Partial and total correctness as greatest and least fixed points

John Wickerson

Imperial College London

Abstract. This paper studies Hoare triples in the context of any programming language specified by a small-step, possibly non-deterministic, operational semantics. We explain how the partial correctness interpretation of the triple can be characterised as the greatest fixed point of a function, and how the total correctness interpretation can be seen as the least fixed point of that very same function. In the latter case, we provide a necessary and sufficient condition for the characterisation to be accurate: that the programming language admits no infinite branching.

1 Introduction

In the context of Hoare Logic [4], a (possibly non-deterministic, and possibly non-terminating) program satisfies a *total correctness* specification when each of its traces from a state that satisfies the given precondition reaches a terminal state satisfying the given postcondition within a finite number of execution steps. That program satisfies a *partial correctness* specification when each of its traces either meets the requirement above or is infinite.

Meanwhile, in the context of order theory, an object is in the *least fixed point* of a continuous function if (roughly speaking) it can be shown meet some condition within a finite number of iterations. That object is in the *greatest fixed point* if it is in the least fixed point or is, in some sense, infinite.

The descriptions above have been deliberately crafted to emphasise a connection between total/partial correctness and least/greatest fixed points. This paper seeks to state that connection precisely, and investigate conditions under which it holds.

Section 3 provides characterisations of partial and total correctness, that differ only in that the former takes a function's greatest fixed point where the latter takes its least. Section 4 describes a condition that is necessary for the fixed point characterisation of *total* correctness to be accurate; namely, that there is no infinite branching. Section 5 discusses related work, mainly focussing on how our result extends a similar observation made by Edmund Clarke in 1979. We begin, in Section 2, by establishing some preliminary definitions.

2 Preliminaries

We assume a small-step transition relation $\rightsquigarrow$ between configurations $C = \langle c, \sigma \rangle$, which comprise a command $c \in \mathsf{Cmd}$ and a state $\sigma \in \mathsf{State}$. We impose no constraints on the forms taken by commands and states. Let $\mathsf{Config} = \mathsf{Cmd} \times \mathsf{State}$ be the set of all configurations. We use the abbreviation $\mathit{next}(C) \overset{\text{def}}{=} \{C'.\, C \rightsquigarrow C'\}$ for the set of configurations immediately reachable from C, and we write $\mathit{stuck}(C)$ if C admits no further transitions. The functions fst and snd serve to project the components of a pair.

Modal-μ calculus We employ the following constructions from the modal-μ calculus [6] to describe properties of our transition relation. In the following, we suppose that $p \in \mathcal{P}(\mathsf{Config})$ and that $\varphi : \mathcal{P}(\mathsf{Config}) \to \mathcal{P}(\mathsf{Config})$ is a monotone function.

$$\begin{aligned} \Box p &= \{C.\, \forall C' \in \mathit{next}(C).\, C' \in p\} \\ \Diamond p &= \{C.\, \exists C' \in \mathit{next}(C).\, C' \in p\} \\ \mu X.\, \varphi(X) &= \bigcap \{S.\, \varphi(S) \subseteq S\} \\ \nu X.\, \varphi(X) &= \bigcup \{S.\, S \subseteq \varphi(S)\} \end{aligned}$$

Possibly-infinite sequences Given a set X, a *possibly-infinite sequence* is a partial function $\pi : \mathbb{N} \to X \cup \{\bot\}$ whose domain of definition is either the entirety of $\mathbb{N}$ or an initial subset thereof. In the latter case, we define $\mathrm{len}(\pi) = j + 1$ when j is the greatest natural in π's domain. We shall sometimes refer to an element of a sequence by writing π_i instead of $\pi(i)$.

