Artificial Intelligence Project--RLE and MIT Computation Center

Symbol Manipulating Language--Memo 15

Examples of Proofs by Recursion Induction

by

John McCarthy

Recursion induction has turned out to have certain bugs and some restrictions have to be imposed. The proofs given in the sections of my notes reproduced below probably will turn out to satisfy whatever restrictions have to be imposed.

4. Recursion Induction. This is the main method for proving assertions about recursively defined functions. We state it as follows:

Suppose f is a function defined by the equation

$$f(x_1,\ldots,x_n) = \mathcal{E} .$$

where $\mathcal{E}$ is an expression which may contain f. Suppose that we wish to prove that $f(x_1,\ldots,x_n)$ has some property for those $x_1,\ldots,x_n$ for which it is defined. Assume that f has the property for the arguments occurring on the right side of the equation and prove on this hypothesis that it also has the property for the arguments $x_1,\ldots,x_n$. Then we may conclude that f has the property for all arguments for which it is defined.

Before justifying the rule, we will give an example of its application.

We define

$$f(n,m,p) = (m = n \rightarrow p, T \rightarrow f(n,m+1,(m+1)p))$$

where $n,m,$ and p are integers. We shall prove that for any n,m,p for which $f(n,m,p)$ is defined we have

$$f(n,m,p) = p.$$

Proof by recursion induction:

We have assuming $f(n,m+1,(m+1)p) = \frac{n!}{m!}(m+1)p$

$$(m = n \rightarrow p, T \rightarrow f(n,m+1,(m+1)p)) = (m=n \rightarrow \frac{n!}{m!}p, T \rightarrow \frac{n!}{(m+1)!}(m+1)p)$$

$$= (m=n \rightarrow \frac{n!}{m!}p, T \rightarrow \frac{n!}{m!}p)$$

$$= \frac{n!}{m!}p$$

which is the desired formula. Hence the theorem.

The general reaction to proofs by recursion induction seems to be suspicion. However, we shall justify the method.

Let M be the class of n-tuplets $x_1,\ldots,x_n$ for which $f(x_1,\ldots,x_n) = \mathcal{E}$ is defined. We express it as the union of disjoint classes $M_0, M_1, \ldots$, etc. defined as follows. $(x_1,\ldots,x_n)$ is in M_0 if the evaluation of $f(x_1,\ldots,x_n)$ does not involve the evaluation of any $f(y_1,\ldots,y_n)$. (In the example M_0 consists of the triplets (n,m,p) for which $n = m$.) M_k consists of those n-tuplets $(x_1,\ldots,x_n)$ for which the evaluation of $f(x_1,\ldots,x_n)$ involves the evaluation of at least one $f(y_1,\ldots,y_n)$ with $(y_1,\ldots,y_n) \in M_{k-1}$ and all $(y_1,\ldots,y_n)$ which occur in $M_0 \cup \ldots \cup M_{k-1}$.

Suppose now that the hypothesis of recursion induction is satisfied. Then $f(x_1,\ldots,x_n)$ has the desired property for $(x_1,\ldots,x_n)$ in M_0 because the assumption made about f on the right side plays no role in computing $f(x_1,\ldots,x_n)$. (In the example, transforming $(m = n \rightarrow p$, etc.) to $(m = n \rightarrow \frac{n!}{m!}p$, etc.) did not involve the induction hypotheses.) This is the basis of a mathematical induction on k.

Namely, suppose that for all $(y_1,\ldots,y_n) \in M_0 \cup \ldots \cup M_{k-1}$, $f(y_1,\ldots,y_n)$ has the desired property. Then we assert that $f(x_1,\ldots,x_n)$ has the property for $(x_1,\ldots,x_n) \in M_k$ because the assumption made in the recursion induction is involved in the proof only for $(y_1,\ldots,y_n)$ in M_{k-1}. This concludes the justification.

The important point to remember about recursion induction is that it says nothing about whether $f(x_1,\ldots,x_n)$ is defined.

5. Properties of the Functions of the Integers. In chapter 2, section 6 we defined some of the elementary functions of the integers recursively. We shall now prove some of their properties.

