Reconstructing Propositional Proofs in Type Theory

Jonathan Prieto-Cubides
Advisor: Andrés Sicard-Ramírez

Master in Applied Mathematics
Universidad EAFIT
Medellín, Colombia

November 15, 2017

Research

Goal

Formalization in type theory of the classical propositional derivations generated by the `Metis` theorem prover

Research

Goal

Formalization in type theory of the classical propositional derivations generated by the `Metis` theorem prover

Topics

- Automatic reasoning using automatic theorem provers (ATPs) (e. g., `Metis`, `EProver`)
- Interactive proving using Proof-assistants (e. g., `Agda`, `Coq`)
- Formal methods to verify outputs of ATPs in Proof-assistants

Related Work

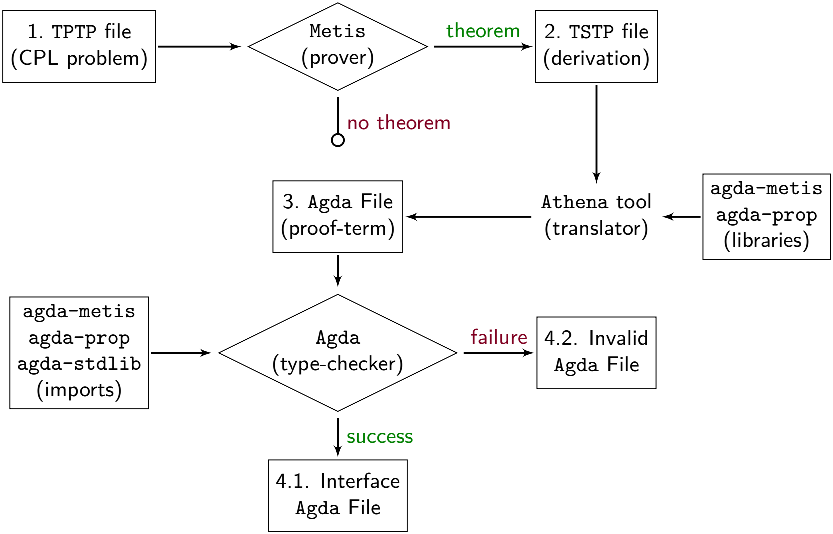
In type theory:

- Kanso in [5] reconstructs in `Agda` propositional proofs generated by `EProver` and `Z3`
- Foster and Struth in [2] describe proof-reconstruction in `Agda` for equational logic of `Waldmeister` prover
- Bezem, Hendriks, and Nivelle in [1] transform a proof produced by the first-order prover `Bliksem` in a `Coq` proof-term

In classical logic:

- Paulson and Susanto in [6] introduce `SledgeHammer`, a tool ables to reconstructs proofs of well-known ATPs: `EProver`, `Vampire`, among others using `SystemOnTPTP` server
- Hurd in [3] integrates the first-order resolution prover `Gandalf` prover for `HOL` proof-assistant
- Kaliszyk and Urban in [4] reconstruct proofs of different ATPs for `HOL Light`

Proof Reconstruction: Overview



Metis Theorem Prover

Metis is an automatic theorem prover for first-order logic with equality.

- Open source implemented
- Reads problems in TPTP format
- Outputs *detailed* proofs in TSTP format
- For the propositional logic, Metis has only three inference rules:

$$\frac{}{\Gamma \vdash \varphi_1 \lor \cdots \lor \varphi_n}\ \textsf{axiom}\ \varphi_1, \cdots, \varphi_n$$

$$\frac{}{\Gamma \vdash \varphi \lor \neg\varphi}\ \textsf{assume}\ \varphi$$

$$\frac{\Gamma \vdash \varphi_1 \lor \cdots \lor l \lor \cdots \lor \varphi_n \qquad \Gamma \vdash \psi_1 \lor \cdots \lor \neg\, l \lor \cdots \lor \psi_m}{\Gamma \vdash \varphi_1 \lor \cdots \lor \varphi_n \lor \psi_1 \lor \cdots \lor \psi_m}\ \textsf{resolve}\ l$$

