Theorema: The Induction Prover over Lists

Bruno Buchberger, Daniela Vasaru

Research Institute for Symbolic Computation
A4232 Schloss Hagenberg, Austria
Bruno.Buchberger@risc.uni−linz.ac.at
Daniela.Vasaru@risc.uni−linz.ac.at

Abstract: We describe the inductive prover for equalities over lists implemented in the *Theorema* system. The structure of the prover is very similar to the one of the induction prover over natural numbers. The main difference is the use of "sequence" variables for handling lists. The prover recurs over the universally quantified sequence variables in the equality to be proven and uses the equalities in the knowledge base for rewriting. The order used in rewriting is a recursive path ordering.

1. Quick Introduction by Examples

We start presenting our prover by means of examples. This should help in understanding for which type of problems and how the induction prover over lists can be used. A formal description of the problem addressed and the method used is given in sections 2 and 3. As the implementation is very similar to the one of the induction prover for natural numbers, see (Buchberger 1997b), we only give some details in section 4. Further improvements that we plan to introduce in our prover are given in section 5. Annexa I contains a list of properties of operations over lists that can be proved with the help of our prover. An example of a double induction proof automatically generated by our prover is given in Annexa II.

1.1. Notations

For a detailed description of the language in the *Theorema* project, see (Buchberger, Kriftner 1997). Here we only give a list of the notations that we use in this prover:

◇ equality is denoted by "≡";
◇ lists are enclosed in angle brackets "$\langle\rangle$";
◇ overbars are used for denoting sequence symbols: $\bar{x}$;
◇ if x is a list, "$\langle\bar{x}\rangle$" is called its **sequence representation**;
◇ labeled formulae are written in the form: `⟨ label,formula⟩,` where `label` is a string;
◇ the quantified variables are written in a list subscripted to the quantifier, even if there is only one variable.

We use "new" function symbols for denoting functions, i.e. function symbols that do not have any built−in meaning in Mathematica, in order to avoid that any knowledge other than the one explicitly stated is used in the proof.

1.2. Example of a Call of the Prover

Suppose we want to prove that, under the assumptions

$$
\begin{array}{ll}
(\frown) & \forall_{\langle \bar{y}, z\rangle}\ \langle\bar{y}\rangle \frown z \equiv \langle\bar{y}, z\rangle, \\
(|\langle\rangle|) & |\langle\rangle| \equiv 0, \\
(|\langle\ .\ ,\ \ldots\rangle|) & \forall_{\langle x, \bar{y}\rangle}\ |\langle x, \bar{y}\rangle| \equiv |\langle\bar{y}\rangle|^{+}\ .
\end{array}
$$

also the following formula is true

$$\text{(T)} \qquad \forall_{\langle x,y\rangle}\ |x \frown y| \equiv |x|^{+}$$

where "$\frown$" denotes the operation of appending an element to a list, "$|x|$" denotes the length of a list x and "$^{+}$" denotes the successor function for natural numbers. The equality $(\frown)$ defines the append operation and the next two equalities define inductively the length of a list.

For this we call our prover in the following way:

$$\text{Proof-by-induction}[\langle \text{"T"},\ \forall_{\langle x,y\rangle}\ |x \frown y| \equiv |x|^{+}\rangle,$$
$$\langle\langle \text{"}\frown\text{"},\ \forall_{\langle \bar{y},z\rangle}\ \langle \bar{y}\rangle \frown z \equiv \langle \bar{y},\ z\rangle\rangle,$$
$$\langle \text{"}|\langle\rangle|\text{"},\ |\langle\ \rangle| \equiv 0\rangle,$$
$$\langle \text{"}|\langle .,\ldots\rangle|\text{"},\ \forall_{\langle x,\bar{y}\rangle}\ |\langle x,\ \bar{y}\rangle| \equiv |\langle \bar{y}\rangle|^{+}\rangle\rangle]$$

The output is a notebook containing the proof shown "in linear form" in the following subsection but with additional cell brackets so that various parts of the proof can be clicked "open" or "closed" in order to make it easier for keeping an overview on the proof. More details on the "nested cell representation" of a proof can be found in (Buchberger 1997a).

1.3. A Proof Notebook

PROPOSITION:

(T) $\forall_{\langle x,y\rangle}\ |x \frown y| \equiv |x|^{+}$

under the assumptions

($\frown$) $\forall_{\langle \bar{y},z\rangle}\ \langle \bar{y}\rangle \frown z \equiv \langle \bar{y},\ z\rangle$,

($|\langle\rangle|$) $|\langle\ \rangle| \equiv 0$,

($|\langle .,\ldots\rangle|$) $\forall_{\langle x,\bar{y}\rangle}\ |\langle x,\ \bar{y}\rangle| \equiv |\langle \bar{y}\rangle|^{+}$.

PROOF:

We represent list variables in sequence notation.

We try to prove by simplification:

(T.V) $\forall_{\langle \bar{x},y\rangle}\ |\langle \bar{x}\rangle \frown y| \equiv |\langle \bar{x}\rangle|^{+}$

For this, we take all variables arbitrary but fixed and try to prove

(T.V.F) $|\langle \overline{x_0}\rangle \frown y_0| \equiv |\langle \overline{x_0}\rangle|^{+}$

An attempt for proving this by simplification fails.

