Integrating Automated Provers in Proof Assistants

Mohamed Yacine EL HADDAD

June 9, 2018

LSV, ENS Paris-Saclay, CNRS, INRIA, University of Paris-Saclay, DigiCosme

Table of contents

Introduction

Introduction

Theorem provers :

- Interactive : *Coq*, *Matita*, *Isabelle...*
- Automated : *Vampire*, *Z3*, *Zenon*, **ZenonModulo**, *E-Prover*, *iProverModulo...*

Introduction

Theorem provers :

- Interactive : *Coq*, *Matita*, *Isabelle*...
- Automated : *Vampire*, *Z3*, *Zenon*, **ZenonModulo**, *E-Prover*, *iProverModulo*...

How to check the proofs generated by the provers?

Introduction

Theorem provers :

- Interactive : *Coq*, *Matita*, *Isabelle...*
- Automated : *Vampire*, *Z3*, *Zenon*, **ZenonModulo**, *E-Prover*, *iProverModulo...*

How to check the proofs generated by the provers?
Proof checkers

Introduction

Dedukti is a **proof checker** based on $\lambda\Pi^{\equiv}$ Theory, an extension of the classical $\lambda\Pi$-*calculus*.
Features :

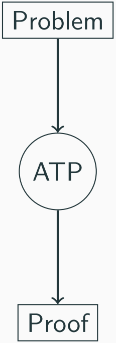
- Very expressive
- Fast
- Higher Order Rewriting

Introduction

Automated Theorem Prover (ATP) : A program that takes as input a formula and generate a **proof** or a **proof trace**.

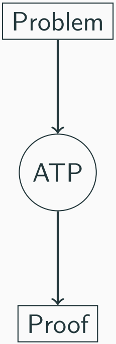
Introduction

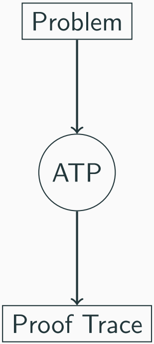
Automated Theorem Prover (ATP) : A program that takes as input a formula and generate a **proof** or a **proof trace**.



Introduction

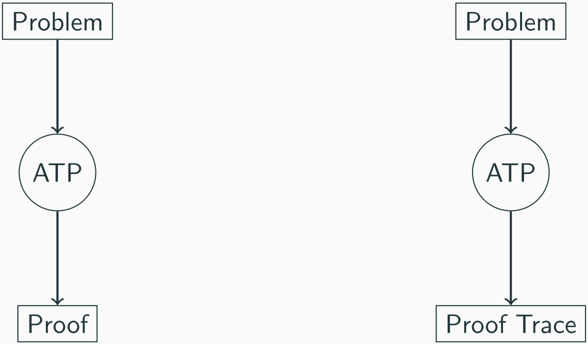
Automated Theorem Prover (ATP) : A program that takes as input a formula and generate a **proof** or a **proof trace**.





Introduction

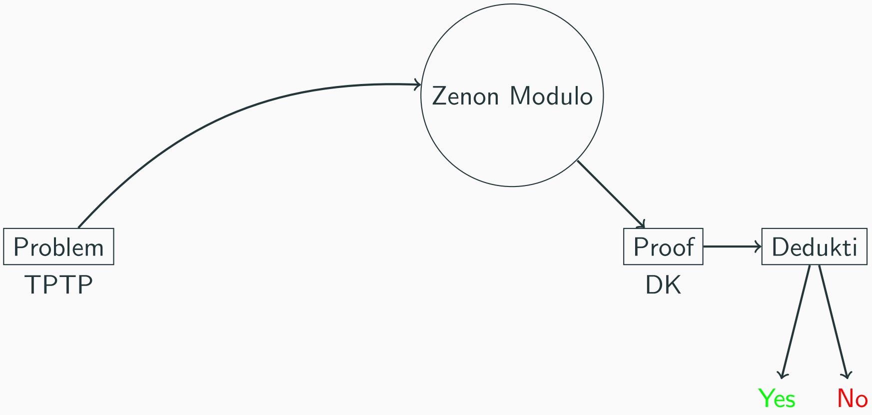
Automated Theorem Prover (ATP) : A program that takes as input a formula and generate a **proof** or a **proof trace**.



