```
#|

 Copyright (C) 1994 by Computational Logic, Inc. and Paolo Pecchiari.  All
 Rights Reserved.

 This script is hereby placed in the public domain, and therefore unlimited
 editing and redistribution is permitted.

 NO WARRANTY

 Computational Logic, Inc. AND Paolo Pecchiari PROVIDE ABSOLUTELY NO WARRANTY.
 THE EVENT SCRIPT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER
 EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF
 MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.  THE ENTIRE RISK AS TO
 THE QUALITY AND PERFORMANCE OF THE SCRIPT IS WITH YOU.  SHOULD THE SCRIPT
 PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR
 CORRECTION.  IN NO EVENT WILL Computational Logic, Inc. OR Paolo Pecchiari BE
 LIABLE TO YOU FOR ANY DAMAGES, ANY LOST PROFITS, LOST MONIES, OR OTHER
 SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR
 INABILITY TO USE THIS SCRIPT (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR
 DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY THIRD PARTIES), EVEN IF
 YOU HAVE ADVISED US OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY
 ANY OTHER PARTY.
|#

; Events from CLI Technical Report 100, ``Interaction with the Boyer-Moore
; Theorem Prover: A Tutorial Study Using the Arithmetic-Geometric Mean
; Theorem,'' by Matt Kaufmann and Paolo Pecchiari.
```

EVENT: Start with the library `"naturals"` using the compiled version.

DEFINITION:
prodlist (lst)
= **if** listp (lst) **then** car (lst) $*$ prodlist (cdr (lst))
  else 1 **endif**

DEFINITION:
sumlist (lst)
= **if** listp (lst) **then** car (lst) $+$ sumlist (cdr (lst))
  else 0 **endif**

DEFINITION:
length (x)

$=$ **if** listp (x) **then** $1 +$ length (cdr (x))
else 0 **endif**

DEFINITION:
maxlist (x)
$=$ **if** listp (x)
then if listp (cdr (x)) **then** max (car (x), maxlist (cdr (x)))
else fix (car (x)) **endif**
else 0 **endif**

DEFINITION:
min (x, y)
$=$ **if** $x < y$ **then** fix (x)
else fix (y) **endif**

DEFINITION:
minlist (x)
$=$ **if** listp (x)
then if listp (cdr (x)) **then** min (car (x), minlist (cdr (x)))
else fix (car (x)) **endif**
else 0 **endif**

DEFINITION:
delete1 (elt, x)
$=$ **if** listp (x)
then if $elt =$ car (x) **then** cdr (x)
else cons (car (x), delete1 (elt, cdr (x))) **endif**
else x **endif**

THEOREM: maxlist-delete1-rearrange
$(b \in x) \rightarrow (\text{maxlist}(\text{cons}(b, \text{delete1}(b, x))) = \text{maxlist}(x))$

EVENT: Let us define the theory *induction-fn-disables* to consist of the following events: sumlist, times, length, minlist, maxlist, delete1, occurrences.

THEOREM: max-greater-than-average
$(\text{listp}(x) \land (\text{sumlist}(x) \geq (k * \text{length}(x))) \land (\text{fix}(k) \neq \text{maxlist}(x)))$
$\rightarrow \quad (k < \text{maxlist}(x))$

THEOREM: min-less-than-average
$(\text{listp}(x) \land (\text{sumlist}(x) \leq (k * \text{length}(x))) \land (\text{fix}(k) \neq \text{minlist}(x)))$
$\rightarrow \quad (\text{minlist}(x) < k)$

THEOREM: times-monotone-1
$(u \not< v) \rightarrow ((u * y) \not< (v * y))$

THEOREM: lessp-times-preserved-in-first-arg
$((a \not< (u * y)) \land (u \not< v)) \rightarrow (a \not< (v * y))$

THEOREM: minlist-main-property
$\text{sumlist}(x) \not< (\text{length}(x) * \text{minlist}(x))$

THEOREM: minlist-not-maxlist-implies-minlist-lessp-average-lemma
$(\text{minlist}(x) \neq \text{maxlist}(x)) \rightarrow ((\text{minlist}(x) * \text{length}(x)) < \text{sumlist}(x))$

THEOREM: minlist-not-maxlist-implies-minlist-lessp-average
$$\begin{array}{ll} (\text{listp}(x) & \\ \land & (\text{sumlist}(x) = (k * \text{length}(x))) \\ \land & (\text{minlist}(x) \neq \text{maxlist}(x))) \\ \rightarrow & (\text{minlist}(x) < \text{fix}(k)) \end{array}$$