Traces A trace is a possibly-infinite sequence of configurations, successively related by $\rightsquigarrow$. The set of traces beginning from a configuration C, written $\mathit{traces}(C)$, comprises those sequences π for which $\pi_0 = C$ and for all i:

$$\frac{\pi_i = \bot}{\pi_{i+1} = \bot} \qquad\qquad \frac{\mathit{stuck}(\pi_i)}{\pi_{i+1} = \bot} \qquad\qquad \frac{\neg \mathit{stuck}(\pi_i)}{\pi_{i+1} \in \mathit{next}(C)}$$

Termination A configuration C *always terminates* if every trace from C reaches a terminal configuration.

$$C \in \mathit{always\text{-}terminates} \overset{\text{def}}{=} \forall \pi \in \mathit{traces}(C).\, \exists j > 0.\, \mathrm{len}(\pi) = j$$

Safe configurations If $Q \in \mathcal{P}(\mathsf{State})$ is a postcondition, we say that a configuration C is *safe for* Q, written $C \in \mathit{safe}_Q$, if whenever a trace starting from that configuration reaches a terminal configuration, the state is in Q.

$$C \in \mathit{safe}_Q \overset{\text{def}}{=} \forall c', \sigma'.\, ((C \rightsquigarrow^* \langle c', \sigma' \rangle) \wedge \mathit{stuck}\langle c', \sigma' \rangle) \Rightarrow \sigma' \in Q$$

Partial and total correctness Suppose $P, Q \in \mathcal{P}(\mathsf{State})$ and $c \in \mathsf{Cmd}$. We write $\{P\}\, c\, \{Q\}$ to mean that whenever c is executed from a state in P, then whenever it reaches a terminal configuration, the state is in Q. We write $[P]\ c\ [Q]$ to mean that whenever c is executed from a state in P, then it reaches a terminal configuration, and whenever it reaches a terminal configuration, the state is in Q.

$$\begin{aligned} \{P\}\, c\, \{Q\} &\overset{\text{def}}{=} \forall \sigma \in P.\, \langle c, \sigma \rangle \in \mathit{safe}_Q \\ [P]\ c\ [Q] &\overset{\text{def}}{=} \forall \sigma \in P.\, \langle c, \sigma \rangle \in \mathit{always\text{-}terminates} \cap \mathit{safe}_Q \end{aligned}$$

3 Main result

Definition 1. *We characterise partial/total correctness (with respect to postcondition Q) as the greatest/least fixed point of the function φ_Q, defined as follows:*

$$\varphi_Q(X) \stackrel{\text{def}}{=} \{\langle c, \sigma\rangle.\, \mathit{stuck}\langle c, \sigma\rangle \Rightarrow \sigma \in Q\} \cap \Box X.$$

We now establish a few properties of φ_Q. The first enables the use of the greatest *post*-fixed point and the least *pre*-fixed point of φ_Q as proxies, respectively, for its greatest and least fixed points.

Lemma 1 (Monotonicity). *φ_Q is monotone; i.e., $X \subseteq X'$ implies $\varphi_Q(X) \subseteq \varphi_Q(X')$.*

The next lemmas allow φ_Q's fixed points to be constructed via series of approximants.

Lemma 2 (GLB-preservation). *φ_Q preserves greatest lower bounds. That is, for any ascending chain $x_0 \subseteq x_1 \subseteq \ldots$, we have $\varphi_Q(\bigcap_k x_k) = \bigcap_k \varphi_Q(x_k)$.*

Lemma 3 (LUB-preservation). *If our transition system has finite branching, then φ_Q preserves least upper bounds. That is, if $\mathit{finite}(\mathit{next}(C))$ holds for all C, then for any ascending chain $x_0 \subseteq x_1 \subseteq \ldots$, we have $\varphi_Q(\bigcap_k x_k) = \bigcap_k \varphi_Q(x_k)$.*

The following lemma states that in the absence of infinite branching, every always-terminating command has a longest trace.