Simplification of the lhs term:

$|\langle \overline{x_0}\rangle \frown y_0| =$ by ($\frown$)

$|\langle \overline{x_0},\ y_0\rangle|$ ⌋

Simplification of the rhs term:

$|\langle \overline{x_0}\rangle|^{+}$ ⌋

Now we prove the simplified formula

(T.V.S) $\forall_{\langle \bar{x},y\rangle}\ |\langle \bar{x},\ y\rangle| \equiv |\langle \bar{x}\rangle|^{+}$

by induction on $\bar{x}$.

Induction base for $\bar{x}$:

We prove

(T.V.S.IB) $\forall_{\langle y \rangle} \; |\langle y \rangle| \equiv |\langle\rangle|^{+}.$

For this, we take in (T.V.S.IB) all variables arbitrary but fixed and prove

(T.V.S.IB.F) $|\langle y_0 \rangle| \equiv |\langle\rangle|^{+}$

A proof by simplification works.

Simplification of the lhs term:

$|\langle y_0 \rangle| =$ by ($|\langle .,\ldots\rangle|$)

$|\langle\rangle|^{+} \rfloor$

Simplification of the rhs term:

$|\langle\rangle|^{+} \rfloor$

Induction hypothesis for $\bar{x}$:

(T.V.S.IH) $\forall_{\langle y \rangle} \; |\langle \overline{x_0}, y \rangle| \equiv |\langle \overline{x_0} \rangle|^{+}.$

Induction step for $\bar{x}$:

We prove

(T.V.S.IS) $\forall_{\langle y \rangle} \; \left|\left\langle \overset{\bullet}{x}_0, \overline{x_0}, y \right\rangle\right| \equiv \left|\left\langle \overset{\bullet}{x}_0, \overline{x_0} \right\rangle\right|^{+}.$

For this, we take in (T.V.S.IS) all variables arbitrary but fixed and prove

(T.V.S.IS.F) $\left|\left\langle \overset{\bullet}{x}_0, \overline{x_0}, y_0 \right\rangle\right| \equiv \left|\left\langle \overset{\bullet}{x}_0, \overline{x_0} \right\rangle\right|^{+}$

A proof by simplification works.

Simplification of the lhs term:

$\left|\left\langle \overset{\bullet}{x}_0, \overline{x_0}, y_0 \right\rangle\right| =$ by ($|\langle .,\ldots\rangle|$)

$|\langle \overline{x_0}, y_0 \rangle|^{+} =$ by (T.V.S.IH)

$(|\langle \overline{x_0} \rangle|^{+})^{+} \rfloor$

Simplification of the rhs term:

$\left|\left\langle \overset{\bullet}{x}_0, \overline{x_0} \right\rangle\right|^{+} =$ by ($|\langle .,\ldots\rangle|$)

$(|\langle \overline{x_0} \rangle|^{+})^{+} \rfloor$

1.4. Remarks about the Proof

The proof shown in the previous subsection is given in natural language. The goal, assumptions and labels are in different colors for easier understanding. Note also that we generate some systematic extensions for the labels of the formulae: "V" for "variant", "F" for "arbitrary but fixed", "S" for "simplified", "IB" for "induction base", "IH" for "induction hypothesis" and "IS" for "induction step".

One of the most important features of the proof is its presentation in nested cells. This allows the reader to compress entire subproofs by just clicking "closed" the corresponding cell. Thus, one can decide by itself to which level of detail one wants to study the proof. We give now an example of one of the possible views of the above proof where the cells corresponding to simplification are "closed":

PROPOSITION:

(T) $\forall_{\langle x,y\rangle} |x \frown y| \equiv |x|^{+}$

under the assumptions

($\frown$) $\forall_{\langle \bar{y},z\rangle} \langle \bar{y}\rangle \frown z \equiv \langle \bar{y}, z\rangle$,

($|\langle\rangle|$) $|\langle\,\rangle| \equiv 0$,

($|\langle .,\ldots\rangle|$) $\forall_{\langle x,\bar{y}\rangle} |\langle x, \bar{y}\rangle| \equiv |\langle \bar{y}\rangle|^{+}$.

PROOF:

We represent list variables in sequence notation.

We try to prove by simplification:

(T.V) $\forall_{\langle \bar{x},y\rangle} |\langle \bar{x}\rangle \frown y| \equiv |\langle \bar{x}\rangle|^{+}$

For this, we take all variables arbitrary but fixed and try to prove

(T.V.F) $|\langle \overline{x_0}\rangle \frown y_0| \equiv |\langle \overline{x_0}\rangle|^{+}$

An attempt for proving this by simplification fails.

Now we prove the simplified formula

(T.V.S) $\forall_{\langle \bar{x},y\rangle} |\langle \bar{x}, y\rangle| \equiv |\langle \bar{x}\rangle|^{+}$

by induction on $\bar{x}$.

Induction base for $\bar{x}$:

We prove

(T.V.S.IB) $\forall_{\langle y\rangle} |\langle y\rangle| \equiv |\langle\,\rangle|^{+}$.