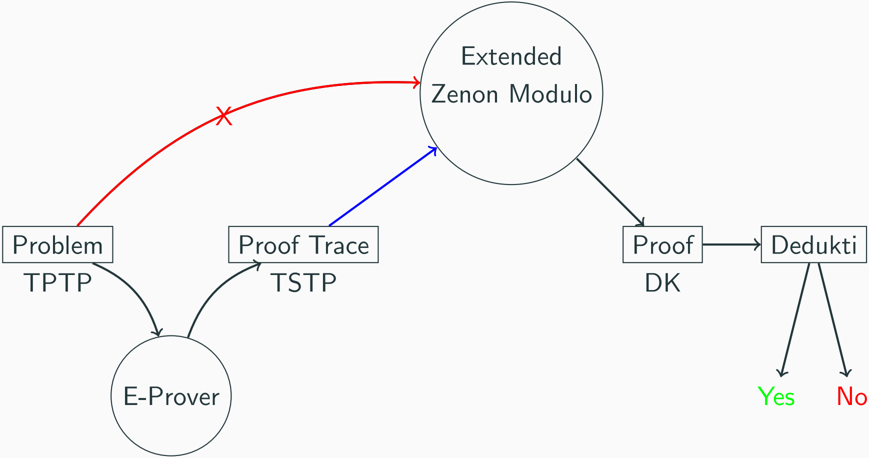
Easily checkable | Hard to check

Less powerful | More powerful

Introduction



Introduction



Zenon Modulo

Zenon Modulo

Zenon Modulo :

- Build easily **checkable** proofs
- Use the **Tableau** method
- Accept **TPTP** format
- Generate a **low level** format of proofs

Zenon Modulo

Building a proof of the formula F with Zenon Modulo :

- Negate F
- Use the Tableau method to demonstrate the insatisfiablity.
- **MLproof** as inference rules

$$\frac{\bot}{\odot}\;\odot_{\bot} \qquad \frac{\neg\top}{\odot}\;\odot_{\neg\top} \qquad \frac{P \quad \neg P}{\odot}\;\odot$$

$$\frac{\neg\neg P}{P}\;\alpha_{\neg\neg} \qquad \frac{}{P \mid \neg P}\;\mathit{cut}$$

$$\frac{P \wedge Q}{P, Q}\;\alpha_{\wedge} \qquad \frac{\neg(P \vee Q)}{\neg P, \neg Q}\;\alpha_{\neg\vee} \qquad \frac{\neg(P \Rightarrow Q)}{P, \neg Q}\;\beta_{\neg\Rightarrow}$$

$$\frac{P \vee Q}{P \mid Q}\;\beta_{\vee} \qquad \frac{\neg(P \wedge Q)}{\neg P \mid \neg Q}\;\beta_{\neg\wedge} \qquad \frac{P \Rightarrow Q}{\neg P \mid Q}\;\beta_{\Rightarrow}$$

Figure 1: MLproof inference rules

Zenon Modulo

Prove a from $(\neg b \vee a) \wedge b$

$$\{(\neg b \vee a) \wedge b, \neg a\}$$

Zenon Modulo

Prove a from $(\neg b \vee a) \wedge b$

$$\{(\neg b \vee a) \wedge b, \neg a\}$$

$$\begin{array}{c} (\neg b \vee a) \wedge b \\ | \\ \neg a \end{array}$$

Zenon Modulo

Prove a from $(\neg b \lor a) \land b$

$$\{(\neg b \lor a) \land b, \neg a\}$$

$$(\neg b \lor a) \land b$$

$$\neg a$$

$$\neg b \lor a \qquad \alpha_\land$$

$$b$$

Zenon Modulo

Prove a from $(\neg b \vee a) \wedge b$

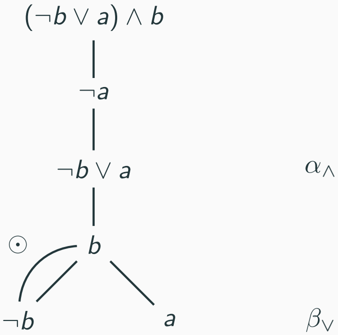
$$\{(\neg b \vee a) \wedge b, \neg a\}$$

$$
\begin{array}{ccc|c}
 & (\neg b \vee a) \wedge b & & \\
 & \neg a & & \\
 & \neg b \vee a & & \alpha_{\wedge} \\
 & b & & \\
\neg b & & a & \beta_{\vee}
\end{array}
$$

