A New Structural Induction Scheme for Proving Properties of Mutually Recursive Concepts

Peiya Liu
Siemens Research and Technology Laboratories
105 College Road East
Princeton, NJ 08540
Tel: 609-734-3349

ABSTRACT
Structural induction schemes have been used for mechanically proving properties of self-recursive concepts in previous research. However, based on those schemes, it becomes very difficult to automatically generate the right induction hypotheses whenever the conjectures are involved with mutually recursive concepts. This paper will show that the difficulties come mainly from the weak induction schemes provided in the past, and a strong induction scheme is needed for the mutually defined concepts. Furthermore, a generalized induction principle is provided to smoothly integrate both schemes. Thus, in this mechanical induction, hypotheses are generated by mixing strong induction schemes with weak induction schemes. While the weak induction schemes are suggested by self-recursive concepts, the strong induction schemes are suggested by mutually recursive concepts.

I. Introduction
There are currently two good ways of programming based on formal logic, namely: (1) programs based on recursive functions, such as LISP, and (2) programs based on non-recursive relations, such as PROLOG. Structural induction schemes have been provided for proving properties of self-recursive functions [Bourbakie 68] [Hurstall 69] [Drozdz 74] [Boyer 75] [Aspin 76] [Cartwright 76] [Boyer 76] and of self-recursive relations [Clark 77] [Brown & Liu 86] [Brown 86] [Liu 86]. Both schemes are applicable to recursively defined data objects such as natural numbers, lists, and trees [Hoare 72] [Boyer 70]. However, it is hard to apply these schemes to mutually recursive functions and relations. In this paper, a new structural induction scheme is introduced for proving properties of mutually recursive concepts. In addition, we shall show that the new scheme can be smoothly integrated with the old scheme in a generalized structural induction principle.

II. Self-Recursive and Mutually Recursive Concepts
Before formally stating the recursive concepts, some definitions are necessary. S is a term if it is a variable, a sequence of a function symbol of n arguments followed by n terms, or a sequence of a universal quantifier ALL or existential quantifier EX of two arguments followed by a variable and a term. The scope of a quantifier occurring in the term is the subterm to which the quantifier applies. For example, the scope of the quantifier ALL in the term \((\text{ALL } X(\text{FOO } X Y))\) is \((\text{FOO } X Y)\). A variable is free in the term if at least one occurrence of it is not within the scope of a quantifier employing the variable. A term \(t\) governs an occurrence of term \(s\) if either there is a subterm \((\text{IF } t \ p \ q)\) and the occurrence of \(s\) is in \(p\), or there is a subterm \((\text{IF } t' \ p \ q)\) and the occurrence of \(s\) in \(q\), where \(t'\) is \((\text{NOT } t)\). A term \(t\) is \(f\)-free if the symbol \(f\) does not occur in the term as a function symbol. \((\text{ALL-LIST } (x_1 \ldots x_k x_{k+1} \ldots x_n))\) is an abbreviation for \((\text{ALL } x_1 (\text{ALL } x_2 (\ldots (\text{ALL } x_k (\ldots \text{ALL } x_p))))\) (EX-LIST(x1 ... xk)) for \((\text{EX } x_1 (\text{EX } x_2 (\ldots (\text{EX } x_p))))\), and \((\text{ALL } (x_1 \ldots x_p))\) for a sequence of \(n\) mixed quantifiers over \(p\), its negated form \((\text{EX } x_1 \ldots x_p)\) (NOT p)). NIL is considered to be false and \(T\) denotes true. The symbols EQUAL and IF are two primitive operators. Informally speaking, if \(X\) is NIL, then \((X \ Y Z)\) is equal to \(Z\), and if \(X\) is not NIL, then \((X \ Y Z)\) is equal to \(Y\). The logic operators AND, IMPLIES, OR, and NOT can also be represented by IF formulae.

