstraction (semantics) of a loop is difficult, but showing the correctness of any candidate abstraction is much easier. For functional programs, Mills has described a checking method known as the while statement verification rule. A programming theorem for iterative constructs is proposed, proved, demonstrated and applied for an example. This theorem can be considered as a generalization of the while statement verification to nondeterministic loops.

Keywords: while loop; demonic semantics, relational abstraction.

Council for Innovative Research
Peer Review Research Publishing System
Journal: Journal of Advances in Mathematics

Vol 3, No 3
editor@cirworld.com
www.cirworld.com, member.cirworld.com

ISSN 2347-1921

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 $\sqcap$ (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.

The semantics of a while loop is given as a fixed point of a monotonic function involving the demonic operators. While there 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. For functional programs, Mills [23, 24] has described a checking method known as the *while statement verification rule*. We generalize this rule to nondeterministic loops.

The rest of the paper is organized as follows. In Section 2, we present our mathematical tool, namely relation algebra [10, 29, 31]. First, we recall the basic laws (Subsection 2.1), then we present notions related to infinite looping (Subsection 2.3) followed by a description of our refinement ordering (Subsection 2.5). In Section 3, we present a generalization of this theorem with an example; we note here that half of the generalized theorem is demonstrated by Sekerinski [30], who uses an approach based on predicative programming [17]. We conclude in Section 5 with prospects for future research.

2 Relation algebras

2.1 Definition and basic laws

Our mathematical tool is abstract relation algebra [10, 29, 31], which we now introduce.

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

- $(R,\cup,\cap,^{-})$) is a complete atomic 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: $PQ \subseteq R \Leftrightarrow \breve{P};\overline{R} \subseteq \overline{Q} \Leftrightarrow \overline{R};\breve{Q} \subseteq \overline{P}$.
- $L;R;L = L \Leftrightarrow R \neq \emptyset$ (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

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

The precedence of the relational operators from highest to lowest is the following: $^{-}$ and $^{\smile}$ 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., [8, 10, 29]). We assume these rules to be known and simply recall a few of them.

(2) Theorem. Let P,Q,R be relations. Then,

ISSN 2347-1921

$(a)\ \overline{Q \cup R} = \overline{Q} \cap \overline{R},$
$(b)\ \overline{Q \cap R} = \overline{Q} \cup \overline{R},$
$(c)\ Q \cap R \cup \overline{R} = Q \cup \overline{R},$
$(d)\ P \cap Q \subseteq R \ \Leftrightarrow\ P \subseteq \overline{Q} \cup R,$
$(e)\ Q \subseteq R \ \Leftrightarrow\ \overline{R} \subseteq \overline{Q},$
$(f)\ P(Q \cap R) \subseteq PQ \cap PR,$
$(g)\ (P \cap Q)R \subseteq PR \cap QR,$
$(h)\ P(Q \cup R) = PQ \cup PR,$
$(i)\ (P \cup Q)R = PR \cup QR,$
$(j)\ Q \subseteq R \ \Rightarrow\ PQ \subseteq PR,$
$(k)\ Q \subseteq R \ \Rightarrow\ QP \subseteq RP.$
$(l)\ \overline{RL}L = \overline{RL},$
$(m)\ PQ \cap R \subseteq P(Q \cap PR),$
$(n)\ (P \cap QL)R = PR \cap QL,$
$(o)\ (\bigcap_{i \in X} R_i L)L = \bigcap_{i \in X} R_i L.$

We now give a definition of various properties of relations.

(3) Definition. **A relation** R **is** *functional* **iff** $\breve{R}R \subseteq I$ **. A relation** v **is a** *vector* **[29] 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 $RL = \{(0,0),(0,1),(2,0),(0,2),(2,1),(2,2)\}$; 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 [35, 33, 34]

(4) Definition. A binary operator $\lhd$, called *relative implication* [36], is defined as follows :

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

This operator has a dual operator (2.2), $\rhd$, given by :

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

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 $x \in \text{dom}(Q \lhd RL) \Leftrightarrow x.Q \subseteq \text{dom}(R)$.

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 verified by the operator $\lhd$. The properties of $\rhd$ can be obtained by dualization of those of the operator $\lhd$ [36].

(5) Lemma. *Let* P, Q *and* R *be relations and* v *a vector.*

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

ISSN 2347-1921

- $P \triangleleft Q \subseteq PQ \cup \overline{PL}$,
- $P \subseteq Q \Rightarrow Q \triangleleft R \subseteq P \triangleleft R$,
- $P \subseteq Q \Rightarrow R \triangleleft P \subseteq R \triangleleft Q$

We note that the properties (2.2) and (2.2) are similar to those of the logical operators $\rightarrow$, $\wedge$ and $\vee$. For example, the property (2.2) corresponds to $(P \wedge Q \rightarrow R) \leftrightarrow (P \rightarrow (Q \rightarrow R))$.

Proof. The properties (2.2) and (2.2) are directly deduced from a Boolean law on the complementation. The property (2.2) is deduced from the laws 2.1(2.1,2.1,2.1, 2.1). In the following we give the proofs of the other properties.

(e) $PQ \triangleleft R$
$= \quad \langle$ Definition 4. $\rangle$
$\overline{PQ\overline{R}}$
$= \quad \langle$ Lemma 5(a). $\rangle$
$\overline{\overline{PQ \triangleleft R}}$
$= \quad \langle$ Definition 4. $\rangle$
$P \triangleleft (Q \triangleleft R)$

(f) $PQ \cap P \triangleleft R$
$= \quad \langle$ Definition 4 and $P\overline{R} \cap P \triangleleft R = \emptyset$. $\rangle$
$(PQ \cup P\overline{R}) \cap P \triangleleft R$
$= \quad \langle$ Theorem 2(h). $\rangle$
$P(Q \cup \overline{R}) \cap P \triangleleft R$

(g) $PL = P(Q \cup \overline{Q})$
$\Rightarrow \quad \langle$ Theorem 2(h). $\rangle$
$PL \subseteq PQ \cup P\overline{Q}$
$\Leftrightarrow \quad \langle$ Theorem 2(d) (applied twice). $\rangle$
$P \triangleleft Q \subseteq PQ \cup \overline{PL}$

(h) $P \subseteq Q$
$\Rightarrow \quad \langle$ Theorem 2(j). $\rangle$
$P\overline{R} \subseteq Q\overline{R}$
$\Leftrightarrow \quad \langle$ Theorem 2(e). $\rangle$
$\overline{Q\overline{R}} \subseteq \overline{P\overline{R}}$
$\Leftrightarrow \quad \langle$ Definition 4. $\rangle$
$Q \triangleleft R \subseteq P \triangleleft R$

(i) Can be proved in a similar way.

