Unifying *wp* and *wlp*

Carroll Morgan* and Annabelle McIver†

Abstract

Boolean-valued predicates over a state space are isomorphic to its characteristic functions into $\{0, 1\}$. Enlarging that range to $\{-1, 0, 1\}$ allows the definition of extended predicates whose associated transformers generalise the conventional *wp* and *wlp*.

The correspondingly extended healthiness conditions include the new 'sub-additivity', an arithmetic inequality over predicates.

Keywords: Formal semantics, program correctness, weakest precondition, weakest liberal precondition, Egli-Milner order.

1 Introduction

The weakest- and weakest liberal preconditions of a program [1] express respectively its total and partial correctness. Neither can be derived from the other: using $[\!]$ for non-deterministic choice, **abort** $[\!]$ **skip** is distinguished from **skip** by *wp* but not by *wlp*, and it is distinguished from **abort** by *wlp* but not by *wp*.

Beyond their use as practical tools for program derivation, however, predicate transformers have an interesting theory in their own right. We contribute to that theory by identifying a new healthiness condition that allows a simple definition of 'extended weakest precondition' *ewp*, generalising both *wp* and *wlp* and distinguishing all three programs above.

The new healthiness condition 'sub-additivity' is numeric rather than logical, and acts uniformly over the extended transformers; in spite of that uniformity it still links the special cases *wp* and *wlp* in the appropriate way. Further, the structure of the new transformers, and the associated programming language semantics, is only slightly more complex mathematically than either of the originals separately.

*Morgan was partially supported during this work by the Department of Computer Science and the SVRC at the University of Queensland.

†Both authors are members of the Programming Research Group at Oxford University: {*carroll,anabel*}@*comlab.ox.ac.uk*. McIver is supported by the EPSRC.

$$
\begin{array}{rcll}
\textbf{abort}.s & := & \{\bot\} & \\
\textbf{skip}.s & := & \{s\} & \\
(\textbf{assign}\ f).s & := & \{f.s\} & \text{for function } f \text{ in } S \to S \\
s'' \in (r_0; r_1).s & := & (\exists s' : r_0.s \cdot s'' = s' = \bot \ \vee \ s'' \in r_1.s') & \\
(r_0 \mathbin{[\!]} r_1).s & := & r_0.s \cup r_1.s & \text{(non-deterministic choice)}
\end{array}
$$

$$
(\textbf{if}\ \pi\ \textbf{then}\ r_0\ \textbf{else}\ r_1\ \textbf{fi}).s \quad := \quad r_0.s \quad \text{if } (\pi.s = 1) \text{ else } \quad r_1.s
$$

$$
(\textbf{mu}\ \mathcal{B}) \quad := \quad \mu\mathcal{B} \qquad \text{for } \sqsubseteq\text{-monotonic } \mathcal{B} : \mathcal{R}S \to \mathcal{R}S
$$

Figure 1: Examples of relational denotations.

2 Extended predicates and *ewp*

We consider a relational model of programs over a state space S, defined as

$$
\mathcal{R}S \quad := \quad S \to \mathbb{P}^+ S_\bot \quad \text{(typical element } r) \ ,
$$

where $S_\bot$ is S together with an extra element $\bot$ and $\mathbb{P}^+$ forms non-empty subsets; an element of $S_\bot$ is *proper* if it is not $\bot$. For program r in $\mathcal{R}S$ and initial state s in S, the function application $r.s$ gives the set of final states that execution of r might produce, including $\bot$ if it might fail to terminate.

We define predicates on S as characteristic functions,

$$
\mathcal{P}S \quad := \quad S \to \{0, 1\} \quad \text{(typical element } \pi) \ .
$$

which view is isomprphic to the usual, and later allows access to predicate arithmetic. For any $S' \subseteq S$ we define the predicate $\chi_{S'}.s \ := \ 1$ if $(s \in S')$ else 0.

Fig. 1 gives relational semantics for a simple language with non-determinism, in which the least fixed point $\mu\mathcal{B}$ is taken over the order $\sqsubseteq$, defined as follows for $\mathbb{P}^+ S_\bot$ and $\mathcal{R}S$:

$$
\begin{array}{rcll}
S_0 \sqsubseteq S_1 & := & \quad\ \bot \notin S_0 \Rightarrow S_1 \subseteq S_0 & \text{(Egli-Milner on sets)} \\
 & & \wedge \quad S_0 - \{\bot\} \subseteq S_1 & \\
r_0 \sqsubseteq r_1 & := & r_0.s \sqsubseteq r_1.s \ \text{ for all } s \ . & \text{(pointwise extension)}
\end{array}
\tag{1}
$$

See *e.g.* Nelson [4] on the Egli-Milner order, where it is shown that any $\mathcal{B}$ made from our programming operators is $\sqsubseteq$-monotonic.

