Proof Reconstruction

Alejandro Gómez-Londoño

Advisor - Andrés Sicard-Ramírez

EAFIT University

Project proposal, February 24 2015

Introduction
A (very) general idea of the context

Proof assistant

ATP

Proof assistants are computer systems that allow a user to do mathematics on a computer, but not so much the computing aspect of mathematics but the aspects of proving and defining [1]

[1] H. Geuvers. "Proof assistants: History, ideas and future".

Deals with the development of computer programs that show that some statement (the conjecture) is a logical consequence of a set of statements (the axioms and hypotheses). [2]

[2] Geoff Sutcliffe's, "What is Automated Theorem Proving?". http://www.cs.miami.edu/ tptp/OverviewOfATP.html

Introduction
Examples

ATPs:

- Vampire
- E
- Metis
- SPASS
- Equinox

Proof assistants:

- Coq
- Agda
- Isabelle
- Mizar
- NuPRL

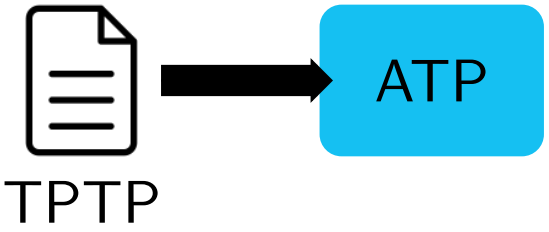
Proofs
in Agda

```
left-dist : ∀ m n p → m * (n + p) ≡ m * n + m * p
left-dist zero    _ _ = refl
left-dist (suc m) n p =
  begin
    (n + p) + m * (n + p)
      ≡⟨ cong (λ x → (n + p) + x) (left-dist m n p)⟩
    (n + p) + (m * n + m * p)
      ≡⟨ cong (λ x → x + (m * n + m * p)) (+-comm n p)⟩
    (p + n) + (m * n + m * p)
      ≡⟨ +-assoc (p + n) (m * n) (m * p)⟩
    ((p + n) + m * n) + m * p
      ≡⟨ cong (λ x → x + m * p) (sym (+-assoc p n (m * n)))⟩
    (p + (suc m * n)) + m * p
      ≡⟨ cong (λ x → x + m * p) (+-comm p (suc m * n))⟩
    ((suc m * n) + p) + m * p
      ≡⟨ sym (+-assoc (suc m * n) p (m * p))⟩
    suc m * n + suc m * p
  ∎
```

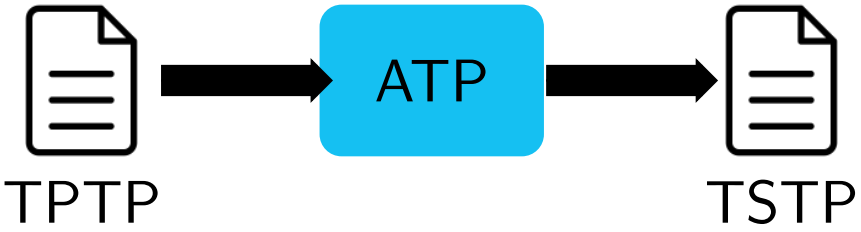
Proofs
ATPs input/output

ATP

Proofs
ATPs input/output



Proofs
ATPs input/output



Proof
Which ATP?

[3] http://www.gilith.com/software/metis/

Proof
Which ATP?

We choose Metis[3] for a couple of reasons:

- Uses TPTP as input format.

[3] http://www.gilith.com/software/metis/

Proof
Which ATP?

We choose Metis[3] for a couple of reasons:

- Uses TPTP as input format.
- Outputs proofs in TSTP format.

[3] http://www.gilith.com/software/metis/

Proof
Which ATP?

We choose Metis[3] for a couple of reasons:

- Uses TPTP as input format.
- Outputs proofs in TSTP format.
- Each refutation step is one of 6 rules.

[3] http://www.gilith.com/software/metis/

Proof
Which ATP?

We choose Metis[3] for a couple of reasons:

- Uses TPTP as input format.
- Outputs proofs in TSTP format.
- Each refutation step is one of 6 rules.
- Has respectable performance.

