Cyclic Proofs in Hoare Logic and its Reverse

James Brotherston [a] Quang Loc Le[a] Gauri Desai[a] Yukihiro Oda[b]

[a] *Department of Computer Science, University College London, London, United Kingdom*
[b] *Tohoku University, Sendai, Japan*

Abstract

We examine the relationships between axiomatic and cyclic proof systems for the partial and total versions of Hoare logic and those of its dual, known as reverse Hoare logic (or sometimes incorrectness logic).
In the axiomatic proof systems for these logics, the proof rules for looping constructs involve an explicit loop invariant, which in the case of the total versions additionally require a well-founded termination measure. In the cyclic systems, these are replaced by rules that simply unroll the loops, together with a principle allowing the formation of cycles in the proof, subject to a global soundness condition that ensures the well-foundedness of the circular reasoning. Interestingly, the cyclic soundness conditions for partial Hoare logic and its reverse are similar and essentially coinductive in character, while those for the total versions are also similar and essentially inductive. We show that these cyclic systems are sound, by direct argument, and relatively complete, by translation from axiomatic to cyclic proofs.

Keywords: Cyclic proofs, Hoare logic, reverse Hoare logic, incorrectness logic

1 Introduction

Hoare logic [18] is long established as a formal framework in which to specify and reason about correctness properties of computer programs. Typically, these properties, written as triples $\{P\}\,C\,\{Q\}$, are given a *partial correctness* interpretation, meaning that certain unwanted behaviours of the program C cannot happen: namely those in which C runs on a state satisfying the *precondition* P but results in a state not satisfying the *postcondition* Q. Sometimes one works instead with a *total correctness* interpretation of judgements, in which the termination of C from states satisfying P is additionally guaranteed.

More recently gaining in popularity is the dual notion of *reverse Hoare logic* proposed in [14] for specifying and proving the presence of specified behaviours in programs, as opposed to their absence. In the original version of reverse Hoare logic one gives a reachability or "incorrectness" interpretation to triples $[P]\,C\,[Q]$, meaning that certain behaviours of C do happen: namely, there are computations of C reaching all states satisfying Q from states satisfying P. O'Hearn went on to propose an extension of this logic dubbed *incorrectness logic* which supports the specification of various types of program errors in the postcondition Q and so can be used as the basis of automatic bug detection via proof search [22,19]. A weaker *partial reachability* interpretation of reverse Hoare triples can also be given, in which validity also admits the possibility of program divergence from states satisfying P.

Turning to proof theory, Hoare logic was originally formulated as an axiomatic proof system with rules for each program construct and a small number of generic rules; proofs themselves are then as usual finite derivation trees of proof judgements. Of particular note is the partial Hoare logic rule for `while` loops, which requires the prover to invent a suitable *loop invariant* that is preserved over any execution of the loop body (and entails the desired postcondition). Total Hoare logic possesses a similar rule, except that

we must also provide a well-founded *termination measure* t that decreases on each iteration of the loop:

$$\frac{\{P \wedge B\}\, C\, \{P\}}{\{P\}\, \texttt{while}\ B\ \texttt{do}\ C\, \{P \wedge \neg B\}}\ (\text{Inv}) \qquad \frac{\{P \wedge B \wedge t = n\}\, C\, \{P \wedge t < n\}}{\{P\}\, \texttt{while}\ B\ \texttt{do}\ C\, \{P \wedge \neg B\}}\ (\text{Inv-Total})$$

Reverse Hoare logic possesses analogous "backwards" versions of these proof rules in its partial and total variants, respectively (cf. Section 3). Devising suitable loop invariants and termination measures is challenging and a serious obstacle to automated proof search, and has been the subject of much work in the literature on program verification [20].

An alternative way of approaching these challenges is to instead adopt systems based on *cyclic proof*, which were first introduced to reason about process calculi [24,21] and later used to develop proof systems for various logics with (co)inductively defined constructs (see e.g. [12,4,6,8,1,13,15,16]). Cyclic proofs are essentially non-wellfounded, regular derivation trees known as "pre-proofs" — typically presented as finite trees with "backlinks" from open leaves to interior nodes — subject to a *global soundness condition* typically ensuring that each infinite path in the pre-proof implicitly embodies a valid argument by infinite descent [5,10]. It is perhaps less widely known than it might be that both partial and total Hoare logic can be formulated as cyclic proof systems sharing exactly the same proof rules but employing two different global soundness conditions (an observation going back at least to [8]).

In this paper, we present axiomatic and cyclic proof systems for the total and partial versions of Hoare logic and reverse Hoare logic, examine their commonalities and symmetries, and we show all of our systems sound and relatively complete [1]. For partial correctness, we stipulate that any infinite path in the pre-proof must contain infinitely many symbolic command executions, which (with local soundness of the proof rules) ensures that any putative counterexample to $\{P\}\, C\, \{Q\}$, i.e. a finite execution of C from P not resulting in Q, in fact cannot be finite and therefore is not a counterexample after all. For total correctness, we instead require that any infinite path must contain an infinite descending chain of well-founded values in the preconditions along the path. Thus, any putative counterexample to $\{P\}\, C\, \{Q\}$, i.e. an execution of C from P that either is infinite, or finite but does not result in Q, in fact *must* be infinite and yield an infinite descending chain of well-founded values, again a contradiction. For reverse Hoare logic, we obtain similar global soundness conditions: the condition for partial reverse Hoare logic is similar to the condition for standard partial Hoare logic, and similarly for the total versions. Soundness of the cyclic systems follows by direct arguments of the usual type for such systems (cf. [5]) and their relative completeness follows by explicit transformations from standard proofs to cyclic proofs. The developments in our paper are shown diagrammatically in Figure 1.

The remainder of this paper is structured as follows. Section 2 describes our programming language and its semantics. Section 3 presents the partial and total versions of Hoare logic and their correspondents in reverse Hoare logic, both semantically and as standard axiomatic proof systems. Section 4 and Section 5 present our cyclic proof systems for these logics, along with their corresponding soundness and relative completeness arguments. Finally, Section 6 discusses future work and concludes.

2 Programming language

In this section, we give the syntax and semantics of a simple *while* programming language (without procedures or function calls).

We assume an infinite set Var of *variables* ranged over by $x, y, z, \ldots$. *Expressions* are first-order terms built from variables, numerals and possibly function symbols ($+$, $\times$, etc). The syntax of *Boolean conditions* B and *programs* C is then given by the following grammar:

$$B := E = E \mid E \leq E \mid \neg B \mid B \wedge B$$

$$C := \texttt{skip} \mid x := E \mid C; C \mid \texttt{if}\ B\ \texttt{then}\ C\ \texttt{else}\ C \mid \texttt{while}\ B\ \texttt{do}\ C$$

[1] Note to referees: In fact we have not yet fully established the relative completeness of our axiomatic system for partial reverse Hoare logic with respect to validity, but it seems highly plausible.

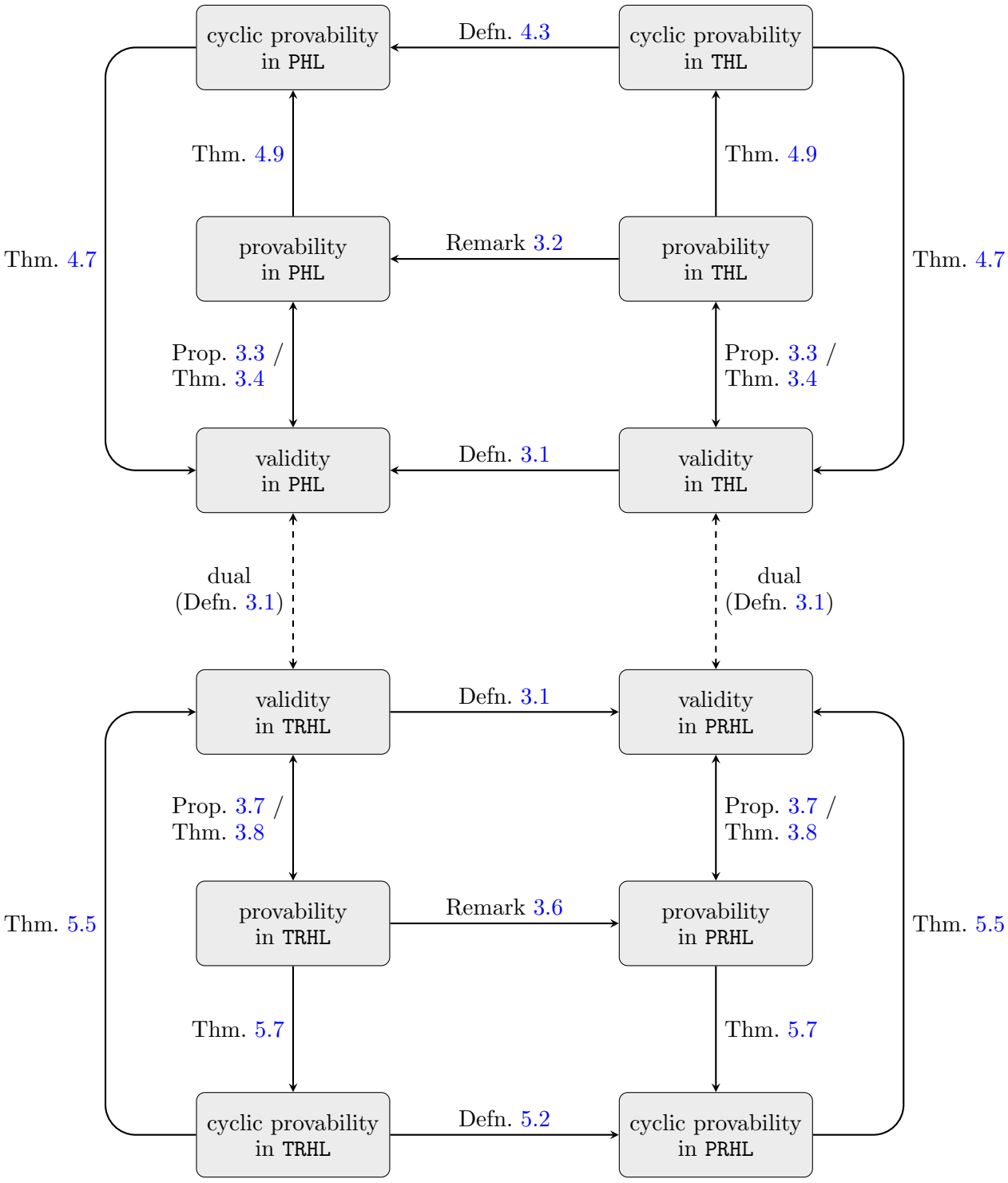

Fig. 1. Summary of our developments. PHL (resp. THL) is partial (resp. total) Hoare logic, and PRHL (resp. TRHL) is partial (resp. total) reverse Hoare logic.