Zenon Modulo

Prove a from $(\neg b \vee a) \wedge b$

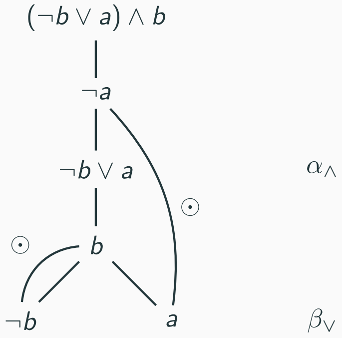
$\{(\neg b \vee a) \wedge b, \neg a\}$



Zenon Modulo

Prove a from $(\neg b \vee a) \wedge b$

$\{(\neg b \vee a) \wedge b, \neg a\}$



Zenon Modulo

$$\textit{MLproof} \Longrightarrow \textbf{LLproof}$$

- Low level format
- Easy to translate by Dedukti

Zenon Modulo

Variables and functions :

$$t ::= x \mid f \;\; t_1 ... t_n$$

Formulas :

$$F ::= \bot \mid \top \mid \neg F \mid F \vee F \mid F \wedge F \mid F \Rightarrow F \mid F \Leftrightarrow F \mid \forall x F \mid \exists x F \mid t_1 = t_2 \mid P \; t_1 ... t_n$$

Figure 2: Syntax of *LLproof* formula

$$\frac{}{\bot \vdash \bot}\ R_{\bot} \qquad \frac{}{\neg\top \vdash \bot}\ R_{\neg\top} \qquad \frac{}{\Gamma, P, \neg P \vdash \bot}\ R_{ax}$$

$$\frac{\Gamma, P \vee Q, P \vdash \bot \quad \Gamma, P \vee Q, Q \vdash \bot}{\Gamma, P \vee Q \vdash \bot}\ R_{\vee}$$

$$\frac{\Gamma, P \wedge Q, P, Q \vdash \bot}{\Gamma, P \wedge Q \vdash \bot}\ R_{\wedge} \qquad \frac{\Gamma, P \vdash \bot \quad \Gamma, \neg P \vdash \bot}{\Gamma \vdash \bot}\ R_{cut}$$

$$\frac{\Gamma, \neg P, P \Rightarrow Q \vdash \bot \quad \Gamma, Q, P \Rightarrow Q \vdash \bot}{\Gamma, P \Rightarrow Q \vdash \bot}\ R_{\Rightarrow}$$

Figure 3: LLproof rules

$$\dfrac{\dfrac{\dfrac{}{(\neg b \lor a) \land b, \neg a, \neg b \lor a, {\color{red}b}, {\color{red}\neg b} \vdash \bot}\,R_{ax} \qquad \dfrac{}{(\neg b \lor a) \land b, {\color{red}\neg a}, \neg b \lor a, b, {\color{red}a} \vdash \bot}\,R_{ax}}{(\neg b \lor a) \land b, \neg a, {\color{blue}\neg b \lor a}, {\color{blue}b} \vdash \bot}\,R_{\lor}}{(\neg b \lor a) \land b, \neg a \vdash \bot}\,R_{\land}$$