Inference Rules of `Metis`

`TSTP` derivations by `Metis` exhibit the following inferences:

`Metis` **rule**	**Purpose**
`strip`	Strip a goal into subgoals
`conjunct`	Takes a formula from a conjunction
`resolve`	A general form of the resolution theorem
`canonicalize`	Normalization of the formula
`clausify`	Performs clausification
`simplify`	Simplify definitions and theorems

Metis Algorithm

Algorithm 1 Metis Refutation Strategy

```
procedure METIS
input: a goal and a set of premises $a_i$
output: maybe a derivation when $\{a_i\} \vdash goal$, otherwise nothing.
    Strip the goal into a list of subgoals.
    for each subgoal $s_i$ do
        Try to find by a refutation of $\neg s_i$:
    end for
end procedure
```

Proof Reconstruction

Stripping a Goal

Splitting a Conjunct

Resolution

Canonicalize

Clausification

Simplification

Formalization Challenges

- Terminating of functions to reconstruct inference rules
- Intuitionistic logic implementation

Complete Example

The problem[1]:

$$(p \Rightarrow q) \land (q \Rightarrow p) \vdash (p \lor q) \Rightarrow (p \land q)$$

In TPTP syntax:

```
fof(a_1, axiom, (p ⇒ q) ∧ (q ⇒ p)).
fof(goal, conjecture, (p ∨ q) ⇒ (p ∧ q)).
```

Its TSTP solution using Metis:

```
fof(a_1, axiom, (p ⇒ q) ∧ (q ⇒ p)).
fof(goal, conjecture, (p ∨ q) ⇒ (p ∧ q))).
fof(s_1, (p ∨ q) ⇒ p, inf(strip, goal)).
fof(s_2, ((p ∨ q) ∧ p) ⇒ q, inf(strip, goal)).
...
```

[1] Problem No. 13 in Disjunction Section in [7]

```
fof(s_1, (p ∨ q) ⇒ p, inf(strip, goal)).
fof(s_2, ((p ∨ q) ∧ p) ⇒ q, inf(strip, goal)).
fof(neg_1, ¬ ((p ∨ q) ⇒ p), inf(negate, s_1)).
fof(n00, (¬ p ∨ q) ∧ (¬ q ∨ p), inf(canonicalize, a_1)).
fof(n01, ¬ q ∨ p, inf(conjunct, n00)).
fof(n02, ¬ p ∧ (p ∨ q), inf(canonicalize, neg_1)).
fof(n03, p ∨ q, inf(conjunct, n02)).
fof(n04, ¬ p, inf(conjunct, n02)).
fof(n05, q, inf(simplify,[n03, n04])).
cnf(r00, ¬ q ∨ p, inf(canonicalize, n01)).
cnf(r01, q, inf(canonicalize, n05)).
cnf(r02, p, inf(resolve, q, [r01, r00])).
cnf(r03, ¬ p, inf(canonicalize, n04)).
cnf(r04, ⊥, inf(resolve, p, [r02, r03])).
fof(neg_2, ¬ (((p ∨ q) ∧ p) ⇒ q), inf(negate, s2)).
fof(n10, ¬ q ∧ p ∧ (p ∨ q), inf(canonicalize, neg_2)).
fof(n11, (¬ p ∨ q) ∧ (¬ q ∨ p), inf(canonicalize, a_1)).
fof(n12, ¬ p ∨ q, inf(conjunct, n11)).
fof(n13, ⊥, inf(simplify,[n10, n12])).
cnf(r10, ⊥, inf(canonicalize, n13)).
```

