A **reconstruction mechanisms** for proofs by **automatic provers** to aid interactive **proof authoring**.

Reconstructing veriT Proofs in Isabelle/HOL

Mathias Fleury, Hans-Jörg Schurr

Automated Provers to Assist Interactive Proof Construction

When constructing proofs in an interactive proof assistant, such as Isabelle, users now commonly rely on automated theorem provers to discharge proof obligations.
To maintain the trustworthiness of a proof the automatically found proof must be verified inside the proof assistant. We developed a reconstruction procedure in the proof assistant Isabelle for proofs generated by the satisfiability modulo theories solver veriT which is part of the *smt* tactic.

The Proofs Generated by veriT

veriT is a CDCL(T)-based satisfiability modulo theories solver and supports the theory of uninterpreted functions, linear integer and real arithmetic, and quantifiers.
It uses the SMT-LIB language as input and output language and also utilizes the many-sorted classical first-order logic defined by this language. veriT outputs a proof if it can deduce that the input problem is unsatisfiable.
Figure 1 shows a proof as generated by veriT. Such a proof is formed by a list of steps. A step consists of an index, a formula, a rule name, a possibly empty set of premises, a rule-dependent list of arguments, and a context.

Proof Reconstruction in Isabelle/HOL

Proof reconstruction is implemented as a pipeline (Figure 2):

1. Parse the proof text into an abstract syntax tree.
2. Unfold the names introduced by term sharing.
3. Transform the abstract syntax tree into Isabelle terms.
4. Reconstruct the proof Step-by-step: Encode each rule as an Isabelle lemma, then unify the assumptions of the proof step with the premises of the lemma.

The parsing code is shared with the proof reconstruction code for proofs generated by the SMT solver Z3.

Experimental Results

- Table 1: We replaced all the *smt* calls that are in the Isabelle distribution and are currently powered by Z3, by the version of *smt* with veriT.
- Table 2: We try to generate new veriT-powered *smt* calls by using Sledgehammer, an Isabelle tool able to find proofs. Sledgehammer suggests the fastest method which is able to to find a proof.

```
(assume h1 (not (p a)))
(assume h2 (forall ((z1 U)) (forall ((z2 U)) (p z2))))
...
(anchor :step t9 :args ((:= z2 vr4)))
(step t9.t1 (cl (= z2 vr4)) :rule refl)
(step t9.t2 (cl (= (p z2) (p vr4))) :rule cong :premises (t9.t1))
(step t9 (cl (= (forall ((z2 U)) (p z2)) (forall ((vr4 U)) (p vr4))))
                                        :rule bind)
...
(step t14 (cl (forall ((vr5 U)) (p vr5)))
                                        :rule th_resolution :premises (t11 t12 t13))
(step t15 (cl (or (not (forall ((vr5 U)) (p vr5))) (p a)))
                                        :rule forall_inst :args ((:= vr5 a)))
(step t16 (cl (not (forall ((vr5 U)) (p vr5))) (p a)) :rule or :premises (t15))
(step t17 (cl) :rule resolution :premises (t16 h1 t14))
```

Figure 1 An Proof Generated by veriT

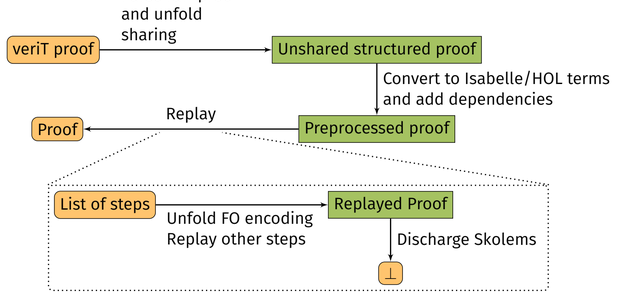

Figure 2 The Reconstruction Pipeline

SMT calls	Number
Successful reconstruction	447
Failed reconstruction	4
veriT timeouts	47
veriT unknown	4

Table 1 Result of using veriT instead of Z3 in existing *smt* calls

Theory	O.R. Prover	SSA
Found proofs	5019	5961
Z3-powered	90	109
veriT-powered	25	4
Oracle	9	63

Table 2 Proofs found by Sledgehammer on two Isabelle formalizations

Fleury, M. & Schurr, H. J. (2019). Reconstructing veriT Proofs in Isabelle/HOL. In G. Reis & H. Barbosa (Eds.) *Proceedings Sixth Workshop on Proof eXchange for Theorem Proving, Natal, Brazil, August 26, 2019*. Open Publishing Association. doi:10.4204/EPTCS.301.6