



# Architecture of the new OMEGA system + The CoRE<sup>+</sup> Calculus

Serge Autexier

`serge@ags.uni-sb.de`

DFKI GmbH & CS Department, Saarland University, Saarbrücken, Germany

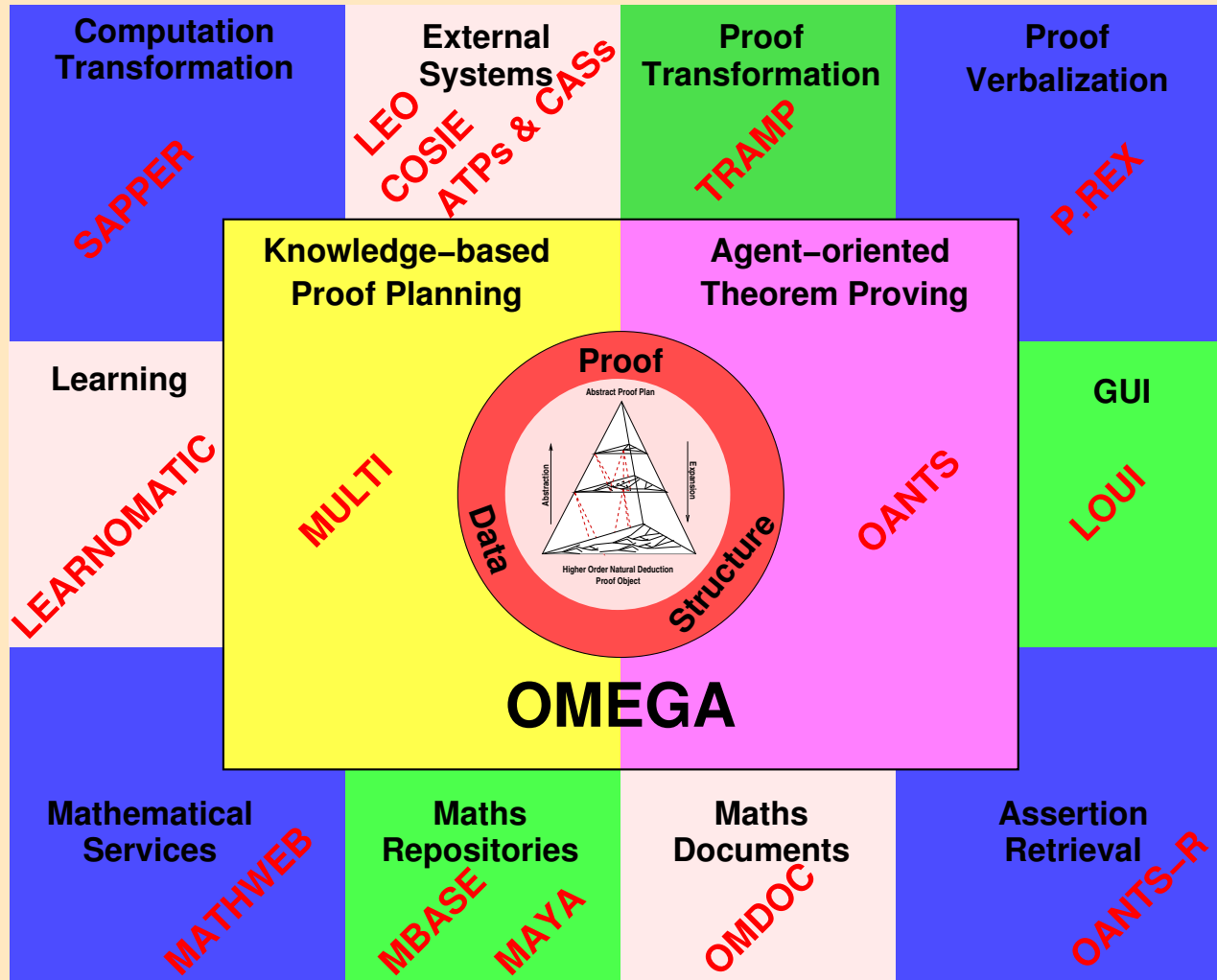
Theorema-Ultra-Omega Workshop, November 15 2005

Saarbrücken, Germany

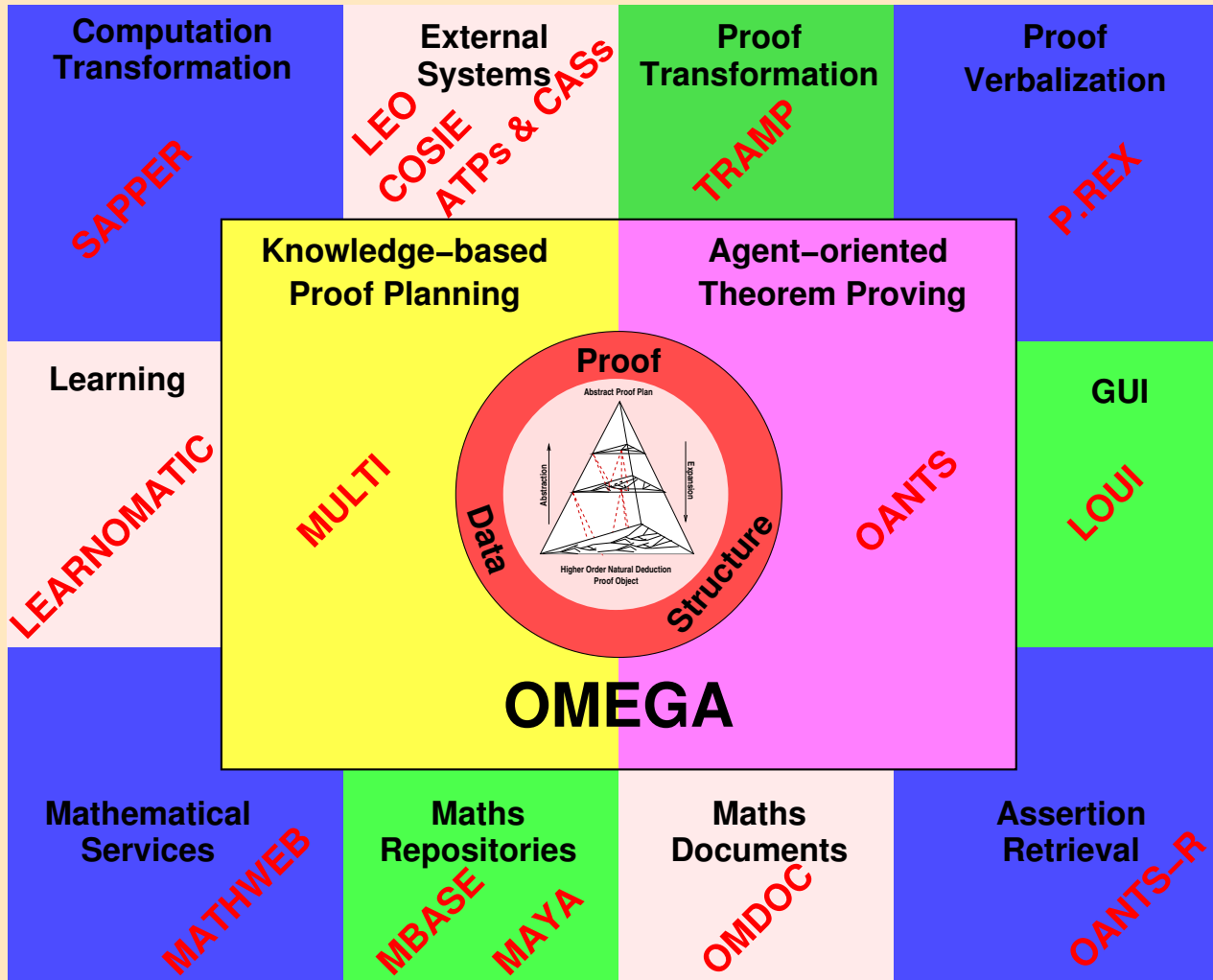


# Architecture of the new OMEGA system

# Old $\Omega$ MEGA

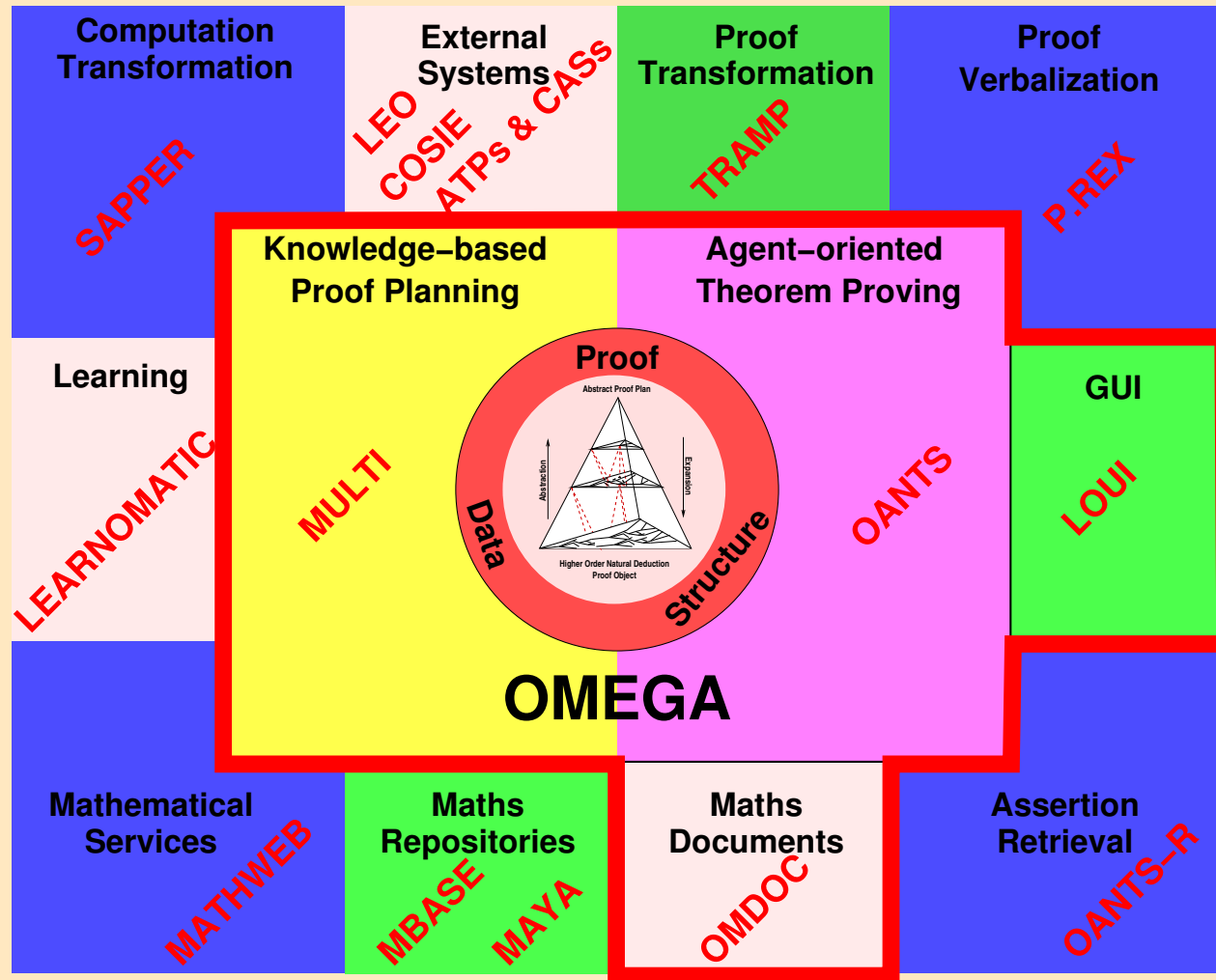


# Old $\Omega$ MEGA



- Base Calculus: Natural Deduction
- Hierarchical proof datastructure (PDS) tailored to
  - ▶ base ND calculus
  - ▶ proof planning methods
  - ▶ tactics
  - ▶ Heavily overcrowded
- Graphical user interface *LΩUI*