The *weakest precondition* of program r with respect to postcondition π is written $wp.r.\pi$; and is usually defined so that for initial state s we have $wp.r.\pi.s = 0$ precisely when either

$$
\bot \in r.s \qquad \text{or} \qquad \pi.s' = 0 \text{ for some proper } s' \text{ in } r.s.
$$

Letting $\pi_\bot$ extend π to domain $S_\bot$ with $\pi_\bot.\bot := 0$, we avoid an explicit check for $\bot$ and conclude that

$$
wp.r.\pi.s \quad = \quad (\bigwedge_{r.s} \pi_\bot) \ , \tag{2}
$$

where $(\wedge_X f)$ denotes the infimum (in the usual arithmetic $\leq$-order) of function f over set X.

For the *weakest liberal precondition* the usual definition ignores $\bot$, so that $wlp.r.\pi.s$ is 0 simply when $\pi.s' = 0$ for some proper s' in $r.s$. Thus

$$\begin{array}{ll} & wlp.r.\pi.s = 0 \\ \text{iff} & (\exists s' \colon r.s \cdot s' \neq \bot \wedge \pi.s' = 0) \\ \text{iff} & (\exists s' \colon r.s \cdot (\mathbf{1} - \pi)_\bot.s' = 1) \\ \text{iff} & (\vee_{r.s}(\mathbf{1} - \pi)_\bot) = 1 \ , \end{array}$$

with $\mathbf{1}$ the constant function and $\vee$ taking $\leq$-suprema. That gives

$$wlp.r.\pi.s \quad = \quad 1 - (\bigvee_{r.s}(\mathbf{1} - \pi)_\bot) \ ,$$

whence (2) and $r.s \neq \emptyset$ allow us to continue

$$= \quad 1 + (\bigwedge_{r.s}(\pi - \mathbf{1})_\bot) \quad = \quad 1 + wp.r.(\pi - \mathbf{1}).s \tag{3}$$

provided we extend *wp* to accept 'predicates' returning values in $\{-1, 0, 1\}$.

Thus our principal motivation for the arithmetic rather than logical view is that we can define *extended predicates* and the *extended weakest precondition*

$$\begin{array}{rcl} \mathcal{E}S & := & S \to \{-1, 0, 1\} \quad \text{(typical element } \varepsilon) \\ ewp.r.\varepsilon.s & := & (\wedge_{r.s}\, \varepsilon_\bot) \ , \end{array} \tag{4}$$

with $\mathcal{E}S$ ordered pointwise by $\leq$ over numbers. Encouraged by (2) and (3), we continue

$$\left.\begin{array}{rcl} wp.r.\pi & := & ewp.r.\pi \\ wlp.r.\pi & := & \mathbf{1} + ewp.r.(\pi - \mathbf{1}) \end{array}\right\} \quad \text{for } \mathbf{0} \leq \pi \ , \tag{5}$$

so unifying *wp* and *wlp*.

3 Extended healthiness conditions

For standard predicate transformers p in $\mathcal{P}S \to \mathcal{P}S$, the healthiness conditions

strictness	$p.\mathbf{0} = \mathbf{0}$	
monotonicity	$p.\pi_0 \leq p.\pi_1$ if $\pi_0 \leq \pi_1$	
positive conjunctivity	$p.(\wedge \Pi) = (\bigwedge_{\pi \in \Pi} p.\pi)$	for non-empty set Π of predicates

are necessary and sufficient for p to be $wp.r$ for some r in $\mathcal{R}S$ [2]. We define the extended-predicate transformers as

$$\mathcal{T}S \quad := \quad \mathcal{E}S \to \mathcal{E}S \quad \text{(typical element } t) \ ,$$

and note that those conditions — extended from $\mathcal{P}S$ to $\mathcal{E}S$ — are still necessary (all elements of the image $ewp.\mathcal{R}S$ satisfy them).

