Research Proposal: Embedding Incorrectness Logic into Dynamic Logic

Kai Phan - U7779236

Abstract

Formal reasoning about programs traditionally focuses on proving correctness, ensuring that no execution leads to undesirable behavior. Incorrectness Logic (IL), introduced by O'Hearn, complements this perspective by providing a framework for proving the presence of bugs through under-approximate reasoning. Meanwhile, Dynamic Logic (DL) offers a powerful modal foundation that subsumes Hoare Logic and supports a wide range of correctness and modal properties. Although O'Hearn conjectured a potential embedding of IL into DL, no formal definition, proof, or mechanisation currently exists.

This project aims to formally define and verify an embedding of Incorrectness Logic into Dynamic Logic, thereby unifying correctness and incorrectness reasoning within a single expressive framework. We will develop a precise semantic translation of IL triples to DL formulas, establish proof-theoretic connections showing how IL inference rules can be derived in DL, and analyze relationships with other formalisms such as Hoare Logic and Dijkstra's weakest precondition calculus. Finally, we will mechanise the embedding in a proof assistant such as Isabelle/HOL to ensure fully rigorous, machine-checked proofs and enable future tool support.

By bridging these two important logics, the project aspires to advance the theoretical foundations of program analysis, support integrated reasoning about correctness and bugs, and promote new directions for compositional and automated verification.

1 Background

Formal reasoning about program behavior lies at the core of programming languages research and modern software verification. For decades, Hoare Logic has served as a trusted tool for specifying and checking the correctness of code. A typical Hoare triple $\{P\}\ C\ \{Q\}$ asserts that, whenever command C starts in a state satisfying precondition P and eventually halts, the resulting state must meet postcondition Q [1]. By chaining such triples together, users can prove that their programs respect the rules they claim to follow and, in turn, rule out hidden bugs.

Yet Hoare Logic remains conservative: it reasons about all possible executions, considering every potential path that might lead to safety violations. This over-approximation guarantees that no harmful outcome can escape detection but significantly increases the proof effort, particularly in large, mutable code bases. As a result, researchers and practitioners often find that while Hoare Logic robustly addresses the overarching question of safety, it does not directly identify a more immediate and practical concern: where the code actually fails or crashes in real executions.

To fill this gap, O'Hearn [2] introduced Incorrectness Logic (IL) as a conceptual twin to Hoare Logic. Rather than demonstrating that a program is always correct, IL employs *under-approximation* to prove that bugs actually exist. An incorrectness triple, written as $[P]\ C\ [Q]$, asserts that there exists at least one execution of C that starts in a state satisfying P and terminates in a state satisfying Q. By focusing explicitly on demonstrating the existence of errors, IL provides a robust foundation for bug-finding tools. These tools can concentrate on reporting only true bugs, thereby producing results without false positives. This strategy aligns precisely with the practical objectives of modern static analysis tools.

Parallel to these developments, Dynamic Logic (DL) has been established as a highly general framework for program reasoning within modal logic. DL subsumes Hoare Logic by treating programs as modal operators. For instance, $[C]\varphi$ states that φ holds after every execution of C, while $\langle C\rangle\varphi$ expresses that there exists at least one execution of C after which φ holds [3]. This expressive power allows DL to capture not only correctness properties but also modal properties, program equivalences, relational specifications, and even temporal perspectives, all within a single uniform logic. Indeed, Hoare Logic embeds naturally into DL via the well-known construction $\{P\}\ C\ \{Q\} \equiv P \to [C]Q$, thus integrating program correctness reasoning into a broader modal framework.

O'Hearn [2] has conjectured that a similar embedding might exist for IL. Embedding IL into DL would unify correctness and incorrectness reasoning, allowing one to leverage DL's expressive power to handle both over- and under-approximate analyses. Such an embedding would facilitate a deeper theoretical understanding of IL, potentially reveal connections to other logics (such as temporal and relational logics), and provide a foundation for developing unified verification tools capable of proving correctness and catching bugs within a single framework.

Despite its potential, no existing work has formally defined or proved an embedding of IL into DL. O'Hearn [2] explicitly suggested this possibility but left it as an open problem. A precise and rigorous embedding would require a careful formalization of IL's under-approximate semantics within the modal structures of DL, as well as proofs of key meta-theoretical properties such as soundness and completeness.

The goal of this project is to first formally define and then verify an embedding of Incorrectness Logic into Dynamic Logic. Concretely, the project involves constructing an explicit mapping from IL triples to DL formulas and proving that this mapping preserves the intended semantics, along with establishing key meta-theoretical properties such as soundness and completeness. In addition, the project aims to mechanise these results in a proof assistant such as Isabelle/HOL, ensuring that the proofs are fully rigorous, independent, and reusable in future tool development. By establishing this theoretical bridge, the project aspires to advance program analysis and support new integrated and compositional reasoning approaches for both correctness and bugs, while enriching the formal proof infrastructure in systems such as Isabelle/HOL.