TSTP Refutation of Subgoal No. 1

```
fof(s_1, (p ∨ q) ⇒ p, inf(strip, goal)).
fof(neg_1, ¬ ((p ∨ q) ⇒ p), inf(negate, s_1)).
fof(n00, (¬ p ∨ q) ∧ (¬ q ∨ p), inf(canonicalize, a_1)).
fof(n01, ¬ q ∨ p, inf(conjunct, n00)).
fof(n02, ¬ p ∧ (p ∨ q), inf(canonicalize, neg_1)).
fof(n03, p ∨ q, inf(conjunct, n02)).
fof(n04, ¬ p, inf(conjunct, n02)).
fof(n05, q, inf(simplify,[n03, n04])).
cnf(r00, ¬ q ∨ p, inf(canonicalize, n01)).
cnf(r01, q, inf(canonicalize, n05)).
cnf(r02, p, inf(resolve, q, [r01, r00])).
cnf(r03, ¬ p, inf(canonicalize, n04)).
cnf(r04, ⊥, inf(resolve, p, [r02, r03])).
```

Tree for the Subgoal No. 1: $(p \vee q) \Rightarrow p$

```
fof(a_1, axiom, (p ⇒ q) ∧ (q ⇒ p)).
...
fof(n00, (¬ p ∨ q) ∧ (¬ q ∨ p), inf(canonicalize, a_1)).
fof(n01, ¬ q ∨ p, inf(conjunct, n00)).
...
```

$$(\mathcal{D}_1) \qquad \cfrac{\cfrac{\cfrac{\cfrac{}{\Gamma \vdash (p \Rightarrow q) \wedge (q \Rightarrow p)}\;\text{axiom } a_1}{\neg s_1 \vdash (p \Rightarrow q) \wedge (q \Rightarrow p)}\;\text{weaken}}{\neg s_1 \vdash (\neg p \vee q) \wedge (\neg q \vee p)}\;\text{canonicalize}}{\neg s_1 \vdash \neg q \vee p}\;\text{conjunct}$$

```
...
fof(s_1, (p ∨ q) ⇒ p, inf(strip, goal)).
fof(neg_1, ¬ ((p ∨ q) ⇒ p), inf(negate, s_1)).
...
fof(n02, ¬ p ∧ (p ∨ q), inf(canonicalize, neg_1)).
fof(n03, p ∨ q, inf(conjunct, n02)).
fof(n04, ¬ p, inf(conjunct, n02)).
...
```

$$(\mathcal{D}_2) \qquad \cfrac{\cfrac{\cfrac{}{\neg s_1 \vdash \neg s_1}\ \textsf{assume}}{\neg s_1 \vdash \neg p \wedge (p \vee q)}\ \textsf{canonicalize}}{\neg s_1 \vdash p \vee q}\ \textsf{conjunct}$$

$$(\mathcal{D}_3) \qquad \cfrac{\cfrac{\cfrac{}{\neg s_1 \vdash \neg s_1}\ \textsf{assume}\ \neg s_1}{\neg s_1 \vdash \neg p \wedge (p \vee q)}\ \textsf{canonicalize}}{\neg s_1 \vdash \neg p}\ \textsf{conjunct}$$

$$(\mathcal{D}_4) \qquad \dfrac{\dfrac{\mathcal{D}_2}{\neg s_1 \vdash p \vee q} \qquad \dfrac{\mathcal{D}_3}{\neg s_1 \vdash \neg p}}{\neg s_1 \vdash q}\ \textsf{simplify}$$

$$(\mathcal{R}_1) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\mathcal{D}_1}{\neg s_1 \vdash \neg q \vee p} \qquad \dfrac{\mathcal{D}_4}{\neg s_1 \vdash q}}{\neg s_1 \vdash p}\ \textsf{resolve } q \qquad \dfrac{\mathcal{D}_3}{\neg s_1 \vdash \neg p}}{\neg s_1 \vdash \bot}\ \textsf{resolve } p}{\Gamma \vdash s_1}\ \textsf{RAA}$$