They are no longer sufficient, however: the extended-predicate transformer defined as $t.\varepsilon := \varepsilon \vee \mathbf{0}$ satisfies them but cannot be expressed as $ewp.r$ for any r, for it behaves like **skip** for non-negative postconditions and like **abort** for the others. We add the healthiness condition that for all t

$$
\begin{array}{rcl}
ewp.\mathbf{abort}.\varepsilon & := & \mathbf{0} \\
ewp.\mathbf{skip}.\varepsilon & := & \varepsilon \\
ewp.(\mathbf{assign}\ f).\varepsilon.s & := & \varepsilon.(f.s) \\
ewp.(r_0; r_1).\varepsilon & := & ewp.r_0.(ewp.r_1.\varepsilon) \\
ewp.(r_0 \parallel r_1) & := & ewp.r_0 \wedge ewp.r_1 \\
\\
ewp.(\mathbf{if}\ \pi\ \mathbf{then}\ r_0\ \mathbf{else}\ r_1\ \mathbf{fi}).s & & \\
 & := & ewp.r_0.s \quad \text{if } (\pi.s = 1) \text{ else } \quad ewp.r_1.s \\
\\
ewp.(\mathbf{mu}\ \mathcal{B}) & := & \mu\mathcal{F} \qquad \text{where } \mathcal{F}.(ewp.r) := ewp.(\mathcal{B}.r)
\end{array}
$$

Figure 2: *ewp*-semantics for a simple language .

sub-additivity $\quad t.(\varepsilon_0 + \varepsilon_1) \geq t.\varepsilon_0 + t.\varepsilon_1 \qquad \text{for } \mathbf{-1} \leq \varepsilon_0 + \varepsilon_1 \leq \mathbf{1}$,

which with strictness excludes the anomaly above by the following contradiction:

$$0 \;=\; t.\mathbf{0}.s \;=\; t.(\mathbf{1}-\mathbf{1}).s \;\geq\; t.\mathbf{1}.s + t.(\mathbf{-1}).s \;=\; 1+0 \;=\; 1\ . \qquad (6)$$

Sub-additivity is satisfied by all $ewp.r$, and in Sec. 5 we show that the four conditions jointly characterise $ewp.\mathcal{R}S$.

As an example of sub-additivity in action, we prove the equality

$$t.\pi \quad = \quad (\mathbf{1} + t.(\pi - \mathbf{1})) \;\wedge\; t.\mathbf{1} \qquad \text{for } \mathbf{0} \leq \pi \qquad (7)$$

which encodes the familiar property $wp.r.\pi = wlp.r.\pi \wedge wp.r.\mathbf{1}$. Note first that from strictness and monotonicity we have $\varepsilon \geq \mathbf{0} \Rightarrow t.\varepsilon \geq \mathbf{0}$ (and $\varepsilon \leq \mathbf{0} \Rightarrow t.\varepsilon \leq \mathbf{0}$). Then for arbitrary s, when $t.\mathbf{1}.s = 0$ we have $t.\pi.s = 0$ as required; when $t.\mathbf{1}.s = 1$ we have

$$
\begin{array}{lll}
 & t.\pi.s & \\
= & 1 + t.\pi.s - 1 & \\
\leq & 1 + t.\pi.s + t.(\mathbf{-1}).s & -1 \leq t.\varepsilon.s \text{ for all } \varepsilon, s \\
\leq & 1 + t.(\pi - \mathbf{1}).s & \text{sub-additivity} \\
= & t.\mathbf{1}.s + t.(\pi - \mathbf{1}).s & \text{assumption} \\
\leq & t.\pi.s\ . & \text{sub-additivity}
\end{array}
$$

4 *ewp* for a simple language

The *ewp* semantics of our language (Fig. 2) is determined by Fig. 1 and (4) — and it looks very like the *wp*. For both *wp* and *wlp*, the definitions induced by (5) agree with the usual.

The least fixed point is taken in an order $\sqsubseteq$ shown by Lem. 4.1 to be Egli-Milner, defined over $\{-1, 0, 1\}$ by $-1 \sqsupset 0 \sqsubset 1$ and lifted to $\mathcal{E}S$ and $\mathcal{T}S$:

$$
\begin{array}{rll}
\varepsilon_0 \sqsubseteq \varepsilon_1 & \text{iff} \quad \varepsilon_0.s \sqsubseteq \varepsilon_1.s & \text{for all states } s \\
t_0 \sqsubseteq t_1 & \text{iff} \quad t_0.\varepsilon \sqsubseteq t_1.\varepsilon & \text{for all extended predicates } \varepsilon\ .
\end{array}
$$

Lem. 4.1 shows that $\sqsubseteq$ over $\mathcal{R}S$ and $\mathcal{T}S$ correspond, thus that $\sqsubseteq$ is Egli-Milner and that the two definitions of $(\mathbf{mu}\ \mathcal{B})$ are consistent.