The recursive concepts are formally defined as follows.

\[\text{DEFQ} \]
\[\text{EQUAL}(f_1 x_1 \ldots x_k x_{k+1} \ldots x_{n_1} \ldots x_{n_1} n_2) \text{ body}_1\]
\[\text{EQUAL}(f_2 x_1 \ldots x_k x_{k+1} \ldots x_{n_2}) \text{ body}_2\]
\[\ldots\]
\[\text{EQUAL}(f_n x_1 \ldots x_k x_{n+1} \ldots x_{n_2} n_n) \text{ body}_n)\]

where \(f_1 \ldots f_n\) are new function symbols of \(n_1 \ldots n_n\) arguments, respectively, and \(1 \leq k \leq n_i\) for \(1 \leq i \leq n\);

\((A)\) \(x_1 \ldots x_k\) are distinct variables;

\((B)\) \(x_1 \ldots x_k\) for \(1 \leq i \leq n\) are
distinct variables;

\((C)\) \(\text{body}_i\) for \(1 \leq i \leq n\) is a term and only
 mentions free variables in \(x_1, \ldots, x_k\),\]
\(x_{i+1} \ldots x_{n_1} n_2\);

\((D)\) there is a well-founded relation \(r\) and a measure function \(m\) of \(k\) arguments;

\((E)\) for each occurrence of a subterm of \((f_1 x_1 \ldots x_n)\), \(1 \leq i \leq n\) in the \(\text{body}_i\), \(1 \leq i \leq n\), it is a
terms that:
\[\text{ALL-LIST} (x_1 \ldots x_k x_{k+1} \ldots x_n)\]
\[\text{ALL-LIST} (x_k x_{k+1} \ldots x_n)\]
\[\text{IMPLIES}(\text{AND} t_1 \ldots t_p)\]
\[r (m y_1 \ldots y_p) (m x_1 \ldots x_p))\]

The definition principle is to describe that \(n\) axioms constitute a recursive definition of some concept. \(n\) axioms of the form:
\[(f_1 x_1 \ldots x_k x_{k+1} \ldots x_{n_1} n_2) = \text{body}_1, \ldots, (f_n x_1 \ldots x_k x_{n+1} \ldots x_{n_2} n_n) = \text{body}_n)\]
\(f_n x_1 \ldots x_k x_{n+1} \ldots x_{n_2} n_n\)
can be shown to be recursive if, according to the same measure \(m\), the complexity of the axioms of every occurrence of \(f_i\) in any \(\text{body}_i\), assuming the hypotheses governing the occurrences in the \(\text{body}_i\) is less than the complexity of \(x_1 \ldots x_k\).

The purpose of requirement \((E)\) in the definition principle is to make recursive concepts terminate, and further, to avoid an inconsistency problem. Note that if \(m=1\), then the principle of definition defines a self-recursive concept.
Example Q0:

```
(DEFQ
  (EQUAL (EVAL (CAR L) ENVRN) 
  (APPLY SUBR (CAR L) 
  (EVAL (CDR L) ENVRN)) 
  L))
```

Example Q1:

```
(DEFQ
  (EQUAL (QSORT R Z W1 W2) 
  (IF (LISTP Z) 
  (EX X (EX Y (IF (PART (CAR Z) (CAR X) X Y) 
  (EX V (IF (QSORT R X W1 (CONS (CAR Z) V)) 
  (QSORT R Y V W2) 
  NIL)) 
  NIL))) 
  NIL)))
```

III. A Generalized Structural Induction Principle

A. Why Strong Induction Schemes are Needed

Essentially mechanical induction reasoning works because the similarity could be contrived between the structures of the recursive definition functions and of the induction schemes. The structures of recursive functions serve as templates for automatically generating the suitable induction hypotheses to prove a conjecture involved with those recursive functions. However, there is often no structure similarity between mutually recursive functions and weak induction schemes provided in previous research. The finite number of hypotheses are needed to be specified explicitly in the weak induction schemes. Using these weak induction schemes often results in the generation of useless induction hypotheses for the conjecture involved with mutually recursive functions. A strong induction form will be shown to be needed and can be generated from the structures of mutually recursive concepts. In the strong induction schemes, the finite number of hypotheses are implicitly described by particular recursive concepts.

