Non Standard Axiomatic Semantics

Patrick Cousot

Courant Institute, New York University

Abstract. We show that the axiomatic semantics (Hoare logic) has non-standard models so does not provide a unique well-defined semantics of programming languages. We propose methods to solve this problem.

Keywords: Hoare logic, Trace semantics, Semantics standardization, Abstract Interpretation.

1 Introduction

We recall Dedekind's categoricity [11] which in addition to Peano's axiomatization [16] of the naturals eliminates Skolem's nonstandard models of arithmetic [17] thus ensuring a unique definition of the naturals (up to an isomorphism) that corresponds to the intended interpretation of the naturals. We show that the axiomatic semantics based on Hoare logic [13] suffers from a similar ambiguity problem. The rule for iteration does not uniquely define the intended meaning of loops (up to isomorphism). We propose two remedies, one based on minimality, as in Dedekind's categoricity, but it does not apply to abstract Hoare logics. The other consists in understanding Hoare logic as an abstraction of a standard trace semantics.

2 Fixpoints and Fixpoint Induction

Tarski's theorem states [18] that an increasing function $f \in L \to L$ on a complete lattice $\langle L, \sqsubseteq, \sqcap \rangle$ has a least fixpoint $\mathsf{lfp}^{\sqsubseteq} f = \bigsqcap \{x \in L \mid f(x) \sqsubseteq x\}$. By definition of "least" and reflexivity, this least fixpoint is unique.

Park's fixpoint induction [15] for an increasing function $f \in L \to L$ on a complete lattice $\langle L, \sqsubseteq, \sqcap \rangle$ and $P \in L$ is $(\mathsf{lfp}^{\sqsubseteq} f \sqsubseteq P) \Leftrightarrow (\exists I \in L \,.\, f(I) \sqsubseteq I \wedge I \sqsubseteq P.)$. For soundness ($\Leftarrow$), $f(I) \sqsubseteq I$ implies $I \in \{x \in L \mid f(x) \sqsubseteq x\}$ and so by Tarski theorem and definition of the greatest lower bound $\sqcap$, $\mathsf{lfp}^{\sqsubseteq} f = \bigsqcap \{x \mid f(x) \sqsubseteq x\} \sqsubseteq I$ so that $I \sqsubseteq P$ implies $\mathsf{lfp}^{\sqsubseteq} f \sqsubseteq P$ by transitivity. For completeness ($\Rightarrow$), take $I = \mathsf{lfp}^{\sqsubseteq} f$. P is called an invariant of f and I an inductive invariant.

3 Nonstandard Models of Arithmetics

Peano defined the naturals $\mathbb{N}$ by $0 \in \mathbb{N}$ and $\forall n \in \mathbb{N} \,.\, \mathcal{S}n \in \mathbb{N}$ where $\mathcal{S}n = n + 1$ is the successor function [16]. The intention was to define the naturals $\mathbb{N}$ as

$0 < 1 < 2 < 3 < \ldots$. Skolem [17] showed that there are nonstandard models of Peano's definition such as the model $S = 0 < 1 < 2 < 3 < \ldots < \ldots - 2 < -1 < 0' < 1' < 2' \ldots$ which obviously satisfies $0 \in S$ and $\forall n \in s \,.\, n + 1 \in S$. So Peano naturals are not well-defined since the axiomatization has more than one model up to isomorphism.

Dedekind solution [11] to ensure the uniqueness of the model (also called categoricity) relies on the chain principle (or second-order induction) that is essentially De Morgan mathematical induction (proofs by recurrence) [14]. From $P(0)$ and $\forall n \,.\, P(n) \Rightarrow P(n+1)$ conclude $\forall n \in \mathbb{N} \,.\, P(n)$ (which is the contrapositive form of Fermat's infinite descent from $\forall n \,.\, \neg P(n+1) \Rightarrow \neg P(n)$, conclude $\forall n \in \mathbb{N} \,.\, P(n)$). Dedekind's additional requirement is that the models of Peano's naturals are those for which proof by recurrence is valid (sound, correct). This eliminates the Skolem's model S since for $P = \{0, 1, 2, \ldots\}$ (which is a definition in extension of a predicate where $P(n)$ denotes $n \in P$) the hypotheses of the recurrence principle are satisfied by P while the conclusion $S \subseteq P$ is not. So the soundness of the principle of proof by recurrence effectively limits the Peano's definition to only those elements generated by the successor function $\mathcal{S}$ starting from 0.

