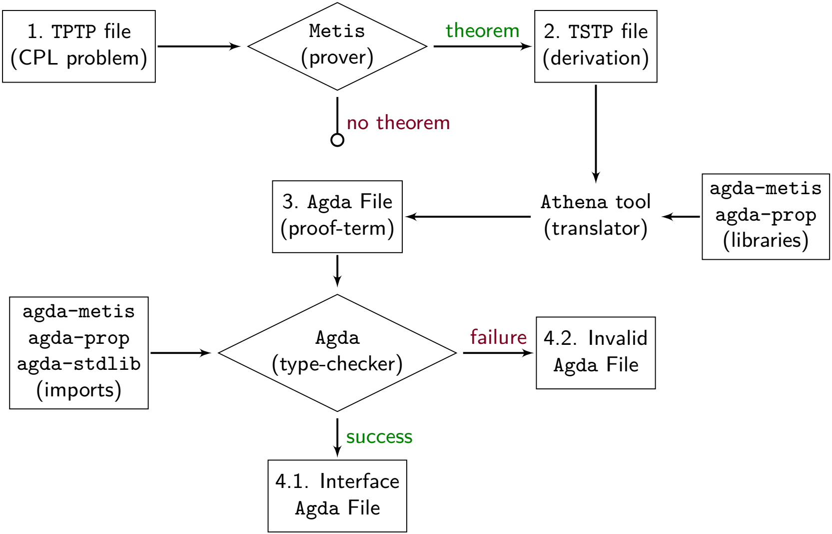
Reconstructing Propositional Proofs in Type Theory

Jonathan Prieto-Cubides
Advisor: Andrés Sicard-Ramírez

Master in Applied Mathematics
Universidad EAFIT
Medellín, Colombia

Novembe

Proof Reconstruction: Overview



`Metis` Theorem Prover

`Metis` is an automatic theorem prover for first-order logic with equality. For the propositional logic, `Metis` has only three inference rules:

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

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

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

Why Metis?

- Open source implemented in `Standard ML`
- Each refutation step is one of the three rules
- Reads problems in `TPTP` format
- Outputs *detailed* proofs in `TSTP` format

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

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:

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

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)).
...
```

```
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)).
```

TSTP Refutation of Subgoal No. 1

```
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])).
```

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

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

$(\mathcal{D}_1)$

$$\cfrac{\cfrac{\cfrac{\cfrac{}{\Gamma \vdash (p \Rightarrow q) \wedge (q \Rightarrow p)}\text{ axiom } a_1}{\Gamma, \neg s_1 \vdash (p \Rightarrow q) \wedge (q \Rightarrow p)}\text{ weaken}}{\Gamma, \neg s_1 \vdash (\neg p \vee q) \wedge (\neg q \vee p)}\text{ canonicalize}}{\Gamma, \neg s_1 \vdash \neg q \vee p}\text{ conjunct}$$

```
...
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)).
...
```

($\mathcal{D}_2$)

$$\dfrac{\dfrac{\dfrac{}{\Gamma, \neg s_1 \vdash \neg s_1}\ \textsf{assume}}{\Gamma, \neg s_1 \vdash \neg p \wedge (p \vee q)}\ \textsf{canonicalize}}{\Gamma, \neg s_1 \vdash p \vee q}\ \textsf{conjunct}$$

($\mathcal{D}_3$)

$$\dfrac{\dfrac{\dfrac{}{\Gamma, \neg s_1 \vdash \neg s_1}\ \textsf{assume}\ \neg s_1}{\Gamma, \neg s_1 \vdash \neg p \wedge (p \vee q)}\ \textsf{canonicalize}}{\Gamma, \neg s_1 \vdash \neg p}\ \textsf{conjunct}$$

$(\mathcal{D}_4)$

$$\dfrac{\dfrac{\mathcal{D}_2}{\Gamma, \neg s_1 \vdash p \vee q} \qquad \dfrac{\mathcal{D}_3}{\Gamma, \neg s_1 \vdash \neg p}}{\Gamma, \neg s_1 \vdash q}\ \textsf{simplify}$$

$(\mathcal{D}_5)$

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

Refutation of Subgoal 2

```
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)).
```

Results

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

- `Athena`[1]: a translator tool for `Metis` proofs to `Agda` in `Haskell`
- `Agda` libraries:
  - `Agda-Metis`[2]: `Metis` prover reasoning for propositional logic
  - `Agda-Prop`[3]: intuitionistic propositional logic with PEM
- Bugs found in `Metis`: see issues No. 2, No. 4, and commit `8a3f11e` in `Metis` official repository[4]

In parallel, we develop:

- `Online-ATPs`[5]: a client for the TPTP world in `Haskell` This tool allowed us to use `Metis` without installing it
- `Prop-Pack`[6]: Compendium of TPTP problems in classical propositional logic used to test `Athena`

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

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

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

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

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

[6] 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

References I

- Is a language[7] 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

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

TSTP Syntax

A TSTP derivation[8]

- 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).
```

[8] 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 $p \vdash p$:

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