For this, we take in (T.V.S.IB) all variables arbitrary but fixed and prove

(T.V.S.IB.F) $|\langle y_0\rangle| \equiv |\langle\,\rangle|^{+}$

A proof by simplification works.

Induction hypothesis for $\bar{x}$:

(T.V.S.IH) $\forall_{\langle y\rangle} |\langle \overline{x_0}, y\rangle| \equiv |\langle \overline{x_0}\rangle|^{+}$.

Induction step for $\bar{x}$:

We prove

(T.V.S.IS) $\forall_{\langle y\rangle} \left|\left\langle \dot{x}_0, \overline{x_0}, y\right\rangle\right| \equiv \left|\left\langle \dot{x}_0, \overline{x_0}\right\rangle\right|^{+}$.

For this, we take in (T.V.S.IS) all variables arbitrary but fixed and prove

(T.V.S.IS.F) $\left|\left\langle \dot{x}_0, \overline{x_0}, y_0\right\rangle\right| \equiv \left|\left\langle \dot{x}_0, \overline{x_0}\right\rangle\right|^{+}$

A proof by simplification works.

1.1.5. Generation of the Proof Object

The proof can also be generated in its "abstract form" as a "proof object", i.e. as a nested Mathematica expression, by specifying the additional parameter "proof object":

Proof-by-induction[$\langle$"T", $\forall_{\langle x,y\rangle}$ $|y \frown x| \equiv |y|^{+}\rangle$,
$\langle\langle$ "$\frown$", $\forall_{\langle \bar{y},z\rangle}$ $\langle \bar{y}\rangle \frown z \equiv \langle \bar{y}, z\rangle\rangle$, $\langle$ "$|\langle\rangle|$", $|\langle\rangle| \equiv 0\rangle$,
$\langle$ "$|\langle .,\ldots\rangle|$", $\forall_{\langle x,\bar{y}\rangle}$ $|\langle x, \bar{y}\rangle| \equiv |\langle \bar{y}\rangle|^{+}\rangle\rangle$, "proof object"]

2. The Problem

This prover solves the following type of problems:

Given: G, a formula of the form

$\forall_{\langle x, y, \ldots\rangle}$ lhs $\equiv$ rhs,

where lhs and rhs are terms containing functions inductively defined by equalities on the domain of lists (over some domain) and x, y, … are variables, some of which range over lists ,

A, a finite sequence of formulae of this form, which typically define a couple of functions over lists inductively and also express some properties of these functions that are already proved..

Find: P, an inductive / simplification proof of G from A (simplification proof if G could be proved just by simplification)
or
an indication of where the proof fails, if it could not prove G.

3. The Method

3.1. Preprocessing

We start by preprocessing the problem in the following way:

◇ In the goal formula G the list variables are represented in sequence notation. Example:

$\forall_{\langle x,y\rangle}$ $|x \frown y| \equiv |x|^{+}$

is transformed into:

$\forall_{\langle \bar{x},y\rangle}$ $|\langle \bar{x}\rangle \frown y| \equiv |\langle \bar{x}\rangle|^{+}$

◇ The assumptions A are transformed into rewrite rules. We use a recursive path ordering in creating the rewrite rules. For more details on how the rewrite rules are generated, see (Buchberger, Marin, Tomuta 1997).

3.2. Induction Principle for Sequences

The induction principle for lists is derived from the following induction principle for sequence variables. If we want to prove:

$\forall_{\langle \bar{x}\rangle}$ G

where G contains $\bar{x}$ unquantified, proceed as follows:

Induction Base: Prove $G_{\bar{x}\leftarrow}$ (i.e. the formula that results from G by replacing $\bar{x}$ by nothing).

Induction Hypothesis: Choose $\dot{x}_0$ and $\overline{x_0}$ ''arbitrary but fixed'' (i.e. introduce a fresh object symbol $\dot{x}_0$ and a fresh sequence symbol $\overline{x_0}$) and assume $G_{\bar{x}\leftarrow \overline{x_0}}$.

Induction Step: Prove $G_{\bar{x} \leftarrow \dot{x}_0, \overline{x_0}}$.

Example: If we want to prove by induction on $\bar{x}$:

$$\forall_{\langle \bar{x}, y\rangle} |\langle \bar{x}, y\rangle| \equiv |\langle \bar{x}\rangle|^{+}$$

we have:

◇ induction base : $\forall_{\langle y\rangle} |\langle y_0\rangle| \equiv |\langle\rangle|^{+}$
◇ induction hypothesis : $\forall_{\langle y\rangle} |\langle \overline{x_0}, y\rangle| \equiv |\langle \overline{x_0}\rangle|^{+}$
◇ induction step : $\forall_{\langle y\rangle} \left|\left\langle \dot{x}_0, \overline{x_0}, y\right\rangle\right| \equiv \left|\left\langle \dot{x}_0, \overline{x_0}\right\rangle\right|^{+}$.

