```
#|

 Copyright (C) 1994 by Robert S. Boyer.  All Rights Reserved.

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

 NO WARRANTY

 Robert S. Boyer 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 Robert S. Boyer 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.

```
;  Talk about journal level proof-checkers.

;  This sequence of lemmas describes the relationship between Ackermann's original
;  function and R. Peter's version of it.  The statement of the theorem
;  appears in a manuscript by George D. Herbert, Department of
;  Computer and Information Sciences, University of North Florida, Jacksonville,
;  Florida.  I was asked to review this result by the Monthly. -- rsb


;  Ackermann's original function.
```

DEFINITION:
$\mathrm{a}(m, n, a)$
$=$ **if** $m \simeq 0$ **then** $n + a$
**elseif** $n \simeq 0$ **then** $0$

**elseif** $n = 1$ **then** $a$
**else** $\mathrm{a}(m - 1, \mathrm{a}(m, n - 1, a), a)$ **endif**

```
;  Peter's simplification of it.
```

DEFINITION:
$\mathrm{p}(m, n)$
$=$ **if** $m \simeq 0$ **then** $1 + n$
**elseif** $n \simeq 0$ **then** $\mathrm{p}(m - 1, 1)$
**else** $\mathrm{p}(m - 1, \mathrm{p}(m, n - 1))$ **endif**

```
;  A stupid lemma forcing the theorem-prover to see the obvious.
```

THEOREM: a-open
$((m \not\simeq 0) \wedge (1 < n)) \rightarrow (\mathrm{a}(m, n, a) = \mathrm{a}(m - 1, \mathrm{a}(m, n - 1, a), a))$

```
;  A nice little lemma used in Herbert's proof.
```

THEOREM: a-2-2->4
$\mathrm{a}(m, 2, 2) = 4$

```
;  Another stupid little lemma.
```

THEOREM: plus-2
$(2 + x) = (1 + (1 + x))$

```
;  The main result.
```

THEOREM: a->p
$(\mathrm{p}(0, n) = (1 + n))$
$\wedge \quad ((0 < m) \rightarrow ((3 + \mathrm{p}(m, n)) = \mathrm{a}(m - 1, 1 + (1 + (1 + n)), 2)))$

```
#|

From cowles%UWYO.BITNET@CUNYVM.CUNY.EDU Wed Sep  7 09:16:01 1988
X-St-Return-Receipt-Requested:
Date:     Tue, 6 Sep 88 22:38:14 MDT
From: cowles%UWYO.BITNET@CUNYVM.CUNY.EDU (John Cowles)
Subject:  Ackermann's Original Function.
To: boyer@cli.com, moore@cli.com, kaufmann@cli.com

In the file basic.events, it is claimed that Ackermann's original function
may be recursively defined on the nonnegative integers by
   ack-h(x,y,z) <== if x=0 then y+z
                    else if z=0 then 0
                    else if z=1 then y
```

```
                    else ack-h( x-1, y, ack-h(x,y,z-1) ).
```

However a look at the English translation, in van Heijenoort's source book in mathematical logic, of Ackermann's 1928 paper shows that the original function may be recursively defined on the nonnegative integers by

```
   ack(x,y,z) <== if x=0 then y+z
                  else if x<=2 and z=0 then x-1
                  else if x>2  and z=0 then y
                  else ack( x-1, y, ack(x,y,z-1) ).
```

For all nonnegative y and z, ack(0,y,z) = y+z = ack-h(0,y,z).

For all nonnegative y and z, ack(1,y,z) = y*z = ack-h(1,y,z).

For all nonnegative y and positive z, ack(2,y,z) = y^z = ack-h(2,y,z).

For all nonnegative y and positive z, ack(3,y,z) = ack-h(3,y,z+1).

For x>3, the exact relationship between ack(x,y,z) and ack-h(x,y,z) seems to be difficult to determine.  In any case, ack and ack-h are not the same function!

From cowles@CORRAL.UWyo.Edu Fri Sep 30 17:17:18 1988
X-St-Return-Receipt-Requested:
Date:     Fri, 30 Sep 88 15:09:58 MDT
From: cowles@CORRAL.UWyo.Edu (John Cowles)
Subject:  more on  Ackermann
To: boyer@cli.com

1. Please feel free to add my note on Ackermann's function as
   a comment.  But see 4 below for more on Ackermann's function.

4.
SEVERAL VERSIONS OF ACKERMANN'S FUNCTION

by J. Cowles and T. Bailey
Dept. of Computer Science
University of Wyoming
Laramie, WY

We have located several versions of Ackermann's function which are listed below.  The claims made below about these functions should be carefully verified (by machine if possible).

The original version (1928) of Ackermann' function may be defined recursively on the nonnegative integers by

```
ack( x, y, z ) <== if x=0 then y+z
                   else if x<=2 and z=0 then x-1
                   else if x>2  and z=0 then y
                   else ack( x-1, y, ack(x,y,z-1) ).
```

Then for all nonnegative integers x, y, and z,