Figure 4: Proof representation in LLproof

$Form : Type$	$term : Type$
$x : term$	$\varphi(x) = x$
$f : term \to ... \to term \to term$	$\varphi(f\ t_1 ... t_n) := f \varphi(t_1)\ ...\ \varphi(t_n)$
$\dot{\bot} : Form$	$\varphi(\bot) := \dot{\bot}$
$\dot{\top} : Form$	$\varphi(\top) := \dot{\top}$
$\dot{\neg} : Form \to Form$	$\varphi(\neg A) := \dot{\neg}\varphi(A)$
$\dot{\wedge} : Form \to Form \to Form$	$\varphi(A \wedge B) := \varphi(A)\ \dot{\wedge}\ \varphi(B)$
$\dot{\vee} : Form \to Form \to Form$	$\varphi(A \vee B) := \varphi(A)\ \dot{\vee}\ \varphi(B)$
$\dot{\Rightarrow} : Form \to Form \to Form$	$\varphi(A \Rightarrow B) := \varphi(A) \dot{\Rightarrow} \varphi(B)$
$\dot{\forall} : (term \to Form) \to Form$	$\varphi(\forall x A) := \dot{\forall}(\lambda x : term.\ \varphi(A))$
$\dot{\exists} : (term \to Form) \to Form$	$\varphi(\exists x A) := \dot{\exists}(\lambda x : term.\ \varphi(A))$
$\dot{=} : term \to term \to Form$	$\varphi(x = y) := \varphi(x) \doteq \varphi(y)$
$P : term \to ... \to term \to Form$	$\varphi(P t_1 ... t_n) := P\ \varphi(t_1) ...\ \varphi(t_n)$

$$\varphi(F) : Form$$

Figure 5: Declaration of **LLproof** symbols in **Dedukti**

$$
\begin{aligned}
&\qquad\qquad\qquad\qquad\qquad \mathit{prf} : \mathit{Form} \to \mathit{Type}\\
&\mathit{prf}\,\bot \hookrightarrow \Pi P : \mathit{Form}.\mathit{prf}\ \ P\\
&\mathit{prf}\,\top \hookrightarrow \Pi P : \mathit{Form}.\mathit{prf}\ \ P \to \mathit{prf}\ \ P\\
&\mathit{prf}\,(\neg A) \hookrightarrow \mathit{prf}\ \ A \to \mathit{prf}\,\bot\\
&\mathit{prf}\,(A\ \wedge\ B) \hookrightarrow \Pi P : \mathit{Form}.(\mathit{prf}\ \ A \to \mathit{prf}\ \ B \to \mathit{prf}\ \ P) \to \mathit{prf}\ \ P\\
&\mathit{prf}\,(A\ \vee\ B) \hookrightarrow \Pi P : \mathit{Form}.(\mathit{prf}\ \ A \to \mathit{prf}\ \ P) \to (\mathit{prf}\ \ B \to \mathit{prf}\ \ P) \to \mathit{prf}\ \ P\\
&\mathit{prf}\,(A\ \Rightarrow\ B) \hookrightarrow \mathit{prf}\ \ A \to \mathit{prf}\ \ B\\
&\quad\vdots
\end{aligned}
$$

$$
\psi(A, ..., B \vdash \bot) := \mathit{prf}\,(\varphi(A)) \to ... \to \mathit{prf}\,(\varphi(B)) \to \mathit{prf}\,\dot{\bot}
$$

Figure 6: Transformation of *LLproof* formulas and judgments in *Dedukti*

Zenon Modulo

For each rule in *LLproof* we will give its representation in Dedukti :

$\dot{R}_\vee : \Pi P, Q : Form.(prf\,\varphi(P) \to prf\,\varphi(P \vee Q) \to prf\,\dot{\perp}) \to (prf\,\varphi(Q) \to prf\,\varphi(P \vee Q) \to prf\,\dot{\perp}) \to prf\,\varphi(P \vee Q) \to prf\,\dot{\perp}$