# Old $\Omega$ MEGA



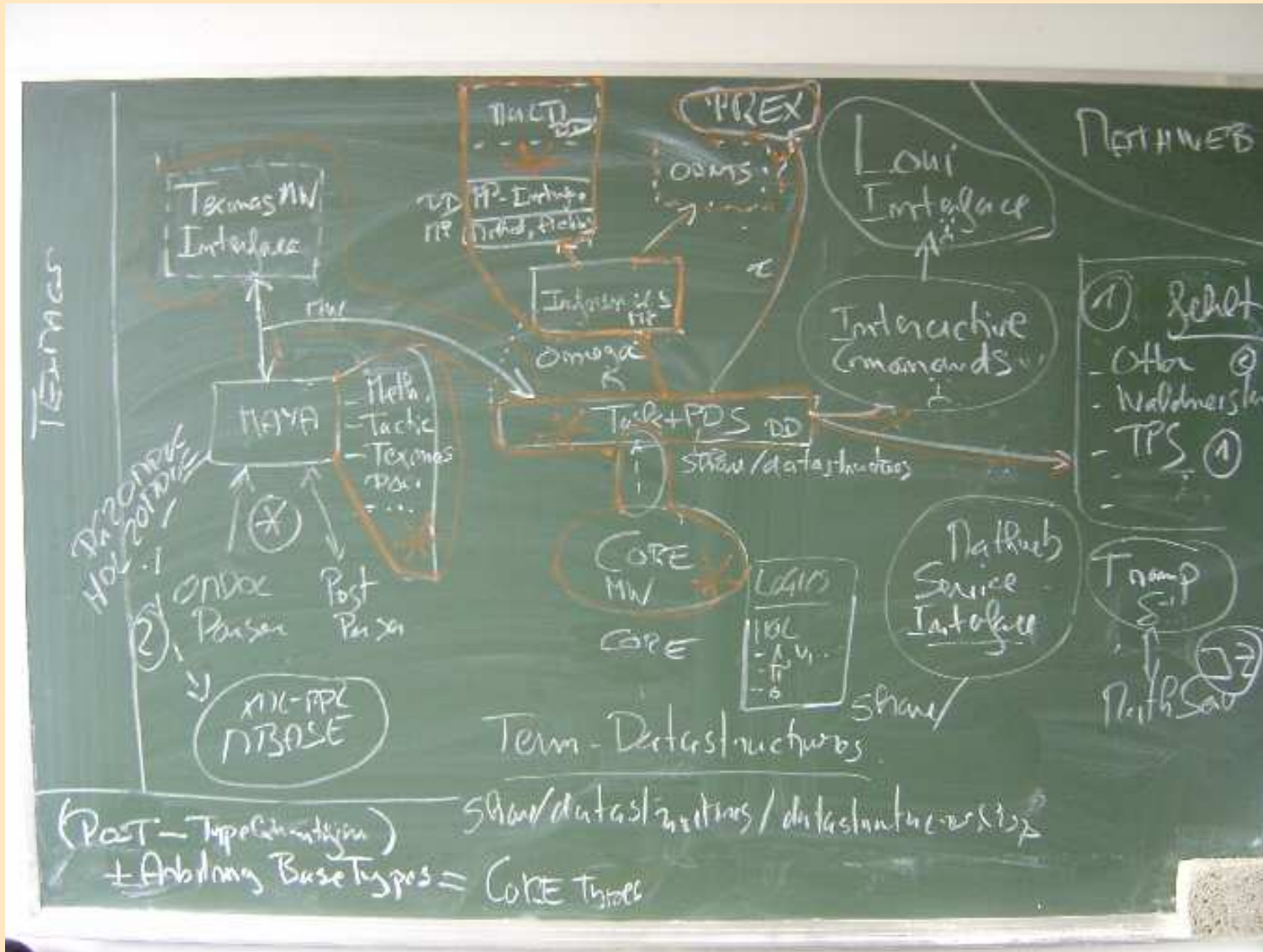
- Base Calculus: Natural Deduction
- Hierarchical proof datastructure (PDS) tailored to
  - ▶ base ND calculus
  - ▶ proof planning methods
  - ▶ tactics
  - ▶ Heavily overcrowded
- Graphical user interface *LΩUI*

# Why and What did we Change?

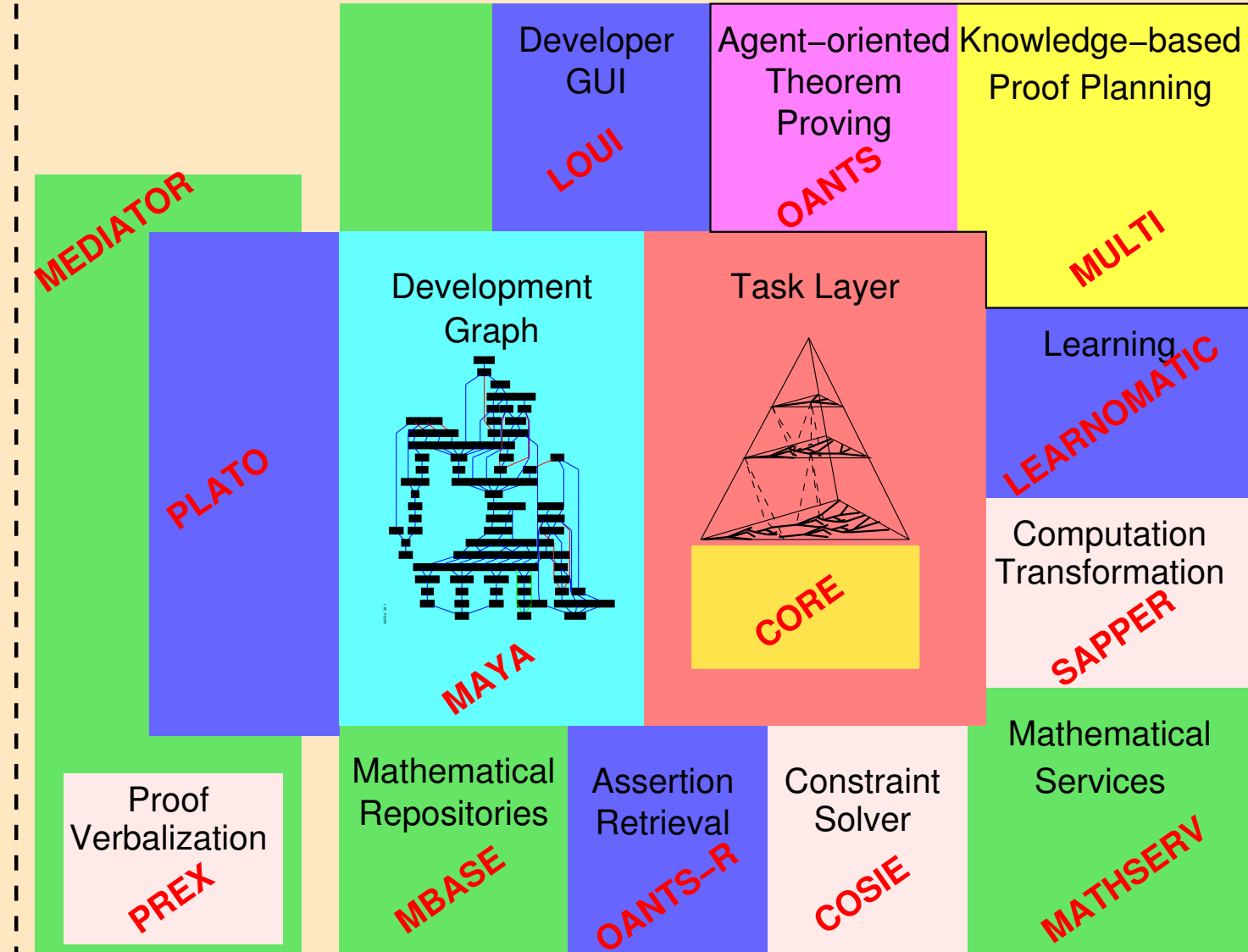
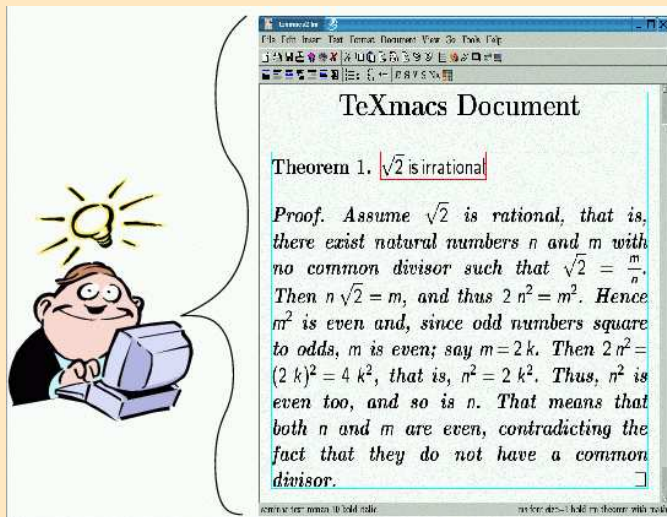
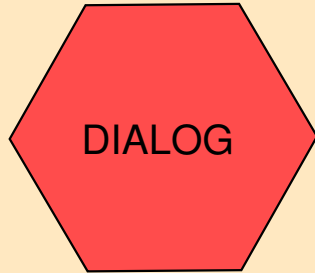


- Natural Deduction Calculus too cumbersome (not natural enough)
  - ▶ Want direct reasoning with the assertions (interactive & automatic)
  - ▶ Natural Deduction not flexible enough
- The proof datastructure was completely overcrowded
  - ▶ New generic proof datastructure
  - ▶ Definition of the task-layer as uniform reasoning platform
- GUI *L $\Omega$ UI* nice, but did not meet requirements of targeted users, say mathematicians
  - ▶ Plug  $\Omega$ MEGA into tools anyway used by users, e.g. TEXMACS
  - ▶  $\Omega$ MEGA as reasoning service provider (analogy: grammar checkers)

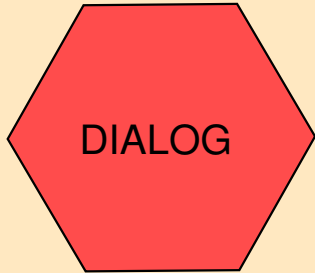
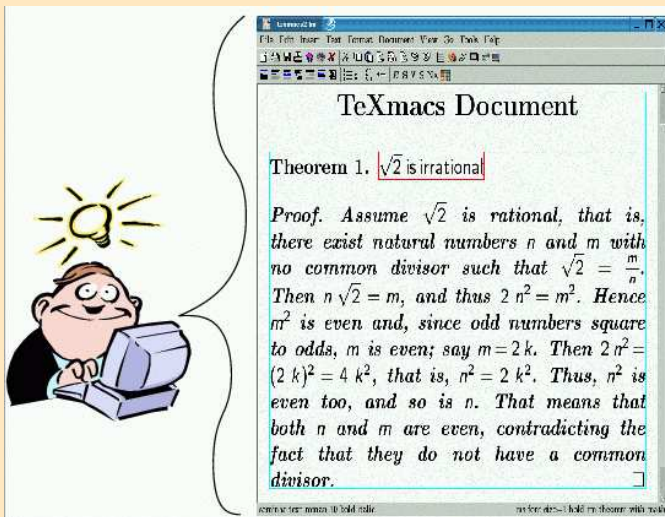
# Towards the New $\Omega$ MEGA