3.3. Proving by Multiple Induction on Lists

Suppose we have a goal G of the form:

$$\forall_{\langle t, \bar{x}, \bar{y}, z\ldots\rangle} \mathrm{lhs}[t, \bar{x}, \bar{y}, z, \ldots] \equiv \mathrm{rhs}[t, \bar{x}, \bar{y}, z, \ldots]$$

and the assumptions are given as rewrite rules A. We try to prove G in the following steps:

Simplification: First, we take t_0, $\overline{x_0}$, $\overline{y_0}$, z_0, … arbitrary but fixed and try to prove

$$\mathrm{lhs}[t_0, \overline{x_0}, \overline{y_0}, z_0, \ldots] \equiv \mathrm{rhs}[t_0, \overline{x_0}, \overline{y_0}, z_0, \ldots]$$

by simplification (rewriting) both sides modulo A. Simplification (rewriting) of equalities modulo equalities plays a crucial role in this prover. We are using a recursive path ordering in the simplification process. For more details on the simplification prover, see (Buchberger, Marin, Tomuta 1997).

If the simplified forms

$$\mathrm{Slhs}[t_0, \overline{x_0}, \overline{y_0}, z_0, \ldots]$$

and

$$\mathrm{Srhs}[t_0, \overline{x_0}, \overline{y_0}, z_0, \ldots]$$

of the two sides are identical, we are done. Otherwise we start an induction on $\bar{x}$:

Induction Base: We try to prove

$$\forall_{\langle t, \bar{y}, z, \ldots\rangle} \mathrm{Slhs}[t, \bar{y}, z, \ldots] \equiv \mathrm{Srhs}[t, \bar{y}, z, \ldots]$$

by recursively calling the prover again (but with one less universally quantified sequence variable!). If this proof fails, we report failure. (In fact we report that

$$\forall_{\langle t, \bar{y}, z, \ldots\rangle} \mathrm{Slhs}[t, \bar{y}, z, \ldots] \equiv \mathrm{Srhs}[t, \bar{y}, z, \ldots]$$

cannot be proved.) If the proof works, we go to the induction step.

Induction hypothesis: We take $\dot{x}_0$, $\overline{x_0}$ arbitrary but fixed and assume:

$$\forall_{\langle t, \bar{y}, z, \ldots\rangle} \mathrm{Slhs}[t, \overline{x_0}, \bar{y}, z, \ldots] \equiv \mathrm{Srhs}[t, \overline{x_0}, \bar{y}, z, \ldots].$$

Induction step: Now we try to prove

$$\forall_{\langle t, \bar{y}, z, \ldots\rangle} \mathrm{Slhs}\left[t, \left(\dot{x}_0, \overline{x_0}\right), \bar{y}, z, \ldots\right] \equiv \mathrm{Srhs}\left[t, \left(\dot{x}_0, \overline{x_0}\right), \bar{y}, z, \ldots\right]$$

by recursively calling the prover again. (Again we have one less universally quantified sequence variable!). If this proof fails, we report failure. (In fact we report that

$$\forall_{\langle t, \bar{y}, z, \ldots\rangle} \mathrm{Slhs}\left[t, \left(\dot{x}_0, \overline{x_0}\right), \bar{y}, z, \ldots\right] \equiv \mathrm{Srhs}\left[t, \left(\dot{x}_0, \overline{x_0}\right), \bar{y}, z, \ldots\right].$$

cannot be proved.) If the proof works, we are done.

3.4. Remarks

At present, the prover proceeds recursively over the universally quantified sequence variables, taking them in the order they are given. No heuristic is used for determining a "good" induction variable. Thus, the proof might succeed / fail, be shorter / longer depending on the order the variables are given.

Sometimes the proof fails because of a lack of knowledge. Looking at the output, the user can get the idea of what other lemmas are needed for proving the theorem and might try to prove those first.

3.5. Abstract Proof Objects and External Proof Representations

The structure of the proof object generated by this prover is very similar to the one generated by the induction prover over natural numbers, see (Buchberger 1997b). A description of the structure of the proof objects can be found in (Buchberger, Jebelean 1997). A proof object is further processed in order to obtain the natural language version of the proof (in nested cell presentation), see (Buchberger 1997a).

4. Implementation Details

4.1. Calls for Proof by Induction

The name of the prover is "`Proof-by-induction`". It calls the function "`ProofByInduction`" which is the main program for creating the proof. We define various versions of "`Proof-by-induction`" for special and general arguments. The principal two arguments of the prover are the labeled universally quantified equality to be proved and a list of labeled equalities that form the knowledge base. The third argument is used for an indication on the type of representation for the proof generated.

If no indication on the type of output is given or, alternatively, the third argument is "nested cells" then the proof generated is given, in a new notebook, in nested cells representation:

```
Clear[Proof-by-induction];
Proof-by-induction[ labeledFormula_, labeledEqualityList_] :=
Proof-by-induction[ labeledFormula, labeledEqualityList,
"nested cells"]

Proof-by-induction[ labeledFormula_, labeledEqualityList_,
"nested cells"] :=
NotebookPut[
Notebook[ 〈
ProofCells[
ProofByInduction[ labeledFormula, labeledEqualityList]]〉,
CellGrouping -> Manual]]
```