An example will illustrate the problem in using weak induction schemes for hypothesis generation. Suppose we try to prove the conjecture (ALL L (EQUAL L (FOO L))), where the mutually recursive functions are defined as follows.

```
(DEFQ
  (EQUAL (FOO L) 
  (IF (LISTP L) 
  (CONS (CAR L) (FOO (CDR L)))) 
  L))
```

```
(DEFQ
  (EQUAL (FOOLIST L) 
  (IF (LISTP L) 
  (CONS (FOO (CAR L)) (FOOLIST (CDR L)))) 
  L))
```

Let (p L) be the term (EQUAL L (FOO L)). In the weak induction schemes, the instantiated terms of (p L) as induction hypotheses are required to be explicitly described. Thus, the induction hypothesis, based on the weak induction scheme and the structure of (FOO L), could be (AND (LISTP L) (p (CDR L))) or (AND (LISTP L) (p (CAR L))). However, if we open the term (p L) in the proof by induction, it will not look like its counterpart in the hypotheses, and the hypotheses will be useless.

Even if we change these functions into a self-recursive function with an extra argument S as follows, our problem still exists.

```
(DEFQ
  (EQUAL (FOOS L S) 
  (IF (EQUAL S 0) 
  (CONS (CAR L) (FOOS (CDR L)))) 
  L))
```

Suppose that (PART I.C I.I 1.2) is true if I.1 is a list of elements of I. less than C, and I.2 is a list of the rest of L. For example, suppose C is 6 and L is a list (2 6 3 0 10), then L.1 is (2 3) and L.2 is (6 0 10). The quick sort concept could be defined as follows.

```
(EX X (EX Y (IF (PART (CAR Z) (CAR X) X Y) 
  (EX V (IF (QSORT R X W1 (CONS (CAR Z) V)) 
  (QSORT R Y V W2) 
  NIL)) 
  NIL))
```

In the predicate (QSORT R Z W1 W2), Z is an input list and the output is the difference list of W1 and W2, which is an ordered list Z. The QSORT.R could be added to the system because (ALL Z (ALL W1 W2 (ALL X (ALL Y (ALL V (IMPLIES (AND (LISTP Z) (PART (CAR Z) (CAR Z) X Y)) (PLESSP (LENGTH X W1 (CONS (CAR Z) V) (LENGTH Z W1 W2)))))))))) and (ALL Z (ALL W1 W2 (ALL X (ALL Y (ALL V (IMPLIES (AND (LISTP Z) (PART (CAR Z) (CAR Z) X Y)) (PLESSP (LENGTH X Y V W2) (LENGTH2 Z W1 W2))))))) hold.
An induction scheme, following the weak induction principle, for the conjecture (ALL (EQUAL (FOOLIST L) (AND (LISTP L) (FOOLIST-IND L)))) will be generated as (ALL (EQUAL (FOO L) (AND (LISTP L) (FOO-IND L)))).

Later on, we will give detailed descriptions of automatically constructing the terms (FOOLIST-IND L) and (FOO-IND L) from mutually recursive concepts. (FOO-IND L) and (FOO-L) are mutually defined as

\[(\text{AND} (\text{LISTP} L) (\text{FOOLIST-IND} (\text{CDR} L)))\]

Intuitively, the term (\text{AND} (\text{LISTP} L) (\text{FOOLIST-IND} (\text{CDR} L))) is actually ANDing the terms (\text{LISTP} L), (\text{p} (\text{CAR} (\text{CDR} L))), \ldots, (\text{p} (\text{CAR}) R L)) together by recursively opening up the term (\text{FOOLIST-IND} (\text{CDR} L)).

Thus, this hypothesis implicitly represents a series of instantiated conjectures and this induction form is actually a strong induction scheme. More importantly, there is an obvious structural similarity between (\text{FOO-IND} L) and (\text{FOOLIST-IND} (\text{CDR} L)).

Later on, we will give detailed descriptions of automatically constructing the terms (\text{FOOLIST-IND} L) and (\text{FOO-IND} L) from mutually recursive concepts. (\text{FOOLIST-IND} (\text{CDR} L)) is obtained from the body of (\text{FOO-IND} L) since (\text{FOO L}) appears in the conjecture, and the corresponding term (\text{FOO-IND} L) suggests the possible induction hypotheses from the recursive structure of its body.