[3] http://www.gilith.com/software/metis/

Proofs
in metis (ATP)

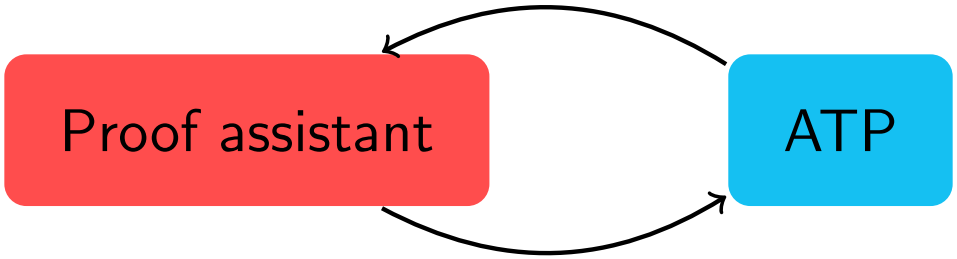
```
$ cat theorem.tptp
fof(id,conjecture,( ! [X] : ( p(X) => p(X) ))).

$ metis --show proof theorem.tptp
------------------------------------------------------------------------------
SZS status Theorem for theorem.tptp

SZS output start CNFRefutation for theorem.tptp
fof(id, conjecture, (! [X] : (p(X) => p(X)))).

fof(subgoal_0, plain, (! [X] : (p(X) => p(X))),
    inference(strip, [], [id])).

fof(negate_0_0, plain, (~ ! [X] : (p(X) => p(X))),
    inference(negate, [], [subgoal_0])).

fof(normalize_0_0, plain, ($false),
    inference(canonicalize, [], [negate_0_0])).

cnf(refute_0_0, plain, ($false),
    inference(canonicalize, [], [normalize_0_0])).
SZS output end CNFRefutation for theorem.tptp
```

Reconstruction

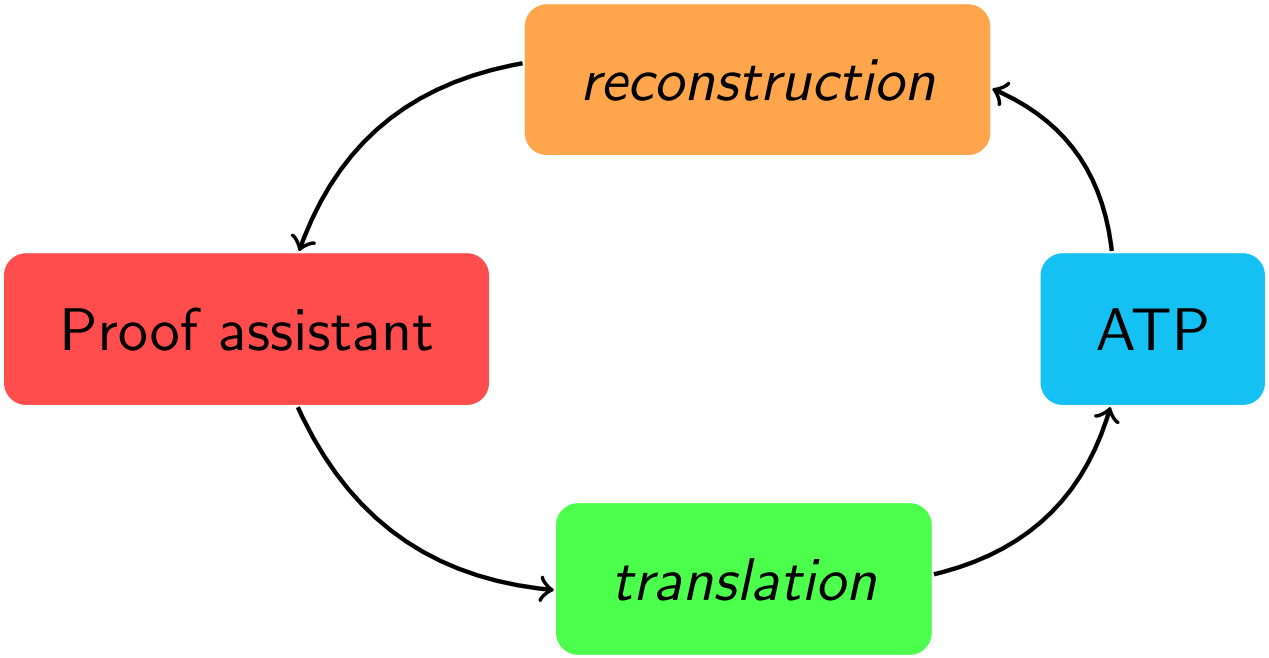
Proof assistant

ATP

Reconstruction



Reconstruction



Conclusion

- We will implement a *proof reconstruction module*.

Conclusion

- We will implement a *proof reconstruction module*.
- With Agda as the proof assistant.

Conclusion

- We will implement a *proof reconstruction module*.
- With Agda as the proof assistant.
- and Metis as the ATP.