$$\frac{}{\langle \texttt{skip}; C, \sigma\rangle \to \langle C, \sigma\rangle} \qquad \frac{}{\langle x := E, \sigma\rangle \to \langle \texttt{skip}, \sigma[x \mapsto [\![E]\!]\sigma]\rangle} \qquad \frac{\langle C_1, \sigma\rangle \to \langle C_1', \sigma'\rangle}{\langle C_1; C_2, \sigma\rangle \to \langle C_1'; C_2, \sigma'\rangle}$$

$$\frac{\sigma \models B}{\langle \texttt{if } B \texttt{ then } C_1 \texttt{ else } C_2, \sigma\rangle \to \langle C_1, \sigma\rangle} \qquad \frac{\sigma \models \neg B}{\langle \texttt{if } B \texttt{ then } C_1 \texttt{ else } C_2, \sigma\rangle \to \langle C_2, \sigma\rangle}$$

$$\frac{\sigma \models B}{\langle \texttt{while } B \texttt{ do } C, \sigma\rangle \to \langle C; \texttt{while } B \texttt{ do } C, \sigma\rangle} \qquad \frac{\sigma \models \neg B}{\langle \texttt{while } B \texttt{ do } C, \sigma\rangle \to \langle \texttt{skip}, \sigma\rangle}$$

Fig. 2. Operational semantics of while programs.

We write $E[E'/x]$ and $B[E'/x]$ to denote the substitution of expression E' for variable x in expression E and Boolean condition B, respectively, and write $fv(C)$ to denote the set of program variables in C. We also employ standard shorthand notations such as $E \neq E$ for $\neg(E_1 = E_2)$ and $E_1 < E_2$ for $E_1 \leq E_2 \wedge \neg E_1 = E_2$.

Semantically, expressions denote (program) *values* in a set Val, which for technical convenience we take in this paper to be the natural numbers. A *(program) state* is a function $\sigma : \text{Var} \to \text{Val}$. The interpretations $[\![E]\!]\sigma \in \text{Val}$ of any expression E and $[\![B]\!]\sigma \in \{\top, \bot\}$ of any Boolean condition B in state σ are then standard: $[\![x]\!]\sigma = \sigma(x)$, $[\![E_1 + E_2]\!]\sigma = [\![E_1]\!]\sigma + [\![E_2]\!]\sigma$, $[\![E_1 \leq E_2]\!]\sigma \Leftrightarrow [\![E_1]\!]\sigma \leq [\![E_2]\!]\sigma$, and so on. Finally, we write $\sigma[x \mapsto E]$ for the state defined exactly as σ except that $\sigma[x \mapsto E](x) = [\![E]\!]\sigma$.

We define a standard small-step operational semantics of our programs. A *program configuration* is a pair $\langle C, \sigma\rangle$, where C is a program and σ a state. The small-step semantics is then given by a binary relation $\to$ on configurations, as shown in Figure 2. An *execution* (of C) is a possibly infinite sequence of configurations $(\gamma_i)_{i\geq 0}$ with $\gamma_0 = \langle C, \_\rangle$ such that $\gamma_i \to \gamma_{i+1}$ for all $i \geq 0$. We write $\to^n$ for the n-step variant of $\to$ to represent $(\gamma_i)_{0\leq i\leq n}$ as $\gamma_0 \to^n \gamma_n$, and $\to^*$ for the reflexive-transitive closure of $\to$ to represent finite executions of arbitrary length. We write $\langle C, \sigma\rangle\uparrow$ if $\langle C, \sigma\rangle$ *diverges*, i.e. there is an infinite execution beginning $\langle C, \sigma\rangle \to \ldots$, and $\langle C, \sigma\rangle\downarrow$ if $\langle C, \sigma\rangle$ *converges*, meaning that there is no such execution. For any program C we may harmlessly identify C with $C; \texttt{skip}$ (but only finitely often).

3 Hoare logic and reverse Hoare logic

In this section, we define Hoare logic and reverse Hoare logic, in both their partial and total forms, first in terms of validity and then as axiomatic proof systems.

Assertions, ranged over by $P, Q, R, \ldots$ are standard formulas of first-order logic, whose terms at least include our expressions E and whose atomic formulas at least include our Boolean conditions B as given in the previous section. We write $\sigma \models P$ to denote satisfaction of assertion P by state σ, defined as usual in first-order logic, and write $[\![P]\!]$ as a shorthand for $\{\sigma \mid \sigma \models P\}$. Then a *Hoare triple* is written as $\{P\}\, C\, \{Q\}$ and a *reverse Hoare triple* as $[P]\, C\, [Q]$, where C is a program and P and Q are assertions.

Definition 3.1 [Validity] First, for a program C and assertion P, define $\mathsf{post}(C)(P)$, the *post-states* of C under P, by

$$\mathsf{post}(C)(P) = \{\sigma' \mid \exists \sigma.\ \sigma \models P \text{ and } \langle C, \sigma\rangle \to^* \langle \texttt{skip}, \sigma'\rangle\}\ .$$

We write $\mathsf{post}(C)(P)\downarrow$ to mean that $\langle C, \sigma\rangle\downarrow$ for all $\sigma \in [\![P]\!]$.

Then we have notions of *validity* for Hoare triples $\{P\}\, C\, \{Q\}$ in *partial* (PHL) and *total* (THL) Hoare logic, and for reverse Hoare triples $[P]\, C\, [Q]$ in total (TRHL) and partial (PRHL) reverse Hoare logic:

- $\{P\}\, C\, \{Q\}$ is *valid in* PHL iff $\mathsf{post}(C)(P) \subseteq [\![Q]\!]$;
- $\{P\}\, C\, \{Q\}$ is *valid in* THL iff $\mathsf{post}(C)(P) \subseteq [\![Q]\!]$ and $\mathsf{post}(C)(P)\downarrow$;
- $[P]\, C\, [Q]$ is *valid in* TRHL iff $\mathsf{post}(C)(P) \supseteq [\![Q]\!]$;
- $[P]\, C\, [Q]$ is *valid in* PRHL iff $\mathsf{post}(C)(P)\downarrow$ implies $\mathsf{post}(C)(P) \supseteq [\![Q]\!]$.

We note that, by definition, validity in THL immediately implies validity in PHL, and validity in TRHL implies validity in PRHL. By unpacking the notations in Defn. 3.1, validity for each of our logics can also be restated in a perhaps more convenient operational form, as follows:

PHL: $\{P\}\,C\,\{Q\}$ is valid iff $\forall \sigma, \sigma'$. if $\sigma \models P$ and $\langle C, \sigma\rangle \to^* \langle \mathtt{skip}, \sigma'\rangle$ then $\sigma' \models Q$;
THL: $\{P\}\,C\,\{Q\}$ is valid iff $\forall \sigma, \sigma'$ if $\sigma \models P$ then $\langle C, \sigma\rangle\!\downarrow$, and if $\langle C, \sigma\rangle \to^* \langle \mathtt{skip}, \sigma'\rangle$ then $\sigma' \models Q$;
TRHL: $[P]\,C\,[Q]$ is valid iff $\forall \sigma'$. if $\sigma' \models Q$ then $\exists \sigma$. $\sigma \models P$ and $\langle C, \sigma\rangle \to^* \langle \mathtt{skip}, \sigma'\rangle$;
PRHL: $[P]\,C\,[Q]$ is valid iff $\forall \sigma'$. if $\sigma' \models Q$ then $\exists \sigma$. $\sigma \models P$ and either $\langle C, \sigma\rangle \to^* \langle \mathtt{skip}, \sigma'\rangle$ or $\langle C, \sigma\rangle\!\uparrow$.