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 :

(6) (a) $\mu f = \bigcap\{X \mid f(X) = X\} = \bigcap\{X \mid f(X) \subseteq X\}$,
(b) $\nu f = \bigcup\{X \mid f(X) = X\} = \bigcup\{X \mid X \subseteq f(X)\}$,
(c) $\mu f \subseteq \nu f$,
(d) $f(Y) \subseteq Y \Rightarrow \mu f \subseteq Y$,
(e) $Y \subseteq f(Y) \Rightarrow Y \subseteq \nu f$.

ISSN 2347-1921

(7) Theorem. *(Knaster-Tarski) Every monotonic endofunction on a complete lattice has a least fixed point, which is equal to its least prefixpoint.*

The comparison of fixed points is sometimes very useful to compare the program semantics. The next proposition will present some results in this meaning.

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

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

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

$$(a) \quad \mu \text{ monotonic} \qquad f = g \;\Rightarrow\; \mu(f) \leq \mu(g),$$
$$(d) \quad \mu - \text{fusion law} \qquad fg = gh \;\Rightarrow\; \mu(f) = g(\mu(h)).$$

Another operation that occurs in the definition of the while program semantics is the

$$(9) \qquad R^* = \mu(X \mapsto I \cup RX).$$

the Knaster-Tarski Theorem (2.2) this operation is well defined and it verifies,

$$(10) \qquad R^* = I \cup RR^* = I \cup R^*R.$$

The unitary operators $*^{\cup}$ and $\bar{}$ bin equally. We can also, define, a similar operation to Called transitive closure, denoted $+$, and defined for every relation R by :

$$(11) \quad (a) \quad R^+ = R^*R = RR^* = R \cup RR^*,$$
$$(b) \quad R^* = I \cup R^+.$$

(4)

The opertions * and + bind equally. The operation * satisfies also

$$(12) \qquad R^* = \cup_{i \geq 0} R^i,$$

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 11 and of the properties of $*$.

$$(13) \qquad R^*Q = \mu(X \mapsto Q \cup RX).$$

(14) Proposition. If *if* $PQP = PQ \Rightarrow (PQ)^*P = PQ^*,$

Proof.

We use Proposition 8(d). Let functions f, g et h defined as follows :

$$f(X) := P \cup PQX, \quad g(X) := PX, \quad h(X) := I \cup QX.$$

It is easy to verify that

$$f ; g = g ; h.$$

ISSN 2347-1921

$$
\begin{aligned}
& \mu(X \mapsto P \cup PQX) \\
= & \quad \langle\ f(X) = P \cup PQX.\ \rangle \\
& \mu(f) \\
= & \quad \langle\ f;g = g;h \text{ and proposition 8(d).}\ \rangle \\
& g(\mu(h)) \\
= & \quad \langle\ g(X) = PX,\ h(X) = I \cup QX \text{ and Equation 9.}\ \rangle \\
& PQ^*
\end{aligned}
$$

We need also the notion of dual function.

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

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

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

2.3 The initial part of a relation

In the following, 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 $L(R)$, 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

(17) Definition. [29] The initial part of a relation R, denoted $L(R)$, is given by :

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

where x takes its value in the set of the vectors (by Theorem 2.1(2.1), $L(R)$ is a vector).

(see [28, 29]); in other words, $L(R)$ is the least fixed point of the $\subseteq$-monotonic function $g(x) := R \lhd x$, where x is a vector (the least fixed point of g exists since the set of vectors is also a complete lattice [29]). A relation R is said to be *progressively finite* iff $L(R) = L$, 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 : L for *loop* because, in the program semantics, $L(R)$ represents the set of states from which no infinite loop is possible.)

We find in [36], an equivalent definition of $L(R)$ on the set of relations instead of vectors. This definition is given in the next proposition

(18) Proposition. *Let* R *be a relation.*

$$
\begin{aligned}
(a) \quad L(R) &= \bigcap\{X \mid R \lhd X = X\} \\
&= \bigcap\{X \mid R \lhd X \subseteq X\} \\
&= \bigcap\{X \mid R\overline{X} = \overline{X}\} \\
&= \mu(X \mapsto R \lhd X)
\end{aligned}
$$

and

ISSN 2347-1921

$$(b)\quad \overline{L(R)} = \bigcup\{X \mid X = RX\}$$
$$= \bigcup\{X \mid X \subseteq RX\}$$
$$= \nu(X \mapsto RX).$$

These results can be easily deduced from the Equations 1, 2.3(a), of the definition of $\lhd$ and certain Boolean laws.

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

(19) Definition. A relation R is said *progressively finite* **[29] iff $L(R) = L$.**

By using the results of the Proposition 2.3, we have :

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

The next proposition presents some properties of the initial part of a relation [36]. The properties (a), (b) and (c) can be found also in [29]. The proofs (a) and (c) are different from those given in [29].

(21) Proposition. *Let Q and R be relations.*

- $L(R) \subseteq R^* \overline{RL}$,
- $Q \subseteq R \Rightarrow L(R) \subseteq L(Q)$,
- $R \lhd L(R) = L(R)$ (equivalent to $R\overline{L(R)} = \overline{L(R)}$),
- $L(R) = L(R^+)$,
- $R^* \lhd L(R) = R^+ \lhd L(R) = L(R)$ (equivalent to $R^*\overline{L(R)} = R^+\overline{L(R)} = \overline{L(R)}$),
- Q progressively finite $\Rightarrow$ $Q \cap R$ progressively finite,
- $R \cap L(R)$ is progressively finite.

