Focus Windows: A New Approach to Presenting Mathematical Proofs (in Automated Theorem Proving Systems)

Theorema Technical Report
January 30, 2000

Bruno Buchberger

RISC
Research Institute for Symbolic Computation
Johannes Kepler University
A4232 Schloss Hagenberg, Austria
Buchberger@RISC.Uni-Linz.ac.a

■ Abstract

We describe a new technique for presenting proofs, in particular proofs generated by automated theorem proving systems like THƎOREM∀. We call this technique "focus windows" technique because with this technique, in each proof step, all the relevant formulae are collected in one window (the "focus window") so that the reader can focus on them. The sequence of focus windows alternates between "attention windows" and "transformation windows". In an attention window, exactly those formulae are displayed - and highlighted - that are relevant for the next proof step. In the subsequent transformation window, in addition to the highlighted formulae, the formulae are displayed that are added as a goal or additional knowledge. Also, a standard natural language text is presented that briefly characterizes the proof technique used. Although the paper presents the idea in terms of THƎOREM∀, the presentation method is applicable to arbitrary automated theorem proving systems that produce proofs as sequences of proof situations consisting of "goals" and "available knowledge".

■ The Problem

TH∃OREM∀ is an automated proof generator. Given an initial "proof situation"

$P = \langle G, K \rangle$

consisting of a "goal" G (the formula to be proved) and a "knowledge base" K (a list of formulae assumed, e.g. axioms, definitions, known properties, temporary assumptions, etc.). The provers of TH∃OREM∀ produce proofs that document how the initial proof situation can be reduced, successively, to other proof situations until each remaining proof situation is trivial. A trivial proof situation is either a proof situation whose goal is contained in the knowledge base or a proof situation for which the decision that the goal is a logical consequence of the knowledge base can be made by calling one of the available black-box decision procedures. The trace of such a reduction is accumulated in a "proof object". In a postprocessing step, TH∃OREM∀ produces a human-readable proof from the proof object.

Although the proofs generated by the current post-processors of TH∃OREM∀ include intermediate natural-language explanatory text and the subproofs, subsubproofs, etc. of a given proof can be "clicked" open and closed, studying such proofs (and also proofs produced by humans in math textbooks) still faces the problem that, in each individual proof step, one has to refer to a couple of formulae that may be quite distant from the current location in the proof text. For example, a typical explanatory text may read as follows:

"From (3), by Definition\ 2.1 and Lemma\ 3.2, we now obtain ..."

Even if the references to the labeled formulae are realized by hyperlinks that produce the referenced formulae in an auxiliary window (as this is done in the current version of TH∃OREM∀) studying proofs is still strenuous even when the proofs are technically correct and pedagogically well presented.

■ The Solution: Focus Windows

We develop here a new approach to solving the above problem. We call this approach "Focus Windows Presentation". The main ideas of this approach are:

1. We arrange the entire presentation in such a way that, in each proof situation, all the relevant information fits into one window, the "focus window". Typically, this window should fit into the screen so that, for studying the current proof situation, no scrolling is necessary. This is achieved by displaying, in a given proof situation, only those formulae that are relevant for the transformation of the current proof situation to the next proof situation. (In case the relevant formulae do not fit into one screen, still, some scrolling or, alternatively, shrinking is necessary. However, we believe that if, in a given prover, proof step needs more information than what fits into a screen then something is wrong with the design of the prover.

2. The proof window can be viewed as the "top of the iceberg", where the iceberg is the current proof situation (consisting of the current goal and all the current knowledge) whereas the top consists only of those formulae of the current proof situation that are relevant in the proof step that transforms the current proof situation into the next one. Correspondingly, the focus window alternates between two states: the state of being an "attention

window" and the state of being a "transformation window". In the state "attention window", the window contains - and highlights - those formulae of the current proof situation that are relevant for the present proof step. In the state "transformation window", the window contains, in addition to the highlighted relevant formulae, the new goal and/or new knowledge generated in this proof step.

3. Both in the attention state and in the transformation state, the proof window shows all the formulae in full text rather than only referencing them by labels.

4. In the transformation windows, a natural language text appears that explains how the new goals and/or formulae in the knowledge base are obtained from the highlighted formulae, i.e. which proof rule has been applied.