Lemma 4.1 For all r_0, r_1 in $\mathcal{R}S$ we have $r_0 \sqsubseteq r_1$ iff $ewp.r_0 \sqsubseteq ewp.r_1$.
Proof: For *only if* we define the two sets $\mathsf{pos}.\varepsilon := \{s \mid \varepsilon.s = 1\}$ and $\mathsf{neg}.\varepsilon := \{s \mid \varepsilon.s = -1\}$ and note that directly from (4) we have

$$\begin{array}{rll} ewp.r.\varepsilon.s = 1 & \text{iff} & r.s \subseteq \mathsf{pos}.\varepsilon \text{ ; and} \\ ewp.r.\varepsilon.s = -1 & \text{iff} & r.s \cap \mathsf{neg}.\varepsilon \neq \emptyset \text{ .} \end{array}$$

The two implications $ewp.r_0.\varepsilon.s = x \Rightarrow ewp.r_1.\varepsilon.s = x$ for $x := \pm 1$ are then straightforward consequences of $r_0.s \sqsubseteq r_1.s$ as defined at (1).

For *if* we apply $ewp.r_0.\varepsilon.s \sqsubseteq ewp.r_1.\varepsilon.s$ to various values of ε: straightforward calculation shows

$$\begin{array}{lcll} \bot \notin r_0.s & \Rightarrow & r_1.s \subseteq r_0.s & \text{using } \varepsilon := \chi_{r_0.s} \text{ ; and} \\ s' \in r_0.s & \Rightarrow & s' \in r_1.s & \text{for all } s' \neq \bot \text{, using } \varepsilon := -\chi_{\{s'\}} \text{ .} \end{array}$$

Together those two facts imply $r_0.s \sqsubseteq r_1.s$. □

Note that the '*wlp* order' defined as $wlp.r_0.\pi \leq wlp.r_1.\pi$ for $\pi \geq \mathbf{0}$ is implied by $ewp.r_0 \sqsupseteq ewp.r_1$, and so the transformer $wlp.(\mathbf{mu}\ \mathcal{B})$ is a *greatest* fixed point for *wlp*.

5 The representation theorem

We prove in Thm. 5.4 that any extended-predicate transformer t satisfying the four conditions of Sec. 3 can be written as $ewp.r$ for some r in $\mathcal{R}S$.

First define, for t in $\mathcal{T}S$, its *representation* $rp.t$ in $\mathcal{R}S$ by

$$\begin{array}{rcll} rp.t.s & := & F.t.s \cup N.t.s \quad \text{where} & \\ F.t.s & := & \{s' \colon S \mid t.(-\chi_{\{s'\}}).s = -1\} & \text{proper } F\text{inal states} \\ N.t.s & := & \emptyset \ \text{ if } (t.\mathbf{1}.s = 1) \text{ else } \ \{\bot\} & \bot \text{ for } N\text{on-termination.} \end{array}$$

The key property of $F.t.s$ is given by Lem. 5.1.

Lemma 5.1 For any subset S' of S,

$$t.(\chi_{S'} - \mathbf{1}).s = 0 \quad \text{iff} \quad F.t.s \subseteq S' \text{ .}$$

Proof:

$$\begin{array}{lll} & F.t.s \subseteq S' & \\ \text{iff} & (\forall s' \colon S - S' \cdot t.(-\chi_{\{s'\}}).s \neq -1) & \text{definition } F.t.s \\ \text{iff} & (\forall s' \colon S - S' \cdot t.(-\chi_{\{s'\}}).s = 0) & t.(-\chi_{\{s'\}}) \leq \mathbf{0} \\ \text{iff} & t.(-\chi_{S-S'}).s = 0 & \text{positive conjunctivity} \\ \text{iff} & t.(\chi_{S'} - \mathbf{1}).s = 0 \text{ .} & \end{array}$$

□

Note that Lem. 5.1 establishes that rp is well-defined when t is healthy: if $F.t.s \cup N.t.s$ were empty we would have $0 = t.(\chi_\emptyset - \mathbf{1}).s = t.(-\mathbf{1}).s$ and $t.\mathbf{1}.s = 1$, reaching a contradiction as at (6).

Next define $\varepsilon^+ := \varepsilon \vee \mathbf{0}$ and $\varepsilon^- := \varepsilon \wedge \mathbf{0}$, the positive and negative components of ε, and observe that for healthy t we have

$$t.(\varepsilon^+) = (t.\varepsilon)^+ \quad \text{and} \quad t.(\varepsilon^-) = (t.\varepsilon)^- \ , \tag{8}$$

since the second is immediate from strictness and positive conjunctivity and for the first we have from sub-additivity

$$(t.\varepsilon)^+ \quad = \quad t.\varepsilon - (t.\varepsilon)^- \quad \geq \quad t.(\varepsilon^+) + t.(\varepsilon^-) - (t.\varepsilon)^- \quad = \quad t.(\varepsilon^+)$$

with the converse $(t.\varepsilon)^+ \leq t.(\varepsilon^+)$ following from monotonicity (since $\varepsilon \leq \varepsilon^+$ and $\mathbf{0} \leq t.(\varepsilon^+)$).