If the third, optional, argument is "nested cells notebook" then the notebook expression for the proof in nested cells representation is produced in the notebook from which the prover is called. This may be useful for test purposes.

```
Proof-by-induction[ labeledFormula_ , labeledEqualityList_,
"nested cells notebook"] :=
 Notebook[ 〈
ProofCells[
ProofByInduction[ labeledFormula, labeledEqualityList]]〉,
CellGrouping -> Manual]
```

If the third, optional, argument is "proof object" then the abstract proof object for the proof is returned. This is necessary if the proof should be processed further in one way or the other, for example if the user wants to analyze the proof for possible shortcuts.

```
Proof-by-induction[ labeledFormula_, labeledEqualityList_,
"proof object"] :=
 ProofByInduction[ labeledFormula, labeledEqualityList]
```

4.2. The Main Program for Induction

"`ProofByInduction`" is the main program for generating the proof object. After a preprocessing part consisting in representing list variables in sequence notation, transforming the equalities from the knowledge base into rewrite rules, it gives the control to "`PI`". "`PI`" is the actual call to the induction rule and it is applied recursively over the universally quantified variables in the equality to be proved.

4.3. Natural Language Proof

"`ProofCells`" is the program that generates a natural language proof (in nested cells representation) from a proof object. The details on how this is done can be found in (Buchberger 1997a).

5. Future Work

The present state of the induction prover for equalities over lists in the *Theorema* project was presented. The prover is not in the final phase. Several improvements of the prover are on the way: finding an heuristic for determining a "good" induction variable, automatizing the detection of the necessary lemmas in the proof.

References

(Buchberger 1997a) B. Buchberger. *Theorema: Natural Language and Nested Cells Representation of Proofs.* In Proceedings of the *Theorema* Workshop, 9–10 June 1997, Hagenberg, Austria.

(Buchberger 1997b) B. Buchberger *The Theorema Prover for Equalities over the Natural Numbers.* In Proceedings of the *Theorema* Workshop, 9–10 June 1997, Hagenberg, Austria.

(Buchberger, Kriftner 1997) B. Buchberger, F. Kriftner *Theorema: The Language.* In Proceedings of the *Theorema* Workshop, 9–10 June 1997, Hagenberg, Austria.

(Buchberger, Jebelean 1997) B. Buchberger, T. Jebelean *Theorema: The Predicate Prover.* In Proceedings of the *Theorema* Workshop, 9–10 June 1997, Hagenberg, Austria.

(Buchberger, Marin, Tomuta 1997) B. Buchberger, M. Marin, E. Tomuta *A Simplification Prover for Noninductive Functors.* Technical Report *Theorema*–97–01–14. Research Institute for Symbolic Computation, A–4232 Schloss Hagenberg, Austria.

Appendix I: A Theory of Lists

A list of the properties of lists operations proven with our prover is given below. Not all were proven only from the definitions, for some of them we had to use lemmas proved before. The meaning of the symbols used is: "$\smile$" – prepend, "$\frown$"– append, "$\asymp$"–concatenate, "$|\,|$"–length, "_"–reversed, "`car`"–head, "`cdr`"–tail, "`cadr`"–second element, "`last`"–last element, "`rotate`"–circular permutation, "$\oplus$"–addition of natural numbers, "$^{+}$"–succesor function for natural numbers.

$$\forall_{\langle x,y,z\rangle}\ (x \smile (y \frown z)) \equiv ((x \smile y) \frown z)$$

$$\forall_{\langle x,y,z\rangle}\ (z \smile (x \asymp y)) \equiv ((z \smile x) \asymp y)$$

$$\forall_{\langle x,y\rangle}\ (x \smile \underline{y}) \equiv \underline{(y \frown x)}$$

$$\forall_{\langle x,y,z\rangle}\ ((x \asymp y) \frown z) \equiv (x \asymp (y \frown z))$$

$$\forall_{\langle x, y\rangle} \ (\underline{x} \frown y) \equiv \underline{(y \smile x)}$$

$$\forall_{\langle x\rangle} \ \langle\rangle \asymp x \equiv x$$

$$\forall_{\langle x\rangle} \ x \asymp \langle\rangle \equiv x$$

$$\forall_{\langle x,y\rangle} \ (\langle x\rangle \asymp y) \equiv (x \smile y)$$

$$\forall_{\langle x,y\rangle} \ (x \asymp \langle y\rangle) \equiv (x \frown y)$$

$$\forall_{\langle x,y,z\rangle} \ ((x \frown y) \asymp z) \equiv (x \asymp (y \smile z))$$

$$\forall_{\langle x, y, z\rangle} \ (x \asymp (y \frown z)) \equiv ((x \asymp y) \frown z)$$

$$\forall_{\langle x,y,z\rangle} \ (x \asymp (y \asymp z)) \equiv ((x \asymp y) \asymp z)$$

$$\forall_{\langle x,y\rangle} \ (x \asymp \underline{y}) \equiv \underline{(y \asymp \underline{x})}$$

$$\forall_{\langle \bar{x},\bar{y},z\rangle} \ |\langle \bar{x}, z, \bar{y}\rangle| \equiv (|\langle \bar{x}\rangle| \oplus |\langle \bar{y}\rangle|)^{+}$$

