IsaRARE: Automatic translation of term rewrite rules into Isabelle/HOL lemmas

Hanna Lachnitt

November 21, 2023

Contents

1 Introduction

IsaRARE is a plugin for Isabelle that transforms rewrite rules in the RARE language into Isabelle lemmas. It serves two main purposes:

1. Verification: Proving a lemma generated by IsaRARE indicates that the corresponding rule is sound.
2. Reconstruction: If rule is used in a proof certificate by an external solver, the generated lemmas can be used by the smt method during the reconstruction of that proof inside of Isabelle.

2 Set-up and Quick Usage

IsaRARE itself does not require any prerequisites but to execute the bitvector examples a copy of the Archive of Formal Proofs (AFP) is needed. The tool can be used simply by importing IsaRARE.thy:

```
theory IsaRARE
  imports HOL−CVC.Smtlib-String HOL−CVC.SMT-CVC
  keywords parse-rare-file parse-rare :: diag
begin
```

The two keywords the theory provides are used as follows

parse-rare <input rare rule as string>

and

parse-rare-file <input rare file, theories names to be imported, target theory name>

Examples:

parse-rare "(define-rule bool-eq-true ((t Bool)) (= t true) t)"

and

parse-rare-file "/IsaRARE/Tests/example_rewrites" "Parent_Theory" "Example_Rewrites"

```
datatype smt-datatype = String string | Int int | Real real
```

3 The RARE language

4 Components

```
ML-file ‹src/isarare-config.ML›
ML-file ‹src/parse-rare.ML›
ML-file ‹src/rare-impl-assump.ML›
ML-file ‹src/rare-lists.ML›
ML-file ‹src/write-rewrite-as-lemma.ML›
```

```
ML ‹
 open Parse-RARE
 open Write-Rewrite-as-Lemma

 fun print-item string-of (modes, arg) = Toplevel.keep (fn state =>
   Print-Mode.with-modes modes (fn () => writeln (string-of state (hd arg))) ())
```

```
(*TODO: Can I use: Library.cat-lines?*)
 fun string-of-rewrite ctxt s
 = (Write-Rewrite-as-Lemma.write-thy (Parse-RARE.parse-rewrites ctxt [s]) THEORY-NAME
IMPORTING-THEORIES ctxt)

 fun print-rewrite (cs:string) (t:Toplevel.transition) : Toplevel.transition =
  Toplevel.keep (fn toplevel => (fn state =>
    Print-Mode.with-modes [] (fn () => writeln (string-of-rewrite state cs)) ())
(Toplevel.context-of toplevel)) t

 val - =
  Outer-Syntax.command command-keyword‹parse-rare› parse a single rule in
rare format (provided as a string) and output lemma
    ( Parse.string >> print-rewrite);

val ISARARE-HOME = OS.FileSys.getDir()

 val semi = Scan.option keyword‹;›; (*TODO: Do not need?*)
val x = OS.Process.getEnv

 val - = Outer-Syntax.local-theory command-keyword‹parse-rare-file› parse file
in rare format and output lemmas. <rare-file, import theories, target-theory>
    (((Parse.string -- Parse.string) -- Parse.string)
    >> (fn ((file-name,theory-imports),theory-name) => fn lthy =>
  let
         (*Built new path*)
         val file-path = Path.explode file-name
         val new-theory-name = theory-name ^ .thy
         val ctxt = Local-Theory.target-of lthy
       val res-path = Path.append (Path.dir file-path) (Path.basic new-theory-name)

         (*Calculate result*)
         (*val lines = raw-explode ( hd (Bytes.contents (Bytes.read file-path))) ;*)
         val lines = Bytes.split-lines (Bytes.read file-path)
        val res = (Write-Rewrite-as-Lemma.write-thy (Parse-RARE.parse-rewrites
ctxt lines) theory-name theory-imports ctxt)
         val - = (Output.writeln res)

         val - =
          Bytes.write
           res-path (Bytes.string res)
         val - = @{print} (done writing to file, res-path)
 in lthy
 end))
›
```

lemmas *cvc-arith-rewrite-defs* = *SMT.z3div-def*

5 Options

```
declare[[IsaRARE-verbose = true]]
declare[[IsaRARE-debug = true]]
declare[[IsaRARE-implAssump = true]]
declare[[IsaRARE-listsAsVar = false]]
declare[[IsaRARE-proofStrategy = Full]]
declare [[ML-print-depth=10000]]
```

6 Test

7 Expansions (Experts)

end