

# Base Technologies for Tutoring

## Experiences from experiments in the *ISAC*-project

Walther Neuper

Institute for Softwaretechnology  
Graz University of Technology

Time Conference 2010  
Jul.8, Malaga

## Outline

- 1 Remarks on tutoring software
  - Status quo in the design
  - Requirements on the design
- 2 Base technologies for tutoring
  - Computer theorem proving
  - Single-stepping program interpretation
  - Human readable math knowledge
- 3 Summary and Conclusion

- 1 Remarks on tutoring software
  - Status quo in the design
  - Requirements on the design
- 2 Base technologies for tutoring
  - Computer theorem proving
  - Single-stepping program interpretation
  - Human readable math knowledge
- 3 Summary and Conclusion

## Status quo in tutoring software

Software for tutoring is concerned with *individuals* ...

- ... **individual learners**
  - on different levels
  - with different pace in learning ...
- ... **individual teachers**
  - with different teaching styles
  - emphasizing specific examples ...
- ... **individual programmers** (frequently teachers)
  - creating an abundant variety of software
  - lack support of general software services

What are the **general requirements** for tutoring ?

What are the **basic technologies** to meet the requirements ?

# Status quo in tutoring software

Software for tutoring is concerned with *individuals* ...

- ... *individual learners*
  - on different levels
  - with different pace in learning ...
- ... *individual teachers*
  - with different teaching styles
  - emphasizing specific examples ...
- ... *individual programmers* (frequently teachers)
  - creating an abundant variety of software
  - lack support of general software services

What are the **general requirements** for tutoring ?

What are the **basic technologies** to meet the requirements ?

# Status quo in tutoring software

Software for tutoring is concerned with *individuals* ...

- ... *individual learners*
  - on different levels
  - with different pace in learning ...
- ... *individual teachers*
  - with different teaching styles
  - emphasizing specific examples ...
- ... *individual programmers* (frequently teachers)
  - creating an abundant variety of software
  - lack support of general software services

What are the **general requirements** for tutoring ?

What are the **basic technologies** to meet the requirements ?

- 1 Remarks on tutoring software
  - Status quo in the design
  - Requirements on the design
- 2 Base technologies for tutoring
  - Computer theorem proving
  - Single-stepping program interpretation
  - Human readable math knowledge
- 3 Summary and Conclusion

## Requirements for tutoring

The general requirements are basically:

- 1 Check user input as generous as possible ...
- 2 Guide the user step by step towards a solution ...
- 3 Explain steps on request by the user ...

... during the stepwise *construction* of the solution  
of some problem in applying mathematics (incl. geometry).

# Requirements for tutoring

The general requirements are basically:

- 1 Check user input as generous as possible ...
- 2 Guide the user step by step towards a solution ...
- 3 Explain steps on request by the user ...

... during the stepwise *construction* of the solution  
of some problem in applying mathematics (incl. geometry).

# Requirements for tutoring

The general requirements are basically:

- 1 Check user input as generous as possible ...
- 2 Guide the user step by step towards a solution ...
- 3 Explain steps on request by the user ...

... during the stepwise *construction* of the solution  
of some problem in applying mathematics (incl. geometry).

# Requirements for tutoring

The general requirements are basically:

- 1 Check user input as generous as possible ...
- 2 Guide the user step by step towards a solution ...
- 3 Explain steps on request by the user ...

... during the stepwise *construction* of the solution  
of some problem in applying mathematics (incl. geometry).

# Requirements for tutoring

The general requirements are basically:

- 1 Check user input as generous as possible ...
- 2 Guide the user step by step towards a solution ...
- 3 Explain steps on request by the user ...

... during the stepwise *construction* of the solution  
of some problem in applying mathematics (incl. geometry).

- 1 Remarks on tutoring software
  - Status quo in the design
  - Requirements on the design
- 2 Base technologies for tutoring
  - Computer theorem proving
  - Single-stepping program interpretation
  - Human readable math knowledge
- 3 Summary and Conclusion

## Requirements and Technologies

The requirements can be met by the **technologies**

- 1 **check user input** as generous and liberal as possible
  - by **Computer Theorem Proving (CTP)**, i.e
  - **Automated Proving (ATP)**: simplifiers, SAT, SMT etc
  - **Interactive Proving (ITP)**: Coq and Isabelle in EU
- 2 **guide the user** step by step towards a solution
  - 
  -
- 3 **explain steps** on request by the user
  -