# New Architecture



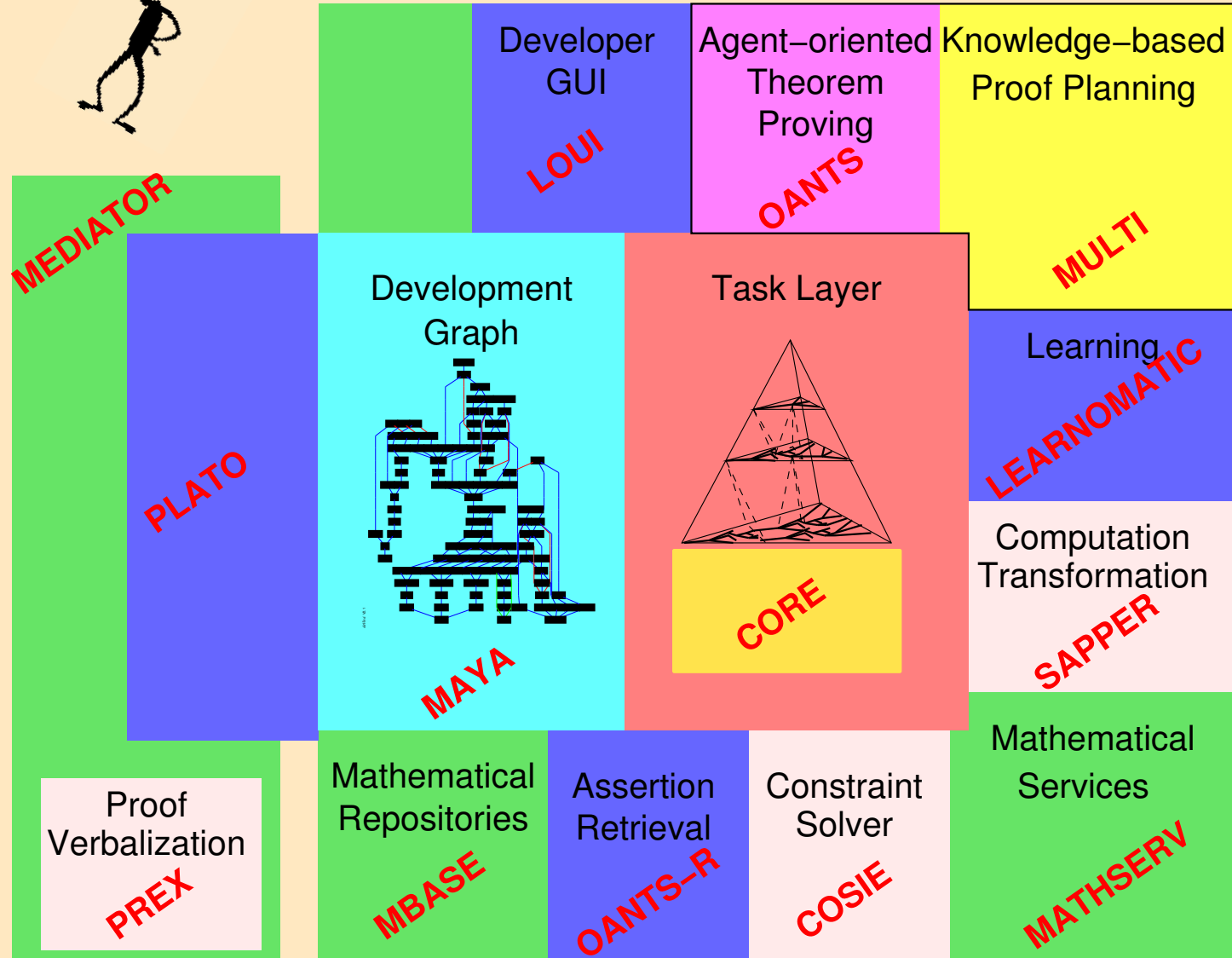
# New Architecture

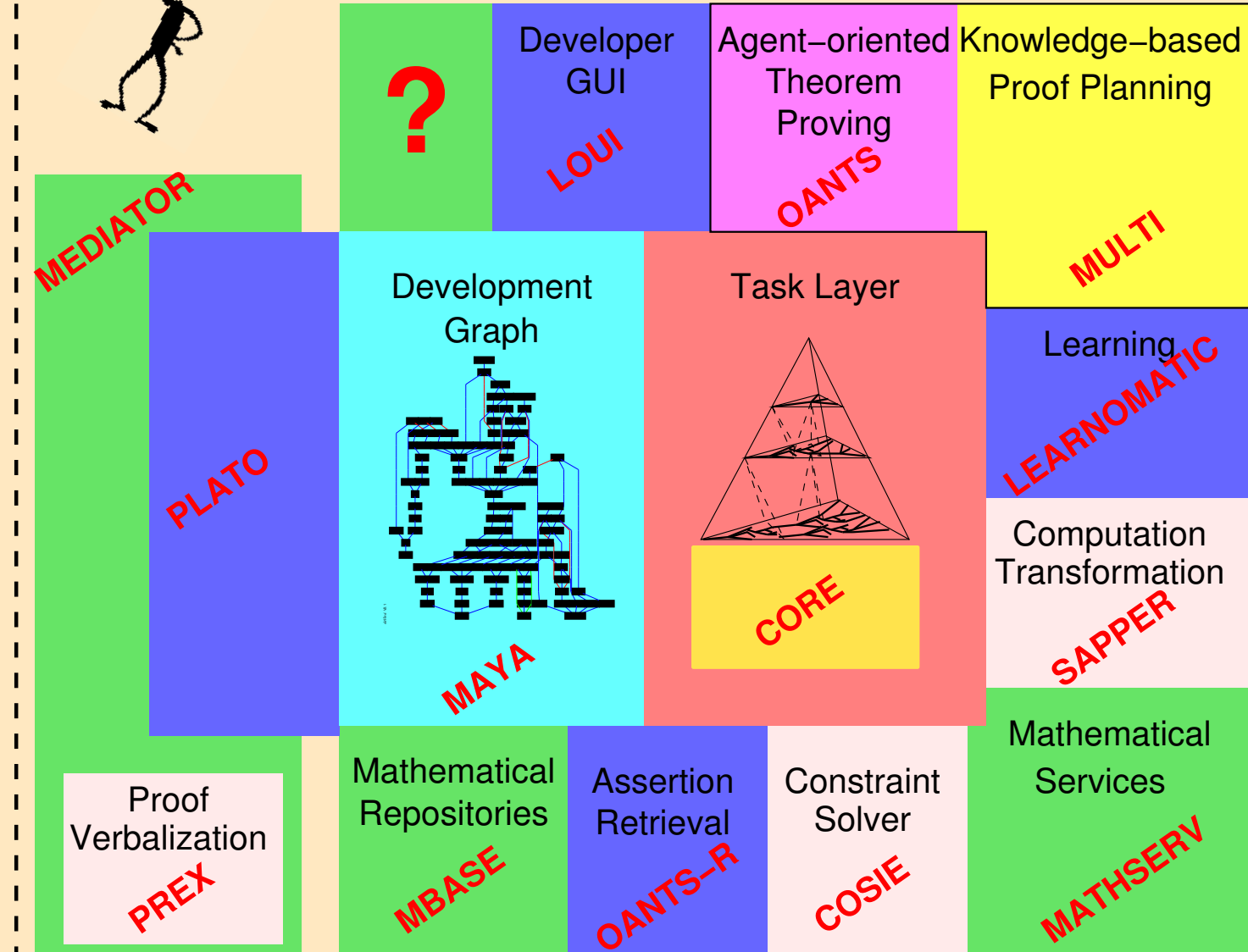
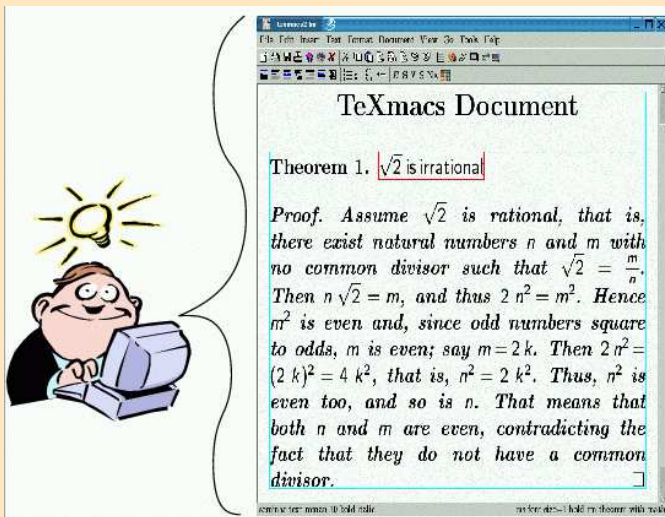
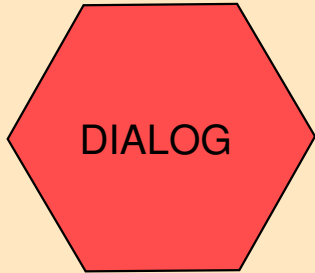
**TeXmacs Document**

Theorem 1.  $\sqrt{2}$  is irrational

*Proof.* Assume  $\sqrt{2}$  is rational, that is, there exist natural numbers  $n$  and  $m$  with no common divisor such that  $\sqrt{2} = \frac{m}{n}$ . Then  $n\sqrt{2} = m$ , and thus  $2n^2 = m^2$ . Hence  $m^2$  is even and, since odd numbers square to odds,  $m$  is even; say  $m = 2k$ . Then  $2n^2 = (2k)^2 = 4k^2$ , that is,  $n^2 = 2k^2$ . Thus,  $n^2$  is even too, and so is  $n$ . That means that both  $n$  and  $m$  are even, contradicting the fact that they do not have a common divisor.  $\square$



# New Architecture



# Talks



Chad Brown (4)

Dominik Dietrich (1)

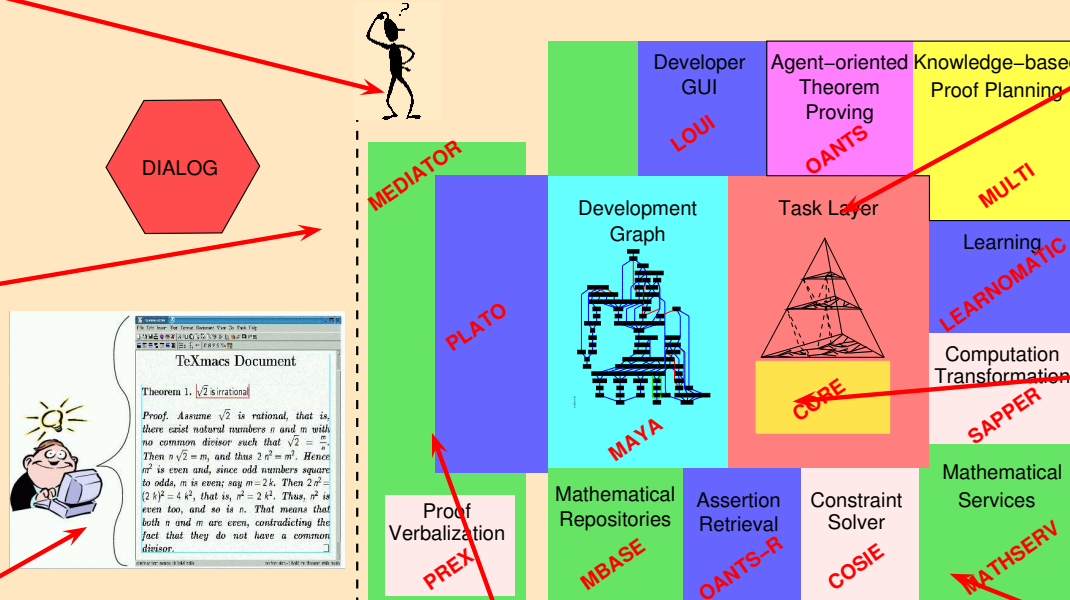
Armin Fiedler (5)

Me

Henri Lesourd (6)

Marc Wagner (2)

Jürgen Zimmer (3)





# The CoRE Calculus

# Context



- Interactive proof assistant system  $\Omega$ MEGA
- Human user assisted by ATPs (including PP)
- Communication between the user and the theorem proving system is crucial
  - (close to CML, “mathural” proof development style)
- ... Especially if human user is not expert in formal logics
- ... How do they do proofs?

# A Textbook Proof

**Theorem.** For any natural numbers  $n$  it holds  $\sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2$ .

**Proof.**

The proof is by induction over  $n$ . For  $n = 0$  the statements holds trivially by definitions. In the induction step by definition we have

$$(n + 1)^3 + \sum_{i=1}^n i^3 = ((n + 1) + \sum_{i=1}^n i)^2$$

$((n + 1) + \sum_{i=1}^n i)^2$  is equivalent  $(n + 1)^2 + 2(n + 1) \sum_{i=1}^n i + (\sum_{i=1}^n i)^2$ , and by induction hypothesis we obtain  $(n + 1)^3 = (n + 1)^2 + 2(n + 1) \sum_{i=1}^n i$ .

...

# Sequent Calculus Proof



$$\begin{array}{c}
 \vdots \\
 \hline
 n = n' + 1 \wedge \sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2 \vdash \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \Rightarrow_R \\
 \vdash (n = n' + 1 \wedge \sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2) \\
 \Rightarrow \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 \vdash \forall n, n'. (n = n' + 1 \wedge \sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2) \quad \forall_R \\
 \Rightarrow \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 \vdash \forall n, n'. (n = n' + 1 \wedge \sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2) \quad \text{Weak} \\
 \Rightarrow \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2, \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 (\forall n, n'. (n = n' + 1 \wedge \sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2)) \\
 \Rightarrow \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \Rightarrow \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \vdash \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 (\forall n. n = 0 \Rightarrow \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2) \\
 \Rightarrow ( (\forall n, n'. (n = n' + 1 \wedge \sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2)) ) \\
 \Rightarrow \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \vdash \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 \forall P. ( \forall n. n = 0 \Rightarrow P(n) ) \\
 \Rightarrow ( (\forall n, n'. (n = n' + 1 \wedge P(n'))) \Rightarrow \forall n. P(n) ) \vdash \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2
 \end{array}$$

$$\begin{array}{c}
 \vdots \\
 \hline
 n = 0 \vdash \sum_{i=1}^0 i^3 = (\sum_{i=1}^0 i)^2, \\
 \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 n = 0 \vdash \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2, \quad \text{Rew} \\
 \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \quad \Rightarrow_R \\
 \vdash n = 0 \Rightarrow \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2, \\
 \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 \vdash \forall n. n = 0 \Rightarrow \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2, \\
 \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \quad \forall_R \\
 \hline
 \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 \Rightarrow_L
 \end{array}$$

# Analysis

- Human users perform proofs by applying definitions, lemmas, theorems (*assertions*), information contained in the formula to (sub-)formulas
- [Huang94] introduced the *assertion level* as basis for NL proof presentation
- **Example:**  $\forall S_1, S_2 : Set. S_1 \subseteq S_2 \Leftrightarrow \forall x. x \in S_1 \Rightarrow x \in S_2.$

This assertion allows us to derive

1.  $a \in S'_2$  from  $a \in S'_1$  and  $S'_1 \subseteq S'_2$
  2.  $S'_1 \not\subseteq S'_2$  from  $a \in S'_1$  and  $a \notin S'_2$
  3.  $\forall x. x \in S'_1 \Rightarrow x \in S'_2$  from  $S'_1 \subseteq S'_2$
- **Drawbacks:**
    - ▶ Proof search is not at the assertion level
    - ▶ No active support what *rules* follow from an assertion

# Goals & Requirements



- Have a calculus where assertion application is a primitive rule
- Requires:
  - ▶ Means to efficiently obtain possible rules from an assertion
  - ▶ Include information contained in subformulas
  - ▶ Means to apply these rules on subformulas
- Build a calculus around a primitive assertion application rule

# Idea

To prove  $\forall n. \sum_{i=1}^n i^3 = \left(\sum_{i=1}^n i\right)^2$

(0) we apply  $\forall P. (\forall n. n = 0 \Rightarrow P(n)) \Rightarrow ((\forall n, n'. (n = n' + 1 \wedge P(n')) \Rightarrow P(n)) \Rightarrow \forall n. P(n))$



to obtain  $\forall n. n = 0 \Rightarrow \sum_{i=1}^n i^3 = \left(\sum_{i=1}^n i\right)^2 \wedge$   
 $\forall n, n'. n = n' + 1 \wedge \sum_{i=1}^{n'} i^3 = \left(\sum_{i=1}^{n'} i\right)^2 \Rightarrow \sum_{i=1}^n i^3 = \left(\sum_{i=1}^n i\right)^2$

(0) then apply  $n = 0$

to obtain  $\forall n. n = 0 \Rightarrow \sum_{i=1}^0 i^3 = \left(\sum_{i=1}^0 i\right)^2 \wedge$   
 $\forall n, n'. n = n' + 1 \wedge \sum_{i=1}^{n'} i^3 = \left(\sum_{i=1}^{n'} i\right)^2 \Rightarrow \sum_{i=1}^n i^3 = \left(\sum_{i=1}^n i\right)^2$

Everything we need is in the formulas,  
 but how to make that information explicit?