Tree for the Subgoal No. 2: $((p \vee q) \wedge p) \Rightarrow q$

```
fof(s_2, ((p ∨ q) ∧ p) ⇒ q, inf(strip, goal)).
fof(neg_2, ¬ (((p ∨ q) ∧ p) ⇒ q), inf(negate, s2)).
fof(n10, ¬ q ∧ p ∧ (p ∨ q), inf(canonicalize, neg_2)).
fof(n11, (¬ p ∨ q) ∧ (¬ q ∨ p), inf(canonicalize, a_1)).
fof(n12, ¬ p ∨ q, inf(conjunct, n11)).
fof(n13, ⊥, inf(simplify,[n10, n12])).
cnf(r10, ⊥, inf(canonicalize, n13)).
```

$$(\mathcal{R}_2)\qquad \dfrac{\dfrac{\dfrac{}{\Gamma, \neg s_2 \vdash \neg s_2}\ \text{assume } (\neg s_2)}{\neg s_2 \vdash \neg q \wedge p \wedge (p \vee q)}\ \text{canonicalize} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash (p \Rightarrow q) \wedge (q \Rightarrow p)}\ \text{axiom } a_1}{\neg s_2 \vdash (p \Rightarrow q) \wedge (q \Rightarrow p)}\ \text{weaken}}{\neg s_2 \vdash (\neg p \vee q) \wedge (\neg q \vee p)}\ \text{canonicalize}}{\neg s_2 \vdash \neg p \vee q}\ \text{conjunct}}{\dfrac{\neg s_2 \vdash \bot}{\Gamma \vdash s_2}\ \text{RAA}}\ \text{simplify}$$

Summarizing the Example

The problem was:

$$(p \Rightarrow q) \wedge (q \Rightarrow p) \vdash (p \vee q) \Rightarrow (p \wedge q)$$

Its `TSTP` solution using `Metis` was:

```
fof(a_1, axiom, (p ⇒ q) ∧ (q ⇒ p)).
fof(goal, conjecture, (p ∨ q) ⇒ (p ∧ q))).
fof(s_1, (p ∨ q) ⇒ p, inf(strip, goal)).
fof(s_2, ((p ∨ q) ∧ p) ⇒ q, inf(strip, goal)).
...
```

The proof is:

$$\dfrac{\dfrac{}{\Gamma \vdash (s_1 \wedge s_2) \Rightarrow \text{goal}}\,\text{strip} \qquad \dfrac{\dfrac{\mathcal{R}_1}{\Gamma \vdash s_1} \qquad \dfrac{\mathcal{R}_2}{\Gamma \vdash s_2}}{\Gamma \vdash s_1 \wedge s_2}\,\wedge\text{-intro}}{\Gamma \vdash \text{goal}}\,\Rightarrow\text{-elim}$$

Results

Academic results: paper (work in progress)
Software related results:

- Athena[2]: a translator tool for Metis proofs to Agda in Haskell
- Agda libraries:
  - Agda-Metis[3]: Metis prover reasoning for propositional logic
  - Agda-Prop[4]: intuitionistic propositional logic with PEM
- Bugs found in Metis: see issues No. 2, No. 4, and commit 8a3f11e in Metis official repository[5]

In parallel, we develop:

- Online-ATPs[6]: a client for the TPTP world in Haskell This tool allowed us to use Metis without installing it
- Prop-Pack[7]: Compendium of TPTP problems in classical propositional logic used to test Athena

[2] https://github.com/jonaprieto/athena.

[3] https://github.com/jonaprieto/agda-metis.

[4] https://github.com/jonaprieto/agda-prop.

[5] https://github.com/gilith/metis.

[6] https://github.com/jonaprieto/online-atps.

[7] https://github.com/jonaprieto/prop-pack.

Future Work

Further research directions include, but are not limited to:

- improve the performance of the `canonicalize` rule
- extend the proof-reconstruction presented in this paper to
  - support the proposition logic with equality of `Metis`
  - support other ATPs for propositional logic like `EProver` or `Z3`. See Kanso's Ph.D. thesis [5]
  - support `Metis` first-order proofs

References I

Marc Bezem, Dimitri Hendriks, and Hans de Nivelle. Automated Proof Construction in Type Theory Using Resolution. Journal of Automated Reasoning 29.3-4 (2002), pp. 253–275. DOI: 10.1023/A:1021939521172 (cit. on p. 4).