$$\forall_{\langle \bar{x},\bar{y},z\rangle} \ |\langle \bar{x}, z, \bar{y}\rangle| \equiv (|\langle \bar{x}, \bar{y}\rangle|)^{+}$$

$$\forall_{\langle x,y\rangle} \ |x \smile y| \equiv |y|^{+}$$

$$\forall_{\langle x,y\rangle} \ |y \frown x| \equiv |y|^{+}$$

$$\forall_{\langle x,y\rangle} \ |x \asymp y| \equiv |x| \oplus |y|$$

$$\forall_{\langle y\rangle} \ |\underline{y}| \equiv |y|$$

$$\forall_{\langle x,y\rangle} \ (\underline{x \asymp y}) \equiv (\underline{y} \asymp \underline{x})$$

$$\forall_{\langle x\rangle} \ \underline{\underline{x}} \equiv x$$

$$\forall_{\langle x\rangle} \ \underline{\underline{\underline{x}}} \equiv \underline{x}$$

$$\forall_{\langle x\rangle} \ \mathrm{car}[x] \smile \mathrm{cdr}[x] \equiv x$$

$$\forall_{\langle x,y\rangle} \ \mathrm{car}[x \smile y] \equiv x$$

$$\forall_{\langle x,y\rangle} \ \mathrm{cdr}[x \smile y] \equiv y$$

$$\forall_{\langle x,y\rangle} \ x \asymp y \equiv \mathrm{car}[x] \smile (\mathrm{cdr}[x] \asymp y)$$

$$\forall_{\langle x\rangle} \ \underline{x} \equiv \underline{\mathrm{cdr}[x]} \frown \mathrm{car}[x]$$

$$\forall_{\langle x,y\rangle} \ \mathrm{last}[x \frown y] \equiv y$$

$$\forall_{\langle x\rangle} \ \mathrm{last}[\underline{x}] \equiv \mathrm{car}[x]$$

$$\forall_{\langle x,y\rangle} \ \mathrm{cadr}[x \smile y] \equiv \mathrm{car}[y]$$

$$\forall_{\langle x,y\rangle} \ \mathrm{rotate}[|x|, x \asymp y] \equiv (y \asymp x)$$

$$\forall_{\langle \bar{x},\bar{y}\rangle} \ \mathrm{rotate}[|\langle \bar{x}\rangle|, \langle \bar{x}, \bar{y}\rangle] \equiv \langle \bar{y}, \bar{x}\rangle$$

$$\forall_{\langle x\rangle} \ \mathrm{rotate}[|x|, x] \equiv x$$

$$\forall_{\langle x, \bar{y}, \bar{z}, t\rangle} \ ((x \smile (\langle \bar{y}\rangle \asymp \langle \bar{z}\rangle)) \frown t) \equiv ((x \smile \langle \bar{y}\rangle) \asymp (\langle \bar{z}\rangle \frown t))$$

$$\forall_{\langle \bar{x},\bar{y},\bar{z}\rangle} \ |(\langle \bar{x}\rangle \asymp \langle \bar{y}\rangle) \asymp \langle \bar{z}\rangle| \equiv (|\langle \bar{x}\rangle| \oplus |\langle \bar{y}\rangle|) \oplus |\langle \bar{z}\rangle|$$

$$\forall_{\langle x,y\rangle} \underline{\underline{(x \asymp y)}} \equiv \left(\underset{=}{x} \asymp \underset{=}{y}\right)$$

$$\forall_{\langle x,y,z\rangle} \lvert (x \frown z) \asymp y\rvert \equiv (\lvert x\rvert \oplus \lvert y\rvert)^{+}$$

$$\forall_{\langle x,y\rangle} \left\lvert \underline{(x \asymp y)}\right\rvert \equiv (\lvert x\rvert \oplus \lvert y\rvert)$$

$$\forall_{\langle x,y\rangle} \underline{(\underset{=}{x} \asymp \langle\rangle)} \equiv x$$

$$\forall_{\langle x,y\rangle} \left(\underset{=}{x} \asymp y\right) \equiv \underline{\underline{(x \asymp y)}}$$

$$\forall_{\langle \bar{x},\bar{y},\bar{z}\rangle} \lvert (\langle\bar{x}\rangle \asymp \langle\bar{y}\rangle) \asymp \langle\bar{z}\rangle\rvert \equiv (\lvert\langle\bar{x}\rangle\rvert \oplus \lvert\langle\bar{y}\rangle\rvert) \oplus \lvert\langle\bar{z}\rangle\rvert$$

$$\forall_{\langle \bar{x},\bar{y}\rangle} \lvert \langle\bar{x}\rangle \asymp \langle\bar{y}\rangle\rvert \equiv \lvert \langle\bar{y}\rangle \asymp \langle\bar{x}\rangle\rvert$$

Appendix II: Example of Double Induction Proof

PROPOSITION:

($\asymp$_) $\forall_{\langle x,y\rangle} (x \asymp \underline{y}) \equiv \underline{y \asymp \underline{x}}$

under the assumptions

($\smile$) $\forall_{\langle x,\bar{y}\rangle} x \smile \langle\bar{y}\rangle \equiv \langle x, \bar{y}\rangle$,