5. In addition to the formulae, we display in concise form the entire proof tree so that, in addition to being able to check the current proof step, the user can also keep the overview on the global development of the proof. Each node in the proof tree represents a proof situation. By clicking, one can navigate through the tree of proof situations and thereby study the details of the proofs. (Note that, here, we speak about displaying and studying proofs that are already completely generated by TH∃OREM∀ or some other automated theorem proving system. Hence, by clicking, the user only decides which part of the proof he wants to *see and study* at a particular moment. Clicking is *not* a user interaction necessary for *generating* the proof!)

Summarizing, the main point in this new proof presentation technique is that, in a proof situation,

- instead of displaying the entire current knowledge base and referring to the relevant formulae by labels

- we only display the relevant parts of the knowledge base but display this part in full text.

Seen in a different way, our new technique adjusts to the psychological fact that not more (or even less) than what fits into one screen can be realistically processed by the reader. We can consider the usual proofs generated by TH∃OREM∀ (or presented in a textbook) as a long tape of paper and we can view our proof window as a magic viewing glass (a "focusing glass") the reader can move over the tape. In each moment, in the magic glass, not only the formulae right under the glass are shown but also all formulae relevant for the proof step under consideration.

Although quite some proof presentation techniques were proposed in the literature on automatomated theorem proving, see for example (Siekmann 1998), our simple approach proposed in this paper seems to be new.

■ Example of a Proof Problem: Correctness of 'Sorting by Merging'

We present, first, an example of a proof problem of realistic size, namely the correctness proof for the merge-sort algorithm. We used this example also for illustrating the use of "logicographic symbols", see (Buchberger 2000). In the present paper, we do not use logicographic symbols because the technique of focus windows proof presentation is independent of the style used for presenting formulae. However, it is clear that the style of logicographic symbols would go particularly well with the style of dynamic proof presentation.

In this section, we present the proof problem by formulating the goal and the knowledge base. In the next section, we will present the proof in the ordinary "linear and static" style of textbooks, which basically is also the style supported by the current version of TH∃OREM∀. In the final section, we will then present the same proof in the "focus windows proof presentation" style.

We first formulate the goal formula:

Theorem["Correctness of Sorting by Merging", any[A],
istv[st[A], A] "stcorr"]

Readings[any[A, B],
istv[A, B] : ⇋ A "is a sorted version of" B
st[A] : ⇋ "sorting" A "by merging"]

Now we formulate the ingredients of a knowledge base that is sufficient for proving the theorem. Naturally, the new notion of "sorting by merging", by a definition, must be connected with notions (like "merged") that are assumed to be known.

Algorithm["Sorting by Merging", any[A],

$$st[A] := \begin{cases} A & \Leftarrow \quad |A| \leq 1 \\ mg[st[p1[A]], st[p2[A]]] & \Leftarrow \quad \text{otherwise} \end{cases} \quad \text{"st:"}]$$

Readings[any[A, B],
mg[A, B] : ⇋ A "merged with" B
p1[A] : ⇋ "the first part of" A
p2[A] : ⇋ "the second part of" A
|A| : ⇋ "the length of" A]

Furthermore, the relation of the notion that describes the crucial property of "sorting by merging", namely "is a sorted version of" with the notions that are assumed to be known should be "completely" known. The parts of this relation that are actually sufficient for proving the above theorem are listed in the following proposition:

Properties["Prerequisites", any[A, B, C, D],

$(\lvert A\rvert \nleq 1) \Longrightarrow (\lvert A\rvert > \lvert p1[A]\rvert)$	"lg1>"
$(\lvert A\rvert \nleq 1) \Longrightarrow (\lvert A\rvert > \lvert p2[A]\rvert)$	"lg2>"
$ipmv[p1[A] \asymp p2[A], A]$	"p1≍p2"
$istv[A, A]$	"refl"
$(istv[A, B] \wedge ipmv[B, C]) \Longrightarrow istv[A, C]$	"trans"
$(istv[A, B] \wedge istv[C, D]) \Longrightarrow istv[mg[A, C], B \asymp D]$	"mgcorr"

]

Readings[any[A, B],
ipmv[A, B] : ⇋ A "is a permuted version of" B
A ≍ B : ⇋ A "concatenated with" B]

■ Example of Static Proof Presentation

The call

```
Prove[
  Theorem["Correctness of Sorting by Merging"],
  using → ⟨Algorithm["Sorting by Merging"], Properties["Prerequisites"]⟩,
  by → course–of–value–induction–on–length]
```

in the current version of TH∃OREM∀ will (should), roughly, generate the following proof:

Proof: We apply course-of-value induction on the length. Let **A** be arbitrary but fixed and assume

$$\underset{|A|<|\mathbf{A}|}{\forall_{A}} \mathrm{istv}[\mathrm{st}[A], A]. \qquad \text{(indhyp)}$$

We have to show

$$\mathrm{istv}[\mathrm{st}[\mathbf{A}], \mathbf{A}]. \qquad \text{(indgoa } \mathbf{A}\text{l)}$$

Case $|\mathbf{A}|\leq 1$: In this case, by (st:), (indgoal) reduces to

$$\mathrm{istv}[\mathbf{A}, \mathbf{A}], \qquad \text{(indgoal } \mathbf{A}\text{ .1)}$$

which is true by (refl).