Observe that $\mathbb{N}$ is the smallest model of Peano's naturals since $\mathbb{N} \setminus \{n\}$, $n \in \mathbb{N}$ does not satisfies Peano's definition. This is Dedekind 's idea of defining $\mathbb{N}$ as the smallest subset of an infinite set such that there exists a transformation (the successor function $\mathcal{S}$) that is injective and a base element 0 not in the range of $\mathcal{S}$. Otherwise stated, the naturals are the least fixpoint $\mathbb{N} = \mathsf{lfp}^{\subseteq} F$ of a transformer $F \in \wp(\mathbb{U}) \to \wp(\mathbb{U})$ defined as $F(X) \triangleq \{0\} \cup \{\mathcal{S}n \mid n \in X\}$ operating on the powerset of a universe $\mathbb{U}$ which can be chosen for example as finite sentences over an alphabet contaning 0, 1, and $\mathcal{S}$ possibly $+$, $-$, $\times$, and any other required letters. Then the powerset $\langle \wp(\mathbb{U}), \subseteq \rangle$ is a complete lattice and F is $\subseteq$-increasing (monotone) so that the $\subseteq$-least fixpoint $\mathsf{lfp}^{\subseteq} F$ of F exists by Tarski's theorem and is unique by definition of "least". This definition allows us to recover proof by recurrence, as follows ($P \in \wp(\mathbb{U})$)

$$
\begin{array}{ll}
& \mathbb{N} \subseteq P \\
\Leftrightarrow & \mathsf{lfp}^{\subseteq} F \subseteq P & \wr \text{fixpoint definition } \mathbb{N} = \mathsf{lfp}^{\subseteq} F \int \\
\Leftrightarrow & \exists I \in \wp(\mathbb{U}) \,.\, F(I) \subseteq I \wedge I \subseteq P & \wr \text{Park's fixpoint induction} \int \\
\Leftrightarrow & \exists I \in \wp(\mathbb{U}) \,.\, \{0\} \cup \{Sn \mid n \in I\} \subseteq I \wedge I \subseteq P & \wr \text{definition of } F \int \\
\Leftrightarrow & \exists I \in \wp(\mathbb{U}) \,.\, 0 \in I \wedge \forall n \in I \,.\, Sn \in I \wedge I \subseteq P & \wr \text{definition of } \cup \text{ and } \subseteq \int
\end{array}
$$

This is exactly second-order induction also called recurrence principle (except that it is emphasized that P might have to be strengthened to I to be provable by recurrence, a situation which is so common in mathematics that it does not even need to be mentioned in the recurrence principle). We conclude that the fixpoint definition of the naturals $\mathbb{N}$ as "the smallest set such that $0 \in \mathbb{N}$ and $\forall n \in \mathbb{N} \,.\, \mathcal{S}n \in \mathbb{N}$" is equivalent to (and much simpler than) the mathematical definition requiring "$0 \in \mathbb{N}$, $\forall n \in \mathbb{N} \,.\, \mathcal{S}n \in \mathbb{N}$, and the recurrence principle is valid in $\mathbb{N}$".

Notice that there are different possible universes but the defined naturals in each case are isomorphic, this is Dedekind's categoricity theorem.

4 Hoare Logic

Hoare logic [13] uses triples $\{P\}\,S\,\{Q\}$ (originally written $P\,\{S\}\,Q$ in [13]) to specify that any execution of statement S from a state in P, which terminates (if ever), will terminate in a state satisfying Q. To avoid the inexpressivity problems of Hoare logic related to the choice of a logic to specify P (or Q) [2,3], we assume that predicates are defined in extension as the set of all states satisfying P (or Q).