($\smile\frown$) $\forall_{\langle x,y,z\rangle} x \smile y \frown z \equiv (x \smile y) \frown z$,

($\asymp$) $\forall_{\langle \bar{x},\bar{y}\rangle} (\langle\bar{x}\rangle \asymp \langle\bar{y}\rangle) \equiv \langle\bar{x}, \bar{y}\rangle$,

($\langle\rangle\asymp$) $\forall_{\langle x\rangle} (\langle\rangle \asymp x) \equiv x$,

(_$\langle\rangle$) $\underline{\langle\rangle} \equiv \langle\rangle$,

(_$\langle .,\ldots\rangle$) $\forall_{\langle x,\bar{y}\rangle} \underline{\langle x, \bar{y}\rangle} \equiv \underline{\langle\bar{y}\rangle} \frown x$,

($\frown\asymp$) $\forall_{\langle x,y,z\rangle} (x \asymp y) \frown z \equiv (x \asymp y \frown z)$,

($\smile$_) $\forall_{\langle x,y\rangle} x \smile \underline{y} \equiv \underline{y \frown x}$.

PROOF:

We represent list variables in sequence notation.

We try to prove by simplification:

($\asymp$_.V) $\forall_{\langle \bar{x},\bar{y}\rangle} (\langle\bar{x}\rangle \asymp \underline{\langle\bar{y}\rangle}) \equiv \underline{\langle\bar{y}\rangle \asymp \underline{\langle\bar{x}\rangle}}$

For this, we take all variables arbitrary but fixed and try to prove

($\asymp$_.V.F) $(\langle\overline{x_0}\rangle \asymp \underline{\langle\overline{y_0}\rangle}) \equiv \underline{\langle\overline{y_0}\rangle \asymp \underline{\langle\overline{x_0}\rangle}}$

An attempt for proving this by simplification fails.

Simplification of the lhs term:

$\langle\overline{x_0}\rangle \asymp \underline{\langle\overline{y_0}\rangle}\rfloor$

Simplification of the rhs term:

$\underline{\langle\overline{y_0}\rangle \asymp \underline{\langle\overline{x_0}\rangle}}\rfloor$

Now we prove the formula

(≍_.V) $\forall_{\langle\bar{x},\bar{y}\rangle} \; (\langle\bar{x}\rangle \asymp \underline{\langle\bar{y}\rangle}) \equiv \underline{\underline{\langle\bar{y}\rangle \asymp \underline{\langle\bar{x}\rangle}}}$

by induction on $\bar{x}$.

Induction base for $\bar{x}$:

We prove

(≍_.V.IB) $\forall_{\langle\bar{y}\rangle} \; (\langle\rangle \asymp \underline{\langle\bar{y}\rangle}) \equiv \underline{\underline{\langle\bar{y}\rangle \asymp \underline{\langle\rangle}}}$.

For this, we take in (≍_.V.IB) all variables arbitrary but fixed and prove

(≍_.V.IB.F) $(\langle\rangle \asymp \underline{\langle\overline{y_0}\rangle}) \equiv \underline{\underline{\langle\overline{y_0}\rangle \asymp \underline{\langle\rangle}}}$

A proof by simplification works.

Simplification of the lhs term:

$\langle\rangle \asymp \underline{\langle\overline{y_0}\rangle} =$ by (⟨⟩≍)

$\underline{\langle\overline{y_0}\rangle}\rfloor$

Simplification of the rhs term:

$\underline{\langle\overline{y_0}\rangle \asymp \underline{\langle\rangle}} =$ by (_⟨⟩)

$\underline{\langle\overline{y_0}\rangle \asymp \langle\rangle} =$ by (≍)

$\underline{\langle\overline{y_0}\rangle}\rfloor$

Induction hypothesis for $\bar{x}$:

(≍_.V.IH) $\forall_{\langle\bar{y}\rangle} \; (\langle\overline{x_0}\rangle \asymp \underline{\langle\bar{y}\rangle}) \equiv \underline{\langle\bar{y}\rangle \asymp \underline{\langle\overline{x_0}\rangle}}$.

Induction step for $\bar{x}$:

We try to prove by simplification:

(≍_.V.IS) $\forall_{\langle\bar{y}\rangle} \; \left(\left\langle \dot{x}_0, \overline{x_0} \right\rangle \asymp \underline{\langle\bar{y}\rangle}\right) \equiv \underline{\langle\bar{y}\rangle \asymp \underline{\left\langle \dot{x}_0, \overline{x_0} \right\rangle}}$

For this, we take all variables arbitrary but fixed and try to prove

(≍_.V.IS.F) $\left(\left\langle \dot{x}_0, \overline{x_0} \right\rangle \asymp \underline{\langle\overline{y_0}\rangle}\right) \equiv \underline{\langle\overline{y_0}\rangle \asymp \underline{\left\langle \dot{x}_0, \overline{x_0} \right\rangle}}$

An attempt for proving this by simplification fails.