3.1 Provability in partial and total Hoare logic

Provability in Hoare logic is defined as derivability using the rules in Fig. 3. The partial version PHL of Hoare logic uses the rule (Inv), which requires the prover to find a *loop invariant* that is preserved by the loop body at each iteration, while the total version THL omits this rule in favour of the rule (Inv-Total) which additionally requires the prover to find a suitable well-founded termination measure t that decreases at each iteration [2]. The remaining rules are standard [18,2], except that our assignment axiom is written in the forward style from Floyd [17].

$$\frac{}{\{P\}\,\mathtt{skip}\,\{P\}}\ (\text{Skip}) \qquad \frac{}{\{P\}\,x := E\,\{P[x'/x] \wedge x = E[x'/x]\}}\ (:=) \qquad \frac{\{P\}\,C_1\,\{R\} \quad \{R\}\,C_2\,\{Q\}}{\{P\}\,C_1; C_2\,\{Q\}}\ (\text{Seq})$$

$$\frac{P' \models P \quad \{P\}\,C\,\{Q\} \quad Q \models Q'}{\{P'\}\,C\,\{Q'\}}\ (\models) \qquad \frac{\{P \wedge B\}\,C_1\,\{Q\} \quad \{P \wedge \neg B\}\,C_2\,\{Q\}}{\{P\}\,\mathtt{if}\ B\ \mathtt{then}\ C_1\ \mathtt{else}\ C_2\,\{Q\}}\ (\text{If})$$

$$\frac{\{P \wedge B\}\,C\,\{P\}}{\{P\}\,\mathtt{while}\ B\ \mathtt{do}\ C\,\{P \wedge \neg B\}}\ (\text{Inv}) \qquad \frac{\{P \wedge B \wedge t = n\}\,C\,\{P \wedge t < n\}}{\{P\}\,\mathtt{while}\ B\ \mathtt{do}\ C\,\{P \wedge \neg B\}}\ (\text{Inv-Total})$$

Fig. 3. Proof rules for Hoare triples. Partial Hoare logic PHL uses the rule (Inv) and total Hoare logic THL uses (Inv-Total), with all other rules shared. Variable x' is fresh in rule (:=) and variable t is fresh in (Inv-Total).

Remark 3.2 Given the premise of (Inv-Total), namely $\{P \wedge B \wedge t = n\}\,C\,\{P \wedge t < n\}$, it is easy to derive the premise of (Inv), namely $\{P \wedge B\}\,C\,\{P\}$: on the one hand we have $P \wedge t < n \models P$, and on the other we have $P \wedge B \models P \wedge B \wedge t = n$, because n is fresh. Thus, provability in total Hoare logic THL immediately implies provability in its partial version PHL as well.

The following results are well known in the literature.

Proposition 3.3 (Soundness) *If $\{P\}\,C\,\{Q\}$ is provable in* PHL *(resp.* THL*) then it is valid in* PHL *(resp.* THL*).*

Theorem 3.4 (Relative completeness [11]) *Assuming an oracle for logical entailment between assertions, if $\{P\}\,C\,\{Q\}$ is valid in* PHL *(resp.* THL*) then it is provable in* PHL *(resp.* THL*).*

Example 3.5 We use the program $\mathtt{while}\ x > 0\ \mathtt{do}\ x := x - 2;$ as a running example. This program has invariant $x \geq 0 \wedge x\%2 = 0$ where the modulo operator % is used to specify even numbers. Here, we show a THL proof of the total correctness triple $\{x \geq 0 \wedge x\%2 = 0\}\,\mathtt{while}\ x > 0\ \mathtt{do}\ x := x - 2;\{x = 0\}$.

$$\dfrac{\dfrac{\dfrac{}{\{x \geq 0 \wedge x\%2 = 0 \wedge x < 2n\}\,\mathtt{skip}\,\{x \geq 0 \wedge x\%2 = 0 \wedge x < 2n\}}\ (\text{Skip})}{\{x' \geq 0 \wedge x'\%2 = 0 \wedge x' > 0 \wedge x' = 2n \wedge x = x' - 2\}\,\mathtt{skip}\,\{x \geq 0 \wedge x\%2 = 0 \wedge x < 2n\}}\ (\models)}{\{x \geq 0 \wedge x\%2 = 0 \wedge x > 0 \wedge x = 2n\}\,x := x - 2;\,\{x \geq 0 \wedge x\%2 = 0 \wedge x < 2n\}}\ (:=)}{\{x \geq 0 \wedge x\%2 = 0\}\,\mathtt{while}\ x > 0\ \mathtt{do}\ x := x - 2;\{x = 0\}}\ (\text{Inv-Total})$$

[2] Note that the termination measure may not become negative because all our values are natural numbers; but, if desired, one can add for safety the requirement that $P \models t \geq 0$.

In this proof, we use the termination measure $t = x/2$.

3.2 Provability in partial and total reverse Hoare logic

We give our proof rules for reverse Hoare triples $[P]\,C\,[Q]$ in Figure 4. Here, PRHL uses rules (RInv-1) and (RInv-2), while TRHL omits these in favour of (RInv-Total). Note that the conditional rules (RIf1) and (RIf2), like those in [19], have stronger preconditions than the ones in [14,22].

$$\frac{}{[P]\,\texttt{skip}\,[P]}\ (\text{RSkip}) \qquad \frac{}{[P]\,x := E\,[P[x'/x] \wedge x = E[x'/x]]}\ (\text{R:=})$$

$$\frac{[P]\,C_1\,[R] \quad [R]\,C_2\,[Q]}{[P]\,C_1; C_2\,[Q]}\ (\text{RSeq}) \qquad \frac{P \models P' \quad [P]\,C\,[Q] \quad Q' \models Q}{[P']\,C\,[Q']}\ (\text{R}\models)$$

$$\frac{[P \wedge B]\,C_1\,[Q]}{[P \wedge B]\,\texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2\,[Q]}\ (\text{RIf1}) \qquad \frac{[P \wedge \neg B]\,C_2\,[Q]}{[P \wedge \neg B]\,\texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2\,[Q]}\ (\text{RIf2})$$

$$\frac{}{[P \wedge \neg B]\,\texttt{while}\ B\ \texttt{do}\ C\,[P \wedge \neg B]}\ (\text{RInv-1}) \qquad \frac{[P \wedge B]\,C\,[P]}{[P \wedge B]\,\texttt{while}\ B\ \texttt{do}\ C\,[P \wedge \neg B]}\ (\text{RInv-2})$$

$$\frac{[P \wedge B \wedge t < n]\,C\,[P \wedge t = n]}{[P]\,\texttt{while}\ B\ \texttt{do}\ C\,[P \wedge \neg B]}\ (\text{RInv-Total})$$

Fig. 4. Proof rules for reverse Hoare triples. Partial reverse Hoare logic $PRHL$ uses the rules (RInv-2) and (RInv-1), while total reverse Hoare logic $TRHL$ uses (RInv-Total), with all other rules shared. Variable x' is fresh in rule (R:=) and variable n is fresh in (RInv-Total).

Remark 3.6 Similarly to the observation in Remark 3.2, provability in TRHL immediately implies provability in PRHL as well.

The proofs of soundness and relative completeness in the total version of reverse Hoare logic appear in [14] and also can be seen as the special *ok* case of incorrectness logic in [22].

Proposition 3.7 (Soundness) *If $[P]\,C\,[Q]$ is provable in* PRHL *(resp.* TRHL*) then it is valid in* PRHL *(resp.* TRHL*).*

Proof. By rule induction on the inference rules in Fig. 4. Generally speaking, for the shared rules, their soundness of TRHL is a special case of the more general soundness for PRHL. Here we just show the more interesting cases.

Case (RSeq)**:** We first consider the case of PRHL. Then, supposing $\sigma' \models Q$, we require to find σ with $\sigma \models P$ such that either $\langle C_1; C_2, \sigma\rangle \to^* \langle \texttt{skip}, \sigma'\rangle$, or $\langle C_1; C_2, \sigma\rangle \uparrow$. By validity of the right premise, we obtain σ'' with $\sigma'' \models R$ and for which there are two possibilities:

- We have $\langle C_2, \sigma''\rangle \to^* \langle \texttt{skip}, \sigma'\rangle$. By validity of the left premise there are two further subcases. In the first we have σ with $\sigma \models P$ and $\langle C_1, \sigma\rangle \to^* \langle \texttt{skip}, \sigma''\rangle$. Then by the operational semantics we obtain $\langle C_1; C_2, \sigma\rangle \to^* \langle \texttt{skip}; C_2, \sigma''\rangle \to^* \langle \texttt{skip}, \sigma'\rangle$ and are done. In the second we have σ with $\sigma \models P$ and $\langle C_1, \sigma\rangle \uparrow$, in which case $\langle C_1; C_2, \sigma\rangle \uparrow$ as well.
- We have $\langle C_2, \sigma''\rangle \uparrow$. Then by the left premise we have σ with $\sigma \models P$ and either $\langle C_1, \sigma\rangle \to^* \langle \texttt{skip}, \sigma''\rangle$ or $\langle C_1, \sigma\rangle \uparrow$. Either way, we have $\langle C_1; C_2, \sigma\rangle \uparrow$ and so are done.

For TRHL, the argument above is simplified: we simply ignore all the subcases involving divergence.

Case (RIf1),(RIf2)**:** These rules are symmetric; we just consider the first. In PRHL, suppose $\sigma' \models Q$. By validity of the rule premise, we obtain σ with $\sigma \models P \wedge B$ obeying one of two possibilities. First, if

$\langle C_1, \sigma\rangle \to^* \langle \texttt{skip}, \sigma'\rangle$ then $\langle \texttt{if } B \texttt{ then } C_1 \texttt{ else } C_2, \sigma\rangle \to^* \langle \texttt{skip}, \sigma'\rangle$ as well, because $\sigma \models B$. Otherwise, if $\langle C_1, \sigma\rangle\uparrow$ then $\langle \texttt{if } B \texttt{ then } C_1 \texttt{ else } C_2, \sigma\rangle\uparrow$ too. (For `TRHL`, we simply ignore the second case.)