2 Research Plan

Aim 1: Define a semantic embedding of Incorrectness Logic into Dynamic Logic

The first aim focuses on establishing a precise and rigorous semantic embedding of Incorrectness Logic (IL) into Dynamic Logic (DL). Specifically, this aim involves constructing a formal translation that maps IL triples, which describe under-approximate behaviors, into modal assertions in DL. While Hoare Logic has an established embedding into DL via the well-known transformation $\{P\}\ C\ \{Q\} \equiv P \to [C]Q$, no such mapping currently exists for IL, which fundamentally differs in its use of under-approximation. To address this gap, we will study the semantic foundations of IL, particularly the meaning of its presumption (initial state set) and result (reachable error states) components, and analyze how they correspond to modal operators in DL such as the diamond modality $\langle C\rangle\varphi$, which intuitively captures existential reachability properties.

We plan to define a formal transformation that systematically rewrites each IL triple into a DL formula that faithfully represents its under-approximate semantics. This process will involve carefully distinguishing between universal modalities, which capture universal properties, and existential modalities, which express the existence of specific executions — the latter being central to IL's reasoning style. Throughout this work, we will develop a series of small, illustrative example programs to validate and refine the embedding approach. By iteratively analyzing these examples and formally proving the equivalence of IL semantics and their DL translations, we will ensure that the proposed mapping captures all essential aspects of incorrectness reasoning. Furthermore, we will document potential limitations and edge cases where this embedding may not directly apply, thus providing a comprehensive and transparent theoretical foundation for the remainder of the project.

Aim 2: Develop proof-theoretic connections

The second aim focuses on establishing rigorous proof-theoretic links between Incorrectness Logic and Dynamic Logic. While the first aim centers on semantic embedding at the formula level, this aim moves one level higher to the structure of proofs and derivations themselves. Specifically, we will investigate how the inference rules and proof principles of IL can be systematically reconstructed or derived as modal theorems within DL. By doing so, we intend to show that IL does not require a completely separate deductive system, but can instead be treated as a specialized fragment or layer inside DL.

To achieve this, we will carefully analyze the core proof rules of IL — including sequential composition, conditional branching, and loops — and identify their equivalents or counterparts within the modal framework of DL. In particular, we will formalize how IL's under-approximate reasoning can be expressed through existential modalities in DL proofs, contrasting this with the universal reasoning used in correctness proofs. We will also investigate how rule derivations in DL may become more succinct or expressive when viewed from an IL perspective, and whether any simplifications or generalizations arise from the modal formulation.

Throughout this process, we plan to construct formal proof sketches and small example derivations, gradually expanding to more complex program structures. These explorations will allow us to validate the correctness and completeness of the embedding at the proof level, rather than just at the semantic level. Ultimately, this aim will provide a strong meta-theoretical foundation for understanding how IL-based bug reasoning can be unified with correctness reasoning inside a single logical framework, thereby opening pathways for integrated proof strategies and tool support.

Aim 3: Analyze relationships with other logics

The third aim seeks to explore and clarify the broader theoretical landscape by analyzing how the proposed embedding of Incorrectness Logic into Dynamic Logic relates to other established logics and reasoning frameworks. First, we will systematically compare the new IL embedding to the classical embedding of Hoare Logic into Dynamic Logic, which relies on universal modalities and over-approximate reasoning. By contrasting the under-approximate semantics of IL with the over-approximate correctness guarantees of Hoare Logic, we aim to highlight both the conceptual duality and the practical distinctions between proving correctness and proving the existence of bugs.

Additionally, we will investigate connections to Dijkstra's weakest precondition calculus (wp-calculus), which traditionally serves as a foundation for backward reasoning about program correctness [4]. Since wp-calculus focuses on computing preconditions that guarantee postconditions under all executions, it offers an interesting contrast to the existential style of IL. We will analyze whether elements of wp-calculus can be reformulated or generalized within our embedding framework and explore potential opportunities for unifying or complementing these approaches.

Beyond these direct comparisons, this aim also includes examining the implications of our embedding for more advanced logical frameworks, such as relational logics and hyperproperties. Relational logics enable reasoning about relationships between multiple program executions or between different programs, while hyperproperties generalize these ideas to sets of execution traces. By studying how incorrectness reasoning interacts with these frameworks, we will assess whether our approach supports or extends reasoning about security properties, information flow, and program equivalence. Through these investigations, this aim will not only situate our work within the broader logical ecosystem but also uncover possible directions for future research and applications in compositional and relational verification.