Hoare defined the semantics of a simple imperative programming language by the following rules, which as those of [13], except that predicates are defined in extension as sets of environments ρ mapping variables $\mathtt{x}$ to their value $\rho(\mathtt{x})$ satisfying these predicates. We define the assignment $\rho[\mathtt{x} \leftarrow v]$ as $\rho[\mathtt{x} \leftarrow v](\mathtt{x}) = v$ and $\rho[\mathtt{x} \leftarrow v](\mathtt{y}) = \rho(\mathtt{y})$ when $\mathtt{y} \neq x$.

- Axiom of Assignment: $\{P\}\,\mathtt{x := f}\,\{\rho[\mathtt{x} \leftarrow \phi(\rho)] \mid \rho \in P \wedge \phi(\rho)$ is the value of expression f when evaluated with values $\rho(\mathtt{y})$ of the variables $\mathtt{y}$ of $\mathtt{f}\}$;
- Rules of Consequence:
  - If $\{P\}\,Q\,\{R\}$ and $R \subseteq S$ then $\{P\}\,Q\,\{S\}$ (where logical implication $R \subseteq S$ is written $R \supset S$ in [13]);
  - If $\{P\}\,Q\,\{R\}$ and $S \subseteq P$ then $\{S\}\,Q\,\{R\}$;
- Rule of Composition:
  If $\{P\}\,Q_1\,\{R_1\}$ and $\{R_1\}\,Q_2\,\{R\}$ then $\{P\}\,Q_1\,\mathtt{;}\,Q_2\,\{R\}$
- Rule of Iteration:
  If $\{P \cap B\}\,S\,\{P\}$ then $\{P\}\,\mathtt{while}\,B\,\mathtt{do}\,S\,\{\neg B \cap P\}$ where B (respectively $\neg B$) stands for $\{\rho \mid$ the evaluation of Boolean expression $\mathtt{B}$ for the values $\rho(\mathtt{x})$ of the variables $\mathtt{x}$ appearing in $\mathtt{B}$ is true$\}$ (respectively false).

The derived intersection rule, from $\forall i \in \Delta\,.\,\{P_i\}\,\mathtt{S}\,\{Q_i\}$ infer $\{\bigcap_{i\in\Delta} P_i\}\,\mathtt{S}\,\{\bigcap_{i\in\Delta} Q_i\}$ can be proved by induction on the structure of programs [10].

It is usually thought that these rule provide an axiomatic definition of the semantics of the programming language defining the possible executions of programs by establishing logical rules, rather than showing how they execute. We show that this is wrong. As was the case for Peano arithmetics, there are different non-standards models of program executions that satisfy the requirements of Hoare logic and so the semantics is not well-defined.

5 Nonstandard Axiomatic Semantics

Let us consider various trace semantics [19] of the programming language. Let Σ be the set of states ρ mapping variables $\mathtt{x}$ to their value $\rho(\mathtt{x})$.

– The standard semantics has domain $D^1 \triangleq [0,n] \to \Sigma$, $n \in \mathbb{N}$ (finite traces) union $\mathbb{N} \to \Sigma$ (infinite traces).
– The nonstandard semantics domain D^2 is D^1 union $-\mathbb{Z} \to \Sigma$ mapping the negative integers to states that is traces that never start but terminate.
– The standard semantics domain D^3 is D^2 union $\mathbb{Z} \to \Sigma$ mapping the integers to states that is traces that never start nor terminate.

Let use show that Hoare logic cannot distinguish between these models (and others e.g. using transfinite traces [4, footnote 17], [12]).

5.1 Counter-example 1

Consider the program `while x<0 do x:=x+1` on integers which traces in D^1 are $\{-n\ -n+1\ \ldots\ 0 \mid n \in \mathbb{N}\}$ and $\{n \mid n \in \mathbb{N}\}$ since the program executions starting with a negative value $-n$ of variable `x` will iteratively increment it by one until reaching 0 and exiting the loop or else the program executions will start with a value n of `x` which is greater than or equal to 0, in which case the loop is not entered and immediately exited.