# Sequent Calculus Proof

Let  $Prop(n) := \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2$

$\frac{}{\vdash \forall n, n'. (n = n' + 1 \wedge Prop(n'))} \text{Ax}$ $\frac{}{\vdash \forall n. Prop(n)} \text{Ax}$ $\Rightarrow \forall n. Prop(n), \forall n. Prop(n)$	$\frac{n = 0 \vdash Prop(0), \forall n. Prop(n)}{n = 0 \vdash Prop(n), \forall n. Prop(n)} \text{Rew}$ $\frac{}{\vdash n = 0 \Rightarrow Prop(n), \forall n. Prop(n)} \Rightarrow_R \alpha$ $\frac{}{\vdash \forall n. n = 0 \Rightarrow Prop(n), \forall n. Prop(n)} \forall_R \delta$
$\frac{(\forall n, n'. (n = n' + 1 \wedge Prop(n')))}{\Rightarrow \forall n. Prop(n)}$ $\frac{}{\vdash \forall n. Prop(n)}$	$\frac{}{\vdash \forall n. n = 0 \Rightarrow Prop(n), \forall n. Prop(n)} \Rightarrow_L \beta$
$\frac{(\forall n. n = 0 \Rightarrow Prop(n)) \Rightarrow (\forall n, n'. (n = n' + 1 \wedge Prop(n')))}{\Rightarrow \forall n. Prop(n) \vdash \forall n. Prop(n)}$	
$\frac{\forall P. (\forall n. n = 0 \Rightarrow P(n))}{\Rightarrow (\forall n, n'. (n = n' + 1 \wedge P(n'))) \Rightarrow \forall n. P(n)} \forall_L P \leftarrow Prop \gamma$	



# Sequent Calculus Proof

Let  $Prop(n) := \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2$

$\frac{}{\vdash \forall n, n'. (n = n' + 1 \wedge Prop(n'))} \text{Ax}$ $\frac{}{\vdash \forall n. Prop(n)} \text{Ax}$ $\frac{}{\Rightarrow \forall n. Prop(n), \forall n. Prop(n)} \text{Ax}$ <hr/> $\frac{}{\vdash \forall n, n'. (n = n' + 1 \wedge Prop(n'))} \text{Ax}$ $\frac{}{\Rightarrow \forall n. Prop(n)} \text{Ax}$ $\frac{}{\vdash \forall n. Prop(n)} \text{Ax}$ <hr/> $\frac{}{\vdash \forall n. n = 0 \Rightarrow Prop(n), \forall n. Prop(n)} \text{Ax}$ <hr/> $\frac{}{\forall P. (\forall n. n = 0 \Rightarrow P(n)) \Rightarrow (\forall n, n'. (n = n' + 1 \wedge P(n')))} \text{Ax}$ $\frac{}{\Rightarrow \forall n. Prop(n), \forall n. Prop(n)} \text{Ax}$ <hr/> $\frac{}{\forall P. (\forall n. n = 0 \Rightarrow P(n)) \Rightarrow (\forall n, n'. (n = n' + 1 \wedge P(n')))} \text{Ax}$ $\frac{}{\Rightarrow \forall n. Prop(n), \forall n. Prop(n)} \text{Ax}$	$\frac{}{\vdash n = 0 \Rightarrow Prop(n), \forall n. Prop(n)} \Rightarrow_R \alpha$ $\frac{}{\vdash n = 0 \Rightarrow Prop(n), \forall n. Prop(n)} \Rightarrow_R \delta$ <hr/> $\frac{}{\vdash \forall n. n = 0 \Rightarrow Prop(n), \forall n. Prop(n)} \Rightarrow_L \beta$ <hr/> $\frac{}{\forall_L P \leftarrow Prop \gamma}$
--	--



# Sequent Calculus Proof

Let  $Prop(n) := \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2$

$\frac{}{\forall n. Prop(n)} \text{Ax}$ $\frac{\vdash \forall n, n'. (n = n' + 1 \wedge Prop(n')) \quad \vdash \forall n. Prop(n)}{\Rightarrow \forall n. Prop(n), \forall n. Prop(n)}$	$\frac{n = 0 \vdash Prop(0), \forall n. Prop(n)}{n = 0 \vdash Prop(n), \forall n. Prop(n)} \text{Rew}$ $\frac{}{\vdash n = 0 \Rightarrow Prop(n), \forall n. Prop(n)} \Rightarrow_R \alpha$
$\frac{(\forall n, n'. (n = n' + 1 \wedge Prop(n')) \Rightarrow \forall n. Prop(n)) \quad \vdash \forall n. Prop(n)}{\Rightarrow \forall n. Prop(n), \forall n. Prop(n)} \Rightarrow_L \beta$	$\frac{}{\vdash \forall n. n = 0 \Rightarrow Prop(n), \forall n. Prop(n)} \forall_R \delta$
$\frac{}{(\forall n. n = 0 \Rightarrow Prop(n)) \Rightarrow (\forall n, n'. (n = n' + 1 \wedge Prop(n')) \Rightarrow \forall n. Prop(n)) \quad \vdash \forall n. Prop(n)} \Rightarrow_L \beta$	
$\frac{}{\forall P. (\forall n. n = 0 \Rightarrow P(n)) \Rightarrow ((\forall n, n'. (n = n' + 1 \wedge P(n'))) \Rightarrow \forall n. P(n)) \quad \vdash \forall n. Prop(n)} \forall_L P \leftarrow Prop \gamma$	



# Sequent Calculus Proof

Let  $Prop(n) := \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2$

$\frac{}{\vdash \forall n, n'. (n = n' + 1 \wedge Prop(n'))} \text{Ax}$ $\frac{}{\vdash \forall n. Prop(n)} \text{Ax}$ $\frac{}{\Rightarrow \forall n. Prop(n), \forall n. Prop(n)} \text{Ax}$	$\frac{n = 0 \vdash Prop(0), \forall n. Prop(n)}{} \text{Rew}$ $\frac{n = 0 \vdash Prop(n), \forall n. Prop(n)}{} \Rightarrow_R \alpha$ $\frac{\vdash n = 0 \Rightarrow Prop(n), \forall n. Prop(n)}{} \forall_R \delta$
$\frac{}{\vdash \forall n, n'. (n = n' + 1 \wedge Prop(n'))} \Rightarrow_L \beta$	$\frac{}{\vdash \forall n. n = 0 \Rightarrow Prop(n), \forall n. Prop(n)} \Rightarrow_L \beta$
$\frac{}{\vdash \forall n. Prop(n)}$	$\frac{}{\Rightarrow_L \beta}$
$\frac{}{\forall P. (\forall n. n = 0 \Rightarrow P(n)) \Rightarrow (\forall n, n'. (n = n' + 1 \wedge P(n')))} \Rightarrow \forall n. Prop(n) \vdash \forall n. Prop(n)$	$\frac{}{\forall_L P \leftarrow Prop \gamma}$
$\frac{}{\forall P. (\forall n. n = 0 \Rightarrow P(n)) \Rightarrow (\forall n, n'. (n = n' + 1 \wedge P(n')))} \Rightarrow \forall n. P(n) \vdash \forall n. Prop(n)$	$\frac{}{\forall_L P \leftarrow Prop \gamma}$

# What Formulas can tell us...

- Formulas are not just a string of symbols. . .





# What Formulas can tell us...

- Formulas are not just a string of symbols. . .
- Formulas are trees, subformulas occur negatively or positively

# What Formulas can tell us...

- Formulas are not just a string of symbols. . .
- Formulas are trees, subformulas occur negatively or positively
- The occurring connectives have a close correspondence to structural properties of the SK proof

# What Formulas can tell us...

- Formulas are not just a string of symbols. . .
- Formulas are trees, subformulas occur negatively or positively
- The occurring connectives have a close correspondence to structural properties of the SK proof
  - ▶ positive  $\wedge$  and negative  $\vee$  to splits in the SK proof  
 $\implies$  their subformulas occur in different sequents  
different logical context

# What Formulas can tell us...

- Formulas are not just a string of symbols. . .
- Formulas are trees, subformulas occur negatively or positively
- The occurring connectives have a close correspondence to structural properties of the SK proof
  - ▶ positive  $\wedge$  and negative  $\vee$  to splits in the SK proof  
 $\implies$  their subformulas occur in different sequents  
different logical context
  - ▶ negative  $\wedge$  and positive  $\vee$  don't  
 $\implies$  their subformulas occur in the same sequent  
same logical context

# What Formulas can tell us...

- Formulas are not just a string of symbols. . .
- Formulas are trees, subformulas occur negatively or positively
- The occurring connectives have a close correspondence to structural properties of the SK proof
  - ▶ positive  $\wedge$  and negative  $\vee$  to splits in the SK proof  
 $\implies$  their subformulas occur in different sequents  
different logical context
  - ▶ negative  $\wedge$  and positive  $\vee$  don't  
 $\implies$  their subformulas occur in the same sequent  
same logical context
  - ▶ negative  $\exists$  quantifiers and positive  $\forall$  quantifiers have an *Eigenvariable* condition

# What Formulas can tell us...

- Formulas are not just a string of symbols. . .
- Formulas are trees, subformulas occur negatively or positively
- The occurring connectives have a close correspondence to structural properties of the SK proof
  - ▶ positive  $\wedge$  and negative  $\vee$  to splits in the SK proof  
 $\implies$  their subformulas occur in different sequents  
different logical context
  - ▶ negative  $\wedge$  and positive  $\vee$  don't  
 $\implies$  their subformulas occur in the same sequent  
same logical context
  - ▶ negative  $\exists$  quantifiers and positive  $\forall$  quantifiers have an *Eigenvariable* condition
  - ▶ positive  $\exists$  quantifiers and negative  $\forall$  quantifiers can be freely instantiated

# What tell us...

Logic of symbols...