Case (RInv-1)**:** Supposing $\sigma' \models P \wedge \neg B$, we immediately have $\langle \texttt{while } B \texttt{ do } C, \sigma'\rangle \to^* \langle \texttt{skip}, \sigma'\rangle$.

Case (RInv-2)**:** This rule is sound for `PRHL` only. Suppose $\sigma' \models P \wedge \neg B$. Since $\sigma' \models P$, we obtain by validity of the rule premise $\sigma \models P \wedge B$ and two possible subcases. First, if $\langle C, \sigma\rangle\uparrow$ then, because $\sigma \models B$, we get $\langle \texttt{while } B \texttt{ do } C, \sigma\rangle \to \langle C; \texttt{while } B \texttt{ do } C, \sigma\rangle\uparrow$ and are done. Otherwise, $\langle C, \sigma\rangle \to^* \langle \texttt{skip}, \sigma'\rangle$. Using the fact that $\sigma' \not\models B$, the operational semantics then gives us

$$\langle \texttt{while } B \texttt{ do } C, \sigma\rangle \to \langle C; \texttt{while } B \texttt{ do } C, \sigma\rangle \to^* \langle \texttt{while } B \texttt{ do } C, \sigma'\rangle \to^* \langle \texttt{skip}, \sigma'\rangle \ .$$

Therefore, $[P \wedge B]\, \texttt{while } B \texttt{ do } C\, [P \wedge \neg B]$ is valid as required.

Case (RInv-Total)**:** This rule is used in `TRHL` only. We first prove the following general statement: for all states σ_k, if $[\![t]\!]\sigma_k = k$ and $\sigma_k \models P$ then $\exists \sigma.\ \sigma \models P$ and $\langle \texttt{while } B \texttt{ do } C, \sigma\rangle \to^* \langle \texttt{while } B \texttt{ do } C, \sigma_k\rangle$. We proceed by complete induction on k, inductively assuming the statement for all $k' < k$ and showing it then holds for k. Since $\sigma_k \models P \wedge t = k$ by assumption, we have by validity of the rule premise $\sigma_{k'}$ with $\sigma_{k'} \models P \wedge B \wedge t < k$ and $\langle C, \sigma_{k'}\rangle \to^* \sigma_k$. Writing $[\![t]\!]\sigma_{k'} = k'$, we then have $k' < k$, and thus by the induction hypothesis we have σ with $\sigma \models P$ and $\langle \texttt{while } B \texttt{ do } C, \sigma\rangle \to^* \langle \texttt{while } B \texttt{ do } C, \sigma_{k'}\rangle$. Thus, because $\sigma_{k'} \models B$, we get $\langle \texttt{while } B \texttt{ do } C, \sigma\rangle \to^* \langle \texttt{while } B \texttt{ do } C, \sigma_k\rangle$ by the operational semantics. This completes the induction.

Now, for the main proof, assume $\sigma' \models P \wedge \neg B$, and write $[\![t]\!]\sigma' = k$ say. By the inductive statement above, we have σ with $\sigma \models P$ and $\langle \texttt{while } B \texttt{ do } C, \sigma\rangle \to^* \langle \texttt{while } B \texttt{ do } C, \sigma'\rangle$. Because $\sigma' \not\models B$, we can then extend this execution to $\langle \texttt{while } B \texttt{ do } C, \sigma\rangle \to^* \langle \texttt{skip}, \sigma'\rangle$ and so are done. □

Theorem 3.8 (Relative completeness) *Assuming an oracle for logical entailment between assertions, if $[P]\, C\, [Q]$ is valid in* `TRHL` *then it is provable.*

The equivalent result for `PRHL` *is presently only conjectured.* [3]

Example 3.9 We show a reverse Hoare proof for the total incorrectness triple

$$[x \geq 0 \wedge (x_0 - x)\%2 = 0]\, \texttt{while } x > 0 \texttt{ do } x := x - 2; [x = 0 \wedge (x_0 - x)\%2 = 0]$$

where x_0 is a logical variable expressing the initial value of x before the loop. This loop has invariant $P = x \geq 0 \wedge x\%2 = 0 \wedge (x_0 - x)\%2 = 0$ and termination measure $t = (x_0 - x)/2$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{[x \geq 0 \wedge (x_0 - x)\%2 = 0 \wedge x_0 - x = 2n]\, \texttt{skip}\, [P \wedge x_0 - x = 2n]}\,(\text{RSkip})}{[x' \geq 0 \wedge (x_0 - x')\%2 = 0 \wedge x' > 0 \wedge x_0 - x' = 2n - 2 \wedge x = x' - 2]\, \texttt{skip}\, [P \wedge x_0 - x = 2n]}\,(\text{R}\models)}{[P \wedge x > 0 \wedge x_0 - x = 2n - 2]\, x := x - 2;\, [P \wedge x_0 - x = 2n]}\,(\text{R:=})}{[P \wedge x > 0 \wedge x_0 - x < 2n]\, x := x - 2;\, [P \wedge x_0 - x = 2n]}\,(\text{R}\models)}{[P]\, \texttt{while } x > 0 \texttt{ do } x := x - 2; [P \wedge \neg x > 0]}\,(\text{RInv-Total})}{[x \geq 0 \wedge (x_0 - x)\%2 = 0]\, \texttt{while } x > 0 \texttt{ do } x := x - 2; [x = 0 \wedge (x_0 - x)\%2 = 0]}\,(\text{R}\models)$$

4 Cyclic proofs in Hoare logic

This section presents a system of cyclic proofs for `PHL` and `THL`, i.e. partial and total Hoare logic, together with their soundness and their subsumption of the corresponding standard proof systems from the previous section.

First, we give the rules of our cyclic proof system for Hoare triples in Figure 5, these being shared for both `PHL` and `THL`. Compared to their standard equivalents, there are three main points of difference. First,

[3] Note to referees: However, we think it ought to be a natural adaptation of the total case, and hope to clear it up soon!

$$\frac{}{\{P\}\,\texttt{skip}\,\{P\}}\ (\text{CSkip}) \qquad \frac{\{P\}\,C\,\{Q\}}{\{P\}\,\texttt{skip};C\,\{Q\}}\ (\text{CSkip2}) \qquad \frac{\{P[x'/x] \wedge x = E[x'/x]\}\,C\,\{Q\}}{\{P\}\,x := E;C\,\{Q\}}\ (\text{C:=})$$

$$\frac{P' \models P \quad \{P\}\,C\,\{Q\} \quad Q \models Q'}{\{P'\}\,C\,\{Q'\}}\ (\text{C}\models) \qquad \frac{\{P \wedge B\}\,C_1;C'\,\{Q\} \quad \{P \wedge \neg B\}\,C_2;C'\,\{Q\}}{\{P\}\,\texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2;C'\,\{Q\}}\ (\text{CIf})$$

$$\frac{\{P\}\,C\,\{Q\}}{\{P[t/z]\}\,C\,\{Q[t/z]\}}\ (\text{Sub}) \qquad \frac{\{P \wedge \neg B\}\,C'\,\{Q\} \quad \{P \wedge B\}\,C;\texttt{while}\ B\ \texttt{do}\ C;C'\,\{Q\}}{\{P\}\,\texttt{while}\ B\ \texttt{do}\ C;C'\,\{Q\}}\ (\text{CInv})$$

Fig. 5. Cyclic proof rules for Hoare triples. x' is fresh in (C:=), and z and t are not in $fv(C)$ in (Sub).

we formulate the rules in "continuation style", where the rule(s) for a program construct C is presented as a rule whose conclusion has the general form $\{P\}\,C;C'\,\{Q\}$. Second, the partial and total rules for `while` loops are replaced with a single rule (CInv) that simply unfolds the loop once (on the left). Third, we include a rule (Sub) for explicit substitution of variables by expressions; this is sometimes included anyway in Hoare logic (see e.g. [2]) but is well known to be necessary or at least useful in general for forming backlinks between judgements that are required to be syntactically identical [5,9].

Next, we define cyclic *pre-proofs* — derivation trees with *backlinks* between judgement occurrences — and the *global soundness conditions* qualifying such structures as either partial or total cyclic proofs.

Definition 4.1 [Pre-proof] A *(cyclic) pre-proof* is a pair $\mathcal{P} = (\mathcal{D}, \mathcal{L})$, where $\mathcal{D}$ is a finite derivation tree constructed according to the proof rules and $\mathcal{L}$ is a *back-link function* assigning to every leaf of $\mathcal{D}$ to which no proof rule has been applied (called a *bud*) another node of $\mathcal{D}$ (called its *companion*) labelled by an identical proof judgement. A *leaf* of $\mathcal{D}$ is called *open* if it is not applied with any proof rule.

We observe that a pre-proof $\mathcal{P}$ can be viewed as a representation of a regular, infinite derivation tree [5].

The global soundness condition qualifying pre-proofs $\mathcal{P}$ as genuine cyclic proofs is very simple in the case of partial Hoare logic: We simply require that a symbolic execution rule (i.e. not (C$\models$) or (Sub)) is applied infinitely often along every infinite path in $\mathcal{P}$. Essentially, this guarantees that any putative counterexample to soundness corresponds to an infinite execution of the program and therefore cannot be a counterexample after all. For total Hoare logic, the situation is a little more complex (cf. [7]); we require that every infinite path in $\mathcal{P}$ contains in the preconditions along the path a *trace* of well-founded measures that decrease, or "progress", infinitely often. This ensures that any putative counterexample cannot after all be terminating and therefore is not a counterexample either. To formulate traces, since we are using expressions interpreted as natural numbers, we adopt Simpson's definition from *cyclic arithmetic* [23].