Lemma 4 (Longest trace). *If $\mathit{next}(C)$ is finite for all C, and $C_0 \in \mathit{always\text{-}terminates}$, then there exists an upper bound M for which $\forall \pi \in \mathit{traces}(C_0).\, \exists j \leq M.\, \mathrm{len}(\pi) = j$.*

Proof. We first recall Kőnig's infinity lemma as it applies to trees – that every tree with infinitely-many vertices, each having finitely-many successor vertices, has at least one infinite trace [3]:

$$\neg \mathit{finite}\{C'.\, C_0 \rightsquigarrow^* C'\} \Rightarrow (\forall C.\, \mathit{finite}(\mathit{next}(C))) \Rightarrow \exists \pi \in \mathit{traces}(C_0).\, \mathrm{dom}(\pi) = \mathbb{N}. \tag{1}$$

Now, if $C_0 \in \mathit{always\text{-}terminates}$, then it has no infinite traces, and hence, by the contrapositive of (1), the number of configurations reachable from C_0 is finite. This number provides a suitable upper bound M on the length of traces from C_0.

Theorem 1. *Partial and total correctness can be characterised as greatest/least fixed points. Note that the third implication below relies on our transition system having the property of finite branching.*

$$\begin{array}{rcll}
\{P\}\, c\, \{Q\} & = & (\{c\} \times P) \subseteq \nu X.\, \varphi_Q(X) & \quad (2) \\
[P]\ c\ [Q] & \Leftarrow & (\{c\} \times P) \subseteq \mu X.\, \varphi_Q(X) & \quad (3) \\
[P]\ c\ [Q] & \Rightarrow & (\{c\} \times P) \subseteq \mu X.\, \varphi_Q(X) \quad \textit{if finite}(\mathit{next}(C))\ \textit{for all } C. & \quad (4)
\end{array}$$

Proof. We begin with the fixed point characterisation of partial correctness. To prove (2), it suffices to prove that safe_Q coincides with φ_Q's greatest post-fixed point.

$$\mathit{safe}_Q = \nu X.\, \varphi_Q(X). \tag{5}$$

For the $(\subseteq)$ direction of (5), it is straightforward to show, by expanding definitions and invoking standard lemmas about reflexive transitive closures, that safe_Q is a post-fixed point (that is, $\mathit{safe}_Q \subseteq \varphi_Q(\mathit{safe}_Q)$), and hence that it is below the *greatest* post-fixed point. To show the $(\supseteq)$ direction, we first observe that we can invoke Lemma 2 to construct φ_Q's greatest post-fixed point as the intersection of a sequence of approximants:

$$\nu X.\, \varphi_Q(X) = \bigcap_k \left(\varphi_Q^k(\mathsf{Config})\right).$$

(The intuition is that approximant $k+1$ contains configurations for which every stuck configuration that is reachable in k steps satisfies the postcondition.) After expanding the definition of *safe*, it remains to show:

$$(\forall k.\, C \in \varphi_Q^k(\mathsf{Config})) \Rightarrow \forall C'.\, ((C \rightsquigarrow^* C') \wedge \mathit{stuck}(C')) \Rightarrow \mathrm{snd}(C') \in Q. \tag{6}$$

After rewriting $C \rightsquigarrow^* C'$ as $\exists n.\, C \rightsquigarrow^n C'$ and then instantiating the k in (6) to $n+1$, it suffices to show, for all n:

$$C \in \varphi_Q^{n+1}(\mathsf{Config}) \Rightarrow \forall C'.\, ((C \rightsquigarrow^n C') \wedge \mathit{stuck}(C')) \Rightarrow \mathrm{snd}(C') \in Q,$$

which can be dispatched via mathematical induction on n (with C universally quantified in the induction hypothesis).

