Comments

A Note on Inconsistent Axioms in Rushby's "Systematic Formal Verification for Fault-Tolerant Time-Triggered Algorithms"

Lee Pike

Abstract—We describe some inconsistencies in John Rushby's axiomatization of time-triggered algorithms that he presented in these transactions and that he formally specifies and verifies in the mechanical theorem-prover PVS. We present corrections for these inconsistencies that have been checked for consistency in PVS.

Index Terms—Formal methods, formal verification, time-triggered algorithms, synchronous systems, PVS.

1 Introduction

This comment's purpose is to make a few minor corrections to John Rushby's paper, "Systematic Formal Verification for Fault-Tolerant Time-Triggered Algorithms," which appeared in volume 25, number 5 of these transactions [1].[1] Rushby presents four principle assumptions (or axioms) about the behavior of time-triggered systems. He describes his use of these axioms in the systematic formal specification and verification of time-triggered systems in the mechanical theorem-prover PVS [2]. Two of these four axioms are inconsistent; in fact, one is inconsistent in three separate ways. Once the axioms are made consistent, one axiom is redundant; it is a corollary of the other. Finally, a contradiction can be derived from another of the four axioms and some other minor axioms in the formal specification. These inconsistencies appear in both the printed paper and the PVS specifications, but, when the printed axioms are ambiguous due to being more informally stated, we defer to the PVS specifications.

We discovered these errors while attempting to interpret these axioms by formally providing a model using theory interpretations in PVS [3]. When a "canonical model" did not satisfy the axioms, we quickly realized these axioms not only fail to model the domain, but are, in fact, inconsistent. Once the errors were discovered, it was fairly straightforward to mend them.[2]

This comment does not suggest a failure of formal verification. Rushby is widely considered to be an expert (if not *the* expert) in the mechanized verification of fault-tolerant real-time systems, particularly in PVS. These errors escaped his attention, despite formally verifying the theory. They also apparently escaped the attention of the reviewers of these transactions, the reviewers of an IEEE workshop,[3] and the numerous researchers who have cited this work, including this author.[4] Because these relatively elementary errors went unnoticed by both Rushby and his peers, this is further evidence that formal verification is crucial to ensure the correctness of a specification. However, a mechanically checked specification and verification is only as sound as one's axioms. The lesson here is the axiomatization of real-time systems is extremely difficult and, to ensure that an axiomatization is consistent and correctly models the intended domain, a formal verification should include a demonstration that some canonical implementation satisfies one's formal specifications.

2 Inconsistencies and Corrections

We begin by stating Rushby's definition of inverse clocks and Clock Drift Rate Axiom.

Definintion 1 (Inverse Clock). *An inverse clock for process p is a total function $C_p : \mathbb{R} \to \mathbb{N}$.*

The domain of an inverse clock is called *realtime* and the range is called *clocktime*. The drift of nonfaulty clocks is bounded by a realtime constant $0 < \rho < 1$.

Axiom 1 (Clock Drift Rate).

$$(1-\rho)(t_1 - t_2) \le C_p(t_1) - C_p(t_2) \le (1+\rho)(t_1 - t_2).$$

Theorem 1. *Axiom 1 is inconsistent.*

Proof. Let $t_2 > t_1$. Then, $(1-\rho)(t_1 - t_2) > (1+\rho)(t_1 - t_2)$. □

Axiom 1 can be revised as follows:

Axiom 2 (Clock Drift Rate (First Revision)). *Let $t_1 \ge t_2$. Then,* $(1-\rho)(t_1 - t_2) \le C_p(t_1) - C_p(t_2) \le (1+\rho)(t_1 - t_2)$.

However, even this is unsatisfiable:

Theorem 2. *Axiom 2 is inconsistent.*

Proof. Let $t_1 > t_2$ such that $(1+\rho)(t_1 - t_2) - (1-\rho)(t_1 - t_2) < 1$ and there exists no $n \in \mathbb{N}$ such that $(1-\rho)(t_1 - t_2) \le n \le (1+\rho)(t_1 - t_2)$. □

We weaken the inequality by taking the floor and ceiling of the drifts:

Axiom 3 (Clock Drift Rate (Second Revision)). *Let $t_1 \ge t_2$. Then,* $\lfloor (1-\rho)(t_1 - t_2) \rfloor \le C_p(t_1) - C_p(t_2) \le \lceil (1+\rho)(t_1 - t_2) \rceil$.

Even with these revisions, no function satisfying Axiom 3 is an inverse clock, as defined by Definition 1.[5]

Theorem 3. *No inverse clock satisfies Axiom 3.*

Proof. By contradiction. The set $\mathbb{N}$ is totally ordered with a least element, so there exists some $t \in \mathbb{R}$ such that $C_p(t) \le C_p(t')$ for all $t' \in \mathbb{R}$. Let $t'' \in \mathbb{R}$, where $t'' < t$, such that $\lfloor (1-\rho)(t - t'') \rfloor > 0$.

1. The mended formal specifications, along with a formal theory interpretation, can be found at http://www.cs.indiana.edu/~lepike/pub_pages/time_triggered.html.

2. Paul Miner of the NASA Langley Formal Methods Group suggested that the revisions made in Axioms 2 and 3 are necessary to axiomatize a canonical clock. He also pointed out that these changes imply Theorem 5 holds.

3. Rushby's paper has not only appeared in these transactions since 1999, but an earlier version appeared in the *IEEE Proceedings of the Sixth Working Conference on Dependable Computing for Critical Applications* [4].