Definition 4.2 [Trace [23]] Let $\mathcal{P} = (\mathcal{D}, \mathcal{L})$ be a pre-proof and $(\{P_k\}\,C_k\,\{Q_k\})_{k\geq 0}$ be a path in $\mathcal{P}$. For terms n and n', we say that n' is a *precursor* of n at k if one of the following holds:

- $\{P_k\}\,C_k\,\{Q_k\}$ is the conclusion of an application of (C:=) or (Sub), and $n' = \theta(n)$ where θ is the substitution used in the rule application; or
- $\{P_k\}\,C_k\,\{Q_k\}$ is the conclusion of another rule, and $n' = n$.

A *trace* following $(\{P_k\}\,C_k\,\{Q_k\})_{k\geq 0}$ is a sequence of terms $(\mathrm{n}_k)_{k\geq 0}$ such that for every $k \geq 0$, the term n_k occurs in P_k and also one of the following conditions holds:

- either n_{k+1} is a precursor of n_k at k; or
- there exists n such that $(\mathrm{n}_{k+1} < n) \in P_k$ and n is a precursor of n_k at k.

When the latter case holds, we say that the trace *progresses* (at $k+1$). An *infinitely progressing trace* is a trace that progresses at infinitely many points.

Definition 4.3 [Cyclic proof] A pre-proof $\mathcal{P}$ is a *cyclic proof* in `PHL` if there are infinitely many applications of symbolic execution rules along every infinite path in $\mathcal{P}$. It is furthermore a cyclic proof in `THL` if in addition there is an infinitely progressing trace along a tail of every path in $\mathcal{P}$.

We remark that, by construction, cyclic provability in THL immediately implies cyclic provability in PHL as well. We now show two examples, of partial and total cyclic proofs respectively.

Example 4.4 Let C stand for the program `while` $x > 0$ `do` $x := x - 2;$. Here we show a cyclic proof of $\{x \geq 0 \wedge x\%2 = 0\}\, C\, \{x = 0\}$ in PHL:

$$\dfrac{\dfrac{\dfrac{}{\{x=0\}\,\texttt{skip}\,\{x=0\}}\,(\text{CSkip})}{\{x \geq 0 \wedge x\%2 = 0 \wedge \neg(x<0)\}\,\texttt{skip}\,\{x=0\}}\,(\text{C}\models) \qquad \dfrac{\dfrac{\{x \geq 0 \wedge x\%2 = 0\}\, C\, \{x=0\}}{\{x' \geq 0 \wedge x'\%2 = 0 \wedge x' > 0 \wedge x = x' - 2\}\, C\, \{x=0\}}\,(\text{C}\models)}{\{x \geq 0 \wedge x\%2 = 0 \wedge x > 0\}\,\texttt{x := x - 2;}\, C\, \{x=0\}}\,(\text{C:=})}{\{x \geq 0 \wedge x\%2 = 0\}\, C\, \{x=0\}}\,(\text{CInv})$$

Example 4.5 Continuing with the program C from the previous example, we show a total cyclic proof of $\{x = 2n\}\, C\, \{x = 0\}$.

$$\dfrac{\dfrac{\dfrac{}{\{x=0\}\,\texttt{skip}\,\{x=0\}}\,(\text{CSkip})}{\{x = 2n \wedge \neg(x>0)\}\,\texttt{skip}\,\{x=0\}}\,(\text{C}\models) \qquad \dfrac{\dfrac{\dfrac{\{x = 2\underline{n}\}\, C\, \{x=0\}}{\{x = 2\underline{(n-1)}\}\, C\, \{x=0\}}\,(\text{Sub})}{\{x' = 2\underline{n} \wedge x' > 0 \wedge x = x' - 2\}\, C\, \{x=0\}}\,(\text{C}\models)}{\{x = 2\underline{n} \wedge x > 0\}\,\texttt{x:=x-2;}\, C\, \{x=0\}}\,(\text{C:=})}{\{x = 2\underline{n}\}\, C\, \{x=0\}}\,(\text{CInv})$$

In this proof, the progressing trace from companion to bud is given by the underlined terms involving n.

Lemma 4.6 *Let $\mathcal{P}$ be a pre-proof of $\{P\}\, C\, \{Q\}$ and suppose that $\{P\}\, C\, \{Q\}$ is invalid. Then there exists an infinite path $(\{P_k\}\, C_k\, \{Q_k\})_{k\geq 0}$ in $\mathcal{P}$, beginning from $\{P\}\, C\, \{Q\}$, such that the following properties hold, for all $k \geq 0$:*

(i) *for all $k \geq 0$, the triple $\{P_k\}\, C_k\, \{Q_k\}$ is invalid, meaning that either*
  (a) *$\exists \sigma, \sigma'.\ \sigma \models P_k$ and $\langle C_k, \sigma\rangle \to^{m_k} \langle \texttt{skip}, \sigma'\rangle$ but $\sigma' \not\models Q_k$; or*
  (b) *(in THL only) $\exists \sigma.\ \sigma \models P_k$ but $\langle C_k, \sigma\rangle \uparrow$.*

(ii) *If the first possibility (i)(a) above holds, then the computation length $m_{k+1} \leq m_k$, and if the rule applied at k is a symbolic execution rule then $m_{k+1} < m_k$.*

(iii) *(in THL only) if the second possibility (i)(b) above instead holds, and there is a trace $(\mathrm{n}_k)_{k\geq i}$ following a tail of the path $(\{P_k\}\, C_k\, \{Q_k\})_{k\geq i}$, then the sequence of natural numbers defined by $(\sigma(\mathrm{n}(k)))_{k\geq i}$ is monotonically decreasing, and strictly decreases at every progress point of the trace.*

Proof. (Sketch) We construct the required path and prove its needed properties inductively, by analysis of each proof rule in Figure 5. □

Theorem 4.7 (Soundness) *If $\{P\}\, C\, \{Q\}$ has a cyclic proof in PHL (resp. THL) then it is valid in PHL (resp. THL).*

Proof. Suppose first that $\{P\}\, C\, \{Q\}$ has a cyclic proof $\mathcal{P}$ in PHL but is invalid there. By Lemma 4.6 we can build an infinite path $(\{P_k\}\, C_k\, \{Q_k\})_{k\geq 0}$ in $\mathcal{P}$, beginning from $\{P\}\, C\, \{Q\}$, satisfying properties (i)(a) and (ii) above. In particular, we have an infinite, monotonically decreasing sequence $(m_k)_{k\geq 0}$ of natural numbers such that for all $k \geq 0$ we have $\langle C_k, \sigma\rangle \to^{m_k} \langle \texttt{skip}, \sigma'\rangle$ (for some σ and σ'). Since $\mathcal{P}$ is a cyclic proof in PHL, this path contains infinitely many applications of symbolic execution rules, and thus by property (ii) the sequence $(m_k)_{k\geq 0}$, which is a contradiction.

Next suppose that $\mathcal{P}$ is also a cyclic proof in THL but is not valid in THL. We apply Lemma 4.6 to obtain an infinite path in $\mathcal{P}$ as above, satisfying properties (i)(b) and (iii); since it is already a cyclic proof in PHL, we may rule out $(i)(a)$ as a possibility (and need not consider (ii) either). Since $\mathcal{P}$ is a cyclic proof in THL, there is an infinitely progressing trace $(\mathrm{n}_k)_{k\geq i}$ following some tail $(k \geq i)$ of this path. Thus by property (iii) there is an infinite, monotonically decreasing sequence $(\sigma(\mathrm{n}_k))_{k\geq i}$ of numbers that moreover strictly decreases infinitely often; again a contradiction. Thus $\{P\}\, C\, \{Q\}$ is valid in THL after all. □

Lemma 4.8 (Proof translation) *If $\{P\}\,C\,\{Q\}$ is provable in* PHL *(resp.* THL*), then for all statements C' and assertions R, there is a cyclic pre-proof of $\{P\}\,C;C'\,\{R\}$ in* PHL *(resp.* THL*) in which all open leaves are occurrences of $\{Q\}\,C'\,\{R\}$:*

$$\begin{array}{c}\vdots\\ \{P\}\,C\,\{Q\}\end{array} \quad\Longrightarrow\quad \begin{array}{c}\{Q\}\,C'\,\{R\}\\ \vdots\\ \{P\}\,C;C'\,\{R\}\end{array}$$

Moreover, any strongly connected subgraph of the pre-proof created by the translation satisfies the global soundness condition for PHL *(resp.* THL*) cyclic proofs.*

Proof. We proceed by structural induction on the Hoare logic proof of $\{P\}\,C\,\{Q\}$, distinguishing cases on the last rule applied in the proof and assuming arbitrary C' and R.

Case (Skip)**:** The proof transformation is as follows:

$$\frac{}{\{P\}\,\texttt{skip}\,\{P\}}\,(\text{Skip}) \quad\Longrightarrow\quad \frac{\{P\}\,C'\,\{R\}}{\{P\}\,\texttt{skip};C'\,\{R\}}\,(\text{CSkip2})$$

The only open leaf in the cyclic pre-proof is an instance of $\{P\}\,C'\,\{R\}$, as required.

Case (:=)**:**

$$\frac{}{\{P\}\,x := E\,\{P[x'/x] \wedge x = E[x'/x]\}}\,(:=) \quad\Longrightarrow\quad \frac{\{P[x'/x] \wedge x = E[x'/x]\}\,C'\,\{R\}}{\{P\}\,x := E;C'\,\{R\}}\,(\text{C}:=)$$

The only open leaf in this pre-proof is an instance of $\{P[x'/x] \wedge x = E[x'/x]\}\,C'\,\{R\}$, as required.