We now turn to the fixed point characterisation of total correctness. To prove (3), it suffices to show that

$$\mathit{safe}_Q \cap \mathit{always\text{-}terminates} \supseteq \mu X.\, \varphi_Q(X),$$

which follows from $\mathit{safe}_Q \cap \mathit{always\text{-}terminates}$ being a pre-fixed point of φ_Q and hence above its *least* pre-fixed point. To show (4), we first use Lemma 3 to equate the least pre-fixed point to the union of a sequence of approximants as follows:

$$\mu X.\, \varphi_Q(X) = \bigcup_k \left(\varphi_Q^k(\emptyset)\right).$$

(The intuition is that approximant $k+1$ contains all configurations for which every trace terminates in no more than k steps and satisfies the postcondition when it does so.) After expanding the definition of *safe*, and invoking Lemma 4 to obtain the length M of C's longest trace, it remains to show:

$$\begin{aligned}&(\exists M.\, \forall \pi \in \mathit{traces}(C).\, \exists j \leq M.\, \mathrm{len}(\pi) = j) \Rightarrow \\ &(\forall C'.\, ((C \rightsquigarrow^* C') \wedge \mathit{stuck}(C')) \Rightarrow \mathrm{snd}(C') \in Q) \Rightarrow (\exists k.\, C \in \varphi_Q^k(\emptyset)).\end{aligned} \tag{7}$$

We choose M as a witness for k in (7). It then suffices to show, for all M:

$$\begin{aligned}&(\forall \pi \in \mathit{traces}(C).\, \exists j \leq M.\, \mathrm{len}(\pi) = j) \Rightarrow \\ &(\forall C'.\, ((C \rightsquigarrow^* C') \wedge \mathit{stuck}(C')) \Rightarrow \mathrm{snd}(C') \in Q) \Rightarrow C \in \varphi_Q^M(\emptyset).\end{aligned}$$

With a little further algebraic manipulation we arrive at:

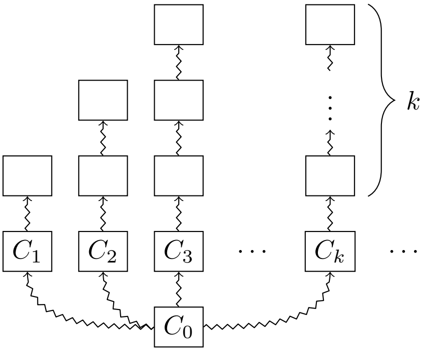
$$(\forall \pi \in \mathit{traces}(C).\, \exists j \leq M.\, \mathrm{len}(\pi) = j \wedge \mathrm{snd}(\pi_{j-1}) \in Q) \Rightarrow C \in \varphi_Q^M(\emptyset)$$

which can be dispatched by mathematical induction on M, generalising C as before, thus completing the proof. □

Proof mechanisation Our main theorem and its accompanying lemmas have been formalised and proved in the Isabelle theorem prover [**?**], with the exception of Lemma 4, of which we currently only have a hand proof. The proof relies on Kőnig's infinity lemma, which we found to be difficult to formalise. The proof is available online.[1]

4 On infinite branching

If the $\rightsquigarrow$ relation allows infinite branching – that is, if a configuration can have infinitely many next configurations to choose from – then the least fixed point calculation does not coincide with total correctness. The technical reason for the failure is that φ_Q no longer preserves least upper bounds (cf. Lemma 3). Intuitively, the failure can be explained by the following counterexample.



Recall that approximant k contains configurations whose traces all terminate in fewer than k steps. No approximant can contain configuration C_0, since there is no bound on the length of its traces. Yet this configuration will be admitted by a total correctness specification whose postcondition is $true$, since it is the case that each of its traces terminates.

Therefore, in the presence of infinite branching, the fixed point characterisation of total correctness must be weakened to an implication. Preservation of greatest lower bounds is unaffected by infinite branching, so the characterisation of partial correctness remains intact.