The method to show that Hoare logic includes any of these traces is for each one to define P to be the set of states along that trace and to prove that P is a loop inductive invariant according to Hoare's rule of iteration.

For example, consider the trace $-n\ -n+1\ \ldots\ 0$ for a given $n \in \mathbb{N}$ where $\rho \in \Sigma$ such that $\rho(\mathtt{x}) = k$ is encoded simply by k. Let $P = \{\rho \mid \rho(\mathtt{x}) \in [-n, 0]\}$ be the set of states along that trace. We have $P \cap \{\rho \mid \rho(\mathtt{x}) < 0\} = \{\rho \in P \mid \rho(\mathtt{x}) < 0\} = \{\rho \mid \rho(\mathtt{x}) \in [-n, -1]\}$ and $\{\{\rho \mid \rho(\mathtt{x}) \in [-n,-1]\}\}\mathtt{x:=x+1}\{\{\rho \mid \rho(\mathtt{x}) \in [-n+1, 0]\}$ by the assignment rule. Moreover $\{\rho \mid \rho(\mathtt{x}) \in [-n+1, 0]\} \subseteq P$ so that $\{P \cap \{\rho \mid \rho(\mathtt{x}) < 0\}\,\mathtt{x:=x+1}\,\{P\}$ holds by the consequence rule. By the iteration rule, $\{P\}\,\mathtt{while\ x>0\ do\ x:=x+1}\,\{\neg(x > 0) \cap P\}$ proving that the trace $-n\ -n+1\ \ldots\ 0$ is feasible for that program. Similarly for any $n \in \mathbb{N}$ we can prove $\{\{\mathrm{n}\}\}$`while x<0 do x:=x+1`$\{\{\mathrm{n}\}\}$ proving the feasibility of the traces $\{n \mid n \in \mathbb{N}\}$.

Now consider the traces in D^2 so that the program semantics now include the trace $\ldots\ -2\ -1\ 0$ that never starts and stops at 0. The invariant is now $P = \{z \in \mathbb{Z} \mid z \leqslant 0\}$ which also satisfies Hoare's verification conditions for iteration so that this trace is also proved to be feasible.

It follows that program `while x<0 do x:=x+1` has the standard semantics $\{-n\ -n+1\ \ldots\ 0 \mid n \in \mathbb{N}\} \cup \{n \mid n \in \mathbb{N}\}$ in D^1 and the nonstandard semantics $\{\ldots\ -2\ -1\ 0\} \cup \{-n\ -n+1\ \ldots 0 \mid\ n \in \mathbb{N}\} \cup \{n \mid n \in \mathbb{N}\}$ in D^2. This shows that for some programs, Hoare logic cannot distinguish between the standard trace semantics on D^1 and the non-standard one on D^2.

5.2 Counter-example 2

Consider now the program `while (true) x:=x+1;` which traces in D^1 are $\{n\ n+1\ n+2\ \ldots \mid n \in \mathbb{Z}\}$ and traces on D^3 also contain the trace $\ldots\ -2\ -1\ 0\ 1\ 2\ \ldots$. Again for all of these traces the set of their states is a valid invariant in Hoare

logic so that Hoare logic cannot distinguish between the standard semantics on D^1 and the non-standard semantics on D^3.

It follows that the popular statement that "Axiomatic semantics is a formal approach in computer science that defines a program's meaning by its effects on logical assertions (pre- and post-conditions), rather than how it executes" is wrong since the defined meaning is not unique up to isomorphism. This is the same problem as for Peano's original definition of the naturals.

6 Standardization of the Axiomatic Semantics

One method to ensure the unicity of the computation model is by explicitly specifying that the set of defined traces is the subset of finite or infinite traces of D_1 such the set of states along each of these traces satisfies the proof rules. This is essentially the method used in Event-B where the shape of traces are explicitly specified [1, Ch. 14.3] and machines are specified by defining the states and invariants satisfying proof obligations which are essentially Hoare proof rules [7].