We defined

(1) $m+n = (n = 0 \rightarrow m, T \rightarrow m'+n^-)$

Theorem 5.1. $m+0 = m$

Proof.

$$\begin{aligned} m+0 &= (0 = 0 \rightarrow m, T \rightarrow m'+0^-) \\ &= (T \rightarrow m, T \rightarrow m'+0^-) \\ &= m \end{aligned}$$

Theorem 5.2. $(m+n)' = m'+n$

Proof.

$$\begin{aligned} m'+n &= (n = 0 \rightarrow m', T \rightarrow (m')'+n^-) && \text{by substituting } m' \text{ for } m \text{ in (1)} \\ &= (n = 0 \rightarrow m', T \rightarrow (m'+n^-)') && \text{recursion induction hypothesis with } m' \text{ for } m \\ &= (n = 0 \rightarrow m, T \rightarrow m'+n^-)' && \text{taking the successor function out of the conditional expression} \\ &= (m+n)' \end{aligned}$$

<u>Theorem 5.3.</u> $(m+n)+p = (m+p)+n$

<u>Proof.</u>

$$(m+n)+p = (n = 0 \rightarrow m, T \rightarrow m'+n)+p$$

$$= (n = 0 \rightarrow m+p, T \rightarrow (m'+n^-)+p) \quad \text{distribution of function}$$

$$= (n = 0 \rightarrow m+p, T \rightarrow (m'+p)+n^-) \quad \text{recursion step}$$

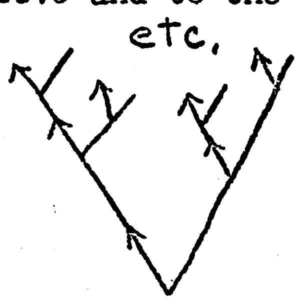
$$= (n = 0 \rightarrow m+p, T \rightarrow (m+p)'+n^-) \quad \text{Theorem 5.2}$$

$$= (m+p)+n$$

Setting $m = 0$ in Theorem 5.3 gives

$$(0+n)+p = (0+p)+n$$

so that if we had $0+m = m$ we would have commutativity of addition. In fact, we cannot prove $0+m = m$ without making some assumptions that take into account that we are dealing with the integers. For suppose our space consisted of the vertices of the binary tree where m' is the vertex just above and to the left and m^- is the vertex just below and



0 is the bottom of the tree. $m+n$ can be defined as above and of course satisfies Theorems 5.1, 5.2, and 5.3 but does not satisfy $0+m = m$.

We shall make the following assumptions:

1. $m' \neq 0$
2. $(m')^- = m$
3. $(m \neq 0) \supset ((m^-)' = m)$

which embody all of Peano's axioms except the induction axiom.

<u>Theorem 5.4.</u> $0+n = n$

<u>Proof.</u>

$$0+n = (n = 0 \rightarrow 0, T \rightarrow 0'+n^-)$$

$$= (n = 0 \rightarrow n, T \rightarrow (0+n^-)') \quad \text{Theorem 5.2}$$

$$= (n = 0 \rightarrow n, T \rightarrow (n^-)') \quad \text{induction hypothesis}$$

$$= (n = 0 \rightarrow n, T \rightarrow n) \quad \text{axiom 3}$$

$$= n$$

Theorem 5.5. $m+n = n+m$

Proof. By Theorems 5.3 and 5.4 as remarked above.

Theorem 5.6. $(m+n)+p = m+(n+p)$

Proof.

$(m+n)+p = (m+p)+n$ — Theorem 5.3

$= (p+m)+n$ — Theorem 5.5

$= (p+n)+m$ — Theorem 5.3

$= m+(n+p)$ — Theorem 5.5 twice

We now have some theorems concerning the product

$$m\times n = (n = 0 \rightarrow 0, T \rightarrow m+m\times n^{-})$$

Theorem 5.7. $m\times 0 = 0$

Proof.

$m\times 0 = (0 = 0 \rightarrow 0, T \rightarrow m, m\ 0^{-})$

$= 0$

Theorem 5.8. $0\times n = 0$

Proof.

$0\times n = (n = 0 \rightarrow 0, T \rightarrow 0+0\times n^{-})$

$= (n = 0 \rightarrow 0, T \rightarrow 0\times n^{-})$ — Theorem 5.4

$= (n = 0 \rightarrow 0, T \rightarrow 0)$ — recursion induction hypothesis

$= 0$

Theorem 5.9. $m\times n = n\times m$

Proof.