Case $|\mathbf{A}|\not\leq 1$: In this case, by (st:), (indgoal) reduces to

$$\mathrm{istv}[\mathrm{mg}[\mathrm{st}[\mathrm{p1}[\mathbf{A}]], \mathrm{st}[\mathrm{p2}[\mathbf{A}]]], \mathbf{A}]. \qquad \text{(indgoal } \mathbf{A}\text{ .2)}$$

From the case assumption, by (lg1>) and (lg2>), we obtain

$$|\mathrm{p1}[\mathbf{A}]| < |\mathbf{A}|, \qquad (1.1)$$

$$|\mathrm{p2}[\mathbf{A}]| < |\mathbf{A}|. \qquad (1.2)$$

From (1.1) and (1.2), by (indhyp), we obtain

$$\mathrm{istv}[\mathrm{st}[\mathrm{p1}[\mathbf{A}]], \mathrm{p1}[\mathbf{A}]], \qquad (2.1)$$

$$\mathrm{istv}[\mathrm{st}[\mathrm{p2}[\mathbf{A}]], \mathrm{p2}[\mathbf{A}]], \qquad (2.2)$$

From (2.1) and (2.2), by (mgcorr), we obtain

$$\mathrm{istv}[\mathrm{mg}[\mathrm{st}[\mathrm{p1}[\mathbf{A}]], \mathrm{st}[\mathrm{p2}[\mathbf{A}]]], \mathrm{p1}[\mathbf{A}] \asymp \mathrm{p2}[\mathbf{A}]]. \qquad (3)$$

Now, from (3), by (p1≍p2) and (trans), we obtain (indgoal **A**.2). □

Note that the proof is relatively short and easy to read because the knowledge base was constructed in a careful way (according to the "complete exploration" principle introduced in (Buchberger 1999)). Nevertheless, in the individual proof steps, quite some jumping between formulae and memorizing the exact text of formulae is still necessary. For example, in the step

"From (2.1) and (2.2), by (mgcorr), we obtain

istv[mg[st[p1[**A**]], st[p2[**A**]]], p1[**A**] ≍ p2[**A**]] (3) ",

although formulae (2.1) and (2.2) are next to the current proof text, we have to jump to the position of formula (3) in order to inspect the structure of this formula and to memorize it for being able to check that the substitution of the constant '**A**' for the variable 'A' was done correctly.

■ Example of a Proof Presentation Using Focus Windows

We explain the principle of dynamic proof presentation by displaying the sequence of windows that should be produced in our example. Note that, in this paper, the proof windows produced are displayed in a linear sequence of sections. In the actual implementation, there will always be only one window open on the screen. It is just this fact, that only one window has to be studied at a time and no reference to any other window is necessary, which makes our approach new and hopefully better than all currently existing approaches to proof presentation.

Each window has an area for the goal(s), an area for the knowledge, and an area for displaying the current state of the proof tree. The nodes of the proof tree are either ● (proof situation that is already processed), ∧ (pending proof situation in a vertical column of proof situations all of which must yield true), or ∨ (pending proof situation in a vertical column of proof situations at least one of which must yield true). In fact, when presenting only the "successful" branches of a proof, we will never encounter proof situations marked by ∨. By clicking into a node, the corresponding proof situation is displayed in the proof window. Typically, one will click into the node highlighted, e.g. ∧. This will produce a "natural" sequence of proof steps.

In an attention window, the formulae relevant for the next proof step are highlighted, e.g.

$$\underset{A}{\forall}\ \mathrm{st}[A] := \begin{cases} A & \Leftarrow\ |A| \leq 1 \\ \mathrm{mg}[\mathrm{st}[\mathrm{p1}[A]], \mathrm{st}[\mathrm{p2}[A]]] & \Leftarrow\ \text{otherwise} \end{cases} \qquad (\mathrm{st}:)$$

In a transformation windwow, in addition to the (highlighted) formulae in the preceding attention window, the (goal and knowledge) formulae produced in this proof step are inserted. New goal(s) appear below the highlighted goal whereas new knowledge appears above the highlighted knowledge so that the new formulae are grouped together in the central part of the window.