# Requirements and Technologies

The requirements can be met by the **technologies**

- ① **check user input** as generous and liberal as possible
  - by **Computer Theorem Proving (CTP)**, i.e
  - **Automated Proving (ATP)**: simplifiers, SAT, SMT etc
  - **Interactive Proving (ITP)**: Coq and Isabelle in EU
- ② **guide the user** step by step towards a solution
  - 
  -
- ③ **explain steps** on request by the user
  -

# Requirements and Technologies

The requirements can be met by the **technologies**

- ① **check user input** as generous and liberal as possible
  - by **Computer Theorem Proving (CTP)**, i.e
  - **Automated Proving (ATP)**: simplifiers, SAT, SMT etc
  - **Interactive Proving (ITP)**: Coq and Isabelle in EU
- ② **guide the user** step by step towards a solution
  - 
  -
- ③ **explain steps** on request by the user
  -



# Requirements and Technologies

The requirements can be met by the **technologies**

- ① **check user input** as generous and liberal as possible
  - by **Computer Theorem Proving (CTP)**, i.e
  - **Automated Proving (ATP)**: simplifiers, SAT, SMT etc
  - **Interactive Proving (ITP)**: Coq and Isabelle in EU
- ② **guide the user** step by step towards a solution
  - 
  -
- ③ **explain steps** on request by the user
  -

# Demo Isabelle from wikipediaedia

- 1 Remarks on tutoring software
  - Status quo in the design
  - Requirements on the design
- 2 Base technologies for tutoring
  - Computer theorem proving
  - Single-stepping program interpretation
  - Human readable math knowledge
- 3 Summary and Conclusion

## Requirements and Technologies

The requirements can be met by the **technologies**

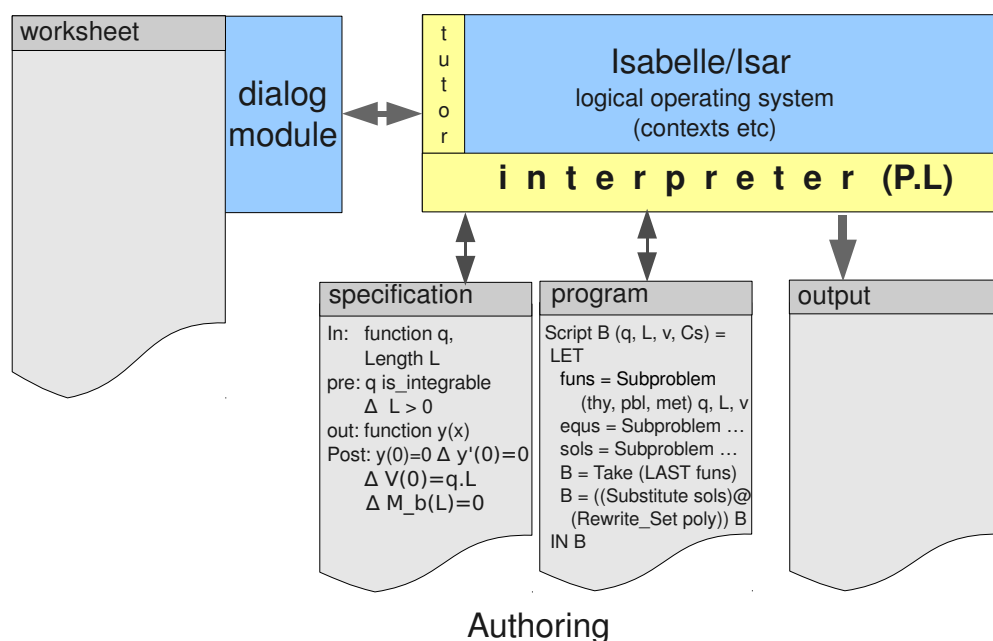
- 1 **check user input** as generous and liberal as possible
  - by Computer Theorem Proving (CTP), i.e
  - Automated Proving (ATP): simplifiers, SAT, SMT etc
  - Interactive Proving (ITP): Coq and Isabelle in EU
- 2 **guide the user** step by step towards a solution
  - by **interpretation in debug-mode** with breakpoints
  -
- 3 **explain steps** on request by the user
  -