### C. Hypothesis Terms

Intuitively, hypothesis terms are those terms allowable to be instantiated as hypothesis in the strong induction schemes. These terms are quite powerful. They can implicitly represent a series of induction hypotheses in mechanical induction proof about the properties of mutually recursive concepts. A formal definition of hypothesis terms is described as follows. A subterm is a call of \(s\) in the term \(t\) if the subterm beginning with the function symbol \(s\) occurs in the term \(t\). (\(f_1 x_1 \cdots x_n x_{n+1} \cdots x_t\), \(f_2 x_1 \cdots x_n x_{n+1} \cdots x_t\), \ldots) are the hypothesis terms of \(f_1, f_2, \ldots, f_t\) with \(P_0\) replacing \(f_1, f_2, \ldots, f_t\), if

1. \(f_1, \ldots, f_t\) are the following mutually recursive functions based on a well-founded relation \(R\) and a measure function \(M\) of a arguments,
   - (\text{EQUAL}(f_1 x_1 \cdots x_n x_{n+1} \cdots x_t) body_1)...
   - (\text{EQUAL}(f_t x_1 \cdots x_n x_{n+1} \cdots x_t) body_t)

2. \((P_0 x_1 \cdots x_n x_{n+1} \cdots x_t)\) is a term; and

3. \((P_1 x_1 \cdots x_n x_{n+1} \cdots x_t), \ldots, (P_d x_1 \cdots x_n x_{n+1} \cdots x_t)\) are obtained in the following way.
   - (\text{EQUAL}(P_1 x_1 \cdots x_n x_{n+1} \cdots x_t) body_1')
   - (\text{EQUAL}(P_2 x_1 \cdots x_n x_{n+1} \cdots x_t) body_2')
   - \ldots
   - (\text{EQUAL}(P_d x_1 \cdots x_n x_{n+1} \cdots x_t) body_d')

where body' = (\(HT body\)) for \(1 \leq i \leq d\) and \(HT\) is recursively defined as follows:

- Suppose the term \(s\) has the form (\(\text{ALL-EX(z)}\) v), then (\(HT s\)) = (\(\text{ALL-EX(s)}(HT v)\))
- Suppose the term \(s\) has the form (\(\text{IF} c (HT u)(HT v)\)). Then (\(HT s\)) = (\(IF c (HT u)(HT v)\)) if the term \(c\) is \(f_1\)-free, otherwise.
- Suppose the term \(s\) is \(f_1\)-free, \(1 \leq i \leq d\), then (\(HT s\)) = (\(T\))
- Suppose \(s\) is a term obtained by replacing every occurrence of \(f_1\) (for \(1 \leq i \leq k\)) as a function symbol in the term \(s\) with the symbol \(P_0\) and by replacing every occurrence of \(f_j\) (for \(j < k\)) as a function symbol in the term \(s\) with the symbol \(P_k\). Then (\(HT s\)) = (\(AND \text{alls of } P_i\) for \(0 \leq i \leq d\) in the term \(s\)) if there is more than one call of \(P_i\), then (\(HT s\)) = (\(call of P_i\) for \(0 \leq i \leq d\) in the term \(s\)).
- Suppose \(s\) is a term obtained by replacing every occurrence of \(f_1\) (for \(1 \leq i \leq k\)) as a function symbol in the term \(s\) with the symbol \(P_0\) and by replacing every occurrence of \(f_j\) (for \(j < k\)) as a function symbol in the term \(s\) with the symbol \(P_k\). Then (\(HT s\)) = (\(AND \text{alls of } P_i, P_k\) for \(0 \leq i \leq d\) in the term \(s\)) if only one call exists.