Quantifiers occur negatively or positively

There is a close correspondence to structural



Polarities

Uniform Types

[Smullyan68, Wallen90]

- ▶ positive  $\wedge$  and negative  $\vee$  to splits in the SK proof  
 $\implies$  their subformulas occur in different sequents  
different logical context
- ▶ negative  $\wedge$  and positive  $\vee$  don't  
 $\implies$  their subformulas occur in the same sequent  
same logical context
- ▶ negative  $\exists$  quantifiers and positive  $\forall$  quantifiers have an *Eigenvariable* condition
- ▶ positive  $\exists$  quantifiers and negative  $\forall$  quantifiers can be freely instantiated

# Polarities & Uniform Types



- Assign polarities to formulas  $\underbrace{\varphi_1, \dots, \varphi_n}_- \vdash \underbrace{\psi_1, \dots, \psi_m}_+$  Signed formulas  $\varphi^-, \psi^+$
- Categorize signed formulas by uniform types

$\alpha$	$\alpha_0$	$\alpha_1$
$(\varphi \vee \psi)^+$	$\varphi^+$	$\psi^+$
$(\varphi \Rightarrow \psi)^+$	$\varphi^-$	$\psi^+$
$(\varphi \wedge \psi)^-$	$\varphi^-$	$\psi^-$
$(\neg\varphi)^+$	$\varphi^-$	—
$(\neg\varphi)^-$	$\varphi^+$	—

$\beta$	$\beta_0$	$\beta_1$
$(\varphi \wedge \psi)^+$	$\varphi^+$	$\psi^+$
$(\varphi \vee \psi)^-$	$\varphi^-$	$\psi^-$
$(\varphi \Rightarrow \psi)^-$	$\varphi^+$	$\psi^-$

$\gamma$	$\gamma_0(c)$
$(\forall x.\varphi)^-$	$(\varphi[x/t])^-$
$(\exists x.\varphi)^+$	$(\varphi[x/t])^+$

$\delta$	$\delta_0(c)$
$(\forall x.\varphi)^+$	$(\varphi[x/c])^+$
$(\exists x.\varphi)^-$	$(\varphi[x/c])^-$

# Signed Formulas

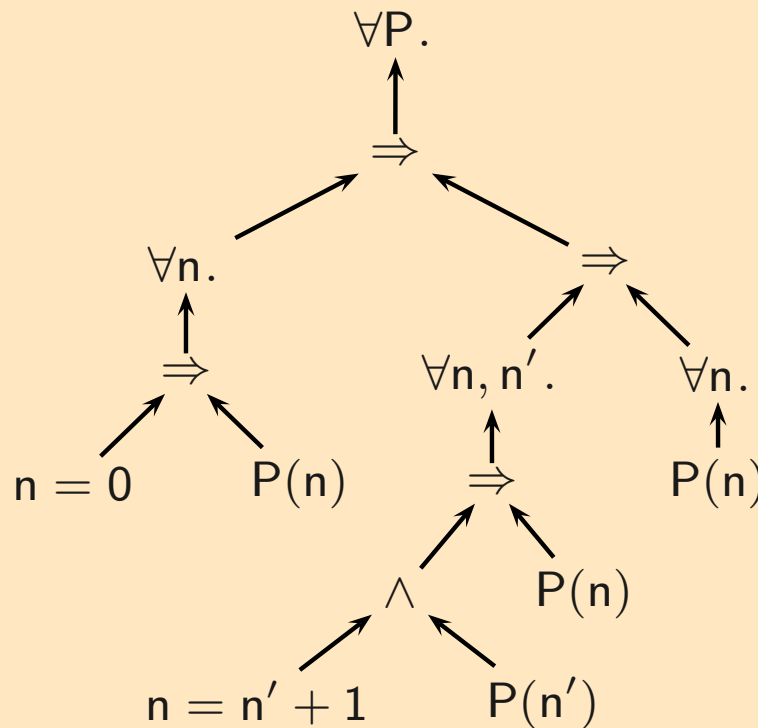


$$\forall P. (\forall n. n = 0 \Rightarrow P(n)) \Rightarrow ((\forall n, n'. (n = n' + 1 \wedge P(n')) \Rightarrow P(n)) \Rightarrow \forall n. P(n)) \vdash \forall n. \sum_{i=1}^n i^3 = \left( \sum_{i=1}^n i \right)^2$$

# Signed Formulas



$$\forall P. (\forall n. n = 0 \Rightarrow P(n)) \Rightarrow ((\forall n, n'. (n = n' + 1 \wedge P(n')) \Rightarrow P(n)) \Rightarrow \forall n. P(n)) \vdash \forall n. \sum_{i=1}^n i^3 = \left(\sum_{i=1}^n i\right)^2$$

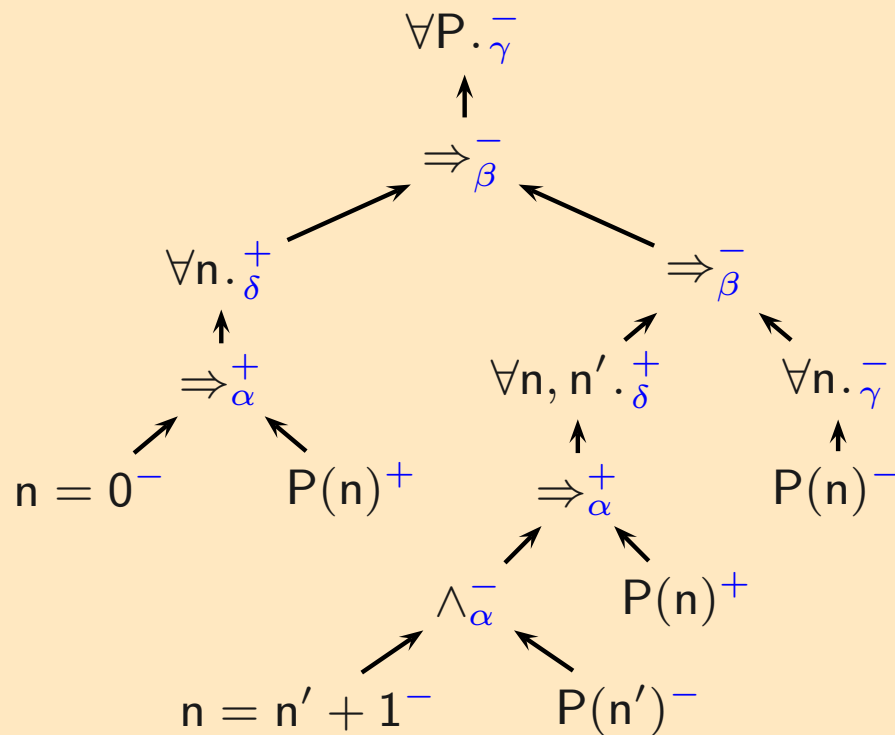


$$\forall n. \sum_{i=1}^n i^3 = \left(\sum_{i=1}^n i\right)^2$$

# Signed Formulas



$$\forall P. (\forall n. n = 0 \Rightarrow P(n)) \Rightarrow ((\forall n, n'. (n = n' + 1 \wedge P(n')) \Rightarrow P(n)) \Rightarrow \forall n. P(n)) \vdash \forall n. \sum_{i=1}^n i^3 = \left(\sum_{i=1}^n i\right)^2$$

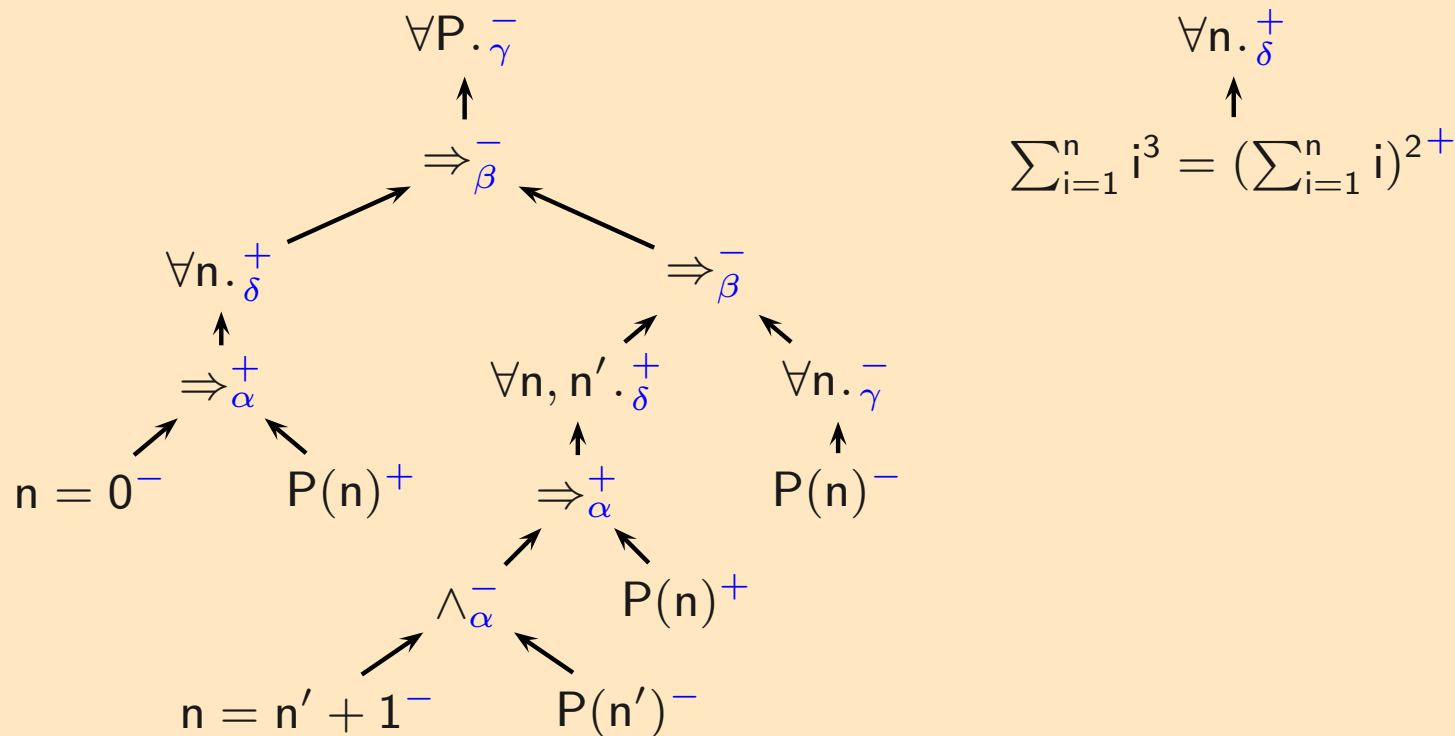


$$\forall n. \delta^+ \quad \uparrow \quad \sum_{i=1}^n i^3 = \left(\sum_{i=1}^n i\right)^2$$

# Signed Formulas



$$\forall P. (\forall n. n = 0 \Rightarrow P(n)) \Rightarrow ((\forall n, n'. (n = n' + 1 \wedge P(n')) \Rightarrow P(n)) \Rightarrow \forall n. P(n)) \vdash \forall n. \sum_{i=1}^n i^3 = \left(\sum_{i=1}^n i\right)^2$$



Replacement rule for  $\forall n. \sum_{i=1}^n i^3 = \left(\sum_{i=1}^n i\right)^2$ :

$$\forall n. P(n)^- \rightarrow \left\langle \forall n. n = 0 \Rightarrow P(n)^+, \quad \forall n, n'. (n = n' + 1 \wedge P(n')) \Rightarrow P(n)^+ \right\rangle$$

$\Rightarrow$  Fix left-hand side  $\forall n. P(n)^-$  and collect **all**  $\beta$ -related formulas

# Application...



By substitution  $[P \leftarrow \lambda x. \sum_{i=1}^x i^3 = (\sum_{i=1}^x i)^2]$  application of

$$\forall n. P(n)^- \rightarrow \left\langle \begin{array}{l} \forall n. n = 0 \Rightarrow P(n)^+, \\ \forall n, n'. (n = n' + 1 \wedge P(n')) \Rightarrow P(n)^+ \end{array} \right\rangle$$

to  $\forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^{2+}$  yields

$$\forall n. n = 0 \Rightarrow \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \wedge$$

$$\forall n, n'. n = n' + 1 \wedge \sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2 \Rightarrow \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2$$

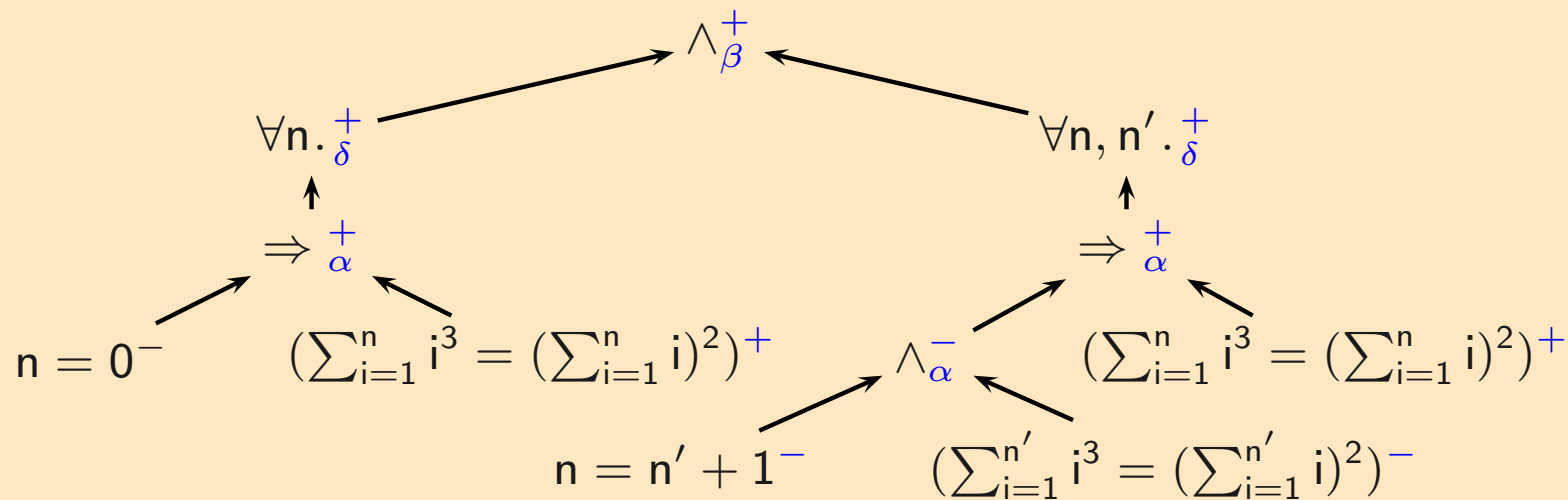
Now: How to enable application of  $n = 0$ ?