Proof.

(a)
$$\begin{array}{ll} & \mathcal{L}(R) \\ = & \quad \langle \text{ Proposition 18(a). } \rangle \\ & \mu(X \mapsto R \lhd X) \\ \subseteq & \quad \langle \text{ Lemma 5(g) and Proposition 8(a). } \rangle \\ & \mu(X \mapsto RX \cup \overline{RL}) \\ = & \quad \langle \text{ Equation 13. } \rangle \\ & R^*\overline{RL} \end{array}$$

(b)
$$\begin{array}{ll} & Q \subseteq R \\ \Rightarrow & \quad \langle \text{ Lemma 5(h). } \rangle \\ & \forall X \ : \ R \lhd X \subseteq Q \lhd X \\ \Rightarrow & \quad \langle \text{ Proposition 8(a). } \rangle \\ & \mu(X \mapsto R \lhd X) \subseteq \mu(X \mapsto Q \lhd X) \\ \Leftrightarrow & \quad \langle \text{ Proposition 18(a). } \rangle \\ & \mathcal{L}(R) \subseteq \mathcal{L}(Q) \end{array}$$

By Proposition 18(a), $L(R)$ is the least relation X verifying $R \lhd X = X$ and, by complementation, we find the other expression.

ISSN 2347-1921

(d)
$$
\begin{aligned}
& R^* \triangleleft \mathcal{L}(R) \\
= & \quad \langle \text{ Equation 11(b) } \rangle \\
& (I \cup R^+) \triangleleft \mathcal{L}(R) \\
= & \quad \langle \text{ Lemma 5(d) and } I \triangleleft \mathcal{L}(R) = \mathcal{L}(R). \rangle \\
& \mathcal{L}(R) \cap R^+ \triangleleft \mathcal{L}(R) \\
= & \quad \langle \text{ Proposition 21(d). } \rangle \\
& \mathcal{L}(R) \cap R^+ \triangleleft \mathcal{L}(R^+) \\
= & \quad \langle \text{ Proposition 21(c). } \rangle \\
& \mathcal{L}(R) \cap \mathcal{L}(R^+) \\
= & \quad \langle \text{ Proposition 21(d). } \rangle \\
& \mathcal{L}(R) \\
= & \quad \langle \text{ Proposition 21(d). } \rangle \\
& \mathcal{L}(R^+) \\
= & \quad \langle \text{ Proposition 21(c). } \rangle \\
& R^+ \triangleleft \mathcal{L}(R^+) \\
= & \quad \langle \text{ Proposition 21(d). } \rangle \\
& R^+ \triangleleft \mathcal{L}(R)
\end{aligned}
$$

The other expression can be easily found by complementation.

(e)
$$
\begin{aligned}
& Q \cap R \subseteq Q \\
\Rightarrow & \quad \langle \text{ Proposition 21(b). } \rangle \\
& \mathcal{L}(Q) \subseteq \mathcal{L}(Q \cap R) \\
\Rightarrow & \quad \langle \mathcal{L}(Q) = L. \rangle \\
& \mathcal{L}(Q \cap R) = L
\end{aligned}
$$

(f) We use the Equation 20. Let X be a certain relation.

$$
\begin{aligned}
& X \subseteq (R \cap \mathcal{L}(R))X \\
\Leftrightarrow & \quad \langle \text{ Theorem 2(n), } \mathcal{L}(R) \text{ is a vector by the Definition 17. } \rangle \\
& X \subseteq RX \cap \mathcal{L}(R) \\
\Leftrightarrow & X \subseteq RX \text{ and } X \subseteq \mathcal{L}(R) \\
\Rightarrow & \quad \langle \text{ Equations 18(b) and 6(b,e). } \rangle \\
& X \subseteq \overline{\mathcal{L}(R)} \text{ and } X \subseteq \mathcal{L}(R) \\
\Leftrightarrow & X = \emptyset
\end{aligned}
$$

The next theorem involves the function $f(X) := Q \cup PX$, which is closely related to the description of iterations. The theorem highlights the importance of progressive finiteness in the simplification of fixed point-related properties.

(22) Theorem. [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 $\nu(f) = \mu(f) = P^*Q$.*

Proof. Since the function f is monotonic and that the algebra is complete, by Knaster-Tarski theorem (7), f has a least and a greatest fixed point. Whence, we deduce that if f has a unique fixed point, then the least and the greatest fixed point coincide. By Equation13, the least fixed point of f is P^*Q. So, It's suffisant to prove that if P is progressively finite, the greatest fixed point of f is also equal to P^*Q; Let X be a fixed point of f ; We will prove $X \subseteq P^*Q$, which is equivalent to $X \cap \overline{P^*Q} = \emptyset$:

ISSN 2347-1921

$$
\begin{array}{ll}
& X \cap \overline{P^*Q} \\
= & \quad \langle\ X = Q \cup PX.\ \rangle \\
& (Q \cup PX) \cap \overline{P^*Q} \\
= & \quad \langle\ Q \subseteq P^*Q.\ \rangle \\
& PX \cap \overline{P^*Q} \\
\subseteq & \quad \langle\ \text{Theorem 2(m) (Dedekind rule).}\ \rangle \\
& P(X \cap \breve{P}\overline{P^*Q}) \\
\subseteq & \quad \langle\ PP^*Q \subseteq P^*Q \Leftrightarrow \breve{P}\overline{P^*Q} \subseteq \overline{P^*Q}\ \text{(Schröder rule).}\ \rangle \\
& P(X \cap \overline{P^*Q})
\end{array}
$$

We proved $X \cap \overline{P^*Q} \subseteq P(X \cap \overline{P^*Q})$ and , as the relation P is progressively finite, by the Equation 7 we have $X \cap \overline{P^*Q} = \emptyset$, then the result.