**Example Q3**: To find out the hypothesis terms of \(\text{FOO}\) and \(\text{FOOLIST}\) with \(P_0\) replacing \(\text{FOO}\).

\[
(\text{EQUAL}(P_1 L) (\text{IF} (\text{LISTP L}) (P_0 (\text{CDR} L)) (T)))
\]

\[
(\text{EQUAL}(P_0 L) (\text{IF} (\text{LISTP L}) (\text{AND}(P_0 (\text{CAR} L)) (P_2 (\text{CDR} L))) (T)))
\]

From the bodies of \(\text{FOO}\) and \(\text{FOOLIST}\), the hypothesis terms \(P_1 L\).
A formal description of the generalized induction principle is contained in Appendix I. The key point in the generalized induction principle is to allow hypothesis terms, in addition to \((P_0, x_1, \ldots, x_n)\), to be instantiated in the induction hypothesis. This extension will make the strong induction forms possible in the hypotheses. The soundness of this principle was shown in \([Liu 86]\). The principle extends the weak induction schemes \([Bayer 79]\ \[Brown & Liu 85]\ to include the strong one. While strong induction schemes are shown to have a close relationship to mutually recursive concepts, weak induction schemes are related to the self-recursive concepts. In the next section, we focus on strong induction schemes and interactions between strong and weak schemes. For the pure weak induction schemes, we refer the readers to prior work \([Boyer 79]\ \[Brown & Liu 85]\ \[Brown 86]\ \[Liu 86]\.

D. Illustrations of Mixing Induction Hypotheses

Once each induction scheme is suggested by any term in the conjecture, we begin to heuristically combine the individual schemes to synthesize the best one for the conjecture. Smooth interactions between induction schemes suggested by self-recursive and mutually recursive concepts are shown below in the synthesis of the final induction scheme.

Suppose that we try to prove the conjecture \((\forall L \in \text{EQUAL}(\text{FOO} L) \mid \text{FOOLIST} L)\). Note that it contains two mutually recursive concepts. Let \((P, L)\) be \((\forall L \in \text{EQUAL}(\text{FOO} L) \mid \text{FOOLIST} L)\). \((P_1, L)\) and \((P_2, L)\) are the hypothesis terms of \((\text{FOO} L)\), \((\text{FOOLIST} L)\) with \(P_0\) replacing \(\text{FOO}, \text{FOOLIST}\).

\[
\begin{align*}
\text{EQUAL}(P_1, L) \quad &\text{(IF \text{LISTP} L)} \\
\quad &\left(\text{P}_0, \text{CDR} L\right) \\
\text{EQUAL}(P_2, L) \quad &\text{(IF \text{LISTP} L)} \\
&\left(\text{AND} \left(\text{P}_0, \text{CAR} L\right), \text{P}_0, \text{CDR} L\right) \quad \forall j
\end{align*}
\]

Therefore, the induction scheme suggested by \((\text{FOO} L)\) is: \((\forall L \in \text{IMP} \left(\text{AND} \left(\text{LISTP} L, \text{FOO} L\right)\right) \mid \left(\text{P}_1, L\right)\)) and the scheme suggested by \((\text{FOOLIST} L)\) is: \((\forall L \in \text{IMP} \left(\text{AND} \left(\text{LISTP} L, \text{FOO} L\right)\right) \mid \left(\text{P}_2, L\right)\)) An interesting thing is shown in this case. Two mutually recursive concepts are supposed to suggest the strong induction schemes. However, since both concepts appear in the conjecture, the strong schemes are collapsed into the weak induction schemes. By merging these two induction hypotheses, we provide one induction step and one base case to cover all the relevant recursive aspects as follows.

