# Das Beweisassistenzsystem Ωmega

Serge Autexier[1]

22. Juni 2007

[1] Joint work with the Ωmega Group

# Zielsetzungen für ein Beweisassistenzsystem (BA)

- Unterstützung für Mathematiker/Wissenschaftler/Software-Entwickler in der Theoriebildung und insbesondere bei der Beweisführung
- Gründe:
  - Verifizierbare Formalisierungen und Beweise
  - Mehrwerte durch maschinelle Unterstützung:
    - Übernahme von Routineaufgaben bei Beweisen
    - Organisation/Handhabung von großen komplexen Beweisen (z.B. Kepler Vermutung durch T. Hales)
    - Semantik-basierte Suche nach math. Konzepten
    - ...

# Herangehensweisen

Bottom-up Vorgehen:

- Logik, Kalkül, und immer neue Komponenten darauf aufbauen
- Hoffnung, daß man irgendwann ein Gesamtsystem hat, dass Benutzer gerne verwenden wollen
- Klassische herangehensweise
- Am Beispiel von Ωmega

Top-down Vorgehen:

- Beginn mit System, das ein Benutzer verwendet
- Füge Funktionalitäten eines BAs hinzu
- Am Beispiel der Integration von Ωmega in den Texteditor TeXmacs

1. Bottom-Up Konstruktion eines BA
   - Logik & Beweiskalküle
   - Beweiskonstruktion
   - Komponenten des $\Omega$mega Systems

# Formelsyntax und Beweiskonstruktionsvorschirften

**Frege, Russell, Hilbert** Prädikatenkalkül und Typentheorie als formale Basis für die Mathematik: $+ : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$

$$\forall x, y, z : \mathbb{N}.(x + (y + z)) = ((x + y) + z)$$

**Gentzen** Kalkül des Natürlichen Schließens (ND)

ND-Regeln (Bsp.)

$$\frac{\mathbf{A} \quad \mathbf{B}}{\mathbf{A} \wedge \mathbf{B}} \wedge I \qquad \frac{\mathbf{A} \Rightarrow \mathbf{B} \quad \mathbf{A}}{\mathbf{B}} \; mp$$

$$\frac{\mathbf{A} \wedge \mathbf{B}}{\mathbf{A}} \wedge E_l \qquad \frac{\begin{array}{c}[\mathbf{A}]_1 \\ \vdots \\ \mathbf{B}\end{array}}{\mathbf{A} \Rightarrow \mathbf{B}} \Rightarrow I^1$$

$$\frac{\mathbf{A} \wedge \mathbf{B}}{\mathbf{B}} \wedge E_r$$

... usw. ...

ND-Beweis für $(\mathbf{A} \wedge \mathbf{B}) \Rightarrow (\mathbf{B} \wedge (\mathbf{C} \vee \mathbf{A}))$

$$\frac{\dfrac{\dfrac{[\mathbf{A} \wedge \mathbf{B}]_1}{\mathbf{B}} \wedge E_r \qquad \dfrac{\dfrac{[\mathbf{A} \wedge \mathbf{B}]_1}{\mathbf{A}} \wedge E_l}{\mathbf{C} \vee \mathbf{A}} \vee I_r}{\mathbf{B} \wedge (\mathbf{C} \vee \mathbf{A})} \wedge I}{(\mathbf{A} \wedge \mathbf{B}) \Rightarrow (\mathbf{B} \wedge (\mathbf{C} \vee \mathbf{A}))} \Rightarrow I^1$$

# Maschinennahe Kalküle

Robinson (1965): Resolutionskalkül Grundlage für Automatisierung

- Nur zwei Schlussregeln
- Formeln müssen erst in Normalform transformiert werden