Simon Foster and Georg Struth. Integrating an Automated Theorem Prover in Agda. In: NASA Formal Methods (NFM 2011). Ed. by Mihael Bobaru et al. Vol. 6617. Lecture Notes in Computer Science. Springer, 2011, pp. 116–130. DOI: 10.1007/978-3-642-20398-5_10 (cit. on p. 4).

Joe Hurd. Integrating Gandalf and HOL. In: Theorem Proving in Higher Order Logics (TPHOLs 2001). Ed. by Yves Bertot, Gilles Dowek, Laurent Théry, and Christine Paulin. Vol. 1690. Lecture Notes in Computer Science. Springer, 2001, pp. 311–321. DOI: 10.1007/3-540-48256-3_21 (cit. on p. 4).

References II

- Cezary Kaliszyk and Josef Urban. PRocH: Proof Reconstruction for HOL Light. In: Automated Deduction (CADE-24). Ed. by Maria Paola Bonacina. Vol. 7898. Lecture Notes in Artifical Intellingence. Springer, 2013, pp. 267–274. DOI: 10.1007/978-3-642-38574-2_18 (cit. on p. 4).

- Karim Kanso. Agda as a Platform for the Development of Verified Railway Interlocking Systems. PhD thesis. Department of Computer Science. Swansea University, 2012. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.310.1502 (cit. on pp. 4, 26).

- Lawrence C. Paulson and Kong Woei Susanto. Source-level Proof Reconstruction For Interactive Theorem Proving. In: TPHOLs. Vol. 4732. Springer. 2007, pp. 232–245 (cit. on p. 4).

- Jonathan Prieto-Cubides. A Collection of Propositional Problems in TPTP Format. June 2017. DOI: 10.5281/ZENODO.817997 (cit. on p. 17).

TPTP Syntax

Thousands of Problems for Theorem Provers

- Is a language[8] to encode problems
- Is the input of the ATPs
- Annotated formulas with the form

```
language(name, role, formula).
```

`language`	FOF or CNF
`name`	to identify the formula within the problem
`role`	axiom, definition, hypothesis, conjecture
`formula`	formula in TPTP format

[8] `http://www.cs.miami.edu/~tptp/TPTP/SyntaxBNF.html`.

TSTP Syntax

A TSTP derivation[9]

- Is a **D**irected **A**cyclic **G**raph where
  - leaf is a formula from the TPTP input
  - node is a formula inferred from parent formula
  - root the final derived formula
- Is a list of annotated formulas with the form

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

where source typically is an inference record

```
inference(rule, useful info, parents).
```

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

TSTP Example

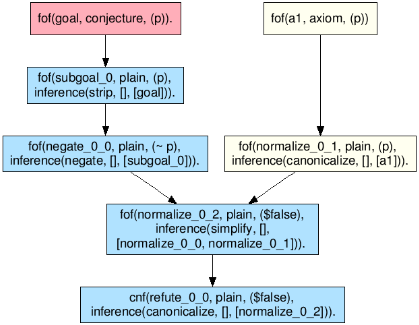
- Proof found by Metis for the problem $p \vdash p$

```
$ metis --show proof problem.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])).
```

DAG Example

By refutation, we proved $p \vdash p$:

$$\cfrac{\cfrac{\cfrac{\cfrac{}{\neg p}\,\text{assume}}{\neg p}\,\text{strip} \qquad \cfrac{\cfrac{}{p}\,\text{axiom}}{p}\,\text{canonicalize}}{\bot}\,\text{simplify}}{\bot}\,\text{canonicalize}$$



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

fof(a1, axiom, (p))

fof(subgoal_0, plain, (p),
inference(strip, [], [goal])).

fof(negate_0_0, plain, (~ p),
inference(negate, [], [subgoal_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])).
```