Case ($\models$)**:**

$$\frac{P \models P' \quad \begin{array}{c}\vdots\\ \{P'\}\,C\,\{Q'\}\end{array} \quad Q' \models Q}{\{P\}\,C\,\{Q\}}\,(\models) \quad\Longrightarrow\quad \frac{\dfrac{\{Q\}\,C\,\{R\}}{\{Q'\}\,C'\,\{R\}}\,(\text{C}\models)\\ \vdots\ (\text{IH})\\ P \models P' \quad \{P'\}\,C;C'\,\{R\}}{\{P\}\,C;C'\,\{R\}}\,(\text{C}\models)$$

On the RHS we first use the consequence rule to transform the precondition P to P', then apply the induction hypothesis (marked (IH) in the derivation) to obtain a cyclic pre-proof with open leaves of the form $\{Q'\}\,C'\,\{R\}$. By applying the consequence rule to each of these open leaves we obtain a pre-proof with open leaves of the form $\{Q\}\,C'\,\{R\}$ as required.

Case (Seq)**:**

$$\frac{\begin{array}{c}\vdots\\ \{P\}\,C_1\,\{S\}\end{array} \quad \begin{array}{c}\vdots\\ \{S\}\,C_2\,\{Q\}\end{array}}{\{P\}\,C_1;C_2\,\{Q\}}\,C \quad\Longrightarrow\quad \begin{array}{c}\{Q\}\,C'\,\{R\}\\ \vdots\ (\text{IH})\\ \{S\}\,C_2;C'\,\{R\}\\ \vdots\ (\text{IH})\\ \{P\}\,C_1;C_2;C'\,\{R\}\end{array}$$

Here, we first use the induction hypothesis with the first premise $\{P\}\,C_1\,\{S\}$ of $\{P\}\,C_1\,\{S\}$ to yield a pre-proof of $\{P\}\,C_1;C_2;C'\,\{R\}$ with open leaves all of form $\{S\}\,C_2;C'\,\{R\}$. Then, we use the induction hypothesis with the second premise to expand these leaves into pre-proofs with open leaves all of form $\{Q\}\,C'\,\{R\}$, as needed.

Case (If)**:**

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ \{P \wedge B\}\, C_1\, \{Q\} & \{P \wedge \neg B\}\, C_2\, \{Q\}\end{array}}{\{P\}\, \texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2\, \{Q\}}\ (\text{If}) \quad \Longrightarrow \quad \dfrac{\begin{array}{cc}\{Q\}\, C'\, \{R\} & \{Q\}\, C'\, \{R\} \\ \vdots\ (\text{IH}) & \vdots\ (\text{IH}) \\ \{P \wedge B\}\, C_1; C'\, \{R\} & \{P \wedge \neg B\}\, C_2; C'\, \{R\}\end{array}}{\{P\}\, \texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2; C'\, \{R\}}\ (\text{CIf})$$

Case (Inv)**:** Here the proof transformation involves creating (possibly many) new backlinks:

$$\dfrac{\begin{array}{c}\vdots \\ \{P \wedge B\}\, C\, \{P\}\end{array}}{\{P\}\, \texttt{while}\ B\ \texttt{do}\ C\, \{P \wedge \neg B\}}\ (\text{Inv}) \quad \Longrightarrow \quad \dfrac{\begin{array}{cc}\multicolumn{2}{c}{\{P\}\, \texttt{while}\ B\ \texttt{do}\ C; C'\, \{R\}} \\ \multicolumn{2}{c}{\vdots\ (\text{IH})} \\ \{P \wedge \neg B\}\, C'\, \{R\} & \{P \wedge B\}\, C; \texttt{while}\ B\ \texttt{do}\ C; C'\, \{R\}\end{array}}{\{P\}\, \texttt{while}\ B\ \texttt{do}\ C; C'\, \{R\}}\ (\text{CInv})$$

In the RHS pre-proof, we first apply the cyclic rule (CInv) to unfold the `while` loop. The resulting left-hand premise is an open leaf of the permitted form, i.e. $\{P \wedge \neg B\}\, C'\, \{R\}$. For the right-hand premise, using the induction hypothesis we can obtain a pre-proof of $\{P \wedge B\}\, C; \texttt{while}\ B\ \texttt{do}\ C; C'\, \{R\}$ with all open leaves of the form $\{P\}\, \texttt{while}\ B\ \texttt{do}\ C; C'\, \{R\}$. These leaves are all back-linked to the conclusion of the pre-proof, which is identical. We additionally note that at least one symbolic execution rule is applied along the path from this companion to any of these buds, namely the instance of (CInv) itself.

Case (Inv-Total)**:** By assumption, we have a proof of the form:

$$\dfrac{\begin{array}{c}\vdots \\ \{P \wedge B \wedge t = n\}\, C\, \{P \wedge t < n\}\end{array}}{\{P\}\, \texttt{while}\ B\ \texttt{do}\ C\, \{P \wedge \neg B\}}\ (\text{Inv-Total})$$

We derive a cyclic pre-proof of $\{P\}\, \texttt{while}\ B\ \texttt{do}\ C; C'\, \{R\}$ as follows.

$$\dfrac{\dfrac{\{P \wedge \neg B\}\, C'\, \{R\}}{\{P \wedge \neg B \wedge t = n\}\, C'\, \{R\}}\ (\text{C}\models) \qquad \begin{array}{c}\dfrac{\dfrac{\dfrac{\{P \wedge t = n\}\, \texttt{while}\ B\ \texttt{do}\ C; C'\, \{R\}}{\{P \wedge t = n'\}\, \texttt{while}\ B\ \texttt{do}\ C; C'\, \{R\}}\ (\text{Sub})}{\{P \wedge t = n' \wedge n' < n\}\, \texttt{while}\ B\ \texttt{do}\ C; C'\, \{R\}}\ (\text{C}\models)}{\{P \wedge t < n\}\, \texttt{while}\ B\ \texttt{do}\ C; C'\, \{R\}}\ (\text{C}\models) \\ \vdots\ (\text{IH}) \\ \{P \wedge B \wedge t = n\}\, C; \texttt{while}\ B\ \texttt{do}\ C; C'\, \{R\}\end{array}}{\{P \wedge t = n\}\, \texttt{while}\ B\ \texttt{do}\ C; C'\, \{R\}}\ (\text{CInv})$$
$$\dfrac{\{P \wedge t = n\}\, \texttt{while}\ B\ \texttt{do}\ C; C'\, \{R\}}{\{P\}\, \texttt{while}\ B\ \texttt{do}\ C; C'\, \{R\}}\ (\text{C}\models)$$

This construction is similar to the previous case, with a little more wrangling to deal with the termination measure t. First, before unfolding the `while` loop we "record" the value of t as a fresh variable n to obtain $t = n$ in the precondition. In the left hand premise of (CInv) this fact is not needed and is discarded again to obtain an open leaf of the permitted form $\{P \wedge \neg B\}\, C'\, \{R\}$. In the right-hand premise, we apply the induction hypothesis to obtain a pre-proof with open leaves all of form $\{P \wedge t < n\}\, \texttt{while}\ B\ \texttt{do}\ C; C'\, \{R\}$. In each of these open leaves we introduce another fresh variable n' to record the new value of t as $t = n'$, where $n' < n$, thus recognising these transformed leaves as substitution instances of the conclusion of (CInv), to which we form backlinks.

Note that we have a trace on n and n' from the companion node in this proof to each of the buds, which progresses when we "jump" from n to n' (at the point where $n' < n$ is introduced). □

$$\frac{}{[P]\,\texttt{skip}\,[P]}\ (\text{CRSkip}) \qquad \frac{[P]\,C\,[Q]}{[P]\,\texttt{skip};C\,[Q]}\ (\text{CRSkip2}) \qquad \frac{[P[x'/x] \wedge x = E[x'/x]]\,C\,[Q]}{[P]\,x := E;C\,[Q]}\ (\text{CR:=})$$

$$\frac{[P]\,C\,[Q]}{[P[t/z]]\,C\,[Q[t/z]]}\ (\text{RSub}) \qquad \frac{P \models P' \quad [P]\,C\,[Q] \quad Q' \models Q}{[P']\,C\,[Q']}\ (\text{C}\models)$$

$$\frac{[P \wedge B]\,C_1;C'\,[Q]}{[P]\,\texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2;C'\,[Q]}\ (\text{CRIf1}) \qquad \frac{[P \wedge \neg B]\,C_2;C'\,[Q]}{[P]\,\texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2;C'\,[Q]}\ (\text{CRIf2})$$

$$\frac{[P \wedge \neg B]\,C'\,[Q]}{[P]\,\texttt{while}\ B\ \texttt{do}\ C;C'\,[Q]}\ (\text{CRInv1}) \qquad \frac{[P \wedge B]\,C;\texttt{while}\ B\ \texttt{do}\ C;C'\,[Q]}{[P]\,\texttt{while}\ B\ \texttt{do}\ C;C'\,[Q]}\ (\text{CRInv2})$$

Fig. 6. Cyclic proof rules for reverse Hoare triples. x' is fresh in (C:=), and z and t are not in $fv(C)$ in (RSub).

Theorem 4.9 (Relative completeness) *If* $\{P\}\,C\,\{Q\}$ *is provable in* PHL *(resp.* THL*) then it has a cyclic proof in* PHL *(resp.* THL*).*

Proof. By taking $C' = \texttt{skip}$ and $R = Q$ in Lemma 4.8 and using the elision of $C;\texttt{skip}$ to C, we obtain a pre-proof $\mathcal{P}$ of $\{P\}\,C\,\{Q\}$ in PHL (resp. THL) in which all open leaves are of the form $\{P\}\,\texttt{skip}\,\{P\}$ and thus can be immediately closed by applications of the rule (CSkip).