In order to obtain a good understanding of what dynamical proof presentation means for the user, you should open only one window in the sequence of windows below. The information in each of the windows should be sufficient to understand what is going on in the corresponding proof step. In fact, it would be sufficient to open and study only the transformation windows because each transformation window completely contains the information in the preceding attention window. However, in the sequel, we display the entire sequence of attention and transformation window in order to convey the flavor of the behavior of the proposed system in which, in the focus window, an attention window will be produced for "setting the stage" and give the user the possibility to "think" about what proof step could be applied in this situation. In the succeeding transformation window, the user will see what proof step is applied in the proof generated by the system.

■ Initial Window

Goal:

$$\underset{A}{\forall} \text{istv}[\text{st}[A], A] \qquad \text{(stcorr)}$$

.____.

∧

■ Attention Window 1

Goal:

$$\underset{A}{\forall} \text{istv}[\text{st}[A], A] \qquad \text{(stcorr)}$$

.____.

Algorithm Definition:

$$\underset{A}{\forall} \text{st}[A] := \begin{cases} A & \Leftarrow \quad |A| \leq 1 \\ \text{mg}[\text{st}[\text{p1}[A]], \text{st}[\text{p2}[A]]] & \Leftarrow \quad \text{otherwise} \end{cases} \qquad \text{(st :)}$$

Properties:

$$\underset{A}{\forall} (|A| \nleq 1) \Longrightarrow (|A| > |\text{p1}[A]|) \qquad \text{(lg1)}$$

$$\underset{A}{\forall} (|A| \nleq 1) \Longrightarrow (|A| > |\text{p2}[A]|) \qquad \text{(lg2)}$$

.____.

∧

▪ Transformation Window 1

As suggested by the structure of definition (st:) and properties (lg1), (lg2), we apply course-of-value induction on the length. Let ***A*** *be arbitrary but fixed:*

Prove:

$\underset{A}{\forall}$ istv[st[A], A] (stcorr)

Induction Goal:

istv[st[**A**], **A**] (indgoal **A**)

Induction Hypothesis:

$\underset{\substack{A \\ |A|<|\mathbf{A}|}}{\forall}$ istv[st[A], A] (indhyp **A**)

Algorithm Definition:

$$\underset{A}{\forall}\ st[A] := \begin{cases} A & \Leftarrow |A| \leq 1 \\ mg[st[p1[A]], st[p2[A]]] & \Leftarrow \text{otherwise} \end{cases}$$ (st :)

Properties:

$\underset{A}{\forall}\ (|A| \nleq 1) \Longrightarrow (|A| > |p1[A]|)$ (lg1)

$\underset{A}{\forall}\ (|A| \nleq 1) \Longrightarrow (|A| > |p2[A]|)$ (lg2)

● ∧

▪ Attention Window 2

Goal:

istv[st[**A**], **A**]. (indgoal **A**)

.____.

Algorithm Definition:

$$\underset{A}{\forall}\ st[A] := \begin{cases} A & \Leftarrow\ |A| \le 1 \\ mg[st[p1[A]], st[p2[A]]] & \Leftarrow\ \text{otherwise} \end{cases} \qquad (st:)$$

.____.

- ∧

■ Transformation Window 2

As suggested by the structure of definition (st:), we consider two cases:

.____

Goal:

istv[st[**A**], **A**] (indgoal)

.____

Algorithm Definition:

$$\underset{A}{\forall}\ st[A] := \begin{cases} A & \Leftarrow\ |A| \le 1 \\ mg[st[p1[A]], st[p2[A]]] & \Leftarrow\ \text{otherwise} \end{cases} \qquad (st:)$$

First Case:

|**A**| ≤ 1 (c1)

.____

- - ∧
∧

■ Attention Window 3

Goal:

istv[st[**A**], **A**] (indgoal)

.____

Algorithm Definition:

$$\underset{A}{\forall}\ st[A] := \begin{cases} A & \Leftarrow\ |A| \leq 1 \\ mg[st[p1[A]], st[p2[A]]] & \Leftarrow\ \text{otherwise} \end{cases} \qquad (st:)$$

Case Assumption:

|**A**| ≤ 1 (c1)

.____

● ● ∧
∧

■ Transformation Window 3

By the case assumption and the algorithm definition (st:), the goal (indgoal ***A****) is reduced to a new goal:*