# Signed Formulas



$$((\forall n. (n = 0^- \Rightarrow (\sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2)^+)_\alpha^+)_\delta^+ \wedge$$

$$(\forall n, n'. ((n = n' + 1 \wedge (\sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2)^-)_\alpha^- \Rightarrow (\sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2)^+)_\delta^+)_\beta^+$$

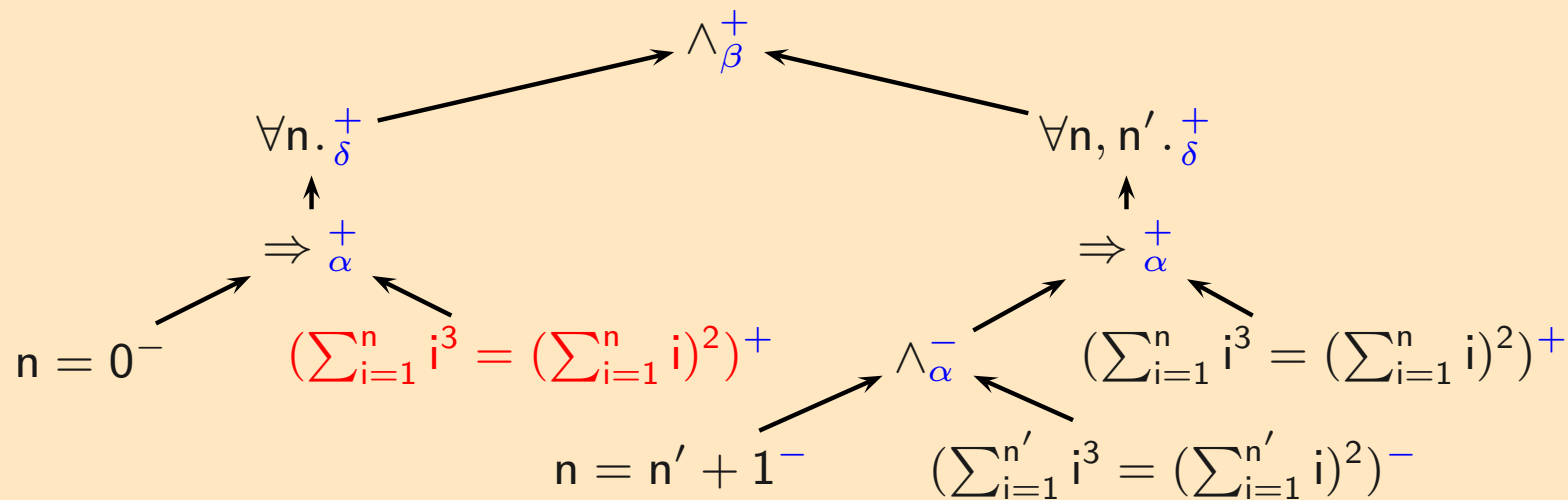


# Signed Formulas



$$((\forall n. (n = 0^- \Rightarrow (\sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2)^+)_\alpha^+)_\delta^+ \wedge$$

$$(\forall n, n'. ((n = n' + 1 \wedge (\sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2)^-)_\alpha^- \Rightarrow (\sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2)^+)_\delta^+)_\beta^+$$



- Formulas connected by  $\alpha$ -type node are in the same context

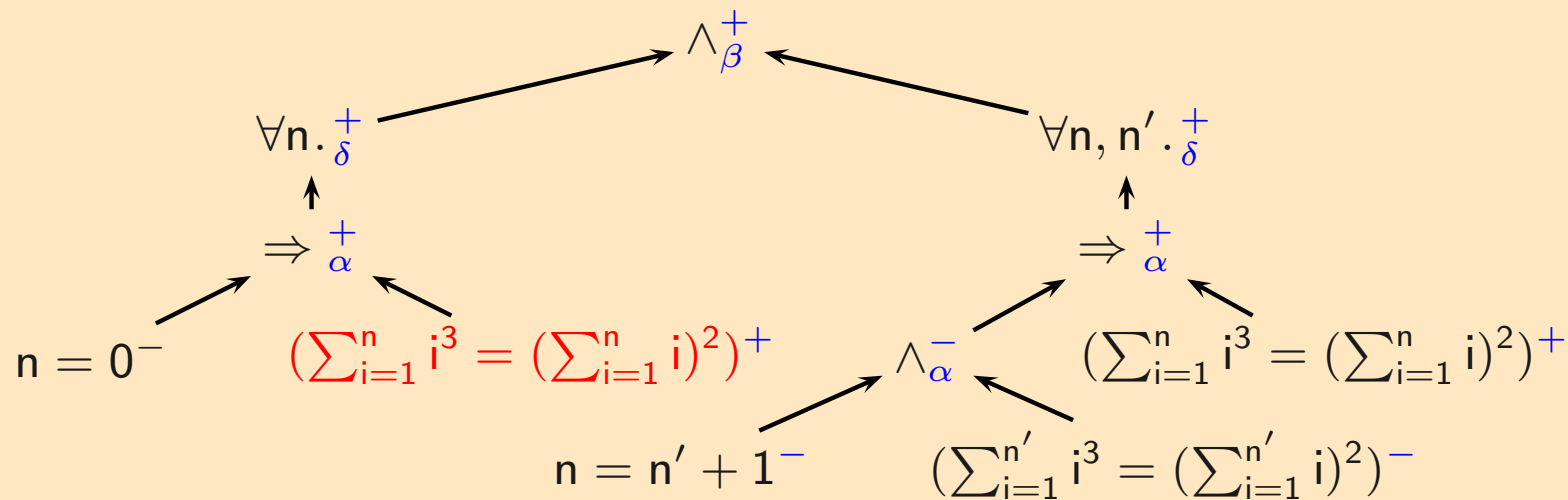
$\implies$  Uniform and static notion of logical context of subformulas!

# Signed Formulas



$$((\forall n. (n = 0^- \Rightarrow (\sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2)^+)_\alpha^+)_\delta^+ \wedge$$

$$(\forall n, n'. ((n = n' + 1 \wedge (\sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2)^-)_\alpha^- \Rightarrow (\sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2)^+)_\delta^+)_\beta^+$$



- Formulas connected by  $\alpha$ -type node are in the same context

$\implies$  Uniform and static notion of logical context of subformulas!

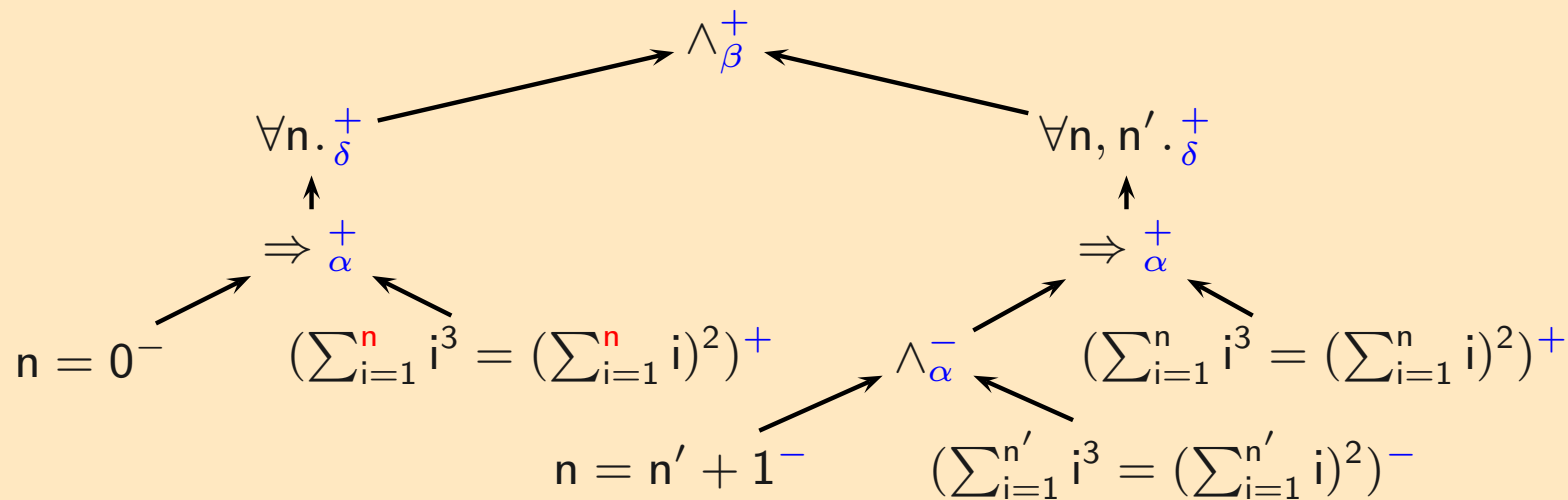
- Derive replacement rules from context

Example:  $n \rightarrow 0$  for  $(\sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2)^+$

# Signed Formulas

$$((\forall n. (n = 0^- \Rightarrow (\sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2)^+)_\alpha^+)_\delta^+ \wedge$$

$$(\forall n, n'. ((n = n' + 1 \wedge (\sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2)^-)_\alpha^- \Rightarrow (\sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2)^+)_\delta^+)_\beta^+$$



- Formulas connected by  $\alpha$ -type node are in the same context

$\implies$  Uniform and static notion of logical context of subformulas!

- Derive replacement rules from context

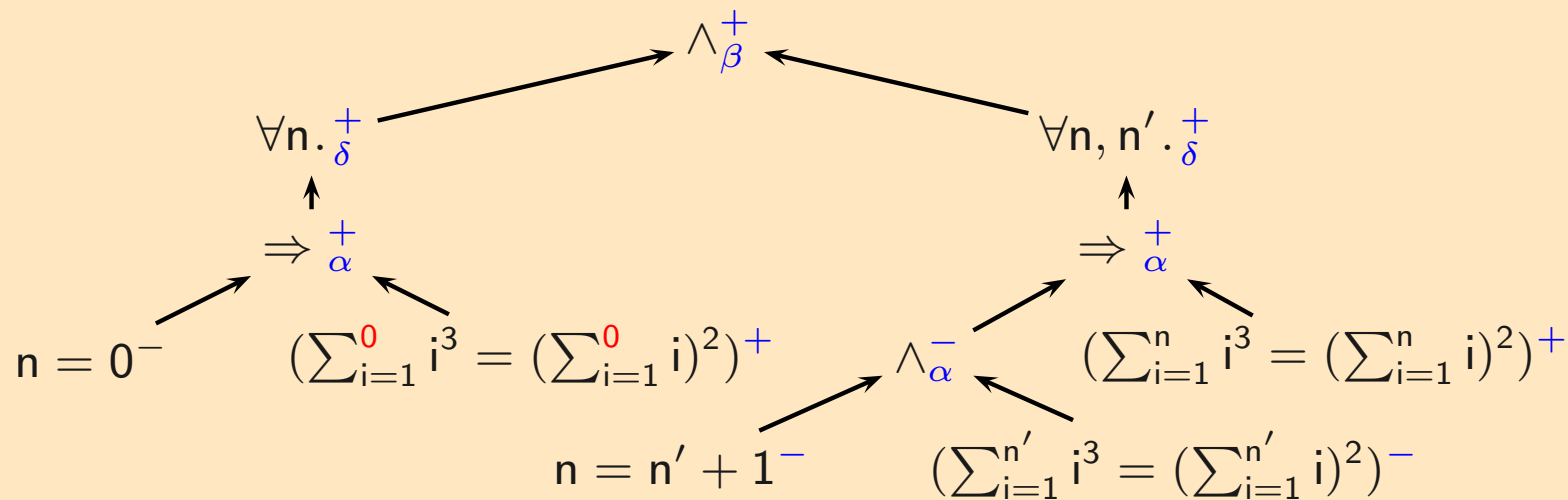
Example:  $n \rightarrow 0$  for  $(\sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2)^+$

# Signed Formulas



$$((\forall n. (n = 0^- \Rightarrow (\sum_{i=1}^0 i^3 = (\sum_{i=1}^0 i)^2)^+)_\alpha^+)_\delta^+ \wedge$$

$$(\forall n, n'. ((n = n' + 1 \wedge (\sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2)^-)_\alpha^- \Rightarrow (\sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2)^+)_\delta^+)_\beta^+$$



- Formulas connected by  $\alpha$ -type node are in the same context

$\implies$  Uniform and static notion of logical context of subformulas!

- Derive replacement rules from context

Example:  $n \rightarrow 0$  for  $(\sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2)^+$

# Turning this into a Calculus



- Signed formulas are close to extensional expansion trees (EET) [[Andrews81](#), [Miller83](#), [Pfenning87](#)] and Wallen's indexed formula trees (IFT) [[Wallen90](#)]
- Uniform integration of IFT and EET
  - ▶ as a starting point and
  - ▶ to deal with quantifiers and admissibility of substitution
- Take a free variable representation of our signed formula
- Define a calculus
  - ▶ starting with primitive rules for assertion application
  - ▶ Further rules to obtain a complete calculus

# Extensional Expansion Trees



- Defined in [\[Pfenning87\]](#), minimal set of logical connectives, all negations pushed to the literals

# Extensional Expansion Trees

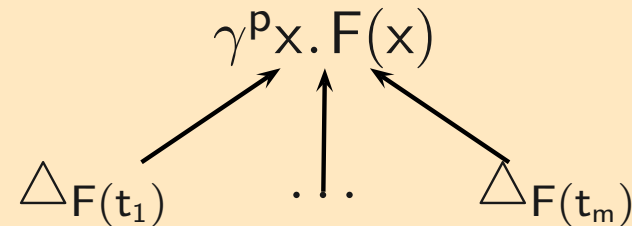


- Defined in [Pfenning87], minimal set of logical connectives, all negations pushed to the literals
- Tree construction follows exactly tree structure of the formula, except. . .

