<AFFIRM>TEDIT..14 30-Sep-81 15:38:10

```
(FILECREATED "26-Sep-81 19:13:01" <AFFIRM>TEDIT..14 21914

     previous date: "26-Sep-81 18:11:17" <AFFIRM>TEDIT..13)


(PRETTYCOMPRINT TEDITCOMS)

(RPAQQ TEDITCOMS [(FNS CheckDefs CreateExpansions EditTheorem ExpandDefs ExpandVisibleDefs
                         ExpandVisibleDefs1 ShortenExpr)
                  (FNS * TEDITPRINTFNS)
                  (* ↑ macros for editor: PPa calls PP with current expression made with short names.
                     Note: current is clobbered,reset. We first try to place the short-named stuff in
                     place of the first element,descend into it, and print. If this fails, we replace
                     the current expression with the shortened thing. If that fails, we punt and
                     print with long names.)
                  (USERMACROS PPa)
                  (* Pa macro does the housekeeping P would do, calls our custom version of LVLPRINT)
                  (USERMACROS Pa)
                  (* CollectAnds converts expressions of the form (AND (AND & &)
                                                                      &)
                     to calls to n-ary AND. Note that this depends on the way RemoveIfs nests them;
                     when the fix is made, this must be switched.)
                  (USERMACROS CollectAnds)
                  (* (invoke ...list of defns...)
                     expands all instances of these definitions in the current expression.)
                  (USERMACROS invoke)
                  (* eval replaces current with evaluated version. Tries to do a :. If that fails,
                     clobbers the first list cell.)
                  (USERMACROS eval)
                  (USERMACROS infix)
                  (VARS NANDOP ExpandAllDefns)
                  (FNS AND\Nary AND1\Nary)
                  (FNS ParityP sense)
                  (FNS DeletableP fixdeleteordinals extractinstead)
                  (* OPERANDTRIM determines when operands may be deleted. OPERATOR2ARY shows when
                     these operators demand 2 arguments (hence we XTR instead))
                  (VARS OPERATORSENSE OPERANDTRIM OPERATOR2ARY)
                  (USERMACROS delete extract hyp con trim)
                  (DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS
                            (ADDVARS (NLAMA AND\Nary)
                                     (NLAML)
                                     (LAMA])
(DEFINEQ
```

1

```
(CheckDefs
  [LAMBDA (defs context)                                    (* R.Bates "14-Dec-79 14:43")
    (PROG (snan ds)
          (snan←(REMOVEDUPLICATES (ShortNamesAndNames context)))
          (ds←(for d in defs when (LITATOM d) and d~='', and (if (FASSOC d snan)
                                                                 else (PRINTLINES T d "?" T))
                collect d))
          (RETURN (for d in ds collect (FASSOC d snan)::1])
```

2

```
(CreateExpansions
  [LAMBDA (e upperalist)                                    (* R.Bates "14-Dec-79 14:43")

          (* Setup translation for user input of