# Requirements and Technologies

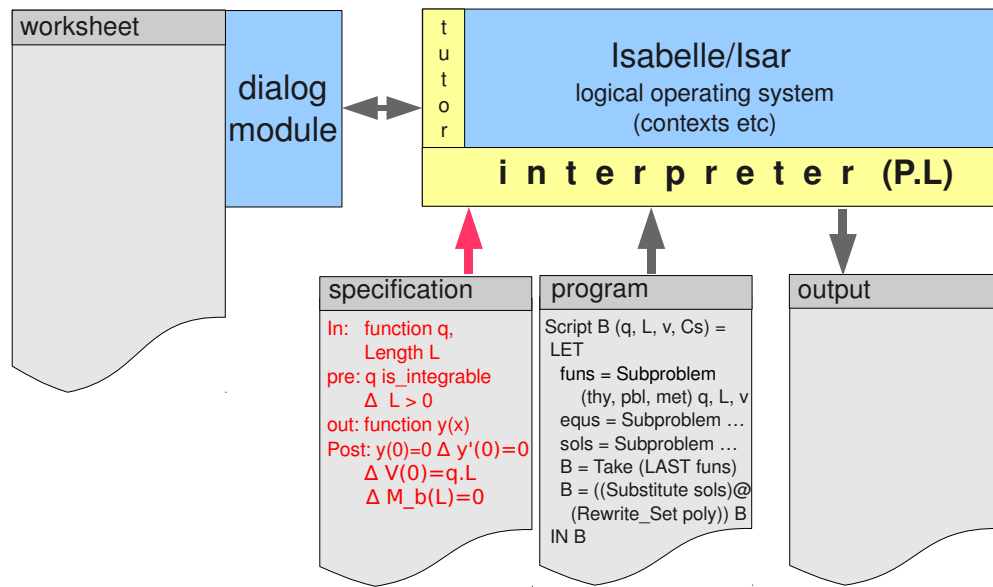
The requirements can be met by the **technologies**

- ① **check user input** as generous and liberal as possible
  - by Computer Theorem Proving (CTP), i.e.
  - Automated Proving (ATP): simplifiers, SAT, SMT etc
  - Interactive Proving (ITP): Coq and Isabelle in EU
- ② **guide the user** step by step towards a solution
  - by **interpretation in debug-mode** with breakpoints
  -
- ③ **explain steps** on request by the user
  -