THEOREM: maxlist-main-property
$(\text{length}(x) * \text{maxlist}(x)) \not< \text{sumlist}(x)$

THEOREM: minlist-not-maxlist-implies-maxlist-greaterp-average-lemma
$(\text{minlist}(x) \neq \text{maxlist}(x)) \rightarrow (\text{sumlist}(x) < (\text{maxlist}(x) * \text{length}(x)))$

THEOREM: minlist-not-maxlist-implies-maxlist-greaterp-average
$$\begin{array}{ll} (\text{listp}(x) & \\ \land & (\text{sumlist}(x) = (k * \text{length}(x))) \\ \land & (\text{minlist}(x) \neq \text{maxlist}(x))) \\ \rightarrow & (\text{fix}(k) < \text{maxlist}(x)) \end{array}$$

THEOREM: minlist-less-than-maxlist-minus-1
$$\begin{array}{ll} (\text{listp}(x) & \\ \land & (\text{sumlist}(x) = (k * \text{length}(x))) \\ \land & (\text{minlist}(x) \neq \text{maxlist}(x))) \\ \rightarrow & (\text{minlist}(x) < (\text{maxlist}(x) - 1)) \end{array}$$

THEOREM: maxlist-delete1-leq
$\text{maxlist}(lst) \not< \text{maxlist}(\text{delete1}(a, lst))$

THEOREM: member-implies-maxlist-geq
$(a \in x) \rightarrow (\text{maxlist}(x) \not< a)$

THEOREM: delete1-preserves-maxlist-when-maxlist-occurs-more-than-once
$$\begin{array}{ll} (\mathtt{1} < \text{occurrences}(\text{maxlist}(x), x)) & \\ \rightarrow & (\text{maxlist}(\text{delete1}(\textit{any-element}, x)) = \text{maxlist}(x)) \end{array}$$

THEOREM: delete1-occurrences
$$\begin{array}{ll} \text{occurrences}(a, \text{delete1}(b, x)) & \\ = & \textbf{if } (a = b) \land (b \in x) \textbf{ then } \text{occurrences}(a, x) - 1 \\ & \textbf{else } \text{occurrences}(a, x) \textbf{ endif} \end{array}$$

THEOREM: occurrence-implies-listp
$(\texttt{1} < \text{occurrences}\,(a,\ x)) \rightarrow (\text{listp}\,(\text{delete1}\,(a,\ x)) = \mathbf{t})$

THEOREM: maxlist-geq-minlist
$\text{maxlist}\,(x) \not< \text{minlist}\,(x)$

THEOREM: induction-fn-help-2-max-occurs-twice-lemma-1
$$
\begin{array}{ll}
& ((\text{sumlist}\,(x) = (k * \text{length}\,(x))) \\
\wedge & (\text{minlist}\,(x) \neq \text{maxlist}\,(x)) \\
\wedge & (\texttt{1} < \text{occurrences}\,(\text{maxlist}\,(x),\ x))) \\
\rightarrow & (\text{maxlist}\,(\text{cons}\,(\text{maxlist}\,(x) - 1, \\
& \quad \text{cons}\,(1 + \text{minlist}\,(x), \\
& \quad\quad \text{delete1}\,(\text{maxlist}\,(x),\ \text{delete1}\,(\text{minlist}\,(x),\ x))))) \\
& \quad = \ \text{maxlist}\,(x))
\end{array}
$$

THEOREM: member-delete1
$$
\begin{array}{ll}
& (a \in \text{delete1}\,(b,\ c)) \\
= & \textbf{if}\ a = b\ \textbf{then}\ \texttt{1} < \text{occurrences}\,(b,\ c) \\
& \textbf{else}\ a \in c\ \textbf{endif}
\end{array}
$$

THEOREM: member-implies-listp
$(a \in x) \rightarrow \text{listp}\,(x)$

THEOREM: induction-fn-help-2-max-occurs-twice-lemma-2
$$
\begin{array}{ll}
& ((\text{sumlist}\,(x) = (k * \text{length}\,(x))) \\
\wedge & (\text{minlist}\,(x) \neq \text{maxlist}\,(x)) \\
\wedge & (\texttt{1} < \text{occurrences}\,(\text{maxlist}\,(x),\ x))) \\
\rightarrow & ((\text{occurrences}\,(\text{maxlist}\,(x), \\
& \quad \text{cons}\,(\text{maxlist}\,(x) - 1, \\
& \quad\quad \text{cons}\,(1 + \text{minlist}\,(x), \\
& \quad\quad\quad \text{delete1}\,(\text{maxlist}\,(x),\ \text{delete1}\,(\text{minlist}\,(x),\ x))))) \\
& \quad < \ \text{occurrences}\,(\text{maxlist}\,(x),\ x)) \\
& \quad = \ \mathbf{t})
\end{array}
$$

EVENT: Disable member-implies-listp.