To see that $\mathcal{P}$ is a genuine cyclic proof, observe that the only strongly connected subgraphs (i.e. collections of cycles) in $\mathcal{P}$ are created by translations of (Inv) (in PHL) and (Inv-Total) (in THL); these are disjoint from one another by construction, and satisfy the global soundness conditions for PHL and THL respectively. Thus $\mathcal{P}$ itself is a cyclic proof in PHL / THL as required. □

5 Cyclic proofs in reverse Hoare logic

This section presents a system of cyclic proofs for PRHL and TRHL, i.e. partial and total reverse Hoare logic, together with their soundness and their subsumption of the corresponding standard proof systems from Section 3. By and large, their development mirrors that of the cyclic proof systems for standard Hoare logic in the previous section.

First, we give our cyclic proof rules for reverse Hoare triples in Figure 6. Like the cyclic proof rules for standard Hoare triples in the previous section, they are formulated in "continuation style" (with conclusions of general form $[P]\,C;C'\,[Q]$, with a rule for explicit substitution (RSub) and an unfolding rule for `while` loops (CRInv2).

Definition 5.1 [Trace] Let $\mathcal{P}$ be a pre-proof and $([P_k]\,C_k\,[Q_k])_{k\geq 0}$ be a path in $\mathcal{P}$. For terms n and n', we say that n' is a *precursor* of n at k if one of the following holds:

- either $[P_k]\,C_k\,[Q_k]$ is the conclusion of an application of (RSub) or (CR:=) and $n' = \theta(n)$, where θ is the substitution used in the rule application; or
- $[P_i]\,C_i\,[Q_i]$ is the conclusion of another rule, and $n' = n$.

A *trace* following $([P_k]\,C_k\,[Q_k])_{k\geq 0}$ is a sequence of terms $(\mathrm{n}_k)_{k\geq 0}$ such that for every $k \geq 0$, the term n_k occurs in either P_k or Q_k and one of the following conditions holds:

- either n_{i+1} is a precursor of n_k at k; or
- there exists $(t = \mathrm{n}_{k+1}) \in P_i$ such that $\mathrm{n}_k < \mathrm{n}_{k+1}$ where n_k is a precursor of n_{k+1} at i.

In the latter case, we say that the trace *progresses* at $k+1$. An *infinitely progressing trace* is a trace that progresses at infinitely many points.

A pre-proof is a genuine cyclic proof if it satisfies the following global soundness condition(s).

Definition 5.2 [Cyclic proof] A pre-proof $\mathcal{P}$ is a *cyclic proof* in PRHL if there are infinitely many symbolic execution rule applications along every infinite path in $\mathcal{P}$. If in addition there is an infinitely progressing trace along a tail of every infinite path in $\mathcal{P}$, it is also a cyclic proof in TRHL.

Similarly to Definition 4.3, every cyclic proof in TRHL is also a cyclic proof in PRHL.

Example 5.3 We show a cyclic proof in TRHL for the reverse Hoare triple

$$[x = x_0]\ \texttt{while}\ x > 0\ \texttt{do}\ x := x - 2; [x = x_0 - 2k]$$

$$\dfrac{\dfrac{\dfrac{\dfrac{[x = \underline{x_0}]\ C\ [x = x_0 - 2k]r}{[x = \underline{x_0 - 2})]\ C\ [x = (x_0 - 2) - 2(k-1)]}\,(\text{RSub})}{[x' = \underline{x_0} \wedge x' > 0 \wedge x = x' - 2]\ \texttt{while}\ x > 0\ \texttt{do}\ x := x - 2; [x = x_0 - 2k]}\,(\text{R}\models)}{[x = \underline{x_0} \wedge x > 0]\ x := x - 2; \texttt{while}\ x > 0\ \texttt{do}\ x := x - 2; [x = x_0 - 2k]}\,(\text{CR}:=)}{[x = \underline{x_0}]\ \texttt{while}\ x > 0\ \texttt{do}\ x := x - 2; [x = x_0 - 2k]}\,(\text{CRInv2})$$

Note that, in this proof, the trace from companion to bud is underlined. It can be confirmed that the induced infinite path includes infinitely instances of symbolic application and infinite progressing points.

Lemma 5.4 *Let $\mathcal{P}$ be a pre-proof of $[P]\,C\,[Q]$ and suppose that $[P]\,C\,[Q]$ is invalid. Then there exists an infinite path $([P_k]\,C_k\,[Q_k])_{k\geq 0}$ in $\mathcal{P}$, beginning from $[P]\,C\,[Q]$, such that the following properties hold, for all $k \geq 0$:*

(i) *for all $k \geq 0$, the triple $[P_k]\,C_k\,[Q_k]$ is invalid, meaning that*
  (a) *$\forall \sigma.\ \sigma \models P_k$, $\langle C_k, \sigma\rangle\downarrow$ within a maximum of m_k steps, but there is a state $s_k \models Q_k$ that is not reachable from $\langle C, \sigma\rangle$ for any $\sigma \models P_k$; or*
  (b) *(in TRHL only) $\forall \sigma'.\ \sigma \models Q_k$, there exists $\sigma.\ \sigma \models P_k \wedge \langle C_k, \sigma\rangle \to^* \langle \texttt{skip}, \sigma'\rangle$, but $\exists \sigma.\ \sigma \models P_k$, $\langle C_k, \sigma\rangle\uparrow$.*

(ii) *If the first possibility (i)(a) above holds, then the computation length $m_{k+1} \leq m_k$, and if the rule applied at k is a symbolic execution rule then $m_{k+1} < m_k$.*

(iii) *(in TRHL only) if the second possibility (i)(b) above holds, and there is a trace $(\mathrm{n}_k)_{k\geq i}$ following a tail of the path $([P_k]\,C_k\,[Q_k])_{k\geq i}$, then the sequence of natural numbers defined by $(\sigma(\mathrm{n}(k)))_{k\geq i}$ is monotonically decreasing, and strictly decreases at every progress point of the trace.*

Theorem 5.5 (Soundness) *If $[P]\,C\,[Q]$ has a cyclic proof in PRHL (resp. TRHL) then it is valid in PRHL (resp. TRHL).*

Proof. Similar to Theorem 4.7, using Lemma 5.4. □

Lemma 5.6 (Proof translation) *If $[P]\,C\,[Q]$ is provable in PRHL (resp. TRHL) then, for all statements C' and assertions R, there is a cyclic pre-proof of $[P]\,C; C\,[R]$ in which all open leaves are occurrences of $[Q]\,C'\,[R]$:*

$$\begin{array}{c}\vdots\\ {[P]\,C\,[Q]}\end{array} \quad\Longrightarrow\quad \begin{array}{c}{[Q]\,C'\,[R]}\\ \vdots\\ {[P]\,C; C'\,[R]}\end{array}$$

Moreover, any strongly connected subgraph of the pre-proof created by the translation satisfies the global soundness condition for PRHL (resp. TRHL) cyclic proofs.

Proof. We proceed by structural induction on the Hoare logic proof of $\{P\}\,C\,\{Q\}$, distinguishing cases on the last rule applied in the proof and assuming arbitrary C' and R. The rules (RSkip) and (RInv-1) are trivial, while the cases of (R:=), (RSeq), (R⊨) are similar to their counterparts in standard Hoare logic (Lemma 4.8).

Case (RIf1), (RIf2)**:** These cases are symmetric. For (RIf1), the transformation is as follows:

$$\dfrac{\begin{array}{c}\vdots\\ [P \wedge B]\, C_1\, [Q]\end{array}}{[P]\, \texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2\, [Q]}\ (\text{RIf1}) \quad \Longrightarrow \quad \dfrac{\begin{array}{c}[Q]\, C\, [R]\\ \vdots\ (\text{IH})\\ [P]\, C_1; C\, [R]\end{array}}{[P]\, \texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2; C\, [Q]}\ (\text{CRIf1})$$

Case (RInv-2)**:**

$$\dfrac{\begin{array}{c}\vdots\\ [P \wedge B]\, C\, [P]\end{array}}{[P]\, \texttt{while}\ B\ \texttt{do}\ C\, [P \wedge \neg B]}\ (\text{RInv-2}) \quad \Longrightarrow \quad \dfrac{\begin{array}{c}[P]\, \texttt{while}\ B\ \texttt{do}\ C; C'\, [R]\\ \vdots\ (\text{IH})\\ [P \wedge B]\, C; \texttt{while}\ B\ \texttt{do}\ C; C'\, [R]\end{array}}{[P]\, \texttt{while}\ B\ \texttt{do}\ C; C'\, [R]}\ (\text{CRInv2})$$

In the pre-proof above, by using the induction hypothesis, a pre-proof of $[P \wedge B]\, C; \texttt{while}\ B\ \texttt{do}\ C; C'\, [R]$ contains open leaves which are occurrences of $[P]\, \texttt{while}\ B\ \texttt{do}\ C; C'\, [R]$. As those occurrences are involved in a back-link, the pre-proof has no open leaves. (In this case, as the first condition in global soundness holds, this pre-proof is a PRHL cyclic proof.)