$m\times n = (n = 0 \rightarrow 0, T \rightarrow m+m\times n^{-})$

$= (n = 0 \rightarrow 0, T \rightarrow m+n^{-}\times m)$ — induction

$= (n = 0 \rightarrow 0, T \rightarrow m+(m = 0 \rightarrow 0,$
$\quad T \rightarrow n^{-}+n^{-}\times m^{-}))$

$= (n = 0 \rightarrow 0, T \rightarrow (m = 0 \rightarrow m+0,$
$\quad T \rightarrow m+n^{-}+n^{-}\times m^{-})$

$= (n = 0 \rightarrow 0, m = 0 \rightarrow 0, T \rightarrow (m^{-}+n^{-}+n^{-}\times m^{-})')$

$= (m = 0 \rightarrow 0, n = 0 \rightarrow 0, T \rightarrow (n^{-}+m^{-}+m^{-}\times n^{-})')$

The steps can now be worked backwards with m and n interchanged.

Theorem 5.10. $m\times n' = m+m\times n$

Proof. $m\times n' = (n' = 0 \to 0, T \to m+m\times(n')^-)$

$= m+m\times n$

Theorem 5.11. $m\times(n+p) = mn+mp$

Proof. $m\times(n+p) = m(p = 0 \to n, T \to n'+p^-)$

$= (p = 0 \to mn, T \to m(n'+p^-))$

$= (p = 0 \to mn, T \to mn'+mp^-)$ induction hypothesis

$= (p = 0 \to mn, T \to mn+(m+mp^-)$

$= (p = 0 \to mn, T \to mn+(p = 0 \to 0, T \to m+mp^-))$

$= (p = 0 \to mn+mp, T \to mn+mp)$

$= mn+mp$

Theorem 5.12. $(mn)p = m(np)$

Proof. $(mn)p = (p = 0 \to 0, T \to mn+(mn)p^-)$

$= (p = 0 \to 0, T \to mn+m(np^-))$

$= (p = 0 \to 0, T \to m(n+np^-)$

$= m(p = 0 \to 0, T \to n+np^-)$

$= m(np)$

This proof is by induction on p. Induction on m or n would also work.

We make the definition

$$m \leq n = (m = 0) \vee (n \neq 0) \wedge (m^- \leq n^-)$$

Theorem 5.13. $0 \leq n$

Proof. $0 \leq n = (0 = 0) \vee (n \neq 0) \wedge (0^- \leq n^-)$

$= T$

<u>Theorem 5.14.</u> $(m+p \le n+p) = (m \le n)$

<u>Proof.</u>

$$
\begin{aligned}
(m+p \le n+p) &= (p = 0 \to m, T \to m'+p^-) \\
&\qquad (p = 0 \to n, T \to n'+p^-) \\
&= (p = 0 \to m \le n, T \to m'+p^- \le n'+p^-) \\
&= (p = 0 \to m \le n, T \to m' \le n') \\
&= (p = 0 \to m \le n, T \to (m' = 0) \vee \sim \\
&\qquad (n' = 0) \wedge ((m')^- \le (n')^-) \\
&= (p = 0 \to m \le n, T \to m \le n) \\
&= m \le n
\end{aligned}
$$

6. <u>Properties of Functions of Symbolic Expressions</u>. In this section we shall use recursion induction to prove a number of properties of functions of symbolic expressions. We start with the basic identities

$$
\begin{aligned}
&\mathrm{car}[\mathrm{cons}[x;y]] = x \\
&\mathrm{cdr}[\mathrm{cons}[x;y]] = y \\
&\mathrm{atom}[x] \vee [\mathrm{cons}[\mathrm{car}[x];\mathrm{cdr}[x]] = x] \\
&\sim \mathrm{atom}[\mathrm{cons}[x;y]] \\
&\mathrm{null}[x] = x[x = \mathrm{NIL}]
\end{aligned}
$$

where in this section we use $=$ for equality of S-expressions in general and not just for atomic S-expressions.

We shall write $x*y$ for <u>conc</u>[x;y] and our first objective is to prove that concatenation is associative, i.e.

<u>Theorem 6.1.</u> $[x*y]*z = x*[y*z]$

<u>Proof.</u> The definition of $x*y$ can be written

$$x*y = [\mathrm{null}[x] \to y; T \to \mathrm{cons}[\mathrm{car}[x];\mathrm{cdr}[x]*y]]$$

We have

$$
\begin{aligned}
\mathrm{null}[x*y] &= [\mathrm{null}[x] \to \mathrm{null}[y]; T \to \mathrm{null}[\mathrm{cons}[\mathrm{car}[x];\mathrm{cdr}[x]*y]]] \\
&= [\mathrm{null}[x] \to \mathrm{null}[y]; T \to F] \\
&= \mathrm{null}[x] \wedge \mathrm{null}[y]
\end{aligned}
$$

car[x*y] = [null[x] → car[y];T → car[cons[car[x];cdr[x]*y]]]
         = [null[x] → car[y];T → car[x]]
cdr[x*y] = [null[x] → cdr[y];T → cdr[cons[car[x];cdr[x]*y]]]
         = [null[x] → cdr[y];T → cdr[x]*y]