The second method is to require that the trace semantics satisfy the strongest invariant. For example 1, this is $\{-n \ -n+1 \ \ldots \ 0 \mid n \in \mathbb{N}\} \cup \{n \mid n \in \mathbb{N}\}$ thus eliminating the non-standard trace $\ldots \ -2 \ -1 \ 0$ in D^2. For example 2, the strongest invariant is $\{n \ n{+}1 \ n{+}2 \ \ldots \mid n \in \mathbb{Z}\}$ thus eliminating the nonstandard trace $\ldots \ -2 \ -1 \ 0 \ 1 \ 2 \ \ldots$ in D^3.

For all statements but iteration, the strongest invariant is given by Hoare's rules defined so as to provide the strongest postcondition provide the consequence rule is not used [8]. For iteration. the strongest invariant is characterized as

Theorem 1. *The strongest invariant J of the iteration* `while B do S` *is* $J \triangleq \bigcap\{I \in \wp(\Sigma) \mid \{I \cap B\}\, \mathtt{S}\, \{I\}\}$.

Proof (of th. 1). By definition of intersection, J is stronger than any invariant since $\{I \cap B\}\, \mathtt{S}\, \{I\}$ implies $J \subseteq I$. It remains to show that J is a loop invariant. Let $\langle I^k,\ k \in \Delta\rangle$ be the family of all $I \in \wp(\Sigma)$ such that $\{I \cap B\}\, \mathtt{S}\, \{I\}$. If empty, the iteration rule is not applicable. Otherwise, we have $\forall k \in \Delta\ .\ \{I^k \cap B\}\, \mathtt{S}\, \{I^k\}$ which implies $\{(\bigcap_{k\in\Delta} I^k) \cap B\}\, \mathtt{S}\, \{(\bigcap_{k\in\Delta} I^k)\}$ by the intersection rule, and therefore $\{J \cap B\}\, \mathtt{S}\, \{J\}$ by definition of J.

So the strongest J eliminates the nonstandard models of iteration but unfortunately, the fact that the strongest invariant should be used to select the valid traces in the trace semantics is not directly expressible in the Hoare logic itself.

7 Abstract Hoare logics

Abstract Hoare logics have been introduced in [10] with the idea that state properties in $\wp(\Sigma)$ in Hoare triples, can be replaced by abstract properties chosen

in an abstract domain $\langle L, \sqsubseteq \rangle$ which meaning is given by a Galois connection $\langle \wp(\Sigma), \subseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle L, \sqsubseteq \rangle$ [5, Ch. 11]. If γ is increasing but not join-preserving then the Hoare rules may be sound while the intersection rule is incorrect [10, p. 219]. This means that there is no strongest invariant in the abstract able to eliminate nonstandard executions (which may also be introduced by the abstraction α anyway). In that case the categorization of Hoare logic, that is (in the sense of Dedekind) defining a unique model of execution, up to isomorphism, can only be done by explicitly providing this model of execution (as finite and infinite traces), which is the approach of [6].

8 Conclusion

Hoare postulated the logic named after him and relied implicitly on a model of computation which defines possible computations of programs. When later proved sound and complete, the same implicit model of computation was used [2,3]. The design of logic by abstract interpretation of a trace semantics [8,6] makes the same hypothesis. So non-standard computation models were never considered. A solution is to make this assumption explicit when providing an axiomatic semantics or to use strongest postconditions [9], but the later solution does not apply to abstract Hoare logics.