Case (RInv-Total)**:** By assumption, we have a proof of the form:

$$\dfrac{\begin{array}{c}\vdots\\ [P \wedge B \wedge t < n]\, C\, [P \wedge t = n]\end{array}}{[P]\, \texttt{while}\ B\ \texttt{do}\ C\, [P \wedge \neg B]}\ (\text{Inv-Total})$$

We derive a cyclic pre-proof of $[P]\, \texttt{while}\ B\ \texttt{do}\ C; C'\, [R]$ as follows.

$$\dfrac{\dfrac{[P \wedge \neg B]\, C'\, [R]}{[P \wedge t < n \wedge \neg B]\, C'\, [R]}\ (\text{C}\models) \qquad \begin{array}{c}\dfrac{\dfrac{\dfrac{[P \wedge t < n]\, \texttt{while}\ B\ \texttt{do}\ C; C'\, [R]}{[P \wedge t < n']\, \texttt{while}\ B\ \texttt{do}\ C; C'\, [R]}\ (\text{RSub})}{[P \wedge t < n' \wedge n < n']\, \texttt{while}\ B\ \texttt{do}\ C; C'\, [R]}\ (\text{R}\models)}{[P \wedge t = n]\, \texttt{while}\ B\ \texttt{do}\ C; C'\, [R]}\ (\text{R}\models)\\ \vdots\ (\text{IH})\\ [P \wedge t < n \wedge B]\, C; \texttt{while}\ B\ \texttt{do}\ C; C'\, [R]\end{array}}{[P \wedge t < n]\, \texttt{while}\ B\ \texttt{do}\ C; C'\, [R]}\ (\text{CRInv2})$$
$$\dfrac{[P \wedge t < n]\, \texttt{while}\ B\ \texttt{do}\ C; C'\, [R]}{[P]\, \texttt{while}\ B\ \texttt{do}\ C; C'\, [R]}\ (\text{C}\models)$$

In the pre-proof above, the left hand premise includes an open leaf of the permitted form $[P \wedge \neg B]\, C'\, [R]$. The right hand premise, by applying the induction hypothesis, $[P \wedge t < n \wedge B]\, C; \texttt{while}\ B\ \texttt{do}\ C; C'\, [R]$ has a cyclic pre-proof in which all open leaves are of the form $[P \wedge t = n]\, \texttt{while}\ B\ \texttt{do}\ C\, [R]$. In each of these open leaves, we introduce another fresh variable n' to record the new value of t as $t < n'$, where $n' < n$, thus recognising these transformed leaves as substitution instances of the conclusion of (CRInv2), to which we form backlinks. Similar to the translation in Hoare logic, in this proof we have a trace on n and n' from the companion node to each of the buds, in which the progress point is at point when we "jump" from n to n' and $n' < n$ is introduced. □

Theorem 5.7 (Relative completeness) *If* $\{P\}\, C\, \{Q\}$ *is provable in* PRHL *(resp.* TRHL*) then it has a cyclic proof in* PRHL *(resp.* TRHL*).*

Proof. Similar to Theorem 4.9, using Lemma 5.6. □

6 Conclusions and future work

This paper presents formulations of the partial and total versions of Hoare logic and its reverse, semantically and as axiomatic and cyclic proof systems. We observe in particular that

- the total versions of the logics are special cases of the corresponding partial version;
- partial reverse Hoare logic `PRHL` and total Hoare logic `THL` are semantic duals, as are `TRHL` and `PHL`;
- the partial versions of the logics are proof-theoretically similar, as are the total versions, with their cyclic proof systems sharing a similar soundness condition;
- there is a natural translation from standard to cyclic proofs for each of the logics.

We must make a frank admission: Very little of what we present here is truly new, in that it could in principle have been distilled from previous works on Hoare logic, reverse Hoare logic and cyclic proof. Of course, the partial and total variants of Hoare logic have been extensively studied for decades [18,11,3,2] and their formulations as cyclic proof systems can largely be inferred from previous works, in particular, on Hoare-style proofs in separation logic [7,8]. Meanwhile, reverse Hoare logic and its extension to incorrectness logic has been studied in [14,22,19], both semantically and axiomatically, although not to our knowledge in cyclic proof form. Moreover, Verscht and Kaminski provide a semantic taxonomy of Hoare-like logics covering many more possibilities than the four we examine here [25]. We see our main contribution as being primarily one of compiling and formulating these logics in such a way that their proof-theoretic as well as their semantic relationships become clear.

We can additionally mention a few minor novelties of the present work. First, the partial variant of reverse Hoare logic we consider here, `PRHL`, does not seem to be well known, apparently not being among the logics in the Verscht-Kaminski taxonomy [25] and may even be somewhat new. A very recent short paper by Verscht et al [26] describes "partial incorrectness logic" which *also* does not seem at first sight the same thing as our `PRHL`; however, we believe that our version arises quite naturally as a semantic dual of total correctness (Defn. 3.1) and a proof-theoretic dual of partial correctness (Figure 4). Second, we formulate cyclic proof systems for reverse Hoare logic, observing that the soundness conditions required to make them work correctly are natural analogues of their equivalents in standard Hoare logic. Lastly, our essentially uniform translation of standard Hoare logic and reverse Hoare logic proofs into cyclic proofs is quite pleasant and may provide, we hope, some minor technical interest.

Potential directions for future work might include, for example, the implementation of our cyclic proof systems within a suitable platform such as the Cyclist theorem prover [9], or the extension of existing systems of incorrectness logic with our cyclic proof principles [22,19].

References

[1] Bahareh Afshari and Graham E. Leigh. Cut-free completeness for modal μ-calculus. In *Proceedings of LICS-32*, pages 1–12. IEEE, 2017.

[2] Krzysztof Apt and Ernst-Rüdiger Olderog. Fifty years of Hoare's logic. *Formal Aspects of Computing*, 31(6):751–807, 2019.

[3] Krzysztof R. Apt. Ten years of Hoare's logic: A survey-part I. *ACM Trans. Program. Lang. Syst.*, 3(4):431–483, 1981.

[4] James Brotherston. Cyclic proofs for first-order logic with inductive definitions. In *Proceedings of TABLEAUX-14*, pages 78–92. Springer, 2005.

[5] James Brotherston. *Sequent Calculus Proof Systems for Inductive Definitions*. PhD thesis, University of Edinburgh, November 2006.

[6] James Brotherston. Formalised inductive reasoning in the logic of bunched implications. In *Proceedings of SAS-14*, volume 4634 of *LNCS*, pages 87–103. Springerg, 2007.

[7] James Brotherston, Richard Bornat, and Cristiano Calcagno. Cyclic proofs of program termination in separation logic. In *Proceedings of POPL-35*. ACM, 2008.

[8] James Brotherston and Nikos Gorogiannis. Cyclic abduction of inductively defined safety and termination preconditions. In *Proceedings of SAS-21*. Springer, 2014.

[9] James Brotherston, Nikos Gorogiannis, and Rasmus L. Petersen. A generic cyclic theorem prover. In *Proceedings of APLAS-10*. Springer, 2012.

[10] James Brotherston and Alex Simpson. Sequent calculi for induction and infinite descent. *Journal of Logic and Computation*, 21(6):1177–1216, 2011.

[11] Stephen A. Cook. Soundness and completeness of an axiom system for program verification. *SIAM J. Comput.*, 7(1):70–90, 1978.

[12] Mads Dam and Dilian Gurov. μ-calculus with explicit points and approximations. *Journal of Logic and Computation*, 12(2):255–269, 2002.

[13] Anupam Das and Damien Pous. A cut-free cyclic proof system for Kleene algebra. In *Proceedings of TABLEAUX*, pages 261–277. Springer, 2017.

[14] Edsko de Vries and Vasileios Koutavas. Reverse hoare logic. In *Proceedings of SEFM*, pages 155–171. Springer, 2011.

[15] Simon Docherty and Reuben N.S. Rowe. A non-wellfounded, labelled proof system for propositional dynamic logic. In *Proceedings of TABLEAUX*, pages 335–352. Springer, 2019.

[16] Gadi Tellez Espinosa and James Brotherston. Automatically verifying temporal properties of programs with cyclic proof. *Journal of Automated Reasoning*, 64:555–578, 2019.

[17] Robert W. Floyd. Assigning meanings to programs. In *Proc. Amer. Math. Soc.*, volume 19 of *Symposia in Applied Mathematics*, pages 19–31, 1967.

[18] C. A. R. Hoare. An axiomatic basis for computer programming. *Commun. ACM*, 12(10):576–580, 1969.

[19] Quang Loc Le, Azalea Raad, Jules Villard, Josh Berdine, Derek Dreyer, and Peter W. O'Hearn. Finding real bugs in big programs with incorrectness logic. In *Proceedings of OOPSLA*. ACM, 2022.

[20] Kenneth L. McMillan. Quantified invariant generation using an interpolating saturation prover. In *Proceedings of TACAS-14*, pages 413–427. Springer, 2008.

[21] Damian Niwiński and Igor Walukiewicz. Games for the μ-calculus. *Theoretical Computer Science*, 163:99–116, 1997.

[22] Peter W. O'Hearn. Incorrectness logic. 2019.

[23] Alex Simpson. Cyclic arithmetic is equivalent to Peano arithmetic. In *Proceedings of FoSSaCS*, pages 283–300. Springer, 2017.

[24] C. Stirling and D. Walker. Local model checking in the modal μ-calculus. *Theoretical Computer Science*, 89:161–177, 1991.

[25] Lena Verscht and Benjamin Lucien Kaminski. A taxonomy of Hoare-like logics: Towards a holistic view using predicate transformers and Kleene algebras with top and tests. 2025.

[26] Lena Verscht, Ānrán Wáng, and Benjamin Lucien Kaminski. Partial incorrectness logic, 2025. https://arxiv.org/abs/2502.14626.