Aim 4: mechanisation in proof assistants

The final aim of this project focuses on mechanising the theoretical results within a modern proof assistant, such as Isabelle/HOL. While semantic embeddings and proof-theoretic analyses provide strong theoretical guarantees on paper, formalizing them in a machine-checked environment ensures that these results are completely rigorous and free from hidden assumptions or overlooked corner cases. By developing a prototype formalization of the IL-to-DL embedding in Isabelle/HOL, we will demonstrate that the translation and its key meta-theoretical properties — including soundness and completeness — can be formally verified in a widely used and well-supported proof assistant.

This mechanisation effort will involve encoding the syntax and semantics of both Incorrectness Logic and Dynamic Logic, defining the embedding transformation, and proving that the mapping preserves semantic equivalence. Additionally, we will analyze the degree to which proofs about incorrectness properties can be automated, exploring the potential for integrating IL-style bug reasoning directly into interactive theorem proving workflows. Such an investigation not only provides a concrete validation of the theoretical framework but also lays a foundation for future tool development that integrates automated incorrectness analysis with correctness verification. Ultimately, this aim aspires to bridge the gap between theoretical logic research and practical verification tool support, thereby enriching the formal verification ecosystem and promoting the adoption of under-approximate reasoning in mainstream verification practice.

3 Outcomes and Significance

This project is expected to deliver several key outcomes that advance both the theoretical and practical state of the art in program logic and formal verification. First, by providing a rigorous embedding of Incorrectness Logic into Dynamic Logic, the project will offer a unified theoretical foundation for reasoning about both correctness and incorrectness within a single, expressive modal framework. This unified perspective enables researchers and practitioners to understand the duality between over-approximate (universal) correctness proofs and under-approximate (existential) bug-finding proofs in a precise and principled way.

Second, the project will produce formal proofs of essential meta-theoretical properties, such as soundness and completeness of the embedding. These proofs, further reinforced through mechanisation in a proof assistant like Isabelle/HOL, will serve as an independent and highly reliable foundation that can be confidently reused or extended in future research and tool development. By making these formalizations publicly available, the project will also enrich the ecosystem of machine-checked logical frameworks and support the creation of new verification methodologies that seamlessly incorporate incorrectness reasoning.

Furthermore, by analyzing relationships with Hoare Logic, weakest precondition calculus, relational logics, and hyperproperties, the project will contextualize Incorrectness Logic within a broader landscape of formal methods. This analysis will help clarify its unique strengths and highlight potential opportunities for integration with other verification techniques, including compositional and relational reasoning about security and program equivalence.

Finally, the mechanisation efforts are expected to open avenues for practical adoption, paving the way toward integrating under-approximate reasoning into interactive theorem proving environments and static analysis tools. By providing a rigorous theoretical and mechanised basis for incorrectness reasoning, this project aspires to reduce the gap between theoretical logic research and practical bug-finding tools, ultimately contributing to the development of more reliable and robust software systems.

4 Timeline (First Year)

Table 1: Timeline for Year 1

Period	**Milestones**
Months 1–3	Conduct an in-depth study of the formal semantics of Incorrectness Logic and Dynamic Logic; begin exploring preliminary ideas for embedding IL triples into DL formulas; collect small example programs to guide early design.
Months 4–6	Develop and formally define the initial version of the embedding from IL to DL (Aim 1); analyze its semantic preservation using illustrative examples; refine definitions based on findings.
Months 7–9	Begin proof-theoretic analysis of the embedding (Aim 2); examine how IL proof rules translate to DL inference patterns; draft formal proofs of soundness for selected cases; continue iterative refinement with examples.
Months 10–12	Extend proof-theoretic connections (Aim 2); consolidate theoretical results from Year 1; prepare an internal technical report or a draft publication summarizing the formal embedding and preliminary proof-theoretic findings.

References

[1] C. A. R. Hoare, "An axiomatic basis for computer programming," *Communications of the ACM*, vol. 12, no. 10, pp. 576–580, 1969.

[2] P. W. O'Hearn, "Incorrectness logic," *Proceedings of the ACM on Programming Languages*, vol. 4, no. POPL, pp. 1–32, 2019.

[3] D. Harel, D. Kozen, and J. Tiuryn, "Dynamic logic," *ACM SIGACT News*, vol. 32, no. 1, pp. 66–69, 2001.

[4] E. W. Dijkstra, *A discipline of programming*. Prentice Hall PTR, 1976, vol. 613924118.