Artifact: Owicki-Gries Reasoning for C11 RAR

Sadegh Dalvandi, Simon Doherty, Brijesh Dongol, and Heike Wehrheim

Abstract

The paper *Owicki-Gries Reasoning for C11 RAR* introduces a new proof calculus for the C11 RAR memory model that allows Owicki-Gries proof rules for compound statements, including non-interference, to remain unchanged. The proof method features novel assertions specifying thread-specific views on the state of programs. This is combined with a set of Hoare logic rules that describe how these assertions are affected by atomic program steps.

The artifact includes the Isabelle formalisation of the proof method introduced in the paper. It also contains the formalisation and proof of all case studies presented in the paper. All of the theorems are accompanied with their respective proofs.

1 Introduction

The artifact (link in Section **??**) is a set of Isabelle[1] theories formalising the proof method introduced in the paper "*Owicki-Gries Reasoning for C11 RAR*". This document outlines the structure of the Isabelle theories and their dependencies. It also provides information on how to map the definitions and theorems in the paper to Isabelle. A brief description on how to use the artifact and check the Isabelle theories and replay the proof is also provided.

2 Artifact Structure

As mentioned previously, the artifact is a set of Isabelle theories. A list of these theories is as follows:

- **OpSem.thy** includes the formalisation of the proof method together with all the rules and their proof.
- **LB.thy** includes an encoding of the load buffering litmus test, its proof outline and the associated proof (Figure 8, Section 5.3 of the paper).
- **MP.thy** includes an encoding of the message passing litmus test, its proof outline and the associated proof (Figure 3, Section 5.4 of the paper).
- **RRC_2T.thy** includes an encoding of a two threaded version of the read-read coherence litmus test, its proof outline and the associated proof (Figure 9, Section 5.5 of the paper).
- **RRC_3T.thy** includes an encoding of a three threaded version of the read-read coherence litmus test, its proof outline and the associated proof (not included int the paper).
- **RRC.thy** includes an encoding of a four threaded version of the read-read coherence litmus test, its proof outline and the associated proof (Figure 10, Section 5.5 of the paper).

[1] `https://isabelle.in.tum.de/`

- **Petersons.thy** includes an encoding of the Peterson's mutual exclusion algorithm case study presented in Section 6 of the paper together with its proof outline and the associated proof.

Table **??** provides a mapping between definitions and lemmas in the paper and Isabelle theory *OpSem.thy*.

Table 1: A mapping from paper to the OpSem.thy Isablle theory

Description	**Paper**	**Isabelle Theory**
C11 State	Table 1	surrey_state
Observable writes	Definition 1	visible_writes
Read transition	Figure 5	read_trans
Write transition	Figure 5	write_trans
Update transition	Figure 5	update_trans
Well formedness	Section 4.4	wfs
Well formedness proof	Lemma 1	wfs_preserved
Definite observation	Section 5.3	d_obs_abbr d_obs d_obs_t
Proof rules in Lemma 4	Lemma 4	d_obs_Rd_pres d_obs_other init_d_obs
Possible Observation	Section 5.4	p_obs p_obs_abbr
Conditional Observation	Section 5.4	c_obs c_obs_abbr
Relationship between assertions	Lemma 6	d_obs_implies_p_obs d_obs_p_obs_agree not_p_obs_implies_c_obs d_obs_same_val
Proof rules in Lemma 7	Lemma 7	d_obs_Wr_set c_obs_Wr_intro c_obs_Rd_d_obs not_p_obs_Wr_pres not_pobs_Rd_pres
Possible value order	Section 5.5	p_vorder
Definite value order	Section 5.5	d_vorder
Initial value	Section 5.5	init_val
Encountered value	Section 5.5	enc_t enc
Value occurence	Section 5.5	amo no_val
Properties in Lemma 9	Lemma 9	pvord_to_dvord d_vorder_one_way

Continued on next page

Table 1 – continued from previous page

Description	Paper	Isabelle Theory
Proof rules in Lemma 10	Lemma 10	no_val_write_diff_value_pres d_vorder_Read_pres amo_intro enc_write_intro enc_read_intro d_vorder_intro amo_read_pres p_vorder_write_pres read_pres_p_vorder
Covered writes	Section 6	covered_v cvd_SWAP_new_cvd
Proof rules in Lemma 13	Lemma 13	covered_read_v_pres covered_diff_var_pres cvd_SWAP_d_obs

The OpSem.thy theory is dependent on the Isabelle *Main* library and also *HOL.Rat*, both part of the standard distribution of Isabelle. Theories related to the litmus tests are only dependent on *OpSem.thy*.

Theories that include definition and proof of litmus tests and case studies have the following structure:

1. Each algorithm is encoded semantically using an Isabelle definition called *prog*.
2. The proof outline for each algorithm is encoded using an Isabelle definition called *prog_inv*.
3. Each theory includes proofs of local correctness as well as the proof of interference freedom for each thread.

3 Instructions

Since the artifact is the formalisation of the proof method in Isabelle/HOL, no installation or compilation for the artifact is required. No external library or prover other than those provided by the standard distribution of Isabelle/HOL is used. The theories have been checked against both Isabelle2019/HOL and (the recently released) Isabelle2020/HOL, and they go through without any problem.

Isabelle/HOL can be downloaded from `https://isabelle.in.tum.de/`. Installation of Isabelle is very straightforward. More information on installation can be found in `https://isabelle.in.tum.de/installation.html`. Once Isabelle/HOL is installed, open any of the theories from $File > Open$.

When a theory is loaded, Isabelle invokes various provers in the background and replays the proofs in the theory file. You may need to scroll to the end of a theory to ensure Isabelle checks the whole proof. It may take couple of minutes for Isabelle to prove a theory. In our artifact, the proof of OpSem.thy takes around 5 minutes to go through.