.____

Induction Goal:

istv[st[**A**], **A**] (indgoal **A**)

New Goal:

istv[**A**, **A**] (indgoal **A** .1)

.____

Algorithm Definition:

$$\underset{A}{\forall}\ st[A] := \begin{cases} A & \Leftarrow\ |A| \leq 1 \\ mg[st[p1[A]], st[p2[A]]] & \Leftarrow\ \text{otherwise} \end{cases} \qquad (st:)$$

Case Assumption:

|**A**| ≤ 1. (c1)

. ___

● ● ● ∧
∧

■ Attention Window 4

Goal:

istv[**A**, **A**] (indgoal **A** .1)

. ___

Property:

$\forall_{A}$ istv[A, A] (refl)

. ___

● ● ● ∧
∧

■ Transformation Window 4

By property (refl), the goal is proved.

. ___

Goal:

istv[**A**, **A**] (indgoal **A** .1)

. ___

Property:

$\forall_{A}$ istv[A, A] (refl)

. ___ .

● ● ● ●
∧

■ Attention Window 5 (Backup to Attention Window 2)

Goal:

istv[st[**A**], **A**] (indgoal **A**)

.____.

Algorithm Definition:

$$\underset{A}{\forall}\ st[A] := \begin{cases} A & \Leftarrow\ |A| \leq 1 \\ mg[st[p1[A]], st[p2[A]]] & \Leftarrow\ \text{otherwise} \end{cases}$$ (st :)

.____

● ● ● ●

∧

■ Transformation Window 5

As suggested by the structure of definition (st:), we consider two cases.

.____

Goal:

istv[st[**A**], **A**] (indgoal **A**)

.____

Algorithm Definition:

$$\underset{A}{\forall}\ st[A] := \begin{cases} A & \Leftarrow\ |A| \leq 1 \\ mg[st[p1[A]], st[p2[A]]] & \Leftarrow\ \text{otherwise} \end{cases}$$ (st :)

Second Case:

$|\mathbf{A}| \nleq 1$ (c2)

.____

● ● ● ●

● ∧

■ Attention Window 6

Goal:

istv[st[**A**], **A**] (indgoal **A**)

.____

Algorithm Definition:

$$\underset{A}{\forall}\ \mathrm{st}[A] := \begin{cases} A & \Leftarrow\ |A| \le 1 \\ \mathrm{mg}[\mathrm{st}[\mathrm{p1}[A]], \mathrm{st}[\mathrm{p2}[A]]] & \Leftarrow\ \text{otherwise} \end{cases}$$ (st :)

Case Assumption:

|**A**| ≰ 1 (c2)

.____

● ● ● ●
● ∧

■ Transformation Window 6

*By the case assumption and the algorithm definition the goal (indgoal **A**) is reduced to a new goal:*

.____

Goal:

istv[st[**A**], **A**] (indgoal **A**)

New Goal:

istv[mg[st[p1[**A**]], st[p2[**A**]]], **A**] (indgoal **A** .2)

.____

Algorithm Definition:

$$\underset{A}{\forall}\ \mathrm{st}[A] := \begin{cases} A & \Leftarrow\ |A| \le 1 \\ \mathrm{mg}[\mathrm{st}[\mathrm{p1}[A]], \mathrm{st}[\mathrm{p2}[A]]] & \Leftarrow\ \text{otherwise} \end{cases}$$ (st :)

Case Assumption:

$|\mathbf{A}| \nleq 1$ (c2)

.____

● ● ● ●
● ● ∧

■ Attention Window 7

Goal:

.____

Case Assumption:

$|\mathbf{A}| \nleq 1$ (c2)

Properties:

$\underset{A}{\forall} \; (|A| \nleq 1) \Longrightarrow (|A| > |p1[A]|)$ (lg1)

$\underset{A}{\forall} \; (|A| \nleq 1) \Longrightarrow (|A| > |p2[A]|)$ (lg2)

.____

● ● ● ●
● ● ∧

■ Transformation Window 7

From the case assumption and properties (lg1), (lg2), we obtain new knowledge:

.____

Goal:

.____

New Knowledge:

$|\mathbf{A}| > |p1[\mathbf{A}]|$ (1.1)

$|\mathbf{A}| > |p2[\mathbf{A}]|$ (1.2)

Case Assumption:

$|\mathbf{A}| \nleq 1$ (c2)

Properties

$\underset{A}{\forall} \; (|A| \nleq 1) \Longrightarrow (|A| > |p1[A]|)$ (lg1)

