What Needs to be Done

Dominic Mulligan

September 28, 2007

Introduction

I'll write about what everything is (although most of the code is well commented), what problems/bugs are left to fix, and what else needs doing.

Design Description

Most of the major contextual information functions now require a few extra things:

- An instantiation environment (`instenv`).
- A goal mode information pair (`gmi`).
- The relatedness table (`rtab`).

The `gmi` is a pair - a *type* table and a *position* table. I originally constructed all of these elements in different places, now they're all constructed in the same function (`RRippleTechn.start`, if I remember correctly), so they can all be wrapped up into a single record, like you suggested a while ago. `gmis` (note plural) are a list of `string` * `gmi` pairs, where the `string` is the "combinator name" (i.e. the relation name, *add* in *add a b ab*, for instance), to which these tables are associated.

`ROOT.ML` is "protected" with a flag variable (type `val flag = ref false;`) to stop loading the theory file twice. If you get a `Match` exception when running `ROOT.ML`, it's origin is most likely in `RRippleCInfo.I.put`. Killing the process and rerunning `ROOT.ML` will fix this. I presume this is a bug in *IsaPlanner*?

The flag passed to `RRippleCInfo.start` signifies whether we are working forwards or backwards (for obtaining the skeletons and goals).

I've put a new structure in `src\gproof\prf\IndPrf.ML` for small utility functions dealing with inductive proofs (like checking whether a goal is a stepcase, checking whether an assumption is an inductive hypothesis etc.). You may want to move this elsewhere (I was unsure as to whether it was suitable for `Prf.ML` or not).

I've got two proof scripts, `rr_proof_devel.ML` and `rr_proof_new.ML`. The latter is a reference copy, the former is my "testing ground" for the technique.

Problems

- When reasoning forward, goalnames don't get recorded correctly. I tried fixing this by keeping a list of open goalnames, applying the steps by hand (i.e. by applying a reasoning technique to a sequence of reasoning states). However, this has the disadvantage of "collapsing" all the steps into a single one. I tried fixing this, but couldn't (you'll notice the code I used to do this is still commented out at the bottom of `rr_techn.ML`. The code that's currently in use has the open goalnames coded in, so that I could get on with the rest of the technique and see the individual steps as they are applied. I think this should be an easy fix.
- I couldn't work out how to obtain the name of the induction variable (is this not recorded somewhere?).
- Because the relational rippling code isn't properly integrated into *IsaPlanner* yet, the relational wave-rule database doesn't get passed any wave-rules by *IsaPlanner*. I'm currently adding them manually (see top of `rr_proof_devel.ML`.
- Again, when reasoning forwards (i.e. by applying all possible `DTacs` to a fact), the newly introduced factname is thrown away (see, for example, *confluence_single_ims* in my development proof). I cannot work out how to obtain these introduced names (nor can Moa).
- The contextual information isn't updated after rewriting (I was going to do this next). Furthermore, decisions that I defered, about how the contextual information will be stored correctly, haven't been looked into.

Todo

- Fix the problem with the goalnames working forwards.
- Update cinfo after rewriting.
- Split any new goals/facts, embed again, rewrite.
- Check for measure decrease properly (was going to address this after I got rewriting working).
- Apply IH when measure is (0, 0).
- Ripple right then fertilize.

Contact

We can arrange a meeting after you return. Jamie knows that I owe two days work (he's away for two weeks, anyway). If you have any questions, it is probably best to e-mail me at: `dpm_edinburgh@yahoo.com` as my university account is now due for deletion (I think?). I'll try to get *IsaPlanner* working on my laptop so I can finish some of the tasks.

Dom.