## Interpretation in debug-mode

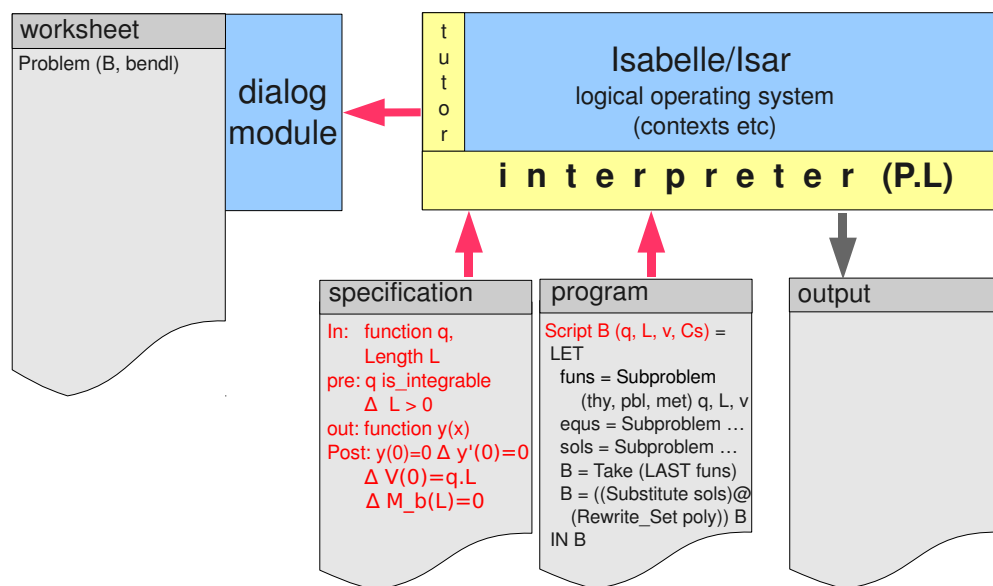


## Example :start tutoring



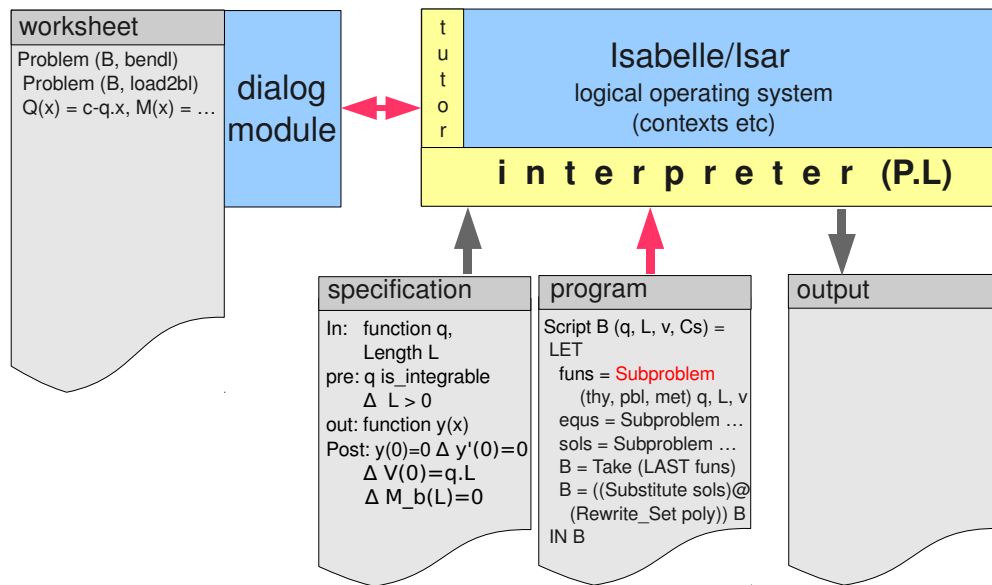
Tutoring

## Example: Tutoring start



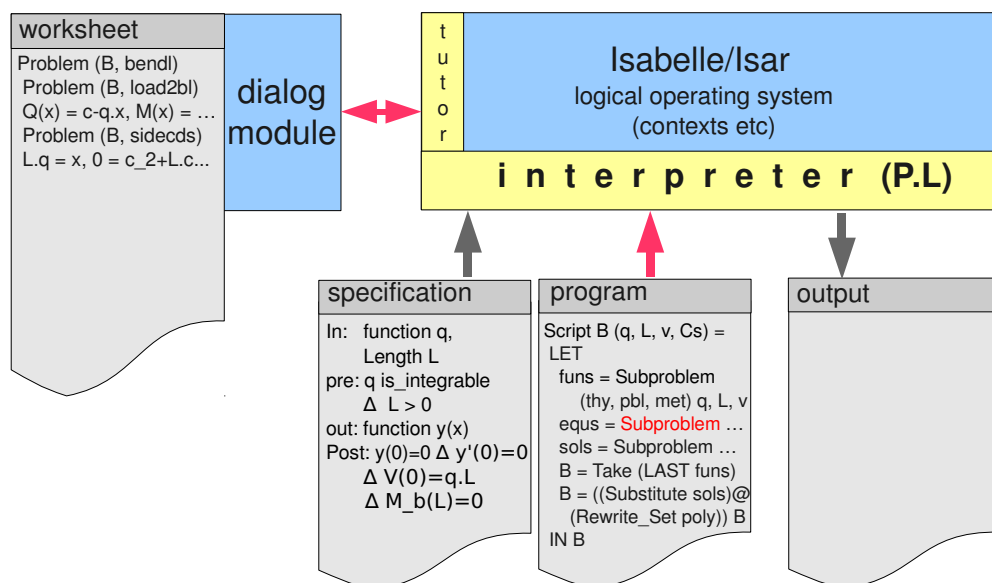
Tutoring

## Example: Tutoring step 1



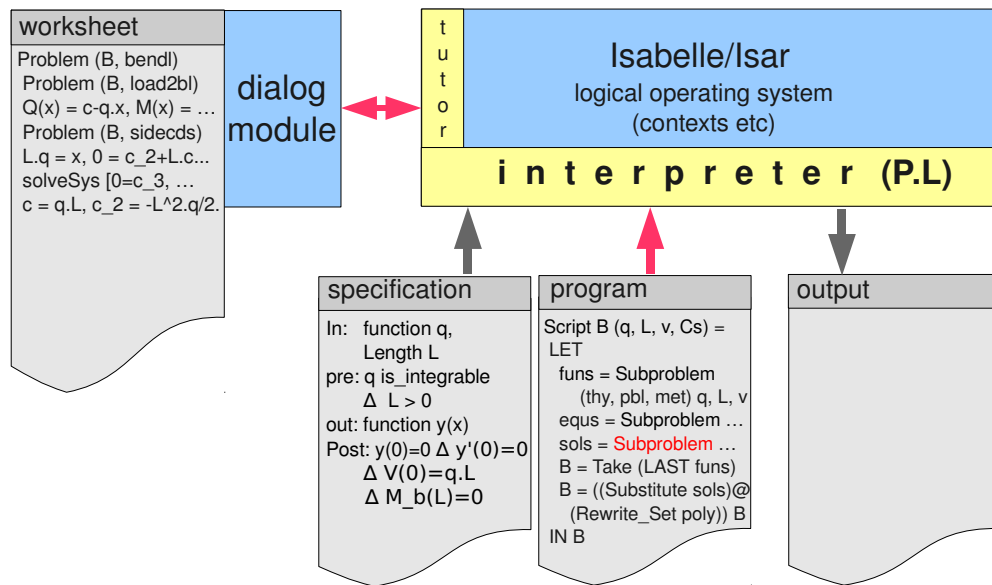
Tutoring

## Example: Tutoring step 2



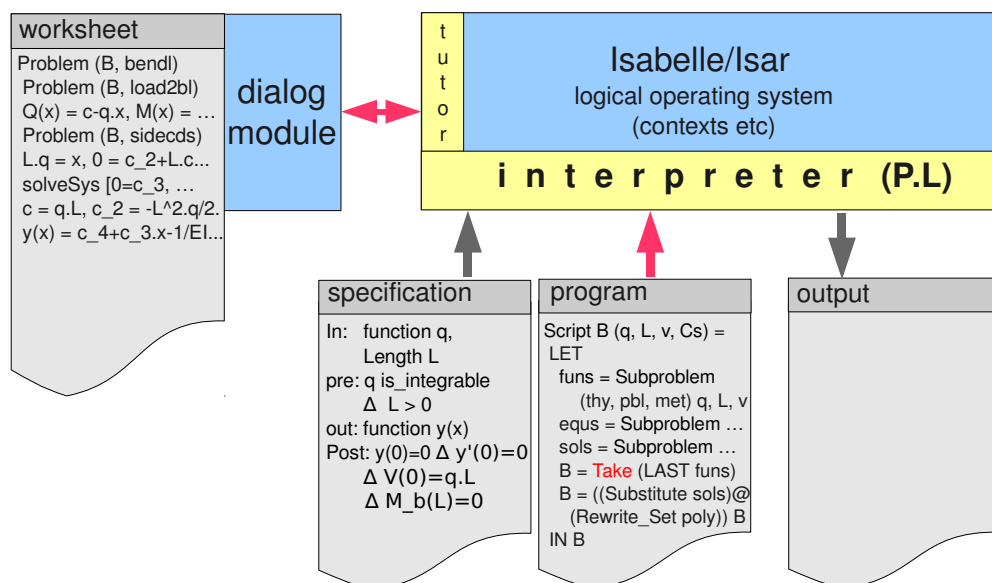
Tutoring

## Example: Tutoring step 3



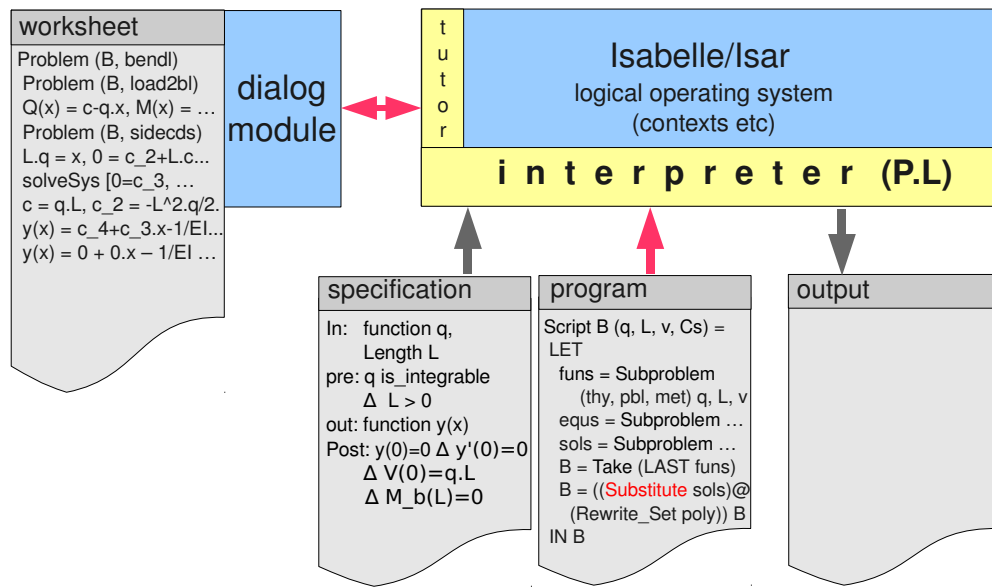
Tutoring

## Example: Tutoring step 4



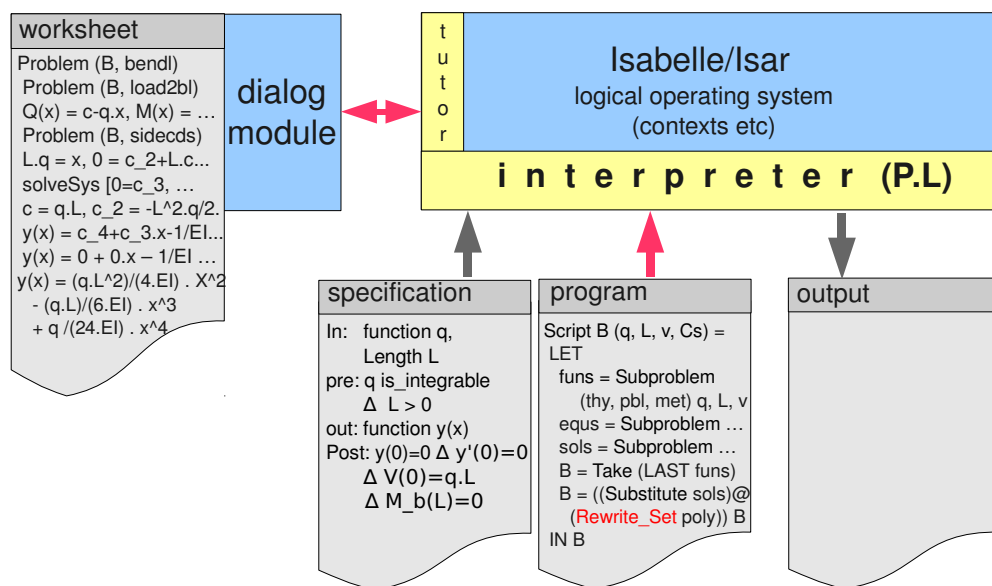
Tutoring

## Example: Tutoring step 5



Tutoring

## Example: Tutoring finished



Tutoring





# Requirements and Technologies

The requirements can be met by the **technologies**

- ① **check user input** as generous and liberal as possible
  - by Computer Theorem Proving (CTP), i.e
  - Automated Proving (ATP): simplifiers, SAT, SMT etc
  - Interactive Proving (ITP): Coq and Isabelle in EU
- ② **guide the user** step by step towards a solution
  - by **interpretation in debug-mode** with breakpoints
  - ? **resume** after liberal user input at breakpoint (ATP !)