Because YL is a vector characterizing the domain of relation Y , the following theorem qualifies the range of domains of fixed points of d . This range is fully determined by the vectors $A(P,Q)$ and $L(P)$. We note that in the case when the relation P is progressively finite ($L(P) = L$), we find the results of Theorem 2.3.

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

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

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

Proof. By Equation 13, P^*Q is the least fixed point of f ; then $P^*Q \subseteq Y$. Let us prove the second inclusion. First, we show that $P^*Q \cup \overline{L(P)}$ is a fixed point of f .

$$
\begin{array}{ll}
& f(P^*Q \cup \overline{\mathcal{L}(P)}) \\
= & \quad \langle\ f(X) = Q \cup PX.\ \rangle \\
& Q \cup P(P^*Q \cup \overline{\mathcal{L}(P)}) \\
= & \quad \langle\ \text{Theorem 2(h).}\ \rangle \\
& Q \cup PP^*Q \cup P\overline{\mathcal{L}(P)} \\
= & \quad \langle\ \text{Theorem 2(h), } I \cup PP^* = P^*\text{, Proposition 21(c).}\ \rangle \\
& P^*Q \cup \overline{\mathcal{L}(P)}
\end{array}
$$

Now, we prove that every fixed point Y of f verifies $Y \subseteq P^*Q \cup \overline{L(P)}$

$$
\begin{array}{ll}
& Y \\
\subseteq & \quad \langle\ Y = f(Y) = Q \cup PY.\ \rangle \\
& Q \cup PY \cup \overline{\mathcal{L}(P)} \\
= & \quad \langle\ \text{Theorem 2(c).}\ \rangle \\
& Q \cup PY \cap \mathcal{L}(P) \cup \overline{\mathcal{L}(P)} \\
= & \quad \langle\ \mathcal{L}(P) \text{ is a vector, Theorem 2(n).}\ \rangle \\
& Q \cup \overline{\mathcal{L}(P)} \cup (P \cap \mathcal{L}(P))Y
\end{array}
$$

By using the result, we have

ISSN 2347-1921

$$
\begin{array}{ll}
 & Y \\
\subseteq & \quad \langle \text{ Law 6(e). } \rangle \\
 & \nu(X \mapsto Q \cup \overline{\mathcal{L}(P)} \cup (P \cap \mathcal{L}(P))X) \\
= & \quad \langle \text{ Theorem 21(g) and Proposition 22. } \rangle \\
 & (P \cap \mathcal{L}(P))^*(Q \cup \overline{\mathcal{L}(P)}) \\
\subseteq & \quad \langle \text{ } ^* \text{ monotonic. } \rangle \\
 & P^*(Q \cup \overline{\mathcal{L}(P)}) \\
= & \quad \langle \text{ Theorem 2(h). } \rangle \\
 & P^*Q \cup P^*\overline{\mathcal{L}(P)} \\
= & \quad \langle \text{ Proposition 21(e). } \rangle \\
 & P^*Q \cup \overline{\mathcal{L}(P)}
\end{array}
$$

The next corollary is about the fixed points of the function $g(X) := Q \cap P \lhd X$.

(24) Corollary. *Every fixed point* Y *of* $g(X) := Q \cap P \lhd X$ *verifies*

$$P^* \lhd Q \cap L(P) \subseteq Y \subseteq P^* \lhd Q$$

$P^* \lhd Q \cap L(P)$ and $P^* \lhd Q$ 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 15) of $f(X) := \overline{Q} \cup PX$. By Lemma 2.2, $\overline{Y}$ is a fixed point of f . By Theorem 2.3, $\overline{Y}$ verifies

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

By applying DeMorgan Laws, this is equivalent to

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

Finaly, by Definition 4,

$$P^* \lhd Q \cap L(P) \subseteq Y \subseteq P^* \lhd Q.$$

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

We introduce the next Abbreviations :

(25) Abbreviation. Let P **and** Q **be relations. The Abbreviations** d , d_L **and** $A(P,Q)$ **are defined as follows (** x **is a vector) :**

$$d(X) := (Q \cup PX) \cap P \lhd XL,$$

$$d_L(x) := (PL \cup QL) \cap P \lhd x,$$

$$A(P,Q) := P^* \lhd (PL \cup QL)$$

$$S := P^*Q \cap A(P,Q) \cap L(P).$$

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

2.4 Intuition

- The function d can be considered as a generalization of the semantic function of the while loop $do\ P \to Q\ od$ but with the hypothesis $PL \cap QL$ is not necessarly empty.

ISSN 2347-1921

- In a nondeterministic while loop while P do Q, P is iteratively applied to a state s until Q is verified. As, P is nondeterministic s can have many outputs If it exists among these outputs a state which can lead outside the domain of P or whose of Q, so this state is exculded (abnormal termination of the loop) . So, $A(P,Q)$ represents the states from which no abnormal termination is possible.
- We note that S is an intersection of three terms; P^*Q, $A(P,Q)$ and $L(P)$. By taking in consideration the intuition behind these terms, it is easy to see that the relation 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) they are excluded by the terms $A(P,Q)$ and $L(P)$.

The following lemma presents the relationship between the fixed points of the functions d and d_L (Abbreviation 2.3).

(26) Lemma. *If Y is a fixed point of d then YL is a fixed point of d_L.*

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

$$
\begin{aligned}
& d_L(YL) \\
= & \quad \langle \text{ Definition of } d_L \text{ (25). } \rangle \\
& (PL \cup QL) \cap P \triangleleft YL \\
= & \quad \langle \ L = YL \cup \overline{YL}. \ \rangle \\
& (QL \cup P(YL \cup \overline{YL}) \cap P \triangleleft YL \\
= & \quad \langle \text{ Boolean laws, Theorem 2(i), Abbreviation 25 and Lemma 5(f) . } \rangle \\
& (Q \cup PY)L \cap P \triangleleft YL \\
= & \quad \langle \text{ Theorem 2(n). } \rangle \\
& ((Q \cup PY) \cap P \triangleleft YL)L \\
= & \quad \langle \text{ Definition of } d \text{ (25) and } d(Y) = Y. \ \rangle \\
& YL
\end{aligned}
$$

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

(27) Theorem. *If Y is a fixed point of d , then*

1. $A(P,Q) \cap L(P) \subseteq YL \subseteq A(P,Q)$,
2. $A(P,Q) \cap L(P)$ and $A(P,Q)$ are fixed points of d_L .

