Proof Reconstruction with Athena (Work in Progress)

Jonathan Prieto-Cubides

Advisor: Andrés Sicard-Ramírez

EAFIT University
Medellín, Colombia

Agda Implementors' Meeting XXV

Keywords

Proof Assistant

is a software program that work together with a human in order to give formal proofs to problems in a wide range of topics.
- Agda, Isabelle/HOL, HOL Light, Coq.

Automatic Theorem Prover

Is software program that tries to prove conjecture based on a list of hypothesis.
- Metis, Z3, Vampire, EProver, iProver, SPSS.

Athena

Is a Haskell program that translates proofs given by Metis ATP in TSTP format to Agda code.

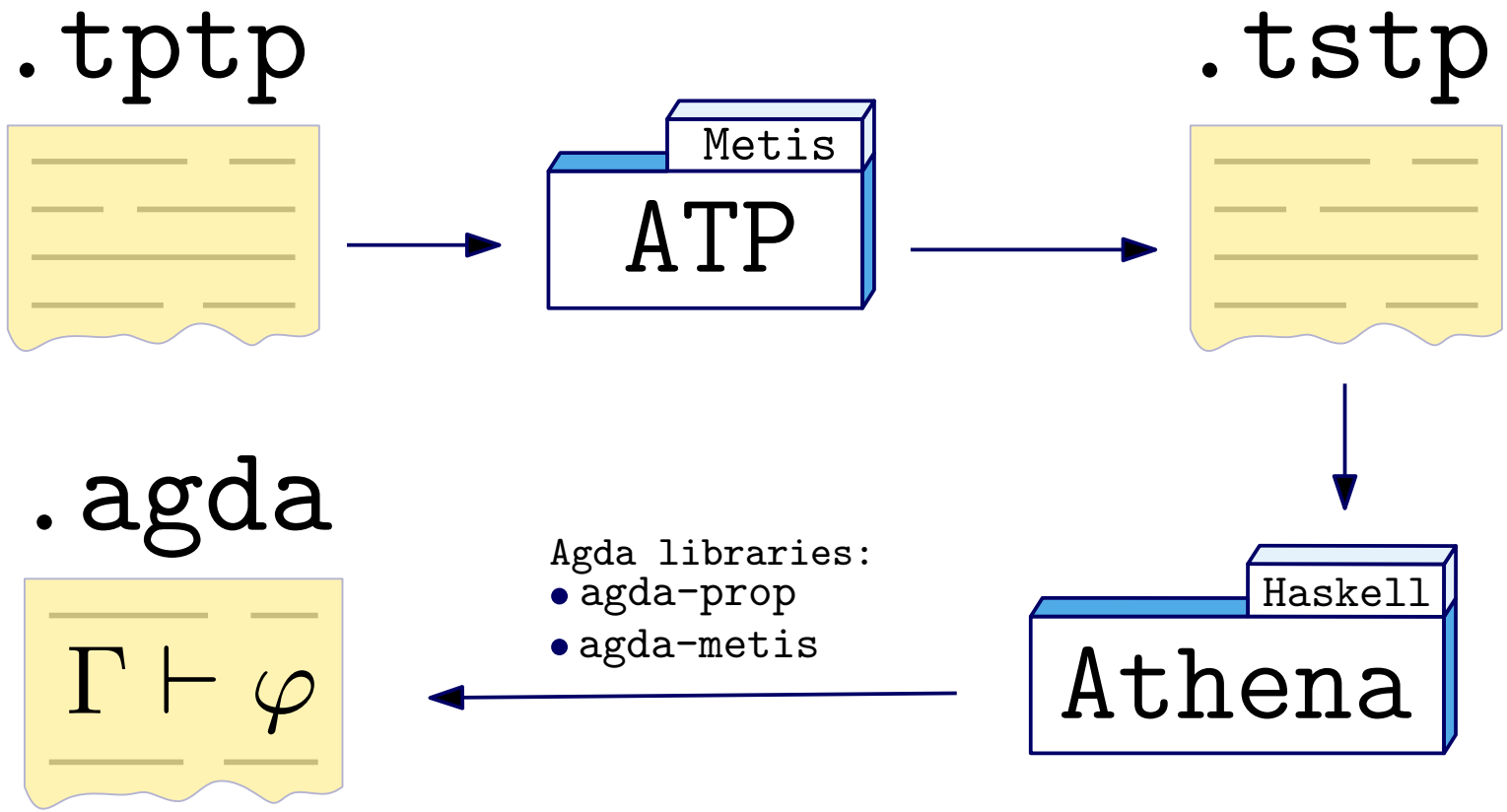

Figure: Proof reconstruction work flow.

TPTP

Is a text format to encode problems in different theories. Its goal is provide a standard for all ATPs in the input file format. The version for derivations is named TSTP but this one is not a standard yet.

A TPTP file looks like:

Listing 1: ../test/prop-pack/problems/basic/basic-4.tptp

```
fof(a1, axiom, p).
fof(goal, conjecture, p).
```

TSTP

A derivation of the previous problem output by Metis ATP.

Listing 2: ../test/prop-pack/problems/basic/basic-4.tstp

```
fof(a1, axiom, (p)).
fof(goal, conjecture, (p)).
fof(subgoal_0, plain, (p),
  inference(strip, [], [goal])).
fof(negate_0_0, plain, (~ p),
  inference(negate, [], [subgoal_0])).
fof(normalize_0_0, plain, (~ p),
  inference(canonicalize, [], [negate_0_0])).
fof(normalize_0_1, plain, (p),
  inference(canonicalize, [], [a1])).
fof(normalize_0_2, plain, ($false),
    inference(simplify, [],
      [normalize_0_0, normalize_0_1])).
cnf(refute_0_0, plain, ($false),
    inference(canonicalize, [],
      [normalize_0_2])).
```

Agda

A verified proof of the previous derivation output by Athena.

```
proof₀ : Γ ⊢ subgoal₀
proof₀ =
  (RAA
    (atp-canonicalize
      (atp-simplify
        (atp-canonicalize
          (atp-strip
            (assume {Γ = Γ}
              (atp-negate subgoal₀))))
        (atp-canonicalize
          (weaken (atp-negate subgoal₀)
            (assume {Γ = ∅} a₁))))))

proof : Γ ⊢ goal
proof =
  ⇒-elim
    atp-splitGoal
    proof₀
```

Figure: Type-checked Proof.

Natural deduction tree

$\Gamma, p \vdash p$

Proof.

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash a_1}{\Gamma, \neg\,\text{subgoal}_0 \vdash a_1}\,(\textbf{weaken}\ \neg\,\textbf{subgoal}_0)}{\Gamma, \neg\,\text{subgoal}_0 \vdash p}\,(\textbf{canonicalize}) \qquad \dfrac{\dfrac{\dfrac{\Gamma \vdash p}{\Gamma, \neg\,\text{subgoal}_0 \vdash \neg p}\,(\textbf{assume})}{\Gamma, \neg\,\text{subgoal}_0 \vdash \neg p}\,(\textbf{canonicali}}{\Gamma, \neg\,\text{subgoal}_0 \vdash \neg p}\,(\textbf{simplify})}{\dfrac{\bot}{\Gamma \vdash p}\,(\textbf{RAA})}$$

□

Future work

- Extend the libraries Agda-Prop and Agda-Metis to handle predicated logic
- Reconstruct proofs of eprover in Agda-EProver
- Support more versions of GHC in Athena

References