# Extensional Expansion Trees

- Defined in [Pfenning87], minimal set of logical connectives, all negations pushed to the literals
- Tree construction follows exactly tree structure of the formula, except...

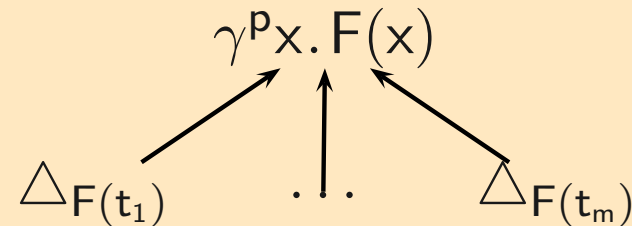
...for existential quantifiers where it allows to have subtrees for arbitrary many instances of the quantifier.



# Extensional Expansion Trees

- Defined in [Pfenning87], minimal set of logical connectives, all negations pushed to the literals
- Tree construction follows exactly tree structure of the formula, except...

...for existential quantifiers where it allows to have subtrees for arbitrary many instances of the quantifier.

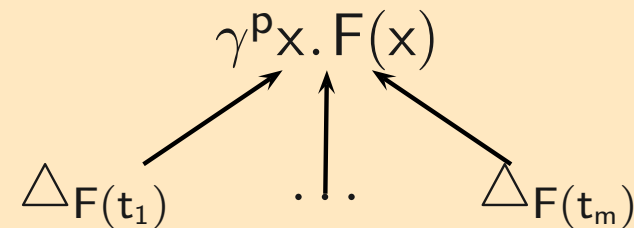


- Rule for functional and Boolean extensionality

# Extensional Expansion Trees

- Defined in [Pfenning87], minimal set of logical connectives, all negations pushed to the literals
- Tree construction follows exactly tree structure of the formula, except...

...for existential quantifiers where it allows to have subtrees for arbitrary many instances of the quantifier.

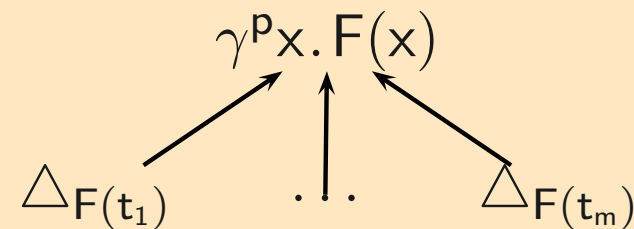


- Rule for functional and Boolean extensionality
- $\Delta_F^\mu$  denotes EET for F and multiplicity  $\mu$

# Extensional Expansion Trees

- Defined in [Pfenning87], minimal set of logical connectives, all negations pushed to the literals
- Tree construction follows exactly tree structure of the formula, except...

...for existential quantifiers where it allows to have subtrees for arbitrary many instances of the quantifier.



- Rule for functional and Boolean extensionality
- $\Delta_F^\mu$  denotes EET for F and multiplicity  $\mu$
- Deep formula**  $DF(\Delta_F^\mu)$ : the formula extracted from  $\Delta_F^\mu$  by omitting all quantifiers.

$$DF \left( \begin{array}{c} \gamma^p x. F(x) \\ \swarrow \quad \uparrow \quad \searrow \\ \Delta_{F(t_1)} \quad \cdot \vdots \cdot \quad \Delta_{F(t_m)} \end{array} \right) := \alpha(DF(\Delta_{F(t_1)}), \dots, DF(\Delta_{F(t_m)}))$$

# Substitutions and Extensions

- $\Delta_F^\mu$ -*admissibility* of substitution (instances of  $\gamma$ -quantifiers)
- An extensional expansion tree  $\Delta_F^\mu$  is an *extensional expansion proof* for F, if, and only if,
  1. its deep formula is a tautology (i.e., all paths through the deep formula are unsatisfiable) and
  2. the associated substitution  $\sigma$  is  $\Delta_F^\mu$ -admissible.
- *Extend* that calculus by
  1. allowing for rich set of logical connectives ( $\Leftrightarrow, =, \top, \perp$ )
  2. use polarities like in Wallen's IFT: overcome restrictions on the position of negations
  3. on the fly increase of the multiplicities  
avoids "guessing" the initial multiplicities and restart. [Tableau05]
- Deep formulas are *quantifier-free (QF) signed formulas*

# Contexts

- *Context in QF-signed formulas*

Two subformulas related by an  $\alpha$ -type connective are in the same logical context

- *Signed formulas obtained by Weakening  $\mathcal{W}(F^p)$*

- ▶  $\mathcal{W}(A \vee (B \wedge C)^-) = \{A \vee (B \wedge C)^-, A \vee B^-, A \vee C^-\}$

- *Conditions of  $L^q$  in  $\varphi(L^q)^p$ :*

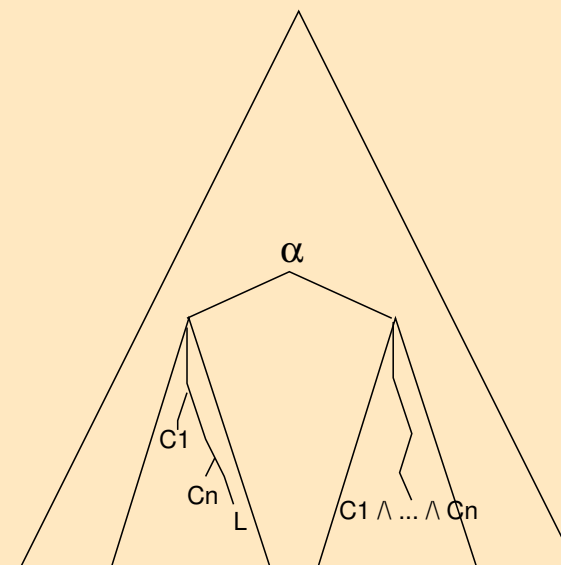
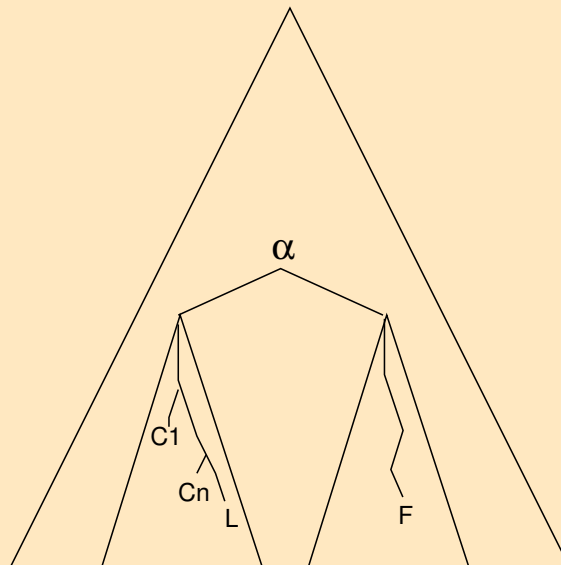
- ▶ If  $G_1^{p_1}, \dots, G_n^{p_n}$  are all maximal signed formulas  $\beta$ -related to  $L^q$  in  $\varphi(L^q)^p$

- ▶ Then the conditions of  $L^q$  are  $\mathcal{C}_{\varphi(\cdot)^p} := \mathcal{W}(G_1^{p_1}) \times \dots \times \mathcal{W}(G_n^{p_n})$

# Replacement rules

## ■ Resolution Replacement Rules

- ▶ Assume  $\psi(\alpha(\varphi(L^{-P}), \varphi'(F^P)))$
- ▶  $(C_1^{P_1}, \dots, C_n^{P_n}) \in \mathcal{C}_{\varphi(\cdot)}$
- ▶ Then  $L^{-P} \rightarrow \langle C_1^{P_1}, \dots, C_n^{P_n} \rangle$  is a replacement rule for  $F^P$



# Replacement rules

## ■ *Resolution Replacement Rules*

- ▶ Assume  $\psi(\alpha(\varphi(L^{-p}), \varphi'(F^p)))$
- ▶  $(C_1^{p_1}, \dots, C_n^{p_n}) \in \mathcal{C}_{\varphi(\cdot)}$
- ▶ Then  $L^{-p} \rightarrow \langle C_1^{p_1}, \dots, C_n^{p_n} \rangle$  is a replacement rule for  $F^p$

## ■ *Rewriting Replacement Rules*

- ▶ Assume  $\psi(\alpha(\varphi(u = v^-), \varphi'(s)))$  or  $\psi(\alpha(\varphi(u \Leftrightarrow v^-), \varphi'(s)))$
- ▶  $(C_1^{p_1}, \dots, C_n^{p_n}) \in \mathcal{C}_{\varphi(\cdot)}$
- ▶ Then  $u \rightarrow \langle v, C_1^{p_1}, \dots, C_n^{p_n} \rangle$  and  $v \rightarrow \langle u, C_1^{p_1}, \dots, C_n^{p_n} \rangle$  are replacement rules for  $s$

# Replacement Rules

*Examples for rules from  $(A \vee B) \Rightarrow (C \wedge D)^-$*

- $A^+ \rightarrow \langle C \wedge D^- \rangle$
- $B^+ \rightarrow \langle C \wedge D^- \rangle$
- $A \vee B^+ \rightarrow \langle C \wedge D^- \rangle$
- $C^- \rightarrow \langle A \vee B^+ \rangle$
- $D^- \rightarrow \langle A \vee B^+ \rangle$
- $C \wedge D^- \rightarrow \langle A \vee B^+ \rangle$

*Weakening right-hand sides:* From  $C \wedge D^-$  also have rules with either C or D

- $A^+ \rightarrow \langle C^- \rangle, A^+ \rightarrow \langle D^- \rangle$
- Also:  $C \wedge D^- \rightarrow \langle A^+ \rangle, C \wedge D^- \rightarrow \langle B^+ \rangle$

*Weakening to enable rule application*

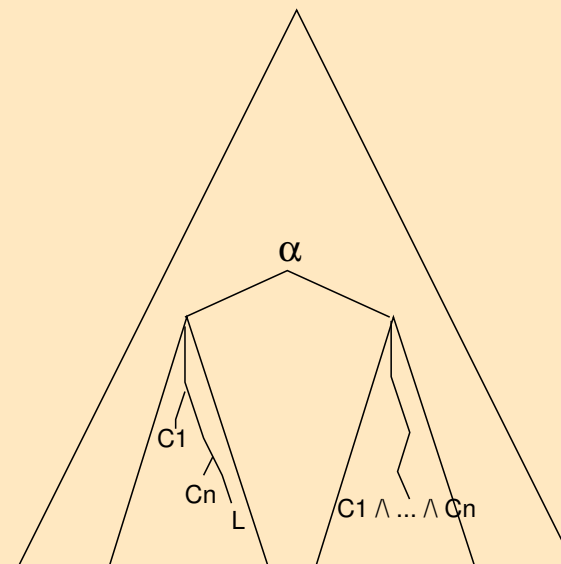
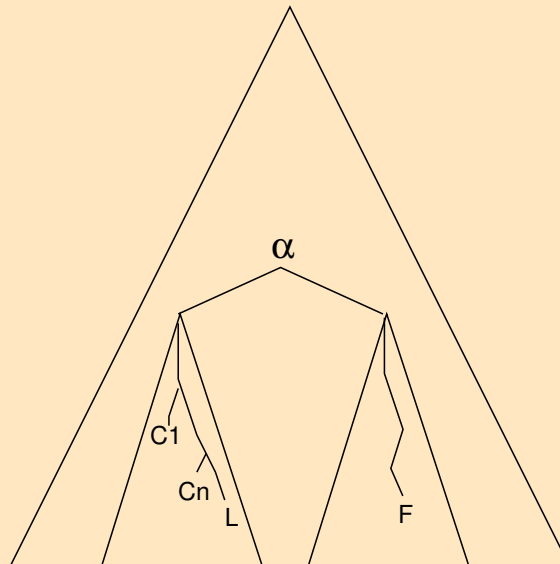
- $A \vee B^+ \rightarrow \langle C \wedge D^- \rangle$  is applicable on  $(A \vee E)^-$  to rewrite it to D
  - ▶  $A^+ \in \mathcal{W}(A \vee B^+)$  and  $A^- \in \mathcal{W}(A \vee E^-)$