Proof.

1. By Lemma 26, YL is a fixed point of d_L . By taking $Q := PL \cup QL$ in the Corrolary 24, we find,

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

by using Abbreviation 25, we find

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

2. By Corollary 24, we deduce that $A(P,Q) \cap L(P)$ and $A(P,Q)$ are fixed points of d_L .

The next theorem characterizes the domain of S (25). This domain is the set of points for which normal termination is guaranteed (no possibility of abnormal termination or infinite loop).

(28) Theorem. *Let S given by the Abbreviation 25. We have*

$$SL = A(P,Q) \cap L(P).$$

Proof.

$$
\begin{array}{ll}
& SL \\
= & \quad \langle \text{ Abbreviation 25. } \rangle \\
& (P^*Q \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P))L \\
= & \quad \langle \text{ Theorem 2(n). } \rangle \\
& P^*QL \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P) \\
= & \quad \langle \text{ Abbreviation 25 and Lemma 5(a). } \rangle \\
& (P^*QL \cup P^*\overline{PL \cup QL}) \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P) \\
= & \quad \langle \text{ Theorem 2(b,h). } \rangle \\
& P^*(QL \cup \overline{PL} \cap \overline{QL}) \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P) \\
= & \quad \langle \text{ Theorem 2(c). } \rangle \\
& P^*(QL \cup \overline{PL}) \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P) \\
= & \quad \langle \text{ Theorem 2(h). } \rangle \\
& P^*QL \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P) \cup P^*\overline{PL} \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P) \\
= & \quad \langle \text{ Proposition 21(a) and Boolean Law. } \rangle \\
& \mathcal{A}(P,Q) \cap \mathcal{L}(P)
\end{array}
$$

2.5 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 ($\diamond$). We give the definitions and needed properties of these operations. For more details on relational demonic semantics and demonic operators, see [4, 8, 6, 7, 12, 13, 36].

(29) Definition. We say that a relation Q *refines* **a relation** R **[22], denoted by** $Q \sqsubseteq R$**, iff**

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

(30) Proposition. *The greatest lower bound (wrt* $\sqsubseteq$*) of relations* Q *and* R *is*

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

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.

Proof. See [9, 13].

Secondly, demonic meet: The existence condition simply means that on the intersection of their domains, Q and R have to agree for at least one value. In the following, we will show that S (Abbreviation 25) is the greatest fixed point with respect to $\sqsubseteq$ of d (Abbreviation 25). In other words we want to prove

(31) Proposition.

$$
\begin{array}{l}
(a)\ Q \text{ deterministic} \Rightarrow Q \diamond R = QR, \\
(b)\ P \text{ deterministic} \Rightarrow P \diamond (Q \sqcap R) = PQ \sqcap PR, \\
(c)\ R \text{ total} \Rightarrow Q \diamond R = QR, \\
(d)\ PL \cap QL = \emptyset \Rightarrow (P \cup Q) \diamond R = P \diamond R \cup Q \diamond R, \\
(e)\ PL \cap QL = \emptyset \Rightarrow P \sqcap Q = P \cup Q, \\
(f)\ P \sqsubseteq Q \ \wedge \ R \sqsubseteq S \ \wedge \\
\quad P \cap RL \cup R \cap SL \subseteq Q \cap PL \cup S \cap RL \Rightarrow P \cup R \sqsubseteq Q \cup S.
\end{array}
$$

Proof. It is easy to verify the condition about the domains ($(Q \cup S)L \subseteq (P \cup R)L$).

ISSN 2347-1921

For the other condition we have,

$$
\begin{array}{ll}
 & (P \cup R) \cap (Q \cup S)L \\
= & \quad \langle \text{ Boolean laws. } \rangle \\
 & P \cap QL \cup P \cap SL \cup R \cap RL \cup R \cap SL \\
\subseteq & \quad \langle \text{ Hypothesis. } \rangle \\
 & Q \cup S
\end{array}
$$

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

(32) Definition. $Q \ \square \ R := QR \cap Q \lhd RL$.

A pair (s,t) belongs to $Q \square R$ 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 $Q \square R = \{(1,3)\}$; the pair $(0,0)$, which belongs to QR, does not belong to $Q \square R$, 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 .

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.

(33) Lemma. *Let Q, R be relations and u ,v vectors. We have,*

$$
\begin{array}{l}
(a)\ (v \cap Q) \circ R = v \cap Q \circ R, \\
(b)\ R \circ v = RL \cap R \lhd v, \\
(c)\ Q \circ (v \cap R) = Q \circ v \cap Q \circ R, \ [16,\ 30] \\
(d)\ Q \sqsubseteq R \Leftrightarrow v \cap Q \sqsubseteq v \cap R \wedge \overline{v} \cap Q \sqsubseteq \overline{v} \cap R, \ ?? \\
(e)\ u \sqsubseteq v \ \Rightarrow \ P \lhd u \cap Q \sqsubseteq P \lhd v \cap Q, \\
(f)\ R \sqsubseteq v \cap R, \\
(g)\ v \cap Q \sqsubseteq R \ \Rightarrow \ Q \sqsubseteq R.
\end{array}
$$

In what follows, we will present some inetresting properties verified by S and relations that have the same domain as S .