References

1. Abrial, J.: Modeling in Event-B: System and Software Engineering. Cambridge University Press (2010)
2. Cook, S.A.: Soundness and completeness of an axiom system for program verification. SIAM J. Comput. **7**(1), 70–90 (1978). https://doi.org/10.1137/0207005, https://doi.org/10.1137/0207005
3. Cook, S.A.: Corrigendum: Soundness and completeness of an axiom system for program verification. SIAM J. Comput. **10**(3), 612 (1981). https://doi.org/10.1137/0210045, https://doi.org/10.1137/0210045
4. Cousot, P.: Constructive design of a hierarchy of semantics of a transition system by abstract interpretation. Theor. Comput. Sci. **277**(1-2), 47–103 (2002). https://doi.org/10.1016/S0304-3975(00)00313-3, https://doi.org/10.1016/S0304-3975(00)00313-3
5. Cousot, P.: Principles of Abstract Interpretation. MIT Press (2021), https://mitpress.mit.edu/9780262044905/principles-of-abstract-interpretation/
6. Cousot, P.: Calculational design of [in]correctness transformational program logics by abstract interpretation. Proc. ACM Program. Lang. **8**(POPL), 175–208 (2024). https://doi.org/10.1145/3632849, https://doi.org/10.1145/3632849
7. Cousot, P.: The Essence of Event-B from an Abstract Interpretation Perspective. Abrial Gedenkschrift, Springer, (to appear) (2026)
8. Cousot, P.: On the design of program logics. In: Dietsch, D., Rybalchenko, A., Schäf, M., Wies, T. (eds.) On the Pursuit of Insight and Elegance - Essays Dedicated to Andreas Podelski on the Occasion of His 65th Birthday. pp. 92–106. Lecture Notes in Computer Science, Springer (2026). https://doi.org/10.1007/978-3-032-13711-1_6, https://doi.org/10.1007/978-3-032-13711-1_6

9. Cousot, P., Cousot, R.: Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: Graham, R.M., Harrison, M.A., Sethi, R. (eds.) Conference Record of the Fourth ACM Symposium on Principles of Programming Languages, Los Angeles, California, USA, January 1977. pp. 238–252. ACM (1977). https://doi.org/10.1145/512950.512973, https://doi.org/10.1145/512950.512973
10. Cousot, P., Cousot, R., Logozzo, F., Barnett, M.: An abstract interpretation framework for refactoring with application to extract methods with contracts. In: Leavens, G.T., Dwyer, M.B. (eds.) Proceedings of the 27th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2012, part of SPLASH 2012, Tucson, AZ, USA, October 21-25, 2012. pp. 213–232. ACM (2012). https://doi.org/10.1145/2384616.2384633, https://doi.org/10.1145/2384616.2384633
11. Dedekind, R.: Was sind und was sollen die Zahlen? Friedr. Vieweg & Sohn, Braunschweig, 4th edition, 1911 edn. (1888)
12. Giacobazzi, R., Mastroeni, I.: Non-standard semantics for program slicing. High. Order Symb. Comput. **16**(4), 297–339 (2003). https://doi.org/10.1023/A:1025872819613, https://doi.org/10.1023/A:1025872819613
13. Hoare, C.A.R.: An axiomatic basis for computer programming. Commun. ACM **12**(10), 576–580 (1969). https://doi.org/10.1145/363235.363259, https://doi.org/10.1145/363235.363259
14. Morgan, A.D.: Mathematical induction. The Penny Cyclopedia of the Society for the Diffusion of Useful Knowledge **12** (1838)
15. Park, D.: Fixpoint induction and proofs of program properties. In: Meltzer, B., Michie, D. (eds.) Machine Intelligence 5, pp. 59–78. Edinburgh University Press (1970)
16. Peano, G.: Arithmetices principia, nova methodo exposita. Fratres Bocca, Torino (1889)
17. Skolem, T.: Über die nicht-charakterisierbarkeit der zahlenreihe mittels endlich oder abzählbar unendlich vieler aussagen mit ausschließlich zahlenvariablen. Fundamenta Mathematicae **23**, 150–161 (1934). https://doi.org/10.4064/fm-23-1-150-161, https://doi.org/10.4064/fm-23-1-150-161
18. Tarski, A.: A lattice-theoretical fixpoint theorem and its applications. Pacific Journal of Mathematics **5**(2), 285–309 (1955). https://doi.org/10.2140/pjm.1955.5.285, https://doi.org/10.2140/pjm.1955.5.285
19. Wegner, P.: Operational semantics of programming languages. In: Proceedings of ACM Conference on Proving Assertions About Programs, Las Cruces, New Mexico, USA, January 6-7, 1972. pp. 128–141. ACM (1972). https://doi.org/10.1145/800235.807081, https://doi.org/10.1145/800235.807081