5 Related work

Edmund Clarke [1] was probably the first to study the correspondence between greatest/least fixed points and partial/total correctness. He remarks, in a 1979 article, that

> the fixedpoints of Γ form a complete lattice under the natural partial ordering on $\mathcal{P}(\mathsf{State})$. The top element of this lattice is the weakest precondition for partial correctness and the bottom element is the weakest precondition for total correctness. [1, p. 279, footnote]

[1] `http://www.doc.ic.ac.uk/~jpw48/Partial_Total_Correctness_As_Fixed_Points.thy`

We note three ways in which the current work extends on Clarke's original observation. First, Clarke works in the setting of a particular programming language that provides syntax for sequencing, conditionals, assignment, and recursive calls to parameterless procedures. In this paper, we need not tackle issues of programming language syntax, since we work at the level of arbitrary transition systems. Second, Clarke's work applies only in a deterministic setting: a program has either a single final state, or none at all. We allow for highly non-deterministic programs, and only restrict to finite non-determinism in the case of total correctness. Third, Clarke's work is in the setting of a big-step semantics; that is, the meaning of each programming construct maps an initial state directly to a final state. We work with a small-step semantics (as described by a transition system), which means that our approach handles parallel programs without further adaptation. Big-step semantics is well-known to be unable to handle parallelism, as acknowledged by Clarke, who remarks that he '[is] currently attempting to extend this fixedpoint theory to additional programming features such as parallelism' [1, p. 292]. The definition of total correctness in a big-step and deterministic setting is trivial, and hence Clarke's Γ functional is straightforward – this perhaps explains why Clarke consigned the observation we quote above to a mere footnote. Our functional, on the other hand, which we call φ, is fairly subtle: experience shows that even minor modifications of Definition 1 quickly lead to either the least or the greatest fixed point becoming degenerate.

Regarding other influences on this work: the idea of using the least and greatest fixed point of the same function has been previously exploited by Paulson [7, §3], who obtains the set of finite lists from a function's least fixed point, and the set of possibly-infinite ('lazy') lists from its greatest fixed point.

Recent work on various Hoare logics for concurrency provide avenues for further development of the current work. These logics typically use (mildly disguised) greatest fixed point calculations to obtain a partial correctness semantics; see, for example, [8, Definition 3.2] and [2, Definition 25]. If these logics were extended to handle total correctness, the result presented in this paper could ease the transition.

Another direction for future work is provided by Jacobs and Gries, who have proposed *general correctness* as a way to unify partial and total correctness [5]. It would be interesting to investigate whether general correctness can be also characterised as a fixed point calculation.

—

Acknowledgements This work was supported by EPSRC grant EP/K011499/1. I would like to thank Edmund Clarke, Alastair Donaldson, Tony Hoare, Peter Lammich, and Andreas Lochbihler for helpful feedback and discussions.

References

1. E. M. Clarke. Program invariants as fixedpoints. *Computing*, 21:273–294, 1979.
2. T. Dinsdale-Young, L. Birkedal, P. Gardner, M. J. Parkinson, and H. Yang. Views: Compositional reasoning for concurrent programs. In R. Giacobazzi and R. Cousot, editors, *The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '13*, pages 287–300. ACM, 2013.
3. M. Franchella. On the origins of Dénes Kőnig's Infinity lemma. *Archive for History of Exact Sciences*, 51:3–27, 1997.
4. C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580, 1969.
5. D. Jacobs and D. Gries. General correctness: A unification of partial and total correctness. *Acta Informatica*, 22:67–83, 1985.
6. D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.
7. T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL - A Proof Assistant for Higher-Order Logic*. Springer-Verlag, 2002.
8. L. C. Paulson. Mechanizing coinduction and corecursion in higher-order logic. *Journal of Logic and Computation*, 7(2):175–204, 1997.
9. V. Vafeiadis. Concurrent separation logic and operational semantics. In J. Ouaknine, editor, *Proceedings of the 27th Annual Conference on the Mathematical Foundations of Programming Semantics (MFPS XXVII)*, volume 276 of *Electronic Notes in Theoretical Computer Science*, pages 335–351. Elsevier, 2011.