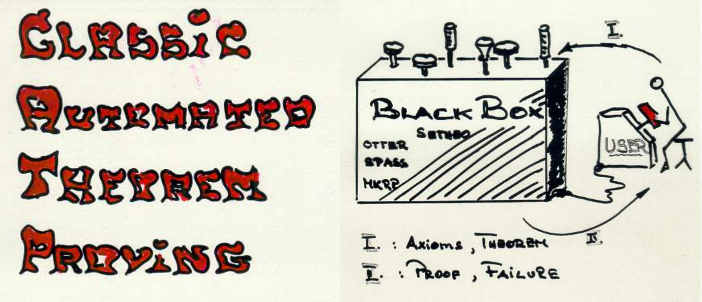



Bild: Jörg Siekmann

# Maschinennahe Kalküle

## Problemeingabe

```
%Here's an input file that gets a proof quickly.
%Note that he has a cancellation rule for multiplication.

set(auto).
set(ur_res).
assign(max_distinct_vars, 1).
list(usable).
x = x.
m(1,x) = x.                          %identity
m(x,1) = x.
m(x,m(y,z)) = m(m(x,y),z).           %associativity
m(x,y) = m(y,x).                     %commutativity
m(x,y) != m(x,z) | y = z.            %cancellation
-d(x,y) | m(x,f(x,y)) = y.           %this and next line define divides
m(x,z) != y | d(x,y).
-d(2,m(x,y)) | d(2,x) | d(2,y).      % 2 is prime  (with 12)
m(a,a) = m(2,m(b,b)).                % a/b = sqrt(2)
-d(x,a) | -d(x,b) | x = 1.           % a/b is in lowest terms
2 != 1.                              % I almost forgot this!
end_of_list.
```

## Beweisausgabe

```
---------------- PROOF ----------------

1 [] m(x,y)!=m(x,z)|y=z.
2 [] -d(x,y)|m(x,f(x,y))=y.
3 [] m(x,y)!=z|d(x,z).
4 [] -d(2,m(x,y))|d(2,x)|d(2,y).
5 [] -d(x,a)| -d(x,b)|x=1.
6 [] 2!=1.
7 [factor,4,2,3] -d(2,m(x,x))|d(2,x).
13 [] m(x,m(y,z))=m(m(x,y),z).
14 [copy,13,flip.1] m(m(x,y),z)=m(x,m(y,z)).
16 [] m(x,y)=m(y,x).
17 [] m(a,a)=m(2,m(b,b)).
18 [copy,17,flip.1] m(2,m(b,b))=m(a,a).
30 [hyper,18,3] d(2,m(a,a)).
39 [para_from,18.1.1,1.1.1] m(a,a)!=m(2,x)|m(b,b)=x.
42 [hyper,30,7] d(2,a).
46 [hyper,42,2] m(2,f(2,a))=a.
48 [ur,42,5,6] -d(2,b).
50 [ur,48,7] -d(2,m(b,b)).
59 [ur,50,3] m(2,x)!=m(b,b).
60 [copy,59,flip.1] m(b,b)!=m(2,x).
145 [para_from,46.1.1,14.1.1.1,flip.1] m(2,m(f(2,a),x))=m(a,x).
189 [ur,60,39] m(a,a)!=m(2,m(2,x)).
190 [copy,189,flip.1] m(2,m(2,x))!=m(a,a).
1261 [para_into,145.1.1.2,16.1.1] m(2,m(x,f(2,a)))=m(a,x).
1272 [para_from,145.1.1,190.1.1.2] m(2,m(a,x))!=m(a,a).
1273 [binary,1272.1,1261.1] $F.

------------ end of proof -------------
```

# Faktenebenen Beweise in $\Omega$mega

## Fakten

- Axiome
- Definitionen
- Theoreme/Korollare/Lemmata

## Fakten als Inferenzregeln

Transitivität von $\subseteq$:
$A \subseteq B \land B \subseteq C \Rightarrow A \subseteq C$ $\quad\Rightarrow\quad$ $\dfrac{A \subseteq B \quad B \subseteq C}{A \subseteq C}\ \mathit{Trans.} \subseteq$

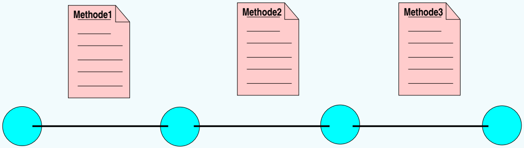
## Anwendung auf Teilformeln

Anwendung von Inferenzregeln auf Teilformeln

$$\frac{P \Rightarrow (A \subseteq B) \vdash Q \Rightarrow (A \subseteq C)}{P \Rightarrow (A \subseteq B) \vdash Q \Rightarrow (P \land (B \subseteq C))}$$

## Automatische Beweissuche

- Beweissuche auf Ebene der Inferenzen (Ordnungen, Heuristiken)
- Verwendung externer Systeme und Transformation der Beweise (i. A. schwierig)
- Beweisplanung: Domänenspezifisches Schließen auf abstrakterer Ebene

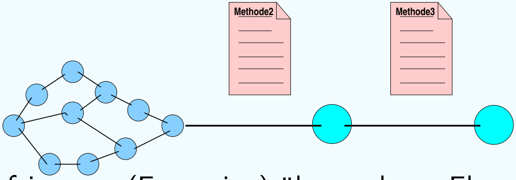



Beispielbeweismethoden: Diagonalisierungsprinzip, Induktionsbeweis + heuristische Steuerung