```
ack( x, y, 0 ) equals if x=1 then 0
                      else if x=2 then 1
                      else y ;

ack( 0, y, z ) equals y+z ;

ack( 1, y, z ) equals y*z ;

ack( 2, y, z ) equals y^z ;

ack( 3, y, z ) equals iter-exp(y,z+1) ;

ack( x, 1, z ) equals if x=0 then z+1
                      else if x=1 then z
                      else 1 .
```

Here the function iter-exp is defined recursively on the nonnegative integers by

```
iter-exp( y, z ) <== if z=0 then 1
                     else y ^ iter-exp(y,z-1).
```

==========================================================

The Herbert version of Ackermann's function may be defined recursively on the nonnegative integers by

```
ack-h( x, y, z ) <== if x=0 then y+z
                     else if z=0 then 0
                     else if z=1 then y
                     else ack-h( x-1, y, ack-h(x,y,z-1) )
```

Then for all nonnegative integers x, y, and z,

```
ack-h( x, y, 0 ) equals if x=0 then y
                        else 0 ;

ack-h( x, y, 1 ) equals if x=0 then y+1
                        else y ;

ack-h( 0, y, z ) equals y+z ;

ack-h( 1, y, z ) equals y*z ;

ack-h( 2, y, z ) equals if z=0 then 0
                        else y^z ;

ack-h( 3, y, z ) equals if z=0 then 0
                        else iter-exp(y,z) ;

ack-h( x, 2, 2 ) equals 4 .
```

```
============================================================
```

The Meyer and Ritchie version (1967) of Ackermann's function may be defined recursively on the nonnegative integers by

```
ack-mr( x, z ) <== if z=0 then 1
                   else if x=0 and z=1 then 2
                   else if x=0 and z>1 then z+2
                   else ack-mr( x-1, ack-mr(z,z-1) ).
```

Then for all nonnegative integers x and z,

```
ack-mr( x, 0 ) equals 1 ;

ack-mr( 0, z ) equals if z=0 then 1
                      else if z=1 then 2
                      else 2+z ;

ack-mr( 1, z ) equals if z=0 then 1
                      else 2*z ;

ack-mr( 2, z ) equals 2^z ;

ack-mr( 3, z ) equals iter-exp(2,z) ;
```

```
ack-mr( x, 1 ) equals 2 ;

ack-mr( x, 2 ) equals 4 .

=======================================================

The R. Peter version (1935) of Ackermann's function may
be defined recirsively on the nonnegative integers by

ack-p( x, z ) <== if x=0 then z+1
                  else if z=0 then ack-p(x-1,1)
                  else ack-p( x-1, ack-p(x,z-1) ).

Then for all nonnegative integers z,

ack-p( 0, z ) equals z+1 ;

ack-p( 1, z ) equals z+2 ;

ack-p( 2, z ) equals 2*z + 3 ;

ack-p( 3, z ) equals 2^(z+3) - 3 ;

ack-p( 4, z ) equals iter-exp(2,z+3) - 3 .

=======================================================

The Z. Manna version (1974) of Ackermann's function may
be defined recursively on the nonnegative integers by

ack-m( x, z ) <== if x*z = 0 then x+z+1
                  else ack-m( x-1, ack-m( x,z-1) ).

Then for all nonnegative integers z,

ack-m( 0, z ) equals z+1 ;

ack-m( 1, z ) equals z+2 ;

ack-m( 2, z ) equals 2*z + 3 ;

ack-m( 3, z ) equals 7*(2^z) - 3 ;
```

```
ack-m( 4, z ) equals h(2,z) - 3 .

Here the function h is defined recursively on
the nonnegative integers by

h( x, z ) <== if z=0 then y^3
              else 7*( y^[h(y,z-1)-3] ).

==========================================================

Since everyone else has a version of Ackermann's function,
it should cause little or no harm if we also define a version.

ack-cb( x, y, z ) <== if x= 0 then y+z
                      else if x=1 and z=0 then 0
                      else if x>1 and z=0 then 1
                      else ack-cb( x-1, y, ack-cb(x,y,z-1) ).

Then for all nonnegative integers x, y, and z,

ack-cb( x, y, 0 ) equals if x=0 then y
                         else if x=1 then 0
                         else 1 ;

ack-cb( 0, y, z ) equals y+z ;

ack-cb( 1, y, z ) equals y*z ;

ack-cb( 2, y, z ) equals y^z ;

ack-cb( 3, y, z ) equals iter-exp(y,z) ;

ack-cb( x, y, 1 ) equals if x=0 then y+1
                         else y ;

ack-cb( x, 2, 2 ) equals 4 .

===================================================

Then for all nonnegative integers x, y, and z,

ack-h( x, y, z+1 ) equals  ack-cb( x, y, z+1 ) ;

ack-mr( x, z+2 )   equals  ack-cb( x, 2, z+2 )
```

```
                   equals  ack-h ( x, 2, z+2 ) ;

ack-p( x+1, z )    equals  ack-mr( x, z+3 ) - 3
                   equals  ack-h ( x, 2, z+3 ) - 3
                   equals  ack-cb( x, 2, z+3 ) - 3 .


|#
```

# Index