Base case: \((\forall L \in \text{IMP} \left(\text{NOT} \left(\text{LISTP} L\right)\right) \mid \left(\text{P}_0, L\right)\))

Induction step: \((\forall L \in \text{IMP} \left(\text{AND} \left(\text{LISTP} L\right)\right) \mid \left(\text{P}_1, L\right)\))

\[
\begin{align*}
&\left(\text{AND} \left(\text{P}_0, \text{CAR} L\right), \text{P}_0, \text{CDR} L\right) \\
&\left(\text{P}_1, \text{L}\right)
\end{align*}
\]

In the second example, there are self-recursive and mutually recursive concepts in the conjecture \((\forall L \in \text{EQUAL}(\text{FOO} L) \mid \text{COPY} L)\), where \((\text{COPY} L)\) is defined as \(\left(\text{IF} \left(\text{LISTP} L\right) \mid \text{CONS} \left(\text{COPY} \left(\text{CAR} L\right)\right) \mid \text{COPY} \left(\text{CDR} L\right)\right) \mid \left(\text{P}_0, L\right)\)) Let \((P, L)\) be \((\forall L \in \text{EQUAL}(\text{FOO} L) \mid \text{COPY} L)\). Thus, the weak induction scheme suggested by the function \((\text{COPY} L)\) is: \((\forall L \in \text{IMP} \left(\text{AND} \left(\text{LISTP} L, \text{FOO} L\right)\right) \mid \left(\text{P}_1, L\right)\)). And the strong induction scheme for the function \((\text{FOO} L)\) is: \((\forall L \in \text{IMP} \left(\text{AND} \left(\text{LISTP} L, \text{FOO} L\right)\right) \mid \left(\text{P}_0, L\right)\)).

IV. Conclusions

A generalized induction principle is provided for the conjectures involved with both self-recursive and mutually recursive concepts. Mechanical induction under the principle could be used as a proof strategy for a theorem prover or logic program interpreter. Two results are shown in this paper for proving properties of recursive concepts: (1) mutually recursive concepts need to suggest strong induction hypotheses, and (2) the relationship between the strong induction forms and the weak induction scheme in mechanical structural induction.

Appendix I: A Formal Description of the Induction Principle

Suppose:

\[
\begin{align*}
(A) \quad & P_0 \text{ is the term } p^0 x_1, \ldots, x_n \text{ with } t \text{ distinct free variables, } 1 \leq n \leq t; \\
(B) & r \text{ is a well-founded relation}; \\
(C) & m \text{ is a measure function of } n \text{ arguments}; \\
(D) & (p_1 \ x_1, \ldots, x_n \ x_{n+1}, \ldots, x_t) \ldots (p_d \ x_1, \ldots, x_n \ x_{n+1}, \ldots, x_t) \text{ are hypothesis terms of any given mutually recursive functions based on } r \text{ and } m \text{ with } p^0 \text{ replacing a subset of } \{p_1, \ldots, p_d\}; \\
(E) & b_1, \ldots, b_k \text{ are non-negative integers}; \\
(F) & \text{for each } i, 1 \leq i \leq k, \text{ variables } x_{1,1}^i, \ldots, x_{n,b_i}^i \text{ are distinct and different from } x_1, \ldots, x_n, x_{n+1}, \ldots, x_t; \\
(G) & q_1, \ldots, q_h \text{ are terms}; \\
(H) & h_1, \ldots, h_k \text{ are positive integers}; \text{ and} \\
(I) & \text{for } 1 \leq i \leq k \text{ and } 1 \leq j \leq h_i, \text{ } s_{i,j}^i \text{ is a substitution and it is a theorem that} \quad \\
& \text{ALL \_LIST}(x_1, \ldots, x_n) \text{ ALL \_LIST}(x_1, \ldots, x_n) \\
& \text{IMPLIES } q_i \quad \left(r(m x_1 \ldots x_n) s_{i,j}^i (m x_1 \ldots x_n))\right).
\end{align*}
\]
Then (A). \(\text{ALL--LIST} (x_1 \ldots x_k) \ p_0\) is a theorem if
for the base case,
\[
(\text{ALL--LIST} (x_1 \ldots x_k) \left(\text{IMPLIES} \ \left(\text{AND} \ \left(\text{NOT} \ (\text{ALL--EX}_1 (z_{1,1} \ldots z_{1,b_1}) q_i)\right) \ \ldots \ \left(\text{NOT} \ (\text{ALL--EX}_k (z_{k,1} \ldots z_{k,b_k}) q_k)\right)\right) \ \ p_0\right)
\]
is a theorem and
for each \(1 \leq i \leq k\) induction step,
\[
(\text{ALL--LIST}(x_1 \ldots x_k) \left(\text{IMPLIES} \ \left(\text{AND} \ \left(\text{NOT} \ (\text{ALL--EX}_1 (z_{1,1} \ldots z_{1,b_1}) q_i)\right) \ \ldots \ \left(\text{NOT} \ (\text{ALL--EX}_k (z_{k,1} \ldots z_{k,b_k}) q_k)\right)\right) \ (\text{AND} \ q_i \ p_1^{l_1/s_{1,1}} \ldots p_{h_i}^{s_{i,1}}) \ p_0\right)
\]
is a theorem.