Finally, the representation theorem itself is proved in two parts, one 'positive' and one 'negative'. We give the positive, then the negative; since they are both *iff*, the 'zero' looks after itself.

Lemma 5.2 $\quad ewp.(rp.t).\varepsilon.s = 1$ iff $t.\varepsilon.s = 1$.
Proof:

	$ewp.(rp.t).\varepsilon.s = 1$	
iff	$F.t.s \cup N.t.s \subseteq \mathsf{pos}.\varepsilon$	definitions
iff	$t.\mathbf{1}.s = 1 \ \wedge \ F.t.s \subseteq \mathsf{pos}.\varepsilon$	$\bot \notin \mathsf{pos}.\varepsilon$
iff	$t.\mathbf{1}.s = 1 \ \wedge \ t.(\chi_{\mathsf{pos}.\varepsilon} - \mathbf{1}).s = 0$	Lem. 5.1
iff	$t.(\varepsilon^+).s = 1$	$\chi_{\mathsf{pos}.\varepsilon} = \varepsilon^+ \geq \mathbf{0}$ and (7)
iff	$t.\varepsilon.s = 1$.	(8)

□

Lemma 5.3 $\quad ewp.(rp.t).\varepsilon.s = -1$ iff $t.\varepsilon.s = -1$.
Proof:

	$ewp.(rp.t).\varepsilon.s = -1$	
iff	$(F.t.s \cup N.t.s) \cap \mathsf{neg}.\varepsilon \neq \emptyset$	definitions
iff	$(\exists s'\colon \mathsf{neg}.\varepsilon \cdot t.(-\chi_{\{s'\}}).s = -1)$	definitions; $\bot \notin \mathsf{neg}.\varepsilon$
iff	$t.(-\chi_{\mathsf{neg}.\varepsilon}).s = t.(\varepsilon^-).s = -1$	positive conjunctivity
iff	$t.\varepsilon.s = -1$.	(8)

□

The two lemmas give us our theorem.

Theorem 5.4 *Representation* Extended-predicate transformer t in $\mathcal{T}S$ satisfies the four healthiness conditions of Sec. 3 if and only if it can be written $ewp.r$ for some r in $\mathcal{R}S$.
Proof: The *only if* is proved in Lemmas 5.2 and 5.3. The *if* is straightforwardly checked from the definition (4) of *ewp*. □

6 Conclusions

We have contributed to the theory of predicate transformers in two ways: first, by extending predicates to $\{-1, 0, 1\}$ we showed (4) that it is possible to define a single predicate transformer *ewp* of which both *wp* and *wlp* are special cases (5).

Second, we have (Thm. 5.4) given an exact characterisation *ewp* for relational programs, just as Dijkstra's original healthiness conditions characterise *wp* for them.

From a contribution to theory does not necessarily follow an immediate change in practice, however: we are not proposing 'three-valued logic' for reasoning about specific programs, nor that *ewp* should replace *wp* and *wlp* for everyday use. Rather we believe that having a single and uniform domain with which both total and partial correctness can be treated is a useful theoretical tool for exploring both their interaction and the algebra of predicate transformers generally, especially given the surprising simplicity of the sub-additivity healthiness condition.

Further, if the predicate transformers are generalised to take arbitrary values in $\mathbb{R}$ instead of $\{-1, 0, 1\}$, the result is a domain for *probabilistic* programming in which sub-additivity generalises to 'sub-linearity' and, rather than being an 'extra' healthiness condition, is then the *only* one: in the probabilistic case it implies all the others [3].

Acknowledgements

We thank Ian Hayes, Dave Carrington, Karen Seidel and JW Sanders for their careful reading and helpful comments. Carroll Morgan is grateful for the support and hospitality of the Department of Computer Science and the SVRC during his sabbatical visit.

References

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

[2] Wim H Hesselink. *Programs, Recursion and Unbounded Choice*, volume 27 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 1992.

[3] C.C. Morgan, A.K. McIver, and K. Seidel. Probabilistic predicate transformers. *ACM Transactions on Programming Languages and Systems*, 18(3):325–53, May 1996.

[4] G. Nelson. A generalization of Dijkstra's calculus. Technical Report 16, Digital Systems Research Center, April 1987.