```
#|
 Copyright (C) 1994 by Ken Kunen.  All Rights Reserved.

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

 NO WARRANTY

 Ken Kunen PROVIDES 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 Ken Kunen 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.
|#
```


EVENT: Start with the initial **nqthm** theory.

```
; load basic definitions and lemmas

;  From kunen@cs.wisc.edu Mon Oct 21 08:56:34 1991
;  Date: Fri, 18 Oct 91 13:20:25 -0500
;  From: kunen@cs.wisc.edu (Ken Kunen)
;  To: boyer@CLI.COM, kaufmann@CLI.COM
;  Subject: nqthm
;  Cc: kunen@cs.wisc.edu
;
;  The following is one of the examples I'm using in my course here
;  to illustrate nqthm.  In particular, note that the representation
;  of a pair of numbers by an ordinal, as described on p. 42, is more
;  complicated than it has to be.
;                                                        Ken
;
;  ------------------------------------------------------------------------
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;                CS761  --  SEMESTER I, 1991-92                       ;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; nqthm contains induction on epsilon_0, so it's stronger than pure primitive
; recursive arithmetic.  Presumably, it can prove Con(PA).
;         LONG project -- do this

; This file -- a simple example -- use recursion on pairs to define the
; Ackermann function, which grows faster than any primitive recursive function
; see Aho-Hopcroft-Ullman, "Data Structures and Algorithms", p. 189



; Representation of a pair of numbers, (i,j), as the ordinal omega^(i+1) + j;
; This is a little simpler than the one described on Boyer-Moore p. 42.
```

DEFINITION: $\mathrm{rep}\,(i,\, j) = \mathrm{cons}\,(1 + i,\, j)$

DEFINITION:
$\mathrm{lex2}\,(i1,\, j1,\, i2,\, j2) = ((i1 < i2) \vee ((i1 = i2) \wedge (j1 < j2)))$

THEOREM: rep-respects-lex
$((i1 \in \mathbf{N}) \wedge (i2 \in \mathbf{N}) \wedge (j1 \in \mathbf{N}) \wedge (j2 \in \mathbf{N}))$
$\rightarrow \quad (\mathrm{lex2}\,(i1,\, j1,\, i2,\, j2) = \text{ord-lessp}\,(\mathrm{rep}\,(i1,\, j1),\, \mathrm{rep}\,(i2,\, j2)))$

DEFINITION:
$\mathrm{ack}\,(x,\, y)$
$=$ **if** $x \simeq 0$ **then** 1
elseif $y \simeq 0$
then if $x = 1$ **then** 2
else $x + 2$ **endif**
else $\mathrm{ack}\,(\mathrm{ack}\,(x - 1,\, y),\, y - 1)$ **endif**

```
; hint
; "fix"  = "cast to numberp"
```

THEOREM: ack-is-positive
$(\mathrm{ack}\,(x,\, y) \simeq 0) = \mathbf{f}$

THEOREM: ack-of-1
$(x \not\simeq 0) \rightarrow (\mathrm{ack}\,(x,\, 1) = (x * 2))$

DEFINITION:
$\text{expt2}\,(x)$
$= \quad \textbf{if } x \simeq 0 \textbf{ then } 1$
$\quad \textbf{else } \text{expt2}\,(x - 1) * 2 \textbf{ endif}$

THEOREM: ack-of-2-aux1
$(x \not\simeq 0) \rightarrow (\text{ack}\,(x, 2) = \text{ack}\,(\text{ack}\,(x - 1, 2), 1))$

;

```
;      ---------------------
```

THEOREM: ack-of-2-aux2
$(x \not\simeq 0) \rightarrow (\text{ack}\,(x, 2) = (\text{ack}\,(x - 1, 2) * 2))$

;

```
;;;;;;;;;;;;;;;;;;;;;;;;;;
```

THEOREM: ack-of-2
$\text{ack}\,(x, 2) = \text{expt2}\,(x)$

```
; ack(x,3) = 2^2^2^ ...^2  (stack of x 2's, ^ assoc to right)
```

Index