THEOREM: induction-fn-help-2-max-occurs-twice
$$
\begin{array}{ll}
\multicolumn{2}{l}{\textbf{let}\ \mathit{x0}\ \textbf{be}\ \text{cons}\,(\text{maxlist}\,(x) - 1,} \\
& \quad \text{cons}\,(1 + \text{minlist}\,(x), \\
& \quad\quad \text{delete1}\,(\text{maxlist}\,(x),\ \text{delete1}\,(\text{minlist}\,(x),\ x)))) \\
\multicolumn{2}{l}{\textbf{in}} \\
& ((\text{sumlist}\,(x) = (k * \text{length}\,(x))) \\
\wedge & (\text{minlist}\,(x) \neq \text{maxlist}\,(x)) \\
\wedge & (\texttt{1} < \text{occurrences}\,(\text{maxlist}\,(x),\ x))) \\
\rightarrow & \text{ord-lessp}\,(\text{cons}\,(1 + \text{maxlist}\,(\mathit{x0}),\ \text{occurrences}\,(\text{maxlist}\,(\mathit{x0}),\ \mathit{x0})), \\
& \quad \text{cons}\,(1 + \text{maxlist}\,(x),\ \text{occurrences}\,(\text{maxlist}\,(x),\ x)))\ \textbf{endlet}
\end{array}
$$

THEOREM: maxlist-not-minlist-implies-listp
$(\text{minlist}(x) \neq \text{maxlist}(x)) \rightarrow \text{listp}(x)$

THEOREM: minlist-less-than-maxlist-minus-1-better
$((\text{sumlist}(x) = (k * \text{length}(x))) \wedge (\text{minlist}(x) \neq \text{maxlist}(x)))$
$\rightarrow \quad (\text{minlist}(x) < (\text{maxlist}(x) - 1))$

EVENT: Disable minlist-less-than-maxlist-minus-1.

THEOREM: maxlist-delete1-delete1
$\text{maxlist}(\text{delete1}(b, x)) \not< \text{maxlist}(\text{delete1}(b, \text{delete1}(a, x)))$

THEOREM: member-maxlist
$(\text{maxlist}(z) \neq \mathtt{0}) \rightarrow (\text{maxlist}(z) \in z)$

THEOREM: lessp-maxlist-delete1-maxlist
$((\mathtt{0} < \text{maxlist}(x)) \wedge (\mathtt{1} \not< \text{occurrences}(\text{maxlist}(x), x)))$
$\rightarrow \quad (\text{maxlist}(\text{delete1}(\text{maxlist}(x), x)) < \text{maxlist}(x))$

THEOREM: induction-fn-help-2-max-occurs-once-main-lemma
$((\text{sumlist}(x) = (k * \text{length}(x)))$
$\wedge \quad (\text{minlist}(x) \neq \text{maxlist}(x))$
$\wedge \quad (\mathtt{1} \not< \text{occurrences}(\text{maxlist}(x), x)))$
$\rightarrow \quad ((\text{maxlist}(\text{cons}(\text{maxlist}(x) - 1,$
$\text{cons}(1 + \text{minlist}(x),$
$\text{delete1}(\text{maxlist}(x), \text{delete1}(\text{minlist}(x), x)))))$
$< \quad \text{maxlist}(x))$
$= \quad \mathbf{t})$

THEOREM: induction-fn-help-2-max-occurs-once
let $x0$ **be** $\text{cons}(\text{maxlist}(x) - 1,$
$\text{cons}(1 + \text{minlist}(x),$
$\text{delete1}(\text{maxlist}(x), \text{delete1}(\text{minlist}(x), x))))$
in
$((\text{sumlist}(x) = (k * \text{length}(x)))$
$\wedge \quad (\text{minlist}(x) \neq \text{maxlist}(x))$
$\wedge \quad (\mathtt{1} \not< \text{occurrences}(\text{maxlist}(x), x)))$
$\rightarrow \quad \text{ord-lessp}(\text{cons}(1 + \text{maxlist}(x0), \text{occurrences}(\text{maxlist}(x0), x0)),$
$\text{cons}(1 + \text{maxlist}(x), \text{occurrences}(\text{maxlist}(x), x)))$ **endlet**