# Beweiskonstruktion I

## Automatische Beweissuche

- Beweissuche auf Ebene der Inferenzen (Ordnungen, Heuristiken)
- Verwendung externer Systeme und Transformation der Beweise (i. A. schwierig)
- Beweisplanung: Domänenspezifisches Schließen auf abstrakterer Ebene

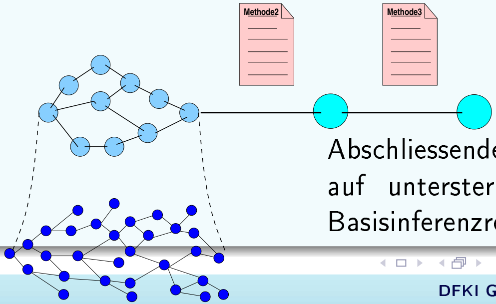



Beweisverfeinerung (Expansion) über meherer Ebenen ...

# Beweiskonstruktion I

## Automatische Beweissuche

- Beweissuche auf Ebene der Inferenzen (Ordnungen, Heuristiken)
- Verwendung externer Systeme und Transformation der Beweise (i. A. schwierig)
- Beweisplanung: Domänenspezifisches Schließen auf abstrakterer Ebene

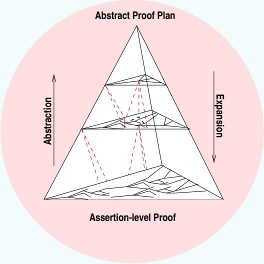



Abschliessende Verifikation auf unterster Ebene mit Basisinferenzregeln

# Beweiskonstruktion I

## Automatische Beweissuche

- Beweissuche auf Ebene der Inferenzen (Ordnungen, Heuristiken)
- Verwendung externer Systeme und Transformation der Beweise (i. A. schwierig)
- Beweisplanung: Domänenspezifisches Schließen auf abstrakterer Ebene




Beweis simultan auf verschiedenen Granularitätsstufen

# Beweiskonstruktion II

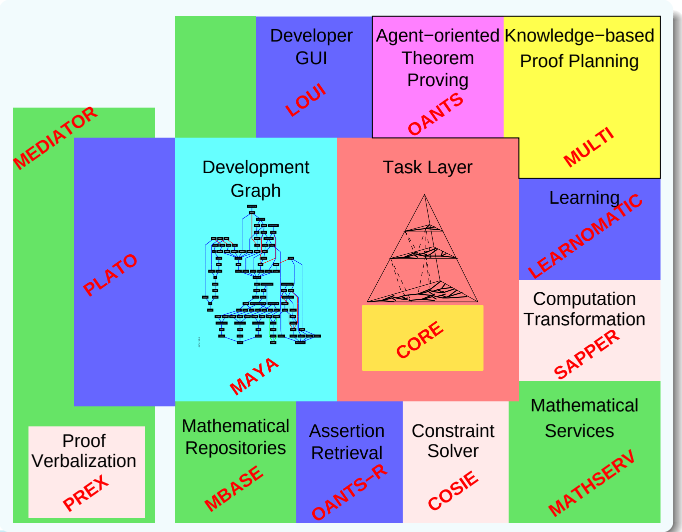
## Interaktive Beweiskonstruktion

- Anwendung einzelner Schritte, Verwendung von automatischen Prozeduren und externen Beweissystemen
- Beweisskizzen

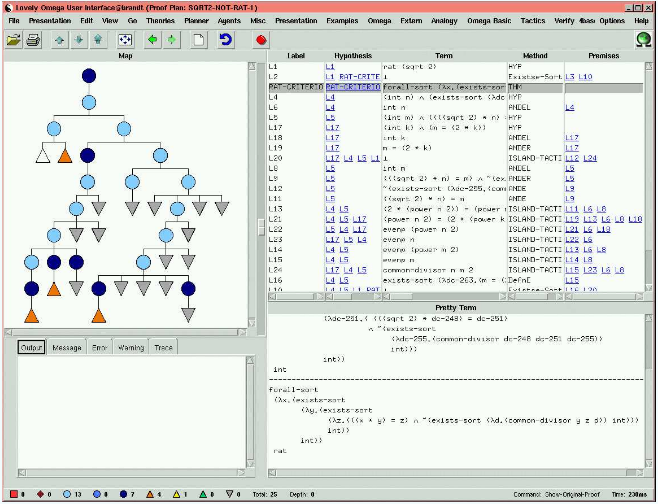
## Verwendung von CAS

- Berechnungen werden gebraucht, aber schlecht unterstützt durch Logikkalküle
- Verwendung von externen CAS
- Problem: BA braucht Beweis für die Berechnung