- ③ **explain steps** on request by the user
  -

## Dialog mechanically generated by CTP-technology

A calculation proceeded to a certain step (no. **3.**):

$$1. \quad \frac{d}{dx}(x^2 + \sin(3 \cdot x^4))$$

$$2. \quad 2 \cdot x^{2-1} + \frac{d}{dx} \sin(3 \cdot x^4)$$

$$3. \quad 2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4)$$

How can we guide the student to the next formula (no. **4.**)

...

$$4. \quad 2 \cdot x + \cos(3 \cdot x^4) \cdot \frac{d}{dx}(3 \cdot x^4)$$

... or some algebraically equivalent formula ?

# Dialog mechanically generated by CTP-technology

A calculation proceeded to a certain step (no. **3.**):

$$1. \quad \frac{d}{dx}(x^2 + \sin(3 \cdot x^4))$$

$$2. \quad 2 \cdot x^{2-1} + \frac{d}{dx} \sin(3 \cdot x^4)$$

$$3. \quad 2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4)$$

How can we guide the student to the next formula (no. **4.**)

...

$$4. \quad 2 \cdot x + \cos(3 \cdot x^4) \cdot \frac{d}{dx}(3 \cdot x^4)$$

...or some algebraically equivalent formula ?

# Dialog mechanically generated by CTP-technology

$$1. \quad \frac{d}{dx}(x^2 + \sin(3 \cdot x^4))$$

$$2. \quad 2 \cdot x^{2-1} + \frac{d}{dx} \sin(3 \cdot x^4)$$

$$3. \quad 2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4)$$

$$4. \quad \underline{\hspace{10em}}$$

# Dialog mechanically generated by CTP-technology

3. 
$$\begin{array}{c} \vdots \\ 2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4) \end{array}$$

4. 
$$2 \cdot \underline{\hspace{2cm}}$$

# Dialog mechanically generated by CTP-technology

3. 
$$\begin{array}{c} \vdots \\ 2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4) \end{array}$$

4. 
$$2 \cdot \underline{\hspace{2cm}}$$

# Dialog mechanically generated by CTP-technology

3. 
$$\vdots$$
$$2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4)$$

4. 
$$2 \cdot x + \underline{\hspace{2cm}}$$

# Dialog mechanically generated by CTP-technology

3. 
$$\vdots$$
$$2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4)$$

4. 
$$2 \cdot x + \underline{\hspace{2cm}}$$

# Dialog mechanically generated by CTP-technology

$$\vdots$$

3.  $2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4)$

4.  $2 \cdot x + \cos(3 \cdot x^4) ???$

Input checked by a **prover**.