π :

$$\dfrac{\dfrac{\pi_1}{\Gamma, P \vee Q, P \vdash \perp} \qquad \dfrac{\pi_2}{\Gamma, P \vee Q, Q \vdash \perp}}{\Gamma, P \vee Q \vdash \perp}\, R_\vee$$

$$\delta(\pi) = \dot{R}_\vee\ P\ Q\ \delta(\pi_1)\ \delta(\pi_2)$$

Figure 7: Transformation of *LLproof* proofs in *Dedukti*

Zenon Modulo

$$\delta\left(\frac{\pi}{\Gamma \vdash \bot}\right) : \psi(\Gamma \vdash \bot)$$

Zenon Modulo

$$\delta\left(\frac{\pi}{\Gamma \vdash \bot}\right) : \psi(\Gamma \vdash \bot)$$

$$\dfrac{\dfrac{}{(\neg b \lor a) \land b, \neg a, \neg b \lor a, b, \neg b \vdash \bot} R_{ax} \qquad \dfrac{}{(\neg b \lor a) \land b, \neg a, \neg b \lor a, b, a \vdash \bot} R_{ax}}{\dfrac{(\neg b \lor a) \land b, \neg a, \neg b \lor a, b \vdash \bot}{(\neg b \lor a) \land b, \neg a \vdash \bot} R_{\land}} R_{\lor}$$

Zenon Modulo

$$\delta\left(\frac{\pi}{\Gamma \vdash \bot}\right) : \psi(\Gamma \vdash \bot)$$

$$\dfrac{\dfrac{}{(\neg b \vee a) \wedge b, \neg a, \neg b \vee a, b, \neg b \vdash \bot}\,R_{ax} \qquad \dfrac{}{(\neg b \vee a) \wedge b, \neg a, \neg b \vee a, b, a \vdash \bot}\,R_{ax}}{\dfrac{(\neg b \vee a) \wedge b, \neg a, \neg b \vee a, b \vdash \bot}{(\neg b \vee a) \wedge b, \neg a \vdash \bot}\,R_{\wedge}}\,R_{\vee}$$

$\delta(exemple) =$
$\dot{R}_{\wedge}\ \varphi(\neg b \vee a)\ \varphi(b)\ (\dot{R}_{\vee}\ \varphi(\neg b)\ \varphi(a)\ (\dot{R}_{ax}\ \varphi(b))(\dot{R}_{ax}\ \varphi(a)))$

Extending Zenon Modulo

Extending ZenonModulo

We will use a TSTP trace to help Zenon Modulo finding a proof.
Exemple of TSTP File :

```
cnf(a_0, axiom, ... ,file('Axioms/SET01.ax',member_of_set2_is_member_of_union)).
cnf(h, hypothesis, ... ,file('Hyothesis/H.p',a_union_a_is_aUa)).
cnf(a_1, axiom, ... ,file('Axioms/SET01.ax',subsets_axiom2)).
cnf(l_0, lemma, ... ,inference(spm,[status(thm)],[a_0, h])).
cnf(l_1, lemma, ... ,inference(spm,[status(thm)],[a_1])).
cnf(p, theorem, ..., inference(sr, [status(thm)], [inference(spm, [status(thm)],
  [l_0, h]), l_1])).
```

Extending ZenonModulo

The TSTP proof trace give informations for ZenonModulo about how the proof is generated. It contains all axioms and hypothesis used in the proof.

Exemple :

$$
\begin{array}{ll}
A_0, H & \vdash L_0 \\
A_1 & \vdash L_1 \\
L_0, L_1 & \vdash P
\end{array}
$$

$$
\begin{array}{llcll}
A_0, H & \vdash L_0 & \longmapsto & A_0, H, \neg L_0 & \vdash \bot \\
A_1 & \vdash L_1 & \longmapsto & A_1, \neg L_1 & \vdash \bot \\
L_0, L_1 & \vdash P & \longmapsto & L_0, L_1, \neg P & \vdash \bot
\end{array}
$$

ZenonModulo will generate a Dedukti proof for each deduction but the proof is not completed since we don't have any link between deductions.

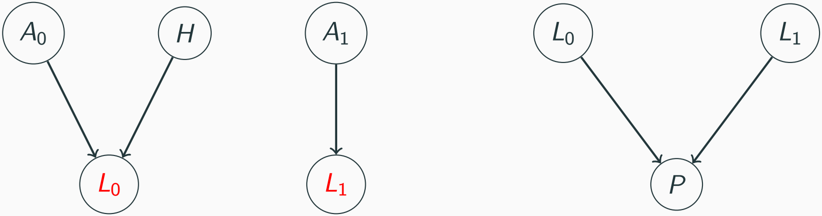


Figure 8: Diagram of proofs generated by ZenonModulo

ZenonModulo will generate a Dedukti proof for each deduction but the proof is not completed since we don't have any link between deductions.