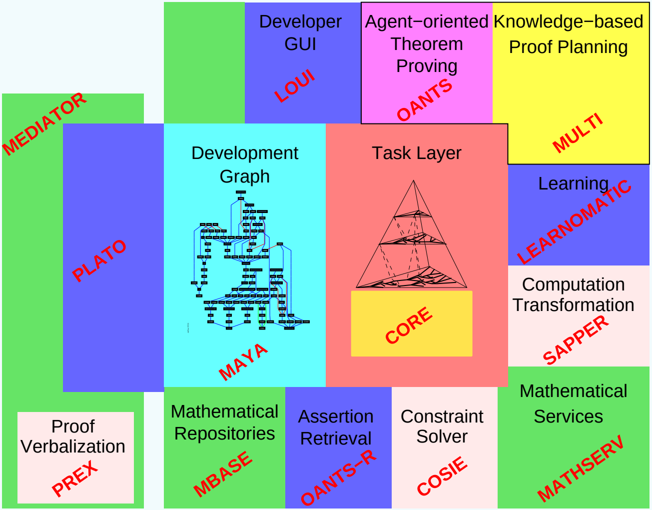
# Komponenten des Ωmega Systems



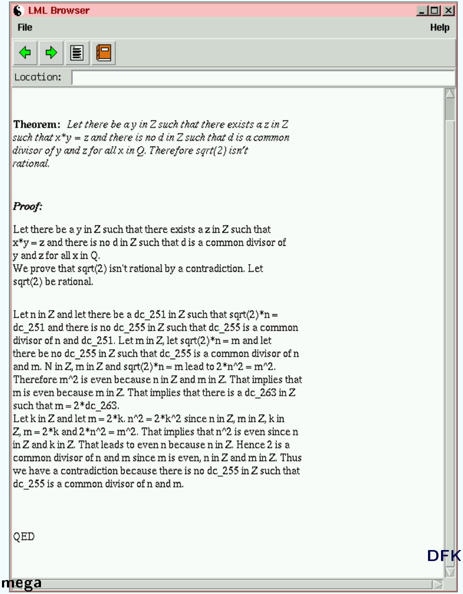
# Graphische Benutzerschnittstelle



# Komponenten des Ωmega Systems



# Beweispräsentation in natürlicher Sprache




**Theorem:** *Let there be a y in Z such that there exists a z in Z such that x*y = z and there is no d in Z such that d is a common divisor of y and z for all x in Q. Therefore sqrt(2) isn't rational.*

***Proof:***

Let there be a y in Z such that there exists a z in Z such that x*y = z and there is no d in Z such that d is a common divisor of y and z for all x in Q.
We prove that sqrt(2) isn't rational by a contradiction. Let sqrt(2) be rational.

Let n in Z and let there be a dc_251 in Z such that sqrt(2)*n = dc_251 and there is no dc_255 in Z such that dc_255 is a common divisor of n and dc_251. Let m in Z, let sqrt(2)*n = m and let there be no dc_255 in Z such that dc_255 is a common divisor of n and m. N in Z, m in Z and sqrt(2)*n = m lead to 2*n^2 = m^2. Therefore m^2 is even because n in Z and m in Z. That implies that m is even because m in Z. That implies that there is a dc_263 in Z such that m = 2*dc_263.
Let k in Z and let m = 2*k. n^2 = 2*k^2 since n in Z, m in Z, k in Z, m = 2*k and 2*n^2 = m^2. That implies that n^2 is even since n in Z and k in Z. That leads to even n because n in Z. Hence 2 is a common divisor of n and m since m is even, n in Z and m in Z. Thus we have a contradiction because there is no dc_255 in Z such that dc_255 is a common divisor of n and m.

QED

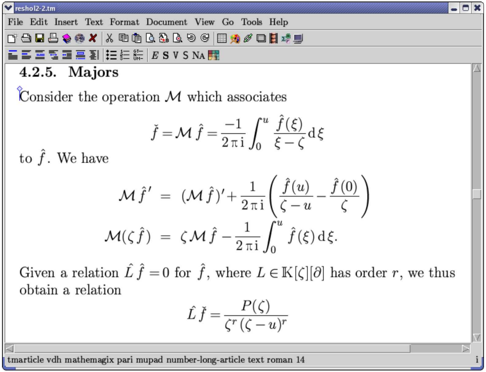
2. Top-down Konstruktion eines BA
   - Der wissenschaftliche Texteditor $\TeX_{macs}$
   - Integration von $\Omega$mega und $\TeX_{macs}$
   - Verfügbare und geplante Funktionalitäten