(E). \(\text{EX--LIST} (x_1 \ldots x_k) \ p_0\) is a theorem if
for the base case,
\[
(\text{EX--LIST}(x_1 \ldots x_k) \left(\text{AND} \ \left(\text{NOT} \ (\text{ALL--EX}_1 (z_{1,1} \ldots z_{1,b_1}) q_i)\right) \ \ldots \ \left(\text{NOT} \ (\text{ALL--EX}_k (z_{k,1} \ldots z_{k,b_k}) q_k)\right)\right) \ p_0\right)
\]
is a theorem or
for some \(1 \leq i \leq k\) induction step,
\[
(\text{EX--LIST}(x_1 \ldots x_k) \left(\text{AND} \ \left(\text{NOT} \ (\text{ALL--EX}_1 (z_{1,1} \ldots z_{1,b_1}) q_i)\right) \ \ldots \ \left(\text{NOT} \ (\text{ALL--EX}_k (z_{k,1} \ldots z_{k,b_k}) q_k)\right)\right) \ (\text{AND} \ q_i \ (\text{NOT} \ p_1^{l_1/s_{1,1}} \ldots p_{h_i}^{s_{i,1}})) \ p_0\right)
\]
is a theorem.

We now illustrate an application of this induction principle to prove the conjecture \(\text{ALL L}(\text{EQUAL L(FOO L))))\). The induction is obtained by the following instantiation of this principle. \(p_0\) is the term \((p^*0 L)\) defined as \((\text{EQUAL L(FOO L)}); \ (p^*1 L)\) and \((p^*2 L)\) are the hypothesis terms of \(\text{FOO}\) and \(\text{FOOLIST}\) with \(p^*0\) replacing \(\text{FOO}\). A well-founded relation \(\text{PLESSP}\): \(m\) is \(\text{LENGTH}\); \(n\) is \(1\); \(t\) is \(1\); \(k\) is \(1\); \(b_1\) is \(0\); \(x_i\) is \(L\); \(q_i\) is the term \((\text{LISTP L}); h_1\) is \(1\); \(s_i\) is \((<L (\text{CDR L}))\); and one theorem required by (A) is: \((\text{ALL L}(\text{IMPLIES (LISTP L)(PLESSP(LENGTH (CDR L))(LENGTH L))}))\). Thus, the base case and the induction step produced by this induction principle are \((\text{ALL L}(\text{IMPLIES (NOT (LISTP L))(PLESSP(LENGTH (CDR L))(LENGTH L))}))\) and \((\text{ALL L}(\text{IMPLIES AND (LISTP L)(p^0(CDR L))}(p^*0 L))\)). The soundness of (A) and this induction principle has been proved [Liu 86]. The proof needs two important properties that hypothesis terms preserve: (1) They satisfy the function definition principle based on the same \(R\) and \(M\), since governing conditions remain unchanged after translation, and (2) Let \(<X_1 \ldots X_t>\) be a t-tuple in the domain of \(D^t\) that is \(\text{RM}\)-smaller than \(<Y_1 \ldots Y_t>\), then \((P_0 Y_1 \ldots Y_t) \ 1 \leq i \leq d\) should not be false for such t-tuples. \(\text{RM}\) is the well-founded relation defined on n-tuples by \((\text{RM} <Z_1 \ldots Z_n> <Y_1 \ldots Y_n>)\) = \((\text{RM} (M Z_1 \ldots Z_n) (M Y_1 \ldots Y_n))\).

References


