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

Lee Pike

The author is with the Formal Methods Group, NASA Langley Research Center.

Index Terms

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

Abstract

I describe some inconsistencies in John Rushby's axiomatization of time-triggered algorithms that he presents in these transactions and that he formally specifies and verifies in a mechanical theorem-prover. I also present corrections for these inconsistencies.

I. Introduction

This note's purpose is to make a few minor corrections to John Rushby's paper, *Systematic Formal Verification for Fault-Tolerant Time-Triggered Algorithms*, appearing in Vol. 25, No. 5 of these transactions [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, I defer to the PVS specifications.

I discovered these errors while attempting to interpret these axioms by formally providing a model using theory interpretations in PVS [3]. When the "canonical model" did not satisfy the axioms,[1] I 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] Rushby's formal proofs do not depend on the inconsistencies. However, these specifications are intended to be systematic and reusable; in the hands of someone without Rushby's expertise, this danger very much exists.

[1]I would like to thank Paul Miner of the NASA Langley Formal Methods Group for suggesting Axioms 2 and 3 are necessary to axiomatize a canonical clock. He also pointed out that these changes imply that Theorem 5 holds.

[2]The mended formal specifications, along with a formal theory interpretation, can be found at <`http://here.com`>.

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 and the numerous researchers who have cited this work, including this author.[3] 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 the axioms are consistent and correctly model the domain, a formal verification should include a demonstration that some (canonical) implementation satisfies one's formal specifications.

II. Inconsistencies and Corrections

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

Definition 1 (Inverse Clock): An inverse clock for process p is a total function $C_p : \mathbb{R} \rightarrow \mathbb{N}$. 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): $(1 - \rho)(t_1 - t_2) \leq C_p(t_1) - C_p(t_2) \leq (1 + \rho)(t_1 - t_2)$.

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 $t_1 \geq t_2$. Then $(1 - \rho)(t_1 - t_2) \leq C_p(t_1) - C_p(t_2) \leq (1 + \rho)(t_1 - t_2)$.

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 $(1 - \rho)(t_1 - t_2) \leq n \leq (1 + \rho)(t_1 - t_2)$. ■

[3]Rushby's paper has not only appeared in these transactions since 1999, but an an earlier version appeared in the IEEE Proceedings of the Sixth Working Conference on Dependable Computing for Critical Applications [4]. The paper has been well-cited, even in the very recent literature. For example, A quick search on Google Scholar finds 36 citations; the author knows of at least three citations appearing in work published in 2004.

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

Axiom 3 (Clock Drift Rate (Second Revision)): Let $t_1 \geq t_2$. Then $\lfloor (1-\rho)(t_1-t_2) \rfloor \leq C_p(t_1) - C_p(t_2) \leq \lceil (1+\rho)(t_1-t_2) \rceil$.

Even with these revisions, no function satisfying Axiom 3 is an inverse clock, as defined by Definition 1.[4]

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 $C_p(t) \leq C_p(t')$ for all $t' \in \mathbb{R}$. Let $t'' \in \mathbb{R}$, where $t'' < t$, such that $\lfloor (1-\rho)(t-t'') \rfloor > 0$. 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. ■

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

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

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 natural numbers (or integers). ■

A satisfiable revision of monotonicity weakens the antecedent 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

[4]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.

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

REFERENCES

[1] J. Rushby, "Systematic formal verification for fault-tolerant time-triggered algorithms," *IEEE Transactions on Software Engineering*, vol. 25, no. 5, pp. 651–660, September 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 Transactions on Software Engineering*, vol. 21, no. 2, pp. 107–125, February 1995.

[3] S. Owre and N. Shankar, "Theory interpretations in PVS," SRI, International, Tech. Rep. SRI-CSL-01-01, April 2001, available at http://pvs.csl.sri.com/documentation.shtml.

[4] J. Rushby, "Systematic formal verification for fault-tolerant time-triggered algorithms," in *Dependable Computing for Critical Applications—6*, vol. 11. IEEE Computer Society, March 1997, pp. 203–222.