# Zielsetzung

- Die Dienste des BA dort anzubieten, wo sie gebraucht werden
- Analogie: Grammatikprüfer, nur interaktiv
- Autoren von Dokumenten sollten nicht Eigenheiten des BA lernen müssen, um diese nutzen zu können
- Das BA soll sich dem Benutzer anpassen, nicht umgekehrt
- Leitgedanke für Integration von $\Omega$mega in $\TeX_{\text{macs}}$

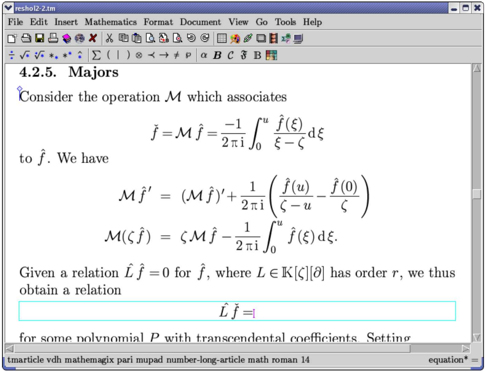
# TeXmacs

- *what-you-see-is-what-you-get* Paradigma
- Struktureditor, gute Unterstützung für math. Formeln
- Konfigurierbar



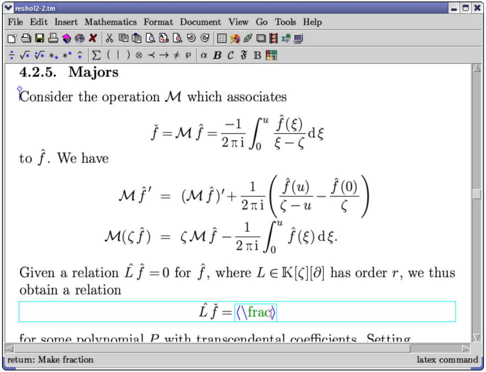
# TeXmacs

- *what-you-see-is-what-you-get* Paradigma
- Struktureditor, gute Unterstützung für math. Formeln
- Konfigurierbar



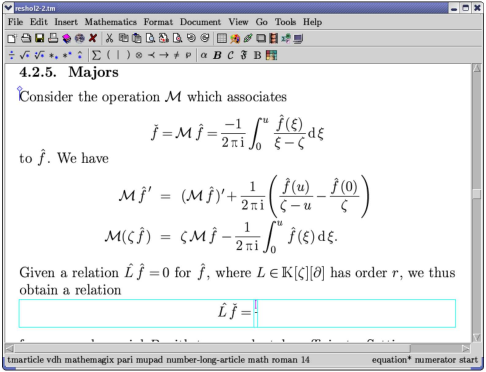
# TeXmacs

- *what-you-see-is-what-you-get* Paradigma
- Struktureditor, gute Unterstützung für math. Formeln
- Konfigurierbar



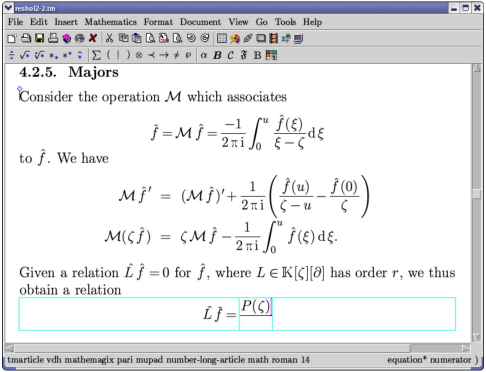
# TeXmacs

- *what-you-see-is-what-you-get* Paradigma
- Struktureditor, gute Unterstützung für math. Formeln
- Konfigurierbar



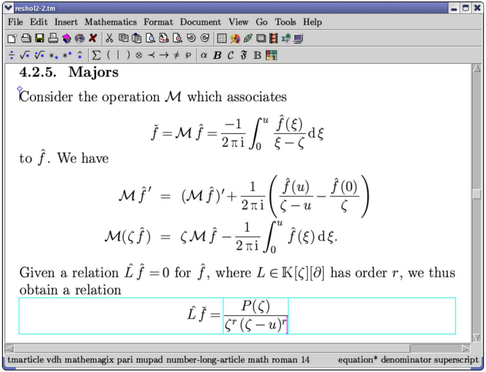
# TeXmacs

- *what-you-see-is-what-you-get* Paradigma
- Struktureditor, gute Unterstützung für math. Formeln
- Konfigurierbar