Now we write

[x*y]z = [null[x*y] → z;T → cons[car[x*y];cdr[x*y]*z]]
       = [null[x] → [null[y] → z;T → cons[car[x*y];cdr[x*y]*z]];
            T → cons[car[x*y];cdr[x*y]*z]]

We substitute the above derived results for car[x*y] and cdr[x*y] making use of the fact that the first occurrences of these quantities have null[x] as a premise and the second occurrences have ∼null[x] as a premise. Thus we get

[x*y]*z = [null[x] → [null[y] → z;T → cons[car[y];cdr[y]*z]];
             T → cons[car[x];[cdr[x]*y]*z]
        = [null[x] → y*z;T → cons[car[x];cdr[x]*[y*z]]]
        = x*[y*z]

Next we have

<u>Theorem 6.2.</u> NIL*x = x
x*NIL = x

<u>Proof.</u> NIL*x = [null[NIL] → x;T → cons[car[NIL];
            cdr[NIL]*x]]
      = x

x*NIL = [null[x] → NIL;T → cons[car[x];
            cdr[x]*NIL]]
      = [null[x] → x;T → cons[car[x];cdr[x]]]
      = [null[x] → x;T → x]
      = x

Next we consider a function <u>reverse</u> defined by
reverse[x] = [null[x] → NIL;T → reverse[cdr[x]]*cons[car[x];NIL]]
which reverses the order of terms in a list.

Theorem 6.3. reverse[x*y] = reverse[y]*reverse[x]

Proof.

reverse[x*y] = [null[x] → reverse[y];T → reverse[cons[car[x];
cdr[x]*y]]]

= [null[x] → reverse[y];T → [null[cons[car[x];
cdr[x]*y]] → NIL;T → reverse[cdr[cons[car[x];cdr[x]*y]]]*cons[
car[cons[car[x];cdr[x]*y]];NIL]]]

= [null[x] → reverse[y]*NIL;T → reverse[cdr[
x]*y]*cons[car[x];NIL]]

= [null[x] → reverse[y]*NIL;T → reverse[y]*
reverse[cdr[x]]*cons[car[x];NIL]]

= reverse[y]*[null[x] → NIL;T → reverse[cdr[
x]*cons[car[x];NIL]]

= reverse[y]*reverse[x]

Theorem 6.4. reverse[reverse[x]] = x

Proof.

reverse[reverse[x]] = [null[x] → reverse[NIL];T → reverse[
reverse[cdr[x]]*cons[car[x];NIL]]]

= [null[x] → NIL;T → reverse[cons[car[
x];NIL]]*reverse[reverse[cdr[x]]]]

= [null[x] → x;T → cons[car[x];NIL]*cdr[x]]

= [null[x] → x;T → cons[car[x];cdr[x]]]

= x

In chapter 2, we defined

equal[x;y] = atom[x]∧atom[y]∧x = y∧¬atom[x]∧¬atom[y]∧equal[cons[x];
car[y]∧equal[cdr[x];cdr[y]

subst[x;y;z] = [atom[z] → [z = y → x;T → z];T → cons[subst[x;y;
car[z]];subst[x;y;cdr[z]]]]

We now define

free[y;z] = [atom[z] → z ≠ y;T → free[y;car[z]]∧free[y;cdr[z]]]

We have

Theorem 6.5. $\mathrm{free}[y;z] \supset \mathrm{equal}[\mathrm{subst}[x;y;z];z]$

Proof.

$$\mathrm{free}[y;z] \supset \mathrm{equal}[\mathrm{subst}[x;y;z];z] = [\mathrm{free}[y;z] \rightarrow \mathrm{equal}[\mathrm{subst}[x;y;z];z];T \rightarrow T]$$

$$= [\mathrm{atom}[z] \rightarrow [\mathrm{free}[y;z] \rightarrow \mathrm{equal}[\mathrm{subst}[x;y;z];z];T \rightarrow T];T \rightarrow [\mathrm{free}[y;z] \rightarrow \mathrm{equal}[\mathrm{subst}[x;y;z];z];T \rightarrow T]]$$