(34) Lemma. Let R be a relation and S given by Abbreviation 25. The next equation is satisfied

$$
\begin{array}{l}
(a)\ R \circ S = RP^*Q \cap R \lhd (\mathcal{A}(P,Q) \cap \mathcal{L}(P)), \\
(b)\ (PL \cup QL) \cap P \lhd (\mathcal{A}(P,Q) \cap \mathcal{L}(P)) = \mathcal{A}(P,Q) \cap \mathcal{L}(P).
\end{array}
$$

Proof.

$$
\begin{array}{ll}
(a) & R \circ S \\
= & \quad \langle \text{ Definition 32 and Theorem 28. } \rangle \\
 & RS \cap R \lhd (\mathcal{A}(P,Q) \cap \mathcal{L}(P)) \\
= & \quad \langle \text{ Abbreviation 25 and Theorem 2(c). } \rangle \\
 & R(P^*Q \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P) \cup \overline{\mathcal{A}(P,Q) \cap \mathcal{L}(P)} \cap R \lhd (\mathcal{A}(P,Q) \cap \mathcal{L}(P)) \\
= & \quad \langle \text{ Boolean law. } \rangle \\
 & RP^*Q \cap R \lhd (\mathcal{A}(P,Q) \cap \mathcal{L}(P))
\end{array}
$$

ISSN 2347-1921

(b) $(PL \cup QL) \cap P \lhd (\mathcal{A}(P,Q) \cap \mathcal{L}(P))$
$= \quad \langle$ Lemma 5(c). $\rangle$
$(PL \cup QL) \cap P \lhd \mathcal{A}(P,Q) \cap P \lhd \mathcal{L}(P)$
$= \quad \langle$ Abbreviation 25 and Proposition 21(c). $\rangle$
$(PL \cup QL) \cap P \lhd (P^* \lhd (PL \cup QL)) \cap \mathcal{L}(P)$
$= \quad \langle$ Definition 4 and Lemma 5(e). $\rangle$
$I \lhd (PL \cup QL) \cap PP^* \lhd (PL \cup QL) \cap \mathcal{L}(P)$
$= \quad \langle$ Lemma 5(d). $\rangle$
$(I \cup PP^*) I \lhd (PL \cup QL) \cap \mathcal{L}(P)$
$= \quad \langle$ Equation 10 and Abbreviation 25. $\rangle$
$\mathcal{A}(P,Q) \cap \mathcal{L}(P)$

In the following theorem, we will show that S is a fixed point of d (Abbreviation 25).

(35) $d(X) = Q \cap P \lhd XL \cup P \sqcap X$

Faire le lien ici avec les travaux anterieurs.

(35) Theorem.

S (Abbreviation 25) is a fixed point of d .

Proof.

$d(S)$
$= \quad \langle$ Abbreviation 25 and Theorem 28. $\rangle$
$(Q \cup PS) \cap P \lhd (\mathcal{A}(P,Q) \cap \mathcal{L}(P))$
$= \quad \langle$ Boolean law. $\rangle$
$Q \cap P \lhd (\mathcal{A}(P,Q) \cap \mathcal{L}(P)) \cup PS \cap P \lhd (\mathcal{A}(P,Q) \cap \mathcal{L}(P))$
$= \quad \langle$ $Q \subseteq QL \cup PL$ and Theorem 28. $\rangle$
$Q \cap (PL \cup QL) \cap P \lhd (\mathcal{A}(P,Q) \cap \mathcal{L}(P)) \cup PS \cap P \lhd SL$
$= \quad \langle$ Definition 32. $\rangle$
$Q \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P) \cup P \circ S$
$= \quad \langle$ Lemma 34(a). $\rangle$
$Q \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P) \cup PP^*Q \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P)$
$= \quad \langle$ Boolean law. $\rangle$
$(Q \cup PP^*Q) \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P)$
$= \quad \langle$ Theorem 2(i) and Equation 10. $\rangle$
$P^*Q \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P)$
$= \quad \langle$ Abbreviation 25. $\rangle$
S

3 A programming theorem

The following theorem is a generalization to a nondeterministic context of the *while statement verification rule* of Mills [23, 24]. It shows that the greatest fixed point W of d (Abbreviation 25) 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* [30]) in a predicative programming set-up.

(37) Theorem. *W is the least fixed point wrt $\subseteq$ of d (Abbreviation 25) ($W = \mu_{\subseteq}(d)$) iff*

$$(a) \quad W = d(W),$$
$$(b) \quad WL \subseteq L(P).$$

Proof.

$(\Rightarrow)$: As W is the least fixed point of d then, (a) is evident. Since $W = \mu(d) \subseteq S$, then $WL = \mu(d)L \subseteq SL$, by using Theorem 28, We have $WL \subseteq L(P)$.

$(\Leftarrow)$: By Hypothesis (a), W is a fixed point of d. Then, by Theorem 27, $A(P,Q) \cap L(P) \subseteq WL \subseteq A(P,Q)$. But, by using Hypothesis (b), $WL \subseteq L(P)$, then $WL = A(P,Q) \cap L(P)$.

$$
\begin{array}{ll}
 & W \\
= & d(W) \\
= & \quad \langle \text{ Abbreviation 25. } \rangle \\
 & (Q \cup PW) \cap P \triangleleft WL \\
= & \quad \langle \ (Q \cup PW) \subseteq (PL \cup QL) \text{ and Theorem 28. } \rangle \\
 & (Q \cup PW) \cap (PL \cup QL) \cap P \triangleleft (\mathcal{A}(P,Q) \cap \mathcal{L}(P,Q)) \\
= & \quad \langle \text{ Lemma 34(b) } \rangle \\
 & (Q \cup PW) \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P) \\
= & \quad \langle \text{ Theorem 2(n). } \rangle \\
 & Q \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P) \cup (P \cap \mathcal{A}(P,Q) \cap \mathcal{L}(P))W
\end{array}
$$

Then W is a fixed point of the function $g(X) := Q \cap A(P,Q) \cap L(P) \cup (P \cap A(P,Q) \cap L(P))X$. Since, by Proposition 21(g,f), $P \cap A(P,Q) \cap L(P)$ is progressively finite. Invoking Theorem 22 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 S is the least fixed point of d wrt $\subseteq$ ($S = \mu_{\subseteq}(d)$).