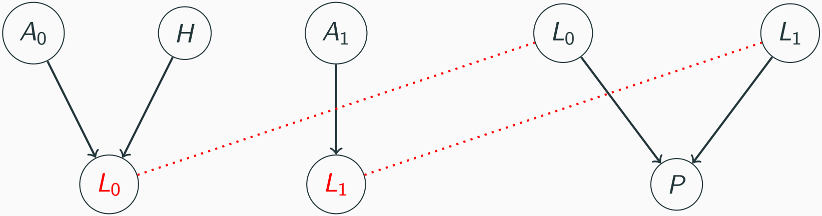


Figure 8: Diagram of proofs generated by ZenonModulo

$$\begin{array}{lll|l} \alpha & := A_0, H, \neg L_0 & \vdash \bot & \delta(\pi_\alpha) : \psi(\alpha) \\ \beta & := A_1, \neg L_1 & \vdash \bot & \delta(\pi_\beta) : \psi(\beta) \\ \gamma & := L_0, L_1, \neg P & \vdash \bot & \delta(\pi_\gamma) : \psi(\gamma) \end{array}$$

$$\begin{aligned} \psi(\alpha) &= \mathit{prf}\,\varphi(A_0) \to \mathit{prf}\,\varphi(H) \to \mathit{prf}\,\varphi(\neg L_0) \to \mathit{prf}\,\dot{\bot} \\ \psi(\beta) &= \mathit{prf}\,\varphi(A_1) \to \mathit{prf}\,\varphi(\neg L_1) \to \mathit{prf}\,\dot{\bot} \\ \psi(\gamma) &= \mathit{prf}\,\varphi(L_0) \to \mathit{prf}\,\varphi(L_1) \to \mathit{prf}\,\varphi(\neg P) \to \mathit{prf}\,\dot{\bot} \end{aligned}$$

Figure 9: Representation of the proof in Dedukti

But what we want is just **one** term of type :

$$\mathit{prf}\,\varphi(A_0) \to \mathit{prf}\,\varphi(H) \to \mathit{prf}\,\varphi(A_1) \to \mathit{prf}\,\varphi(\neg P) \to \mathit{prf}\,\dot{\bot}$$

$$prf\,\varphi(A_0) \to prf\,\varphi(H) \to prf\,\varphi(A_1) \to prf\,\varphi(\neg P) \to prf\,\dot{\bot}$$

Combine $\delta(\pi_\alpha)$, $\delta(\pi_\beta)$ and $\delta(\pi_\gamma)$:

$$\lambda a_0.\lambda h.\lambda a_1.\lambda np.\delta(\pi_\beta)a_1(\lambda y : \varphi(I_1).\delta(\pi_\alpha)a_0 h(\lambda z : \varphi(I_0).\delta(\pi_\gamma)zynp))$$

$$\mathit{prf}\,\varphi(A_0) \to \mathit{prf}\,\varphi(H) \to \mathit{prf}\,\varphi(A_1) \to \mathit{prf}\,\varphi(\neg P) \to \mathit{prf}\,\dot{\perp}$$

Combine $\delta(\pi_\alpha)$, $\delta(\pi_\beta)$ and $\delta(\pi_\gamma)$:

$$\lambda a_0.\lambda h.\lambda a_1.\lambda np.\delta(\pi_\beta)a_1(\lambda y : \varphi(I_1).\delta(\pi_\alpha)a_0 h(\lambda z : \varphi(I_0).\delta(\pi_\gamma)zynp))$$

Proof completed by using only A_0, H, A_1 and P

Further works

- Add **Skolemisation** to the extended ZenonModulo
- Translate Dedukti terms to another format accepted by ATPs

Questions

Questions ?