$$= [\mathrm{atom}[z] \rightarrow [z \neq y \rightarrow \mathrm{equal}[[z = y \rightarrow x;T \rightarrow z];z];T \rightarrow T];T \rightarrow [\mathrm{free}[y;\mathrm{car}[z]] \wedge \mathrm{free}[y;\mathrm{cdr}[z]] \rightarrow \mathrm{equal}[\mathrm{cons}[\mathrm{subst}[x;y;\mathrm{car}[z]];\mathrm{subst}[x;y;\mathrm{cdr}[z]]];T \rightarrow T]]$$

$$= [\mathrm{atom}[z] \rightarrow [z \neq y \rightarrow \mathrm{equal}[z;z];T \rightarrow T];T \rightarrow [\mathrm{free}[y;\mathrm{car}[z]] \wedge \mathrm{free}[y;\mathrm{cdr}[z]] \rightarrow \mathrm{equal}[\mathrm{car}[\mathrm{cons}[\mathrm{subst}[x;y;\mathrm{car}[z]];\mathrm{subst}[x;y;\mathrm{cdr}[z]]]];\mathrm{car}[z]] \wedge \mathrm{equal}[\mathrm{cdr}[\mathrm{cons}[\mathrm{subst}[x;y;\mathrm{car}[z]];\mathrm{subst}[x;y;\mathrm{cdr}[z]]]];\mathrm{cdr}[z]];T \rightarrow T]$$

where we used the definition of <u>equal</u> and atom z and $\sim \mathrm{atom}[\mathrm{cons}[u;v]]$

$$= [\mathrm{atom}[z] \rightarrow T;T \rightarrow [\mathrm{free}[y;\mathrm{car}[z]] \wedge \mathrm{free}[y;\mathrm{cdr}[z]] \rightarrow \mathrm{equal}[\mathrm{subst}[T;y;\mathrm{car}[z]];\mathrm{car}[z]] \wedge \mathrm{equal}[\mathrm{subst}[x;y;\mathrm{cdr}[z]];\mathrm{cdr}[z]];T \rightarrow T]]$$

$$= [\mathrm{atom}[z] \rightarrow T;T \rightarrow [\mathrm{free}[y;\mathrm{car}[z]] \wedge \mathrm{free}[y;\mathrm{cdr}[z]] \rightarrow [\mathrm{free}[y;\mathrm{car}[z] \rightarrow \mathrm{equal}[\mathrm{subst}[x;y;\mathrm{car}[z]];\mathrm{car}[z]];T \rightarrow T] \wedge [\mathrm{free}[y;\mathrm{cdr}[z]] \rightarrow \mathrm{equal}[\mathrm{subst}[x;y;\mathrm{cdr}[z]];\mathrm{cdr}[z]];T \rightarrow T];T \rightarrow T]]$$

$$= [\mathrm{atom}[z] \rightarrow T;T \rightarrow [\mathrm{free}[y;\mathrm{car}[z]] \wedge \mathrm{free}[y;\mathrm{cdr}[z]] \rightarrow T \wedge T;T \rightarrow T]]$$

$$= [\mathrm{atom}[z] \rightarrow T;T \rightarrow T]$$

$$= T$$

CS-TR Scanning Project
Document Control Form

Date : 11 / 30 / 95

Report # AIM - 15

Each of the following should be identified by a checkmark:
Originating Department:

☒ Artificial Intellegence Laboratory (AI)
☐ Laboratory for Computer Science (LCS)

Document Type:

☐ Technical Report (TR) ☒ Technical Memo (TM)
☐ Other: ____________

Document Information

Number of pages: 10 (14 - IMAGES)
Not to include DOD forms, printer intstructions, etc... original pages only.

Originals are:
☒ Single-sided or
☐ Double-sided

Intended to be printed as :
☒ Single-sided or
☐ Double-sided

Print type:
☐ Typewriter ☐ Offset Press ☐ Laser Print
☐ InkJet Printer ☐ Unknown ☒ Other: COPY OF MIMEOGRAPH

Check each if included with document:

☐ DOD Form ☐ Funding Agent Form ☐ Cover Page
☐ Spine ☐ Printers Notes ☐ Photo negatives
☐ Other: ____________

Page Data:

Blank Pages (by page number): ____________

Photographs/Tonal Material (by page number): ____________

Other (note description/page number):

Description :	Page Number:
IMAGE MAP! (1-10)	1-10
	(11-14) SCANCONTROL

Scanning Agent Signoff:
Date Received: 11 / 30 / 95 Date Scanned: 12 / 13 / 95 Date Returned: 12 / 14 / 95

Scanning Agent Signature: Michael W. Cook

Rev 9/94 DS/LCS Document Control Form cstrform.vsd