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

Proof. It suffices to prove that S verifies the conditions of the Theorem 37. By Theorem 36, S is a fixed point of d. So, the condition (a) is satisfied. By Theorem 28, $SL = A(P,Q) \cap L(P)$. So, the condition (b) is easily verified.

4 Application

In Mills approach, the semantics W of a deterministic loop $do\ g \to B\ od$ is given as the least fixed point (wrt $\subseteq$) of the function

$$w_a(X) := g^{\cdot} \cup gBX,$$

where the partial identity g is the semantics of the loop condition g and the relation B is the semantics of the loop body B. The loop $do\ g \to B\ od$ is deterministic if B is deterministic. As we consider a complete relation algebra (see the section 2) and as the function w_a is monotonic (wrt $\subseteq$), by Theorem 7 the least fixed point W of w_a exists and $W = (gB)^* g^{\cdot}$. Calculating the relational abstraction (semantics) of a loop is difficult, but showing the correctness of any candidate abstraction is much easier. For functional programs, Mills [23, 24] has described a checking method known as the *while statement verification rule*. In a nondeterministic context, the abstraction is calculated by considering the worst behavior of the program (*demonic semantics*) [36]. Given a loop condition and a loop body, theorem 37 can be used to verify if a relation W is indeed the semantics of the loop as it will be shown in the next example 41.

By using a similar intuition as in the deterministic case, we have to prove the next equation

(40) $$W = \coprod \{X \mid X \sqsubseteq g^{\sim} \cup (gB) \sqcap X\}.$$

To solve this equation we will use Theorem 3, where $P := gB$ and $Q := g^{\sim}$. Notice that $PL \cap QL = \emptyset$.

The following example is an application of this theorem. It is rather contrived, but it is simple and fully illustrate the various cases that may happen. Consider the following loop, where the unique variable n ranges over the set Z of integers [13] :

ISSN 2347-1921

(41) Example.

```
do n > 0 → if  n = 1 → n := 1   ▯  n = 1 → n := −3
           ▯  n = 3 → n := 2   ▯  n = 3 → n := −1
           ▯  n ≥ 4 → n := −4
           fi
od
```

Notice $n > 0$ such that $n \ mod \ 4 = 1$ may lead to termination with a final value $n' = -3$, but may lead to an infinite loop over the value $n = 1$ these initial values of n do not belong to the domain of the relation giving the semantics of the loop. Note also that all $n > 0$ such that $n \ mod \ 4 = 3$ may lead to termination with a final value $n' = -1$, but may also lead to avalue $n = 2$, for which the loop body is not defined (by the semantics of **if fi**; these n do not belong to the domain of W. Because they also lead to $n = 2$, all $n > 0$ such that $n \ mod \ 4 = 2$ do not belong to the domain of W.

The semantics of the loop condition is given by:

(42)
$$g = \{n > 0 \wedge n' = n\} \quad (g^{\sim} = \{n \leq 0 \wedge n' = n\}). \tag{11}$$

The boody of the loop is :

(43)
$$\begin{aligned} B = \ & \{n = 1 \wedge n' = n\} \ \square \ (\{n' = 1\} \sqcup \{n' = -3\}) \\ & \sqcap \{n = 3 \wedge n' = n\} \ \square \ (\{n' = 2\} \sqcup \{n' = -1\}) \\ & \sqcap \{n \geq 4 \wedge n' = n\} \ \square \ \{n' = n - 4\}. \end{aligned} \tag{12}$$

This expression can be simplified as follows :

(44)
$$\begin{aligned} B = \ & (\{n = 1 \wedge n' = 1\} \sqcup \{n = 1 \wedge n' = -3\}) \\ & \sqcap (\{n = 3 \wedge n' = 2\} \sqcup \{n = 3 \wedge n' = -1\}) \\ & \sqcap \{n \geq 4 \wedge n' = n - 4\}). \end{aligned} \tag{13}$$

Also, by using 31(a,b), it is easy to see that $g \ \square \ B = gB = B$. Let's now show that

(45)
$$W := \{n \leq 0 \wedge n' = n \vee n > 0 \wedge n \ mod \ 4 = 0 \wedge n' = 0\}$$

is the abstraction (semantics) of the loop. By 31(e), W is equal to :

$$W = \{n \leq 0 \wedge n' = n\} \sqcap \{n > 0 \wedge n \ mod \ 4 = 0 \wedge n' = 0\}.$$

We have to verify the conditions (a) and (b) of Theorem 37, *i.e* $d(W) = W$ and $WL \subseteq L(gB)$.

Notice that,

(46)
$$W = g^{\sim} \sqcap \{n > 0 \wedge n \ mod \ 4 = 0 \wedge n' = 0\}.$$

So, to show (a), it is suffisant to prove

(47)
$$B \ \square \ W = \{n > 0 \wedge n \ mod \ 4 = 0 \wedge n' = 0\},$$

as,

$$\begin{aligned} & B \circ W = \{n > 0 \wedge n \bmod 4 = 0 \wedge n' = 0\} \\ \Rightarrow \quad & \langle \ B = g \circ B. \ \rangle \\ & g^{\sim} \sqcap g \circ B \circ W = g^{\sim} \sqcap \{n > 0 \wedge n \bmod 4 = 0 \wedge n' = 0\} \\ \Rightarrow \quad & \langle \text{ Equations 31(e), Abbreviation 25 with } P := gB \text{ and } Q := g^{\sim}, \text{ and Equation 46. } \rangle \\ & d(W) = W \end{aligned}$$

Let's prove the Equation 47.