THEOREM: induction-fn-help-2
let $x0$ **be** $\text{cons}(\text{maxlist}(x) - 1,$
$\text{cons}(1 + \text{minlist}(x),$
$\text{delete1}(\text{maxlist}(x), \text{delete1}(\text{minlist}(x), x))))$
in

$((\mathrm{sumlist}(x) = (k * \mathrm{length}(x))) \wedge (\mathrm{minlist}(x) \neq \mathrm{maxlist}(x)))$
$\rightarrow \quad \text{ord-lessp}(\mathrm{cons}(1 + \mathrm{maxlist}(x0), \mathrm{occurrences}(\mathrm{maxlist}(x0), x0)),$
$\qquad\qquad \mathrm{cons}(1 + \mathrm{maxlist}(x), \mathrm{occurrences}(\mathrm{maxlist}(x), x)))$ **endlet**

Event: Disable theory induction-fn-disables.

Definition:
induction-fn (x, k)
$=$ **if** sumlist $(x) \neq (k * \mathrm{length}(x))$ **then t**
elseif minlist $(x) =$ maxlist (x) **then t**
else induction-fn (cons (maxlist $(x) - 1$,
cons $(1 +$ minlist (x),
delete1 (maxlist (x),
delete1 (minlist (x), x)))),
k) **endif**

Event: Enable theory induction-fn-disablesinduction-fn-disables.

Theorem: sumlist-delete1-plus-version
$(a + \mathrm{sumlist}(\mathrm{delete1}(a, x)))$
$=$ **if** $a \in x$ **then** sumlist (x)
else $a +$ sumlist (x) **endif**

Theorem: sumlist-delete1
sumlist (delete1 (a, x))
$=$ **if** $a \in x$ **then** sumlist $(x) - a$
else sumlist (x) **endif**

Theorem: maxlist-0-is-sumlist-0
$(\mathrm{maxlist}(x) = 0) = (\mathrm{sumlist}(x) = 0)$

Theorem: main-lemma-base-case-lemma-1
$(\mathrm{minlist}(x) = \mathrm{maxlist}(x)) \rightarrow (\exp(\mathrm{minlist}(x), \mathrm{length}(x)) = \mathrm{prodlist}(x))$

Theorem: main-lemma-base-case-lemma-2-lemma
$(\mathrm{minlist}(x) = \mathrm{maxlist}(x)) \rightarrow (\mathrm{sumlist}(x) = (\mathrm{minlist}(x) * \mathrm{length}(x)))$

Theorem: main-lemma-base-case-lemma-2-hack
$(((k * n) = \mathit{sumlist}) \wedge (n \not\simeq 0) \wedge ((n * \mathit{maxlist}) = \mathit{sumlist}))$
$\rightarrow \quad (\mathrm{fix}(k) = \mathrm{fix}(\mathit{maxlist}))$

Theorem: equal-length-0
$(\mathrm{length}(x) = 0) = (\neg\, \mathrm{listp}(x))$

THEOREM: main-lemma-base-case-lemma-2
$$(\text{listp}\,(x) \;\wedge\; (\text{sumlist}\,(x) = (k * \text{length}\,(x))) \;\wedge\; (\text{minlist}\,(x) = \text{maxlist}\,(x))) \;\rightarrow\; (\text{minlist}\,(x) = \text{fix}\,(k))$$

THEOREM: main-lemma-base-case
$$((\text{sumlist}\,(x) = (k * \text{length}\,(x))) \wedge (\text{minlist}\,(x) = \text{maxlist}\,(x))) \;\rightarrow\; (\exp\,(k,\ \text{length}\,(x)) = \text{prodlist}\,(x))$$

THEOREM: length-delete1
$$\text{length}\,(\text{delete1}\,(a,\ x)) = \textbf{if } a \in x \textbf{ then } \text{length}\,(x) - 1 \textbf{ else } \text{length}\,(x) \textbf{ endif}$$

DEFINITION:
$$\text{numberp-listp}\,(x) = \textbf{if } \text{listp}\,(x) \textbf{ then } (\text{car}\,(x) \in \mathbf{N}) \wedge \text{numberp-listp}\,(\text{cdr}\,(x)) \textbf{ else } x = \textbf{nil} \textbf{ endif}$$

THEOREM: member-minlist
$$(\text{listp}\,(x) \wedge \text{numberp-listp}\,(x)) \rightarrow (\text{minlist}\,(x) \in x)$$

THEOREM: numberp-listp-delete1
$$\text{numberp-listp}\,(x) \rightarrow \text{numberp-listp}\,(\text{delete1}\,(a,\ x))$$