# Dialog mechanically generated by CTP-technology

$$\vdots$$

3.  $2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4)$

$$\frac{d}{dx} \sin(u) = \cos(u) \cdot \frac{d}{dx} \underline{\hspace{1cm}}$$

Provers identify and suggest theorems.

# Dialog mechanically generated by CTP-technology

3. 
$$\begin{array}{c} \vdots \\ 2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4) \end{array}$$

$$\frac{d}{dx} \sin(u) = \cos(u) \cdot \frac{d}{dx} \text{---} ???$$

# Dialog mechanically generated by CTP-technology

3. 
$$\begin{array}{c} \vdots \\ 2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4) \end{array}$$

$$\begin{array}{l} \frac{d}{dx} \cos(x) = -\sin(x) \\ \frac{d}{dx} \sin(u) = \cos(u) \cdot \frac{d}{dx} u \\ \frac{d}{dx} x^n = n \cdot x^{n-1} \end{array}$$

Provers operate on theories comprising theorems.

# Dialog mechanically generated by CTP-technology

$$3. \quad \begin{array}{c} \vdots \\ 2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4) \end{array}$$

$$\begin{aligned} \frac{d}{dx} \cos(x) &= -\sin(x) \\ \frac{d}{dx} \sin(u) &= \cos(u) \cdot \frac{d}{dx} u \\ \frac{d}{dx} x^n &= n \cdot x^{n-1} \end{aligned}$$

# Dialog mechanically generated by CTP-technology

$$3. \quad \begin{array}{c} \vdots \\ 2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4) \end{array}$$

$$\frac{d}{dx} \sin(u) = \cos(u) \cdot \frac{d}{dx} u$$

$$4. \quad 2 \cdot x + \cos(3 \cdot x^4) \cdot \frac{d}{dx} \underline{\hspace{2cm}}$$

Provers use “matching” for fill-in gaps.

# Dialog mechanically generated by CTP-technology

$$\vdots$$

3.  $2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4)$

$$\frac{d}{dx} \sin(u) = \cos(u) \cdot \frac{d}{dx} u$$

4.  $2 \cdot x + \cos(3 \cdot x^4) \cdot \frac{d}{dx} ($ \_\_\_\_\_

# Dialog mechanically generated by CTP-technology

$$\vdots$$

3.  $2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4)$

$$\frac{d}{dx} \sin(u) = \cos(u) \cdot \frac{d}{dx} u$$

4.  $2 \cdot x + \cos(3 \cdot x^4) \cdot \frac{d}{dx} (3$ \_\_\_\_\_



# Dialog mechanically generated by CTP-technology

$$\vdots$$

3.  $2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4)$

$$\frac{d}{dx} \sin(u) = \cos(u) \cdot \frac{d}{dx} u$$

4.  $2 \cdot x + \cos(3 \cdot x^4) \cdot \frac{d}{dx} (3 \cdot \underline{\hspace{1cm}})$

# Dialog mechanically generated by CTP-technology

$$\vdots$$

3.  $2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4)$

$$\frac{d}{dx} \sin(u) = \cos(u) \cdot \frac{d}{dx} u$$

4.  $2 \cdot x + \cos(3 \cdot x^4) \cdot \frac{d}{dx} (3 \cdot x \underline{\hspace{1cm}})$

# Dialog mechanically generated by CTP-technology

$$\vdots$$

3.  $2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4)$

$$\frac{d}{dx} \sin(u) = \cos(u) \cdot \frac{d}{dx} u$$

4.  $2 \cdot x + \cos(3 \cdot x^4) \cdot \frac{d}{dx} (3 \cdot x^4 \_$

# Dialog mechanically generated by CTP-technology

$$\vdots$$

3.  $2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4)$

$$\frac{d}{dx} \sin(u) = \cos(u) \cdot \frac{d}{dx} u$$

4.  $2 \cdot x + \cos(3 \cdot x^4) \cdot \frac{d}{dx} (3 \cdot x^4)$

# Dialog mechanically generated by CTP-technology

$$\vdots$$

$$3. \quad 2 \cdot x + \frac{d}{dx} \sin(3 \cdot x^4)$$

$$\frac{d}{dx} \sin(u) = \cos(u) \cdot \frac{d}{dx} u$$

$$4. \quad 2 \cdot x + \cos(3 \cdot x^4) \cdot \frac{d}{dx} (3 \cdot x^4)$$

Provers check, if a formula can be derived in a context.