ISSN 2347-1921

$$
\begin{array}{ll}
 & B \circ W \\
= & \quad \langle \text{ Equations 44. } \rangle \\
 & ((\{n = 1 \wedge n' = 1\} \sqcup \{n = 1 \wedge n' = -3\}) \\
 & \sqcap (\{n = 3 \wedge n' = 2\} \sqcup \{n = 3 \wedge n' = -1\}) \\
 & \sqcap \{n \geq 4 \wedge n' = n - 4\}) \circ W \\
= & \quad \langle \text{ Proposition 31(d,e). } \rangle \\
 & (\{n = 1 \wedge n' = 1\} \sqcup \{n = 1 \wedge n' = -3\}) \circ W \\
 & \sqcap (\{n = 3 \wedge n' = 2\} \sqcup \{n = 3 \wedge n' = -1\}) \circ W \\
 & \sqcap \{n \geq 4 \wedge n' = n - 4\} \circ W \\
= & \quad \langle \text{ Distributivity of } \circ \text{ wrt } \sqcup. \rangle \\
 & (\{n = 1 \wedge n' = 1\} \circ W \sqcup \{n = 1 \wedge n' = -3\} \circ W) \\
 & \sqcap (\{n = 3 \wedge n' = 2\} \circ W \sqcup \{n = 3 \wedge n' = -1\} \circ W) \\
 & \sqcap \{n \geq 4 \wedge n' = n - 4\} \circ W \\
= & \quad \langle \text{ Proposition 31(a) and Expression of } W. \rangle \\
 & (\emptyset \sqcup \{n = 1 \wedge n' = -3\}) \sqcap (\emptyset \sqcup \{n = 3 \wedge n' = -1\}) \\
 & \sqcap \{n > 0 \wedge n \bmod 4 = 0 \wedge n' = 0\} \\
= & \quad \langle \emptyset \text{ is the supremum of the demonic semilattice. } \rangle \\
 & \emptyset \sqcap \emptyset \sqcap \{n > 0 \wedge n \bmod 4 = 0 \wedge n' = 0\} \\
= & \quad \langle \emptyset \text{ is the supremum of the demonic semilattice. } \rangle \\
 & \{n > 0 \wedge n \bmod 4 = 0 \wedge n' = 0\}
\end{array}
$$

This shows part (a) of the theorem. Part (b) can be established informally by noting that the domain of W is

$$\{n \leq 0 \vee n \ mod \ 4 = 0\},$$

and that there is no infinite sequence by gB for any n in the domain of the relation W; in other words, $WL \subseteq L(gB)$

A more satisfying way to show $WL \subseteq L(gB)$ is to calculate $L(gB)$. However, because $L(gB)$ characterizes the domain of guaranteed termination of the associated loop, there is no systematic way to compute it (this would solve the halting problem). To demonstrate termination of the loop from every state in the domain of W, classical proofs based on variant functions or well-founded sets could be given. But formal arguments based on the definition of initial part (Definition 17) can be also used.

We sketch one such argument.

For $k \geq 0$, let $v_k := \{n \leq 0 \vee n \leq 4k \wedge n \ mod \ 4 = 0\}$. Also, let $h(X) := gB \lhd X$.

It is easy to show by induction that the domain of W (Equation 48) is equal to $\bigcup_{k \geq 0} v_k$.

By using the laws of the lattice theory [11], we have :

$$
\begin{array}{ll}
 & WL \\
= & \bigcup_{k \geq 0} v_k \\
\subseteq & \quad \langle \ v_k \subseteq h^{k+1}(\emptyset). \ \rangle \\
 & \bigcup_{k \geq 0} h^{k+1}(\emptyset) \\
\subseteq & \quad \langle \text{ Boolean law. } \rangle \\
 & \bigcup_{k \geq 0} h^{k}(\emptyset) \\
\subseteq & \quad \langle \ h \text{ monotonic and properties of the least fixed point (see [11]). } \rangle \\
 & \bigcap \{X \mid h(X) = X\} \\
= & \quad \langle \text{ Definition 18 and } h(X) = gB \lhd X. \ \rangle \\
 & \mathcal{L}(gB)
\end{array}
$$

In this example, Theorem 3, was used to verify that the guessed semantics W of the loop L was correct, given the semantics c of the loop condition and b of the loop body. The theorem can be used in the other direction. If we are given a specification W, we can guess c and b, and then apply Theorem 3 to verify the correctness of the guess. If it is correct, then a loop of the form do $c \to p$ od, where p is an implementation of b, is correct with respect to W. But in the context of program construction, the theorem is more useful in the other direction: given a specification (relation) W of a loop, one can guess the abstractions t and B of the loop condition and loop body, respectively, and use Theorem 3 to verify that the guess is correct.

5 Conclusion

We presented a theorem that can be also used to find the fixed points of functions of the form $f(X) := (Q \cup PX) \cap P \triangleleft XL$ (no restriction on the domains of P and Q). This theorem can be applied also to the program verification and construction (as in the precedent example). Half of this theorem (the $\Leftarrow$ direction) is also proved by Sekerinski (the *main iteration theorem* [30]) in a predicative programming set-up. Our theorem is more general because there is no restriction on the domains of the relations P and Q.

The approach to demonic input-output relation presented here is not the only possible one. In [18, 19, 20], the infinite looping has been treated by adding to the state space a fictitious state $\perp$ to denote nontermination. In [8, 15, 21, 27], 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.

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, 14].

Acknowledgement

This research project was supported by a grant from the “Research Center of the Center for Female Scientific and Medical Colleges”, Deanship of Scientific Research, King Saud University.

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 and J. van der Woude. Demonic operators and monotype factors. *Mathematical Structures in Comput. Sci. 3*, 4, 417–433, 1993.

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

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

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

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

[9] 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.

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

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

[12] 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.

[13] 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.

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

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

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

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

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

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

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

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

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

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

[24] 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.

[25] 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.

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

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

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

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

[30] 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.

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

[32] F. Tchier. While loop demonic relational semantics monotype/residual style. *2003 International Conference on Software Engineering Research and Practice (SERP'03)*, Las Vegas, Nevada, USA, 23-26, June 2003.

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

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

[35] 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.

[36] 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.