

# Proof Reconstruction: Parsing Proofs

Alejandro Gómez-Londoño

Advisor - Andrés Sicard-Ramírez

EAFIT University

Report, June 5, 2015

# Objective

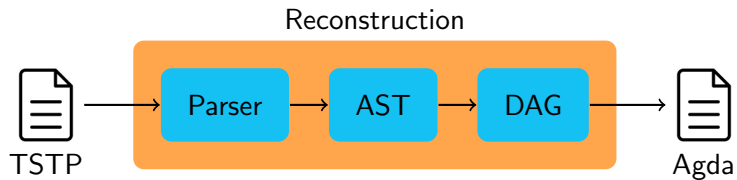
*Translate the output from an ATP-generated proof into idiomatic Agda code.*



# Schedule

- Clarify the problem.
- Select an ATP.
- Determine a development environments.
- Parser implementation. ◀
- AST to DAG translation.
- Typed-DAG construction.
- Proof term reconstruction in Agda.

# Overview



# Example

- Hand written proof

$$\frac{\frac{x \quad y}{x \wedge y} \quad x \wedge y \Rightarrow z}{z}$$

# Example

- Hand written proof

$$\frac{\frac{x \quad y}{x \wedge y} \quad x \wedge y \Rightarrow z}{z}$$

- TPTP problem

```
fof(a_0,axiom,x).  
fof(a_1,axiom,y).  
fof(a_2,axiom,((x & y) => z)).  
fof(c_0,conjecture, z).
```

# Example

- TSTP proof

```
fof(s_0,plain,(x & y),  
    inference(conj,[],[a_0,a_1])).
```

```
fof(s_1,plain,(z),  
    inference(modp,[],[a_2,s_0])).
```

```
fof(r_0,plain,($true),  
    inference(simplify,[],[s_1,c_0])).
```

# Example

- Haskell Type (AST)

```
F {name = "s_0",  
  role = Plain,  
  formula = BinOp  
      (PredApp (AtomicWord "x") [])  
      (:&:)  
      (PredApp (AtomicWord "y") []),  
  annotations = Inference Conj [] ["a_0", "a_1"]  
}
```



# Conclusions

- Metis is our ATP.

# Conclusions

- Metis is our ATP.
- Our proof assistant is Agda.

# Conclusions

- Metis is our ATP.
- Our proof assistant is Agda.
- Haskell is being used for most of the parsing.

# Conclusions

- Metis is our ATP.
- Our proof assistant is Agda.
- Haskell is being used for most of the parsing.
- And `logic-tptp` as parsing library.

# Conclusions

- Metis is our ATP.
- Our proof assistant is Agda.
- Haskell is being used for most of the parsing.
- And `logic-tptp` as parsing library.
- The project is available on github  
(<https://github.com/agomezl/tstp2agda>)