4. At the time of writing, Citeseer (http://citeseer.ist.psu.edu/) finds 16 citations and Google Scholar (http://scholar.google.com/) finds 50; the number of peer-reviewed citations likely falls between these two data points.

5. It should already be intuitive that Definition 1 is incorrect since, e.g., a canonical inverse clock function like the floor function does not satisfy Axiom 3.

• *The author is with Galois Connections, 12725 SW Millikan Way, Suite 290, Beaverton, OR 97005. E-mail: leepike@galois.com.*

Manuscript received 25 Apr. 2005; accepted 22 Mar. 2006; published online 24 May 2006.
Recommended for acceptance by T. Bell.
For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number TSE-0098-0405.

By Axiom 3, $\lfloor(1-\rho)(t-t'')\rfloor + C_p(t'') \leq C_p(t)$. However, because $\lfloor(1-\rho)(t-t'')\rfloor$ is assumed to be strictly greater than zero, $C_p(t'') < C_p(t)$, contradicting our assumption that $C_p(t)$ is least. □

We therefore extend the range of an inverse clock from $\mathbb{N}$ to $\mathbb{Z}$.

Defiinition 2 (Revised Inverse Clock). *An inverse clock for process p is a total function $C_p : \mathbb{R} \rightarrow \mathbb{Z}$.*

Note that the inconsistencies in Axioms 1 and 2 hold regardless of whether an inverse clock is defined by Definition 1 or Definition 2.

A second inconsistent axiom is the Monotonicity Axiom. Nonfaulty clocks are monotonic:

Axiom 4 (Monotonicity). $t_1 < t_2$ *implies* $C_p(t_1) < C_p(t_2)$.

Theorem 4. *Axiom 4 is inconsistent (with respect to either Definition 1 or Definition 2).*

Proof. Because $<$ is a total order over $\mathbb{R}$, Axiom 4 implies that C_p is an injective function, but there exists no injection from the reals into the integers (or natural numbers). □

A satisfiable revision of monotonicity weakens the consequent slightly:

Axiom 5 (Revised Monotonicity). $t_1 < t_2$ *implies* $C_p(t_1) \leq C_p(t_2)$.

Axiom 5 now becomes a corollary of Axiom 3:

Theorem 5. *Let Axiom 3 hold. Prove Axiom 5.*

Proof. By Axiom 3, $C_p(t_2) \geq C_p(t_1) + \lfloor(1-\rho)(t_2-t_1)\rfloor$. □

The third inconsistency can be derived from the axiomatization of when messages are sent and received by nonfaulty processes. Let $sent_p(q, m, t)$ be a relation that holds if process p sends message m to process q at realtime t. Similarly, let $recv_q(p, m, t)$ be a relation that holds if process q receives message m from process p at realtime t. The following axiom relates the delay between when a nonfaulty process sends a message and when a nonfaulty process receives it. Let the maximum delay be a realtime constant such that $\delta \geq 0$.

Axiom 6 (Maximum Delay). $sent_p(q, m, t)$ *if and only if there exists some realtime delay* $0 \leq d \leq \delta$ *such that* $recv_q(p, m, t+d)$.

Theorem 6. *If $\delta > 0$, then Axiom 6, together with other minor axioms and constraints in the formal specification, is inconsistent.*

Proof. (Sketch.) The essential problem is that the existential quantifier is within the scope of the biconditional operator in Axiom 6. As stated, Axiom 6 implies that, for all realtimes t, if there exists a $0 \leq d \leq \delta$ such that $recv_q(p, m, t+d)$, then $sent_p(q, m, t)$. It can be shown that there exists some t such that $recv_q(p, m, t+d)$. Because d ranges over the interval $[0, \delta]$, there exists a realtime t' and realtime delay $0 \leq d' \leq \delta$ such that $d' \neq d$ and $t' + d' = t + d$, implying that $sent_p(q, m, t)$ and $sent_p(q, m, t')$, where the distance between t and t' is less than δ. However, by other constraints, no two separate realtimes within δ of each other satisfy $sent$. □

A possible consistent revision is as follows:

Axiom 7 (Revised Maximum Delay). *There exists some* $0 \leq d \leq \delta$ *such that* $sent_p(q, m, t)$ *implies* $recv_q(p, m, t+d)$ *and there exists some* $0 \leq d' \leq \delta$ *such that* $recv_q(p, m, t)$ *implies* $sent_p(q, m, t-d')$.

Acknowledgments

This work was completed while the author was with the NASA Langley Research Center Formal Methods Group.

References

[1] J. Rushby, "Systematic Formal Verification for Fault-Tolerant Time-Triggered Algorithms," *IEEE Trans. Software Eng.*, vol. 25, no. 5, pp. 651-660, Sept. 1999.

[2] S. Owre, J. Rusby, N. Shankar, and F. von Henke, "Formal Verification for Fault-Tolerant Architectures: Prolegomena to the Design of PVS," *IEEE Trans. Software Eng.*, vol. 21, no. 2, pp. 107-125, Feb. 1995.

[3] S. Owre and N. Shankar, "Theory Interpretations in PVS," Technical Report SRI-CSL-01-01, SRI Int'l, Apr. 2001, http://pvs.csl.sri.com/documentation.shtml.

[4] J. Rushby, "Systematic Formal Verification for Fault-Tolerant Time-Triggered Algorithms," *Dependable Computing for Critical Applications—6*, vol. 11, pp. 203-222, Mar. 1997.