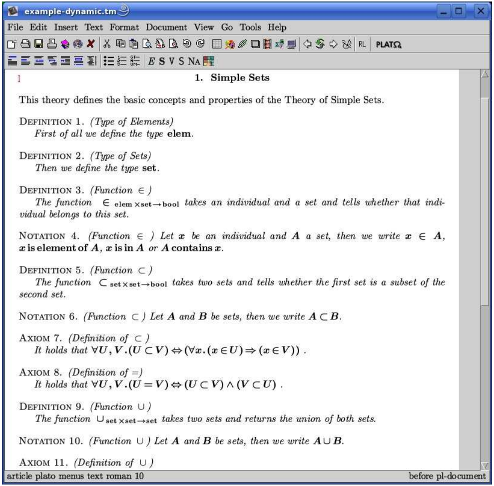


# TeXmacs

- *what-you-see-is-what-you-get* Paradigma
- Struktureditor, gute Unterstützung für math. Formeln
- Konfigurierbar



# Beispieldokument



example-dynamic.tm

File Edit Insert Text Format Document View Go Tools Help

**1. Simple Sets**

This theory defines the basic concepts and properties of the Theory of Simple Sets.

DEFINITION 1. *(Type of Elements)*
*First of all we define the type* **elem**.

DEFINITION 2. *(Type of Sets)*
*Then we define the type* **set**.

DEFINITION 3. *(Function* $\in$ *)*
*The function* $\in_{\mathbf{elem}\times\mathbf{set}\rightarrow\mathbf{bool}}$ *takes an individual and a set and tells whether that individual belongs to this set.*

NOTATION 4. *(Function* $\in$ *) Let* $x$ *be an individual and* $A$ *a set, then we write* $x \in A$, $x$ **is element of** $A$, $x$ **is in** $A$ *or* $A$ **contains** $x$.

DEFINITION 5. *(Function* $\subset$ *)*
*The function* $\subset_{\mathbf{set}\times\mathbf{set}\rightarrow\mathbf{bool}}$ *takes two sets and tells whether the first set is a subset of the second set.*

NOTATION 6. *(Function* $\subset$ *) Let* $A$ *and* $B$ *be sets, then we write* $A \subset B$.

AXIOM 7. *(Definition of* $\subset$ *)*
*It holds that* $\forall U, V.(U \subset V) \Leftrightarrow (\forall x.(x \in U) \Rightarrow (x \in V))$ .

AXIOM 8. *(Definition of* $=$*)*
*It holds that* $\forall U, V.(U = V) \Leftrightarrow (U \subset V) \wedge (V \subset U)$ .

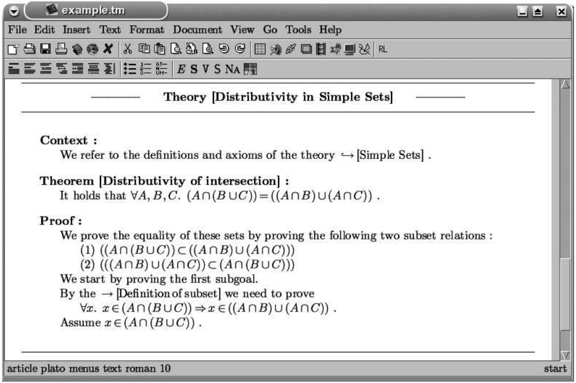
DEFINITION 9. *(Function* $\cup$ *)*
*The function* $\cup_{\mathbf{set}\times\mathbf{set}\rightarrow\mathbf{set}}$ *takes two sets and returns the union of both sets.*

NOTATION 10. *(Function* $\cup$ *) Let* $A$ *and* $B$ *be sets, then we write* $A \cup B$.

AXIOM 11. *(Definition of* $\cup$ *)*

article plato menus text roman 10 before pl-document

# Beweisunterstützung


example.tm
File Edit Insert Text Format Document View Go Tools Help

**Theory [Distributivity in Simple Sets]**

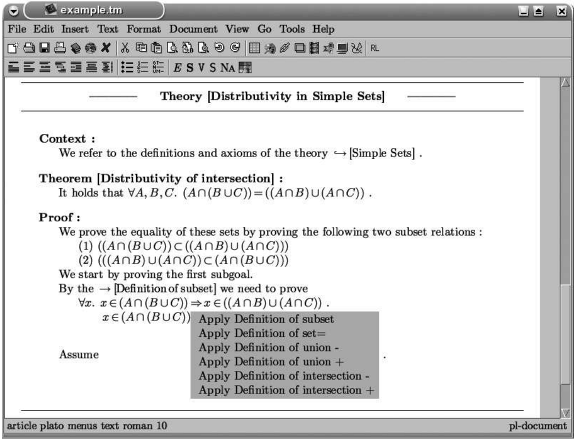
**Context :**
We refer to the definitions and axioms of the theory ↪[Simple Sets] .