## Requirements and Technologies

The requirements can be met by the **technologies**

- ① **check user input** as generous and liberal as possible
  - by Computer Theorem Proving (CTP), i.e
  - Automated Proving (ATP): simplifiers, SAT, SMT etc
  - Interactive Proving (ITP): Coq and Isabelle in EU
- ② **guide the user** step by step towards a solution
  - by **interpretation in debug-mode** with breakpoints
  - ? **resume** after liberal user input at breakpoint (ATP !)
- ③ **explain steps** on request by the user
  -

- 1 Remarks on tutoring software
  - Status quo in the design
  - Requirements on the design
- 2 Base technologies for tutoring
  - Computer theorem proving
  - Single-stepping program interpretation
  - Human readable math knowledge
- 3 Summary and Conclusion

## Requirements and Technologies

The requirements can be met by the **technologies**

- 1 **check user input** as generous and liberal as possible
  - by Computer Theorem Proving (CTP), i.e
  - Automated Proving (ATP): simplifiers, SAT, SMT etc
  - Interactive Proving (ITP): Coq and Isabelle in EU
- 2 **guide the user** step by step towards a solution
  - by interpretation in debug-mode with breakpoints
  - ? resume after liberal user input at breakpoint (ATP !)
- 3 **explain steps** on request by the user
  - by **human-readable knowledge** on a separate language layer.

# Requirements and Technologies

The requirements can be met by the **technologies**

- ① **check user input** as generous and liberal as possible
  - by Computer Theorem Proving (CTP), i.e
  - Automated Proving (ATP): simplifiers, SAT, SMT etc
  - Interactive Proving (ITP): Coq and Isabelle in EU
- ② **guide the user** step by step towards a solution
  - by interpretation in debug-mode with breakpoints
  - ? resume after liberal user input at breakpoint (ATP !)
- ③ **explain steps** on request by the user
  - by **human-readable knowledge** on a separate language layer.

# Demo Isabelle distribution

# Summary

General features and technologies like ...

- 1 **check user input** — comp. theorem proving (CTP)
- 2 **guide the user** — interpretation in debug-mode
- 3 **explain steps** — human-readable knowledge

... provide for novel services in tutoring and authoring.

Austria can contribute significantly !

# Summary

General features and technologies like ...

- 1 **check user input** — comp. theorem proving (CTP)
- 2 **guide the user** — interpretation in debug-mode
- 3 **explain steps** — human-readable knowledge

... provide for novel services in tutoring and authoring.

Austria can contribute significantly !

# Summary

General features and technologies like ...

- ① **check user input** — comp. theorem proving (CTP)
- ② **guide the user** — interpretation in debug-mode
- ③ **explain steps** — human-readable knowledge

... provide for novel services in tutoring and authoring.

Austria can contribute significantly !

# Conclusion ?

Possible Austrian contributions are ...

- ... **interpretation in debug-mode**, *ISAC*+Isabelle
  - TU Graz, Institute for Softwaretechnology
- ... **user-guidance** in single-stepping
  - TU Graz, IICM (H.Maurer)
- ... **Computer Algebra**
  - RISC Linz (B.Buchberger)
- ... **Dynamic Geometry**
  - Uni Linz (GeoGebra, M.Hohenwarter)
- ... **Didactics, teacher training, field tests**
  - ACDCA

Joint development of partners above ?

Start with a course by Ralph-Johan Back ?

## Conclusion ?

Possible Austrian contributions are ...

- ... **interpretation in debug-mode**, *ISAC*+Isabelle
  - TU Graz, Institute for Softwaretechnology
- ... **user-guidance** in single-stepping
  - TU Graz, IICM (H.Maurer)
- ... **Computer Algebra**
  - RISC Linz (B.Buchberger)
- ... **Dynamic Geometry**
  - Uni Linz (GeoGebra, M.Hohenwarter)
- ... **Didactics, teacher training, field tests**
  - ACDCA

Joint development of partners above ?

Start with a course by Ralph-Johan Back ?