# CoRE Calculus



## Rule application

- Replace by conjunction ( $\beta$ ) of subgoals by respecting polarities
- Example: From  $((A \Rightarrow (B \Rightarrow (C \wedge D))) \wedge (B \vee E))^-$   
by  $B^+ \rightarrow \langle A^+, C \wedge D^- \rangle$  on  $(B \vee E)$   
we obtain  $((A \Rightarrow (B \Rightarrow (C \wedge D))) \wedge (A \Rightarrow (C \wedge D)))^-$



# CoRE Calculus Rules:

*Given a conjecture F*

- Construct  $\Delta_F^1$  with singular multiplicity and meta-variables
- Extract deep formula  $DF(\Delta_F^1) := F_X$
- The initial proof state is  $\Delta_F^1; \text{Id} \triangleright F_X$

*Rules*  $\frac{\Delta; \sigma \triangleright G}{\Delta'; \sigma' \triangleright G'} R$  (top-down, deep structural)

- Replacement rule application NNF Resolution and paramodulation style
- Rules from EET functional and Boolean extensionality,  
Substitution of meta-variables, dynamic increase of multiplicities
- Rules to deal with richer set of connectives: Introduction of Leibniz' equality  
Elimination of positive equivalences
- Structural rules Weakening, Contraction, propositional simplification
- Cut and Axiom

# Soundness and Completeness



## *Soundness:*

- Define paths through the QF signed formula
- Initial QF signed formula is deep formula of initial EET
- Show for each rule that it preserves the existence of satisfiable paths

## *Completeness:*

- From completeness of EEP: Guess the right initial multiplicity and substitution, + applications of *Leibniz*, *f-Ext*, *b-Ext*, *ζ-Elim*, and *Cut*.
- The resulting QF signed formula  $F_P$  is propositionally unsatisfiable.
- Show that we can reduce  $\Delta_P; \sigma \triangleright F_P$  to  $\Delta; \sigma \triangleright \top$  (Axiom rule)
  - ▶ *path resolution* [MurrayRosenthal87] is admissible using *Res*, *Contract*, *Weakening* and the simplification rules.
  - ▶ path resolution complete for propositional logic

# Example

## Assume Axioms and Lemmas

$$\forall p. \forall x. \exists y. (p(0) \wedge (p(y) \Rightarrow p(s(y)))) \Rightarrow p(x) \quad (1)$$

$$\forall n. \sum_{i=1}^0 i^n = 0 \quad (3)$$

$$\forall n, m. \sum_{i=1}^{s(m)} i^n = s(m)^n + \sum_{i=1}^m i^n \quad (4)$$

$$\forall a, b. (a + b)^2 = (a^2 + (2 \times b) \times a) + b^2 \quad (5)$$

$$\forall a, b, c. a = b \Rightarrow a + c = b + c \quad (6)$$

$$\forall n. \sum_{i=1}^n i^1 = \frac{n \times s(n)}{2} \quad (7)$$

$$\forall m. 2 \times \frac{m}{2} = m \quad (8)$$

$$\forall q. s(q)^3 = s(q)^2 + (q \times s(q)) \times s(q) \quad (9)$$

$$0 = (0)^2 \quad (10)$$

Conjecture:  $\forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i^1)^2$

## The initial proof state

$$\Delta_0; \text{id} \triangleright \left( \begin{array}{ll} (P(0) \wedge (P(y) \Rightarrow P(s(y)))) \Rightarrow P(X) & (1) \\ \sum_{i=1}^0 i^N = 0 & (3) \\ \sum_{i=1}^{s(M)} i^N = s(M)^N + \sum_{i=1}^M i^N & (4) \\ (A + B)^2 = (A^2 + (2 \times B) \times A) + B^2 & (5) \\ A' = B' \Rightarrow A' + C' = B' + C' & (6) \\ \sum_{i=1}^{N'} i^1 = \frac{N' \times s(N')}{2} & (7) \\ 2 \times \frac{M'}{2} = M' & (8) \\ s(Q)^3 = s(Q)^2 + (Q \times s(Q)) \times s(Q) & (9) \\ 0 = (0)^2 & (10) \end{array} \right)$$

$$\Rightarrow \sum_{i=1}^n i^3 = (\sum_{i=1}^n i^1)^2$$

# Example Proof

1. By *RuleApplication from* (1)  $(P(0) \wedge (P(y) \Rightarrow P(s(y)))) \Rightarrow P(X)$

$\Delta_1; \sigma_1 \triangleright$

$$\left( (1) \wedge (3) \wedge (4) \wedge (5) \wedge (6) \wedge (7) \wedge (8) \wedge (9) \wedge (10) \right)$$

$$\Rightarrow (y' = 0 \Rightarrow \sum_{i=1}^{y'} i^3 = (\sum_{i=1}^{y'} i^1)^2) \wedge (\sum_{i=1}^{y'} i^3 = (\sum_{i=1}^{y'} i^1)^2) \Rightarrow (\sum_{i=1}^{s(y')} i^3 = (\sum_{i=1}^{s(y')} i^1)^2)$$

2. By *RuleApplication from*  $y' = 0$

$\Delta_1; \sigma_1 \triangleright$

$$\left( (1) \wedge \sum_{i=1}^0 i^N = 0 \wedge (4) \wedge (5) \wedge (6) \wedge (7) \wedge (8) \wedge (9) \wedge (10) \right)$$

$$\Rightarrow (y' = 0 \Rightarrow \sum_{i=1}^0 i^3 = (\sum_{i=1}^0 i^1)^2) \wedge (\sum_{i=1}^{y'} i^3 = (\sum_{i=1}^{y'} i^1)^2) \Rightarrow (\sum_{i=1}^{s(y')} i^3 = (\sum_{i=1}^{s(y')} i^1)^2)$$

3. By *RuleApplication from* (3)  $\sum_{i=1}^0 i^N = 0(2 \times)$

$\Delta_2; \sigma_2 \triangleright$

$$\left( (1) \wedge (3) \wedge (4) \wedge (5) \wedge (6) \wedge (7) \wedge (8) \wedge (9) \wedge 0 = (0)^2 \right)$$

$$\Rightarrow (y' = 0 \Rightarrow 0 = (0)^2) \wedge (\sum_{i=1}^{y'} i^3 = (\sum_{i=1}^{y'} i^1)^2) \Rightarrow (\sum_{i=1}^{s(y')} i^3 = (\sum_{i=1}^{s(y')} i^1)^2)$$

# Example Proof (cont'd)

4. By *RuleApplication from* (10)  $0 = (0)^2$

$$\Delta_3; \sigma_3 \triangleright \left( (1) \wedge (3) \wedge (4) \wedge (5) \wedge (6) \wedge (7) \wedge (8) \wedge (9) \wedge (10) \right)$$

$$\Rightarrow (y' = 0 \Rightarrow \top) \wedge (\sum_{i=1}^{y'} i^3 = (\sum_{i=1}^{y'} i^1)^2) \Rightarrow (\sum_{i=1}^{s(y')} i^3 = (\sum_{i=1}^{s(y')} i^1)^2)$$

5. By *Simplify with*  $\Rightarrow_R^\top$  and  $\wedge_L^\top$

$$\Delta_4; \sigma_4 \triangleright \left( (1) \wedge (3) \wedge \sum_{i=1}^{s(M)} i^N = s(M)^N + \sum_{i=1}^M i^N \wedge (5) \wedge (6) \wedge (7) \wedge (8) \wedge (9) \wedge (10) \right)$$

$$\Rightarrow (\sum_{i=1}^{y'} i^3 = (\sum_{i=1}^{y'} i^1)^2) \Rightarrow (\sum_{i=1}^{s(y')} i^3 = (\sum_{i=1}^{s(y')} i^1)^2)$$

⋮

# Compare first two Steps to...



To prove  $\forall n. \sum_{i=1}^n i^3 = \left(\sum_{i=1}^n i\right)^2$

we apply  $\forall P. (\forall n. n = 0 \Rightarrow P(n)) \Rightarrow ((\forall n, n'. (n = n' + 1 \wedge P(n')) \Rightarrow P(n)) \Rightarrow \forall n. P(n))$

to obtain  $\forall n. n = 0 \Rightarrow \sum_{i=1}^n i^3 = \left(\sum_{i=1}^n i\right)^2 \wedge$   
 $\forall n, n'. n = n' + 1 \wedge \sum_{i=1}^{n'} i^3 = \left(\sum_{i=1}^{n'} i\right)^2 \Rightarrow \sum_{i=1}^n i^3 = \left(\sum_{i=1}^n i\right)^2$

then apply  $n = 0$

to obtain  $\forall n. n = 0 \Rightarrow \sum_{i=1}^0 i^3 = \left(\sum_{i=1}^0 i\right)^2 \wedge$   
 $\forall n, n'. n = n' + 1 \wedge \sum_{i=1}^{n'} i^3 = \left(\sum_{i=1}^{n'} i\right)^2 \Rightarrow \sum_{i=1}^n i^3 = \left(\sum_{i=1}^n i\right)^2$

# Sequent Calculus Proof



$$\begin{array}{c}
 \vdots \\
 \hline
 n = n' + 1 \wedge \sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2 \vdash \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \Rightarrow_R \\
 \vdash (n = n' + 1 \wedge \sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2) \\
 \Rightarrow \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 \vdash \forall n, n'. (n = n' + 1 \wedge \sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2) \quad \forall_R \\
 \Rightarrow \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 \vdash \forall n, n'. (n = n' + 1 \wedge \sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2) \quad \text{Weak} \\
 \Rightarrow \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2, \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 (\forall n, n'. (n = n' + 1 \wedge \sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2)) \\
 \Rightarrow \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \Rightarrow \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \vdash \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 (\forall n. n = 0 \Rightarrow \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2) \\
 \Rightarrow ( (\forall n, n'. (n = n' + 1 \wedge \sum_{i=1}^{n'} i^3 = (\sum_{i=1}^{n'} i)^2)) \\
 \Rightarrow \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2) \vdash \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 \forall P. ( \forall n. n = 0 \Rightarrow P(n)) \quad \forall_L \\
 \Rightarrow ( (\forall n, n'. (n = n' + 1 \wedge P(n'))) \Rightarrow \forall n. P(n)) \vdash \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2
 \end{array}$$

$$\begin{array}{c}
 \vdots \\
 \hline
 n = 0 \vdash \sum_{i=1}^0 i^3 = (\sum_{i=1}^0 i)^2, \\
 \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 n = 0 \vdash \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2, \quad \text{Rew} \\
 \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \quad \Rightarrow_R \\
 \vdash n = 0 \Rightarrow \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2, \\
 \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 \vdash \forall n. n = 0 \Rightarrow \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2, \\
 \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \quad \forall_R \\
 \hline
 \forall n. \sum_{i=1}^n i^3 = (\sum_{i=1}^n i)^2 \\
 \hline
 \Rightarrow_L
 \end{array}$$

# Related Work



- Deep Inference paradigm for calculi:
  - ▶ Calculus of Structures (CoS) [Guglielmi,Brünnler04, ...]
  - ▶ Used to study proof theoretic properties, no assertion level reasoning
  - ▶ CORE calculus supports reasoning with assertions, no proof theoretic studies
  - ▶ Maybe combination/integration with CoS could provide a basis for proof checking
  
- Focusing Proof Construction [Andreoli00]
  - ▶ Derives sequent calculus macro steps from available assertions.
  - ▶ Derived rules can only be applied on top-level formulas
  
- IMPS “macetes” [FarmerGuttmanThayer1993]
  - ▶ Builtin procedures to extract very specific application procedures from theorems.

# Implementation



- It is implemented and provides all major features
  - ▶ determine available assertions for arbitrary subformulas
  - ▶ determine possibilities to apply an assertion on arbitrary subformulas
  - ▶ apply assertions on arbitrary subformulas
  - ▶ get additional renamed copies of formulas via dynamic increase of multiplicities

to the Task-Level of  $\Omega$ MEGA

- ... and thus to the user working with TEXMACS, to the planner, to the  $\Omega$ ANTS.

# Conclusion

CoRE Calculus where

- Assertion application is primitive rule (closer to CML, more “mathural”)
- Assertion application is actively supported
- Deep structural
- Technically assertion application corresponds to NNF resolution and paramodulation
  - ▶ Interactive proof construction: Technicalities of assertion application are hidden from the user
  - ▶ Automated proof construction: Directly presentable as a proof at the level of assertions

Here we focused on higher-order logic (Cut not admissible—yet!).

The formulations (Cut-free) for many first-order modal logics considered in [\[Wallen90\]](#) can be found in [\[AutexierPhD03\]](#)

# Future Work

- Define CoRE calculi for further logics, for instance intuitionistic logic
- Define cut-free version of CoRE for HOL
- Define CoRE with primitive equality (so far via Leibniz) [\[Brown,Cade05\]](#)
- Deduction modulo [\[DowekHardinKirchner98\]](#) to enable rule application:
  - ▶ Wish: Take local context into account for Deduction modulo in place
  - ▶ Difficulty: Rules from different contexts for left-hand side of rule and application position
- Transformations of CoRE proofs into SK proofs (proof checking)
- Integrate/combine CoRE and Calculus of Structures (CoS)  
(get independent from expansion trees, proof checking)
- Investigate automated reasoning procedures for CoRE  
Ordering-based techniques like [\[GanzingerStuber03\]](#) for NNF

