Reconstructing Propositional Proofs in Type Theory

Jonathan Prieto-Cubides
Advisor: Andrés Sicard-Ramírez

Master in Applied Mathematics
Universidad EAFIT
Medellín, Colombia

Novembe **UNIVERSIDAD EAFIT**®

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:

- ▶ **Kanso2012** in [**Kanso2012**] reconstructs in `Agda` propositional proofs generated by `EProver` and `Z3`
- ▶ **foster2011integrating** in [**foster2011integrating**] describe proof-reconstruction in `Agda` for equational logic of `Waldmeister` prover
- ▶ **Bezem2002** in [**Bezem2002**] transform a proof produced by the first-order prover `Bliksem` in a `Coq` proof-term

In classical logic:

- ▶ **paulson2007source** in [**paulson2007source**] introduce `SledgeHammer`, a tool ables to reconstructs proofs of well-known ATPs: `EProver`, `Vampire`, among others using `SystemOnTPTP` server
- ▶ **Hurd1999** in [**Hurd1999**] integrates the first-order resolution prover `Gandalf` prover for `HOL` proof-assistant
- ▶ **kaliszyk2013** in [**kaliszyk2013**] reconstruct proofs of different ATPs for `HOL Light`

Proof Reconstruction: Overview
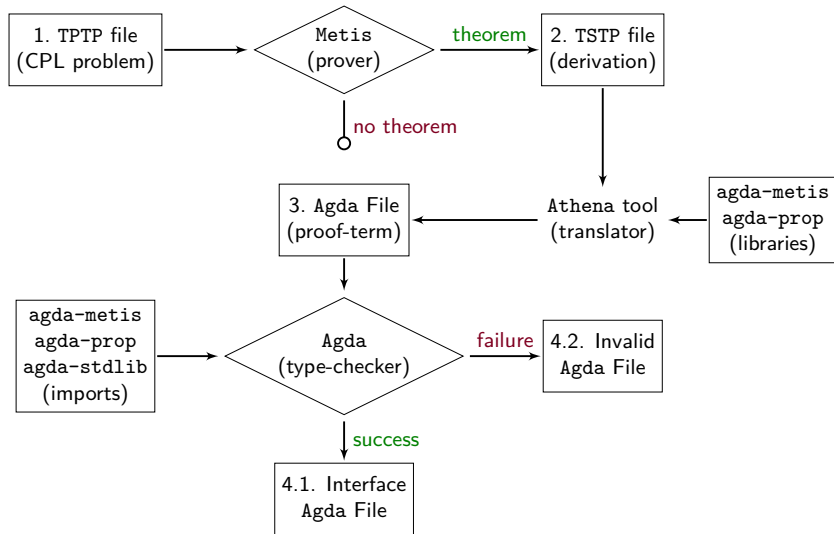
Metis **Theorem Prover**

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:

$$\overline{\Gamma \vdash \varphi_1 \vee \cdots \vee \varphi_n} \text{ axiom } \varphi_1, \cdots, \varphi_n$$

$$\overline{\Gamma \vdash \varphi \vee \neg\varphi} \text{ assume } \varphi$$

$$\frac{\Gamma \vdash \varphi_1 \vee \cdots \vee l \vee \cdots \vee \varphi_n \qquad \Gamma \vdash \psi_1 \vee \cdots \vee \neg l \vee \cdots \vee \psi_m}{\Gamma \vdash \varphi_1 \vee \cdots \vee \varphi_n \vee \psi_1 \vee \cdots \vee \psi_m} \text{ resolve } l$$

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 **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

Resolution

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(a₁, axiom, (p ⇒ q) ∧ (q ⇒ p)).
fof(goal, conjecture, (p ∨ q) ⇒ (p ∧ q)).
```

Its TSTP solution using Metis:

```
fof(a₁, axiom, (p ⇒ q) ∧ (q ⇒ p)).
fof(goal, conjecture, (p ∨ q) ⇒ (p ∧ q)).
fof(s₁, (p ∨ q) ⇒ p, inf(strip, goal)).
fof(s₂, ((p ∨ q) ∧ p) ⇒ q, inf(strip, goal)).
...
```

---

[1]Problem No. 13 in Disjunction Section in [**Prieto-Cubides2017**]