**Theorem [Distributivity of intersection] :**
It holds that $\forall A, B, C.\ (A \cap (B \cup C)) = ((A \cap B) \cup (A \cap C))$ .

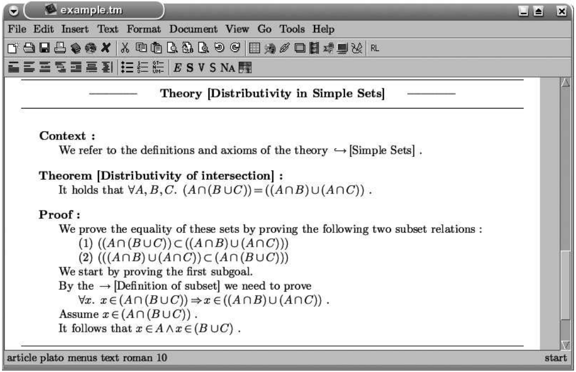
**Proof :**
We prove the equality of these sets by proving the following two subset relations :
(1) $((A \cap (B \cup C)) \subset ((A \cap B) \cup (A \cap C)))$
(2) $(((A \cap B) \cup (A \cap C)) \subset (A \cap (B \cup C)))$
We start by proving the first subgoal.
By the → [Definition of subset] we need to prove
$\forall x.\ x \in (A \cap (B \cup C)) \Rightarrow x \in ((A \cap B) \cup (A \cap C))$ .
Assume $x \in (A \cap (B \cup C))$ .

article plato menus text roman 10 start

# Beweisunterstützung



# Beweisunterstützung




**Theory [Distributivity in Simple Sets]**

**Context :**
We refer to the definitions and axioms of the theory ↪[Simple Sets] .

**Theorem [Distributivity of intersection] :**
It holds that $\forall A, B, C.\ (A \cap (B \cup C)) = ((A \cap B) \cup (A \cap C))$ .

**Proof :**
We prove the equality of these sets by proving the following two subset relations :
(1) $((A \cap (B \cup C)) \subset ((A \cap B) \cup (A \cap C)))$
(2) $(((A \cap B) \cup (A \cap C)) \subset (A \cap (B \cup C)))$
We start by proving the first subgoal.
By the → [Definition of subset] we need to prove
$\forall x.\ x \in (A \cap (B \cup C)) \Rightarrow x \in ((A \cap B) \cup (A \cap C))$ .
Assume $x \in (A \cap (B \cup C))$ .
It follows that $x \in A \wedge x \in (B \cup C)$ .

# Was muß der Autor machen?

- Annotationen am Text in Form von Macros

**Definition (Function $\in$)**

The predicate $\in_{elem \times set \rightarrow bool}$ takes an individual and a set and tells whether that individual belongs to this set.

```
\begin{definition}[Function $\in$]
  The predicate \concept{\in}{elem \times set \rightarrow bool}
  takes an individual and a set and tells whether that
  individual belongs to this set.
\end{definition}
```

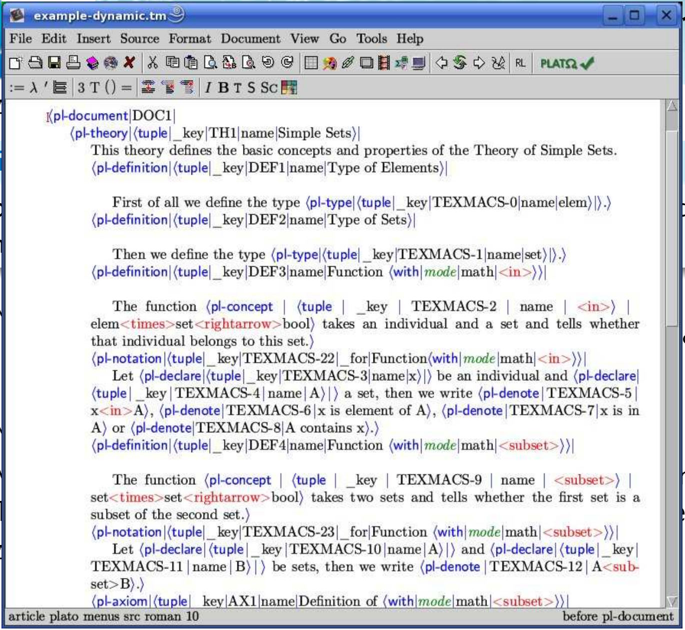
- Muss für alle Semantik-tragenden Teile angegeben werden: Theorien, Definitionen, Theorem, Beweise, Beweisschritte, ...
- Ausnahme: Formeln

?

