Proof Reconstruction in Classical Propositional Logic

Jonathan Prieto-Cubides

Advisor: Andrés Sicard-Ramírez

Universidad EAFIT
Medellín, Colombia

Agda Implementors' Meeting XXV
May 9-15th

Proof Term Reconstruction
At the moment, the communication between Agda and the ATPs is
unidirectional because we use the ATPs as oracles

.tptp

- ▶ Is a language[1] to encode problems in text files
- ▶ Is the input of the ATPs
- ▶ his problems contains formulas with the form

  language(name, role, formula).

language THF, TFF, FOF, or CNF
    name to identify the formula within the problem
    role axiom, definition, hypothesis, conjecture, among others
 formula the logic formula in the language

---

[1]Is available at
http://www.cs.miami.edu/~tptp/TPTP/SyntaxBNF.html

Problems in Propositional Logic:

▶ $p \vdash p$

```
$ cat basic-4.tptp
fof(a, axiom, p).
fof(goal, conjecture, p).
```

▶ $p \wedge q \vdash q \wedge p$

```
$ cat conj-3.tptp
fof(a, axiom, p & q).
fof(goal, conjecture, q & p).
```

▶ $\vdash \neg(p \wedge \neg p) \vee (q \wedge \neg q)$

```
$ cat neg-7.tptp
fof(goal, conjecture, ~ ((p & ~ p) | (q & ~ q))).
```

`.tstp`

A TSTP derivation[2]

▶ Is a Directed Acyclic Graph where

leaf is a formulae from the TPTP input
node is a formulae inferred from parent formulae
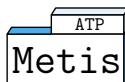**root** the final derived formulae

▶ Is a list of annotated formulae:

language(name,role,formula,source[,useful info]).

source typically is a inference record:

inference(rule, [], [])

- The inference rule name,
- a list of useful inference information, and
- a list of references to its parent formulae.

---

[2]**http://www.cs.miami.edu/~tptp/TPTP/QuickGuide/ Derivations.html**

▶ Proof found by **Metis** ATP for the problem $p \vdash p$

ATP

Metis

```
$ metis --show proof basic-4.tptp
fof(a, 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, [], [a])).
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])).
```