$\underset{A}{\forall} \; (|A| \nleq 1) \Longrightarrow (|A| > |p2[A]|)$ (lg2)

.____

● ● ● ●
● ● ● $\wedge$

■ Attention Window 8

Goal:

.____

Knowledge:

$|\mathbf{A}| > |p1[\mathbf{A}]|$ (1.1)

$|\mathbf{A}| > |p2[\mathbf{A}]|$ (1.2)

Induction Hypothesis:

$\underset{\substack{A \\ |A| < |\mathbf{A}|}}{\forall} \; istv[st[A], A]$ (indhyp **A**)

.____

● ● ● ●
● ● ● $\wedge$

■ Transformation Window 8

From (1.1) and (1.2), by the induction hypothesis (indhyp ***A****), we obtain new knowledge:*

.____

Goal:

.____

New Knowledge:

istv[st[p1[**A**]], p1[**A**]] (2.1)

istv[st[p2[**A**]], p2[**A**]] (2.2)

Knowledge:

|**A**| > |p1[**A**]| (1.1)

|**A**| > |p2[**A**]| (1.2)

Induction Hypothesis:

$\forall_{\substack{A \\ |A|<|\mathbf{A}|}}$ istv[st[A], A] (indhyp **A**)

.____

● ● ● ●

● ● ● ● ∧

■ Attention Window 9

Goal:

.____

Knowledge:

istv[st[p1[**A**]], p1[**A**]] (2.1)

istv[st[p2[**A**]], p2[**A**]] (2.2)

Property:

$\forall_{A,B,C,D}$ (istv[A, B] ∧ istv[C, D]) ⟹ istv[mg[A, C], B ≍ D] (mgcorr)

.___

● ● ● ●
● ● ● ● ∧

■ Transformation Window 9

From (2.1) and (2.2), by (mgcorr), we obtain new knowledge:

.___

Goal:

.___

New Knowledge:

istv[mg[st[p1[**A**]], st[p2[**A**]]], p1[**A**] ≍ p2[**A**]] (3)

Knowledge:

istv[st[p1[**A**]], p1[**A**]] (2.1)

istv[st[p2[**A**]], p2[**A**]] (2.2)

Property:

$\forall_{A,B,C,D}$ (istv[A, B] ∧ istv[C, D]) ⟹ istv[mg[A, C], B ≍ D] (mgcorr)

.___

● ● ● ●
● ● ● ● ● ∧

■ Attention Window 10

Goal:

istv[mg[st[p1[**A**]], st[p2[**A**]]] , **A**] (indgoal **A** .2)

.____

Knowledge:

istv[mg[st[p1[**A**]], st[p2[**A**]]], p1[**A**] ≍ p2[**A**]] (3)

Property:

$\forall_{A}$ ipmv[p1[A] ≍ p2[A], A] (p1 ≍ p2)

$\forall_{A,B,C}$ (istv[A, B] ∧ ipmv[B, C]) ⟹ istv[A, C] (trans)

.____

● ● ● ●
● ● ● ● ● ∧

■ Transformation Window 10

From (3), by (p1≍p2) *and (trans), we obtain the goal:*

.____

Goal:

istv[mg[st[p1[**A**]], st[p2[**A**]]] , **A**] (indgoal **A** .2)

.____

Knowledge:

istv[mg[st[p1[**A**]], st[p2[**A**]]], p1[**A**] ≍ p2[**A**]] (3)

Property:

$$\underset{A}{\forall}\ \mathrm{ipmv}[\mathrm{p1}[A] \asymp \mathrm{p2}[A], A] \qquad (\mathrm{p1} \asymp \mathrm{p2})$$

$$\underset{A,B,C}{\forall}\ (\mathrm{istv}[A, B] \wedge \mathrm{ipmv}[B, C]) \Longrightarrow \mathrm{istv}[A, C] \qquad (\mathrm{trans})$$

.____

● ● ● ●
● ● ● ● ● ●

■ References

(Buchberger 1999) B. Buchberger. *Theorem Proving Versus Theory Exploration.* Invited talk at the Calculemus Meeting, Trento, July 1999.

(Buchberger 2000) B. Buchberger. *Logicographic Symbols: Some Examples of Their Use in Formal Proofs.* Manuscript, January 2000, RISC, A4040 Schloss Hagenberg, Austria.

(Siekmann 1998). J. Siekmann et al. *The Omega System*. DFKI, Saarbrücken.