▶ A

Defini

The p … d tells
wheth




ow bool}

▶ M … n:
T … e, . . .

▶ A

# Eingabe von Formeln

## Notation (Function $\in$)

*Let $x$ be an individual and $A$ a set, then we write $x \in A$, $x$ is element of $A$, $x$ is in $A$ or $A$ contains $x$.*

```
\begin{notation}[Function $\in$]
   Let \declare{x} be an individual and \declare{A} a set,
   then we write \denote{x \in A}, \denote{x is element of A},
   \denote{x is in A} or \denote{A contains x}.
 \end{notation}
```

## Example

Allows to write the inline formulas *x is element of* $(A \cup B)$ as well as $\forall y.y \in (A \cup B)$ in formulas.
But not: *x and y are element of* $(A \cup B)$.

# Integration mit Ωmega

# Verfügbare und geplante Funktionalitäten I

## Allgemeine Eigenschaften

- Sind alle verwendeten Konzepte definiert?
- Sind die vereinbarten Notationen eingehalten worde?
- Verwaltung der Notationen
- Verwendung von Theorien aus anderen Dokumenten (Semantik + Notation) Semantisches Zitieren

## Speziell für Beweise

- In einem offenen Teilbeweis: Was sind mögliche nächste Beweisschritte?
- Aufruf der automatischen Beweissuchverfahren auf einem offenen Beweisziel

# Verfügbare und geplante Funktionalitäten II

- Automatisch gefundene (Teil-)Beweise werden
  - automatisch in das TeXmacs-Dokument integriert
  - für Formeln wird die eingeführte Notation verwendet
- Ist der Beweis fertig?
- Ist der Beweis korrekt?
- Ist der angegebene Beweis ein Beweis für das Theorem?
- Kollaboratives Arbeiten
  - Vorausschau für Effekte von Änderungen
  - Locking-Mechanismus um semantische Aspekte zu schützen
- Manuelles annotieren reduzieren durch Textanalyse-Verfahren

# Zusammenfassung/Ausblick

- Ein BA bottom-up zu bauen umfasst vielfältige Aspekte (ähnlich wie ein CAS)
  - Logik, Kalküle,
  - Repräsentation, Verwendung und Lernen von Beweissuchwissen, Organisation der automat. Beweissuche
  - Verwaltung von mathematischen Wissen, ...
- Top-down Vorgehen
  - orientiert sich an Wünschen des Anwenders
  - bringt zusätzliche Aspekte:
    - Granularität von Beweisen
    - Textanalyse und Textgenerierung
    - Verwaltung von Abhängigkeiten in Dokumenten
    - Mehrwerte für Autoren da Semantik von Dokumenten maschinell verarbeitbar (Suche, Extraktion, Reformulierung, ...)

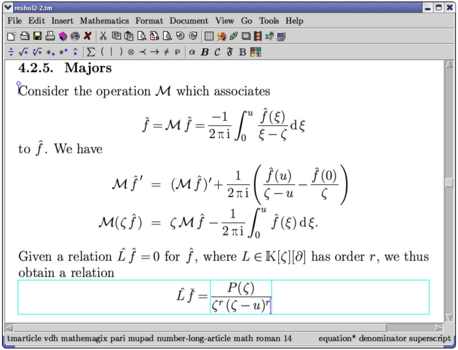
# Die Ωmega Gruppe

- Serge Autexier
- Christoph Benzmüller
- Dominik Dietrich
- Andreas Franke
- Henri Lesourd
- Marvin Schiller
- Ewaryst Schulz
- Jörg Siekmann
- Marc Wagner

+ Studenten

# In eigener Sache. . .

- Top-down Vorgehen orientiert sich an dem was gebraucht wird
- Wir können uns alles mögliche vorstellen, aber . . .
- . . . was wird tatsächlich gebraucht?
- Hilfreich typische Abäufe zu kennen




- Artikel, (Lehr-)Bücher oder Folien zu erstellen (alleine, zusammen mit anderen, Wiederverwendung von Teilen aus anderen Dokumenten, . . . )
- Wir suchen: Leute, die gerne mal ausprobieren ihre Texte damit zu schreiben, um feedback zu bekommen, was gut ist, was unzumutbar ist, was brauchbar wäre, . . .