```
fof(s₁, (p ∨ q) ⇒ p, inf(strip, goal)).
fof(s₂, ((p ∨ q) ∧ p) ⇒ q, inf(strip, goal)).
fof(neg₁, ¬ ((p ∨ q) ⇒ p), inf(negate, s₁)).
fof(n00, (¬ p ∨ q) ∧ (¬ q ∨ p), inf(canonicalize, a₁)).
fof(n01, ¬ q ∨ p, inf(conjunct, n00)).
fof(n02, ¬ p ∧ (p ∨ q), inf(canonicalize, neg₁)).
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(neg₂, ¬ (((p ∨ q) ∧ p) ⇒ q), inf(negate, s2)).
fof(n10, ¬ q ∧ p ∧ (p ∨ q), inf(canonicalize, neg₂)).
fof(n11, (¬ p ∨ q) ∧ (¬ q ∨ p), inf(canonicalize, a₁)).
fof(n12, ¬ p ∨ q, inf(conjunct, n11)).
fof(n13, ⊥, inf(simplify,[n10, n12])).
cnf(r10, ⊥, inf(canonicalize, n13)).
```

```
fof(s₁, (p ∨ q) ⇒ p, inf(strip, goal)).
fof(neg₁, ¬ ((p ∨ q) ⇒ p), inf(negate, s₁)).
fof(n00, (¬ p ∨ q) ∧ (¬ q ∨ p), inf(canonicalize, a₁)).
fof(n01, ¬ q ∨ p, inf(conjunct, n00)).
fof(n02, ¬ p ∧ (p ∨ q), inf(canonicalize, neg₁)).
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: $(p \lor q) \Rightarrow p$

```
fof(a₁, axiom, (p ⇒ q) ∧ (q ⇒ p)).
...
fof(n00, (¬ p ∨ q) ∧ (¬ q ∨ p), inf(canonicalize, a₁)).
fof(n01, ¬ q ∨ p, inf(conjunct, n00)).
...
```

$$(\mathcal{D}_1) \qquad \frac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash (p \Rightarrow q) \land (q \Rightarrow p)} \text{ axiom } a_1}{\neg s_1 \vdash (p \Rightarrow q) \land (q \Rightarrow p)} \text{ weaken}}{\neg s_1 \vdash (\neg p \lor q) \land (\neg q \lor p)} \text{ canonicalize}}{\neg s_1 \vdash \neg q \lor p} \text{ conjunct}$$

```
...
fof(s₁, (p ∨ q) ⇒ p, inf(strip, goal)).
fof(neg₁, ¬ ((p ∨ q) ⇒ p), inf(negate, s₁)).
...
fof(n02, ¬ p ∧ (p ∨ q), inf(canonicalize, neg₁)).
fof(n03, p ∨ q, inf(conjunct, n02)).
fof(n04, ¬ p, inf(conjunct, n02)).
...
```

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

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

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

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

```
fof(s₂, ((p ∨ q) ∧ p) ⇒ q, inf(strip, goal)).
fof(neg₂, ¬ (((p ∨ q) ∧ p) ⇒ q), inf(negate, s2)).
fof(n10, ¬ q ∧ p ∧ (p ∨ q), inf(canonicalize, neg₂)).
fof(n11, (¬ p ∨ q) ∧ (¬ q ∨ p), inf(canonicalize, a₁)).
fof(n12, ¬ p ∨ q, inf(conjunct, n11)).
fof(n13, ⊥, inf(simplify,[n10, n12])).
cnf(r10, ⊥, inf(canonicalize, n13)).
```

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

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(a₁, axiom, (p ⇒ q) ∧ (q ⇒ p)).
fof(goal, conjecture, (p ∨ q) ⇒ (p ∧ q))).
fof(s₁, (p ∨ q) ⇒ p, inf(strip, goal)).
fof(s₂, ((p ∨ q) ∧ p) ⇒ q, inf(strip, goal)).
...
```

The proof is:

$$
\cfrac{
\cfrac{}{\Gamma \vdash (s_1 \wedge s_2) \Rightarrow \mathsf{goal}} \text{ strip}
\quad
\cfrac{
\cfrac{\mathcal{R}_1}{\Gamma \vdash s_1} \quad \cfrac{\mathcal{R}_2}{\Gamma \vdash s_2}
}{\Gamma \vdash s_1 \wedge s_2} \wedge\text{-intro}
}{\Gamma \vdash \mathsf{goal}} \Rightarrow\text{-elim}
$$

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 [**Kanso2012**]
  - ▶ support `Metis` first-order proofs

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 $p \vdash p$:

$$\frac{}{\neg\, p}\ \text{assume} \qquad \frac{}{p}\ \text{axiom}$$

$$\frac{\neg\, p}{\neg\, p}\ \text{strip} \qquad \frac{p}{p}\ \text{canonicalize}$$

$$\frac{}{\bot}\ \text{simplify}$$

$$\frac{\bot}{\bot}\ \text{canonicalize}$$

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])).