Simplification of the lhs term:

$\left\langle \dot{x}_0, \overline{x_0} \right\rangle \asymp \underline{\langle\overline{y_0}\rangle}\rfloor$

Simplification of the rhs term:

$\underline{\langle\overline{y_0}\rangle \asymp \underline{\left\langle \dot{x}_0, \overline{x_0} \right\rangle}} =$ by (_⟨.,…⟩)

$\underline{\langle\overline{y_0}\rangle \asymp \underline{\langle\overline{x_0}\rangle} \frown \dot{x}_0} =$ by (⌢≍)

$\underline{(\langle\overline{y_0}\rangle \asymp \underline{\langle\overline{x_0}\rangle}) \frown \dot{x}_0} =$ by (⌣_)

$\dot{x}_0 \smile \underline{\langle\overline{y_0}\rangle \asymp \underline{\langle\overline{x_0}\rangle}} =$ by (≍_.V.IH)

$\dot{x}_0 \smile (\langle\overline{x_0}\rangle \asymp \underline{\langle\overline{y_0}\rangle})\rfloor$

Now we prove the simplified formula

(≍_.V.IS.S) $\forall_{\langle \bar{y} \rangle} \left(\left\langle \dot{x}_0 ,\ \overline{x_0} \right\rangle \asymp \underline{\langle \bar{y} \rangle} \right) \equiv \dot{x}_0 \smile (\langle \overline{x_0} \rangle \asymp \underline{\langle \bar{y} \rangle})$

by induction on $\bar{y}$.

Induction base for $\bar{y}$:

(≍_.V.IS.S.IB) $\left(\left\langle \dot{x}_0 ,\ \overline{x_0} \right\rangle \asymp \underline{\langle \rangle} \right) \equiv \dot{x}_0 \smile (\langle \overline{x_0} \rangle \asymp \underline{\langle \rangle})$

A proof by simplification works.

Simplification of the lhs term:

$\left\langle \dot{x}_0 ,\ \overline{x_0} \right\rangle \asymp \underline{\langle \rangle} =$ by (_⟨⟩)

$\left\langle \dot{x}_0 ,\ \overline{x_0} \right\rangle \asymp \langle \rangle =$ by (≍)

$\left\langle \dot{x}_0 ,\ \overline{x_0} \right\rangle \rfloor$

Simplification of the rhs term:

$\dot{x}_0 \smile (\langle \overline{x_0} \rangle \asymp \underline{\langle \rangle}) =$ by (_⟨⟩)

$\dot{x}_0 \smile (\langle \overline{x_0} \rangle \asymp \langle \rangle) =$ by (≍)

$\dot{x}_0 \smile \langle \overline{x_0} \rangle =$ by (⌣)

$\left\langle \dot{x}_0 ,\ \overline{x_0} \right\rangle \rfloor$

Induction hypothesis for $\bar{y}$:

(≍_.V.IS.S.IH) $\left(\left\langle \dot{x}_0 ,\ \overline{x_0} \right\rangle \asymp \underline{\langle \overline{y_0} \rangle} \right) \equiv \dot{x}_0 \smile (\langle \overline{x_0} \rangle \asymp \underline{\langle \overline{y_0} \rangle})$.

Induction step for $\bar{y}$:

(≍_.V.IS.S.IS) $\left(\left\langle \dot{x}_0 ,\ \overline{x_0} \right\rangle \asymp \underline{\left\langle \dot{y}_0 ,\ \overline{y_0} \right\rangle} \right) \equiv \dot{x}_0 \smile \left(\langle \overline{x_0} \rangle \asymp \underline{\left\langle \dot{y}_0 ,\ \overline{y_0} \right\rangle} \right)$

A proof by simplification works.

Simplification of the lhs term:

$\left\langle \dot{x}_0 ,\ \overline{x_0} \right\rangle \asymp \underline{\left\langle \dot{y}_0 ,\ \overline{y_0} \right\rangle} =$ by (_⟨.,…⟩)

$\left\langle \dot{x}_0 ,\ \overline{x_0} \right\rangle \asymp \underline{\langle \overline{y_0} \rangle \frown \dot{y}_0} =$ by (⌢≍)

$\left(\left\langle \dot{x}_0 ,\ \overline{x_0} \right\rangle \asymp \underline{\langle \overline{y_0} \rangle} \right) \frown \dot{y}_0 =$ by (≍_.V.IS.S.IH)

$\left(\dot{x}_0 \smile (\langle \overline{x_0} \rangle \asymp \underline{\langle \overline{y_0} \rangle}) \right) \frown \dot{y}_0 =$ by (⌣⌢)

$\dot{x}_0 \smile (\langle \overline{x_0} \rangle \asymp \underline{\langle \overline{y_0} \rangle}) \frown \dot{y}_0 \rfloor$

Simplification of the rhs term:

$\dot{x}_0 \smile \left(\langle \overline{x_0} \rangle \asymp \underline{\left\langle \dot{y}_0 ,\ \overline{y_0} \right\rangle} \right) =$ by (_⟨.,…⟩)

$\dot{x}_0 \smile \left(\langle \overline{x_0} \rangle \asymp \underline{\langle \overline{y_0} \rangle \frown \dot{y}_0} \right) =$ by (⌢≍)

$\dot{x}_0 \smile (\langle \overline{x_0} \rangle \asymp \underline{\langle \overline{y_0} \rangle}) \frown \dot{y}_0 \rfloor$