THEOREM: add1-plus-sub1-second
$$(y \not\simeq 0) \rightarrow ((1 + (x + (y - 1))) = (x + y))$$

THEOREM: sumlist-geq-maxlist
$$\text{sumlist}\,(x) \not< \text{maxlist}\,(x)$$

THEOREM: sumlist-geq-minlist
$$\text{sumlist}\,(x) \not< \text{minlist}\,(x)$$

THEOREM: sumlist-geq-minlist-plus-maxlist
$$(\text{minlist}\,(x) \neq \text{maxlist}\,(x)) \rightarrow (\text{sumlist}\,(x) \not< (\text{minlist}\,(x) + \text{maxlist}\,(x)))$$

EVENT: Disable plus.

THEOREM: plus-times-sub1-second
$$(y \not\simeq 0) \rightarrow ((x + (x * (y - 1))) = (x * y))$$

EVENT: Disable times.

THEOREM: minlist-not-maxlist-implies-length-at-least-2
$(\text{minlist}(x) \neq \text{maxlist}(x)) \rightarrow (1 < \text{length}(x))$

EVENT: Disable plus-add1-arg1.

EVENT: Disable plus-add1-arg2.

EVENT: Disable times-add1.

EVENT: Disable maxlist-0-is-sumlist-0.

THEOREM: times-prodlist-delete1
$(a \in x) \rightarrow ((a * \text{prodlist}(\text{delete1}(a, x))) = \text{prodlist}(x))$

THEOREM: product-of-modified-list-lemma
$$((min \neq max) \land (min \in x) \land (max \in x))$$
$$\rightarrow \quad ((max * min * \text{prodlist}(\text{delete1}(max, \text{delete1}(min, x))))$$
$$= \quad \text{prodlist}(x))$$

EVENT: Disable times-prodlist-delete1.

THEOREM: product-of-modified-list-lemma-2
$$((min < max) \land (max \not\simeq 0))$$
$$\rightarrow \quad (((1 + min) * (max - 1) * rest)$$
$$= \quad ((min * max * rest) + ((max - (1 + min)) * rest)))$$

THEOREM: positive-implies-numberp
$(min < max) \rightarrow (max \in \mathbf{N})$

THEOREM: product-of-modified-list
$$((min < max) \land (min \in x) \land (max \in x))$$
$$\rightarrow \quad \textbf{let } rest \textbf{ be } \text{prodlist}(\text{delete1}(max, \text{delete1}(min, x)))$$
$$\textbf{in}$$
$$((1 + min) * (max - 1) * rest)$$
$$= \quad (\text{prodlist}(x) + ((max - (1 + min)) * rest)) \textbf{ endlet}$$

THEOREM: main-lemma
$$\textbf{let } n \textbf{ be } \text{length}(x)$$
$$\textbf{in}$$
$$(\text{numberp-listp}(x) \land (\text{sumlist}(x) = (k * n)))$$
$$\rightarrow \quad (\exp(k, n) \not< \text{prodlist}(x)) \textbf{ endlet}$$

DEFINITION:
scalar-product ($scalar$, lst)
$=$ **if** listp (lst)
then cons ($scalar * $ car (lst), scalar-product ($scalar$, cdr (lst)))
else nil endif

THEOREM: sumlist-scalar-product
sumlist (scalar-product ($scalar$, lst)) $=$ ($scalar *$ sumlist (lst))

THEOREM: prodlist-scalar-product
prodlist (scalar-product ($scalar$, lst))
$=$ (exp ($scalar$, length (lst)) $*$ prodlist (lst))

THEOREM: length-scalar-product
length (scalar-product (n, a)) $=$ length (a)

THEOREM: numberp-listp-scalar-product
numberp-listp (a) $\rightarrow$ numberp-listp (scalar-product (n, a))

THEOREM: main
numberp-listp (a)
$\rightarrow$ **let** n **be** length (a)
in
exp (sumlist (a), n) $\geq$ (exp (n, n) $*$ prodlist (a)) **endlet**

THEOREM: sumlist-for
sumlist (a)
$=$ **for** x **in** a
sum x **endfor**

THEOREM: prodlist-for
prodlist (a)
$=$ **for** x **in** a
multiply x **endfor**

THEOREM: main-again
(numberp-listp (a) $\wedge$ ($n =$ length (a)))
$\rightarrow$ (exp (**for** x **in** a
sum x **endfor**,
n)
$\geq$ (exp (n, n)
$*$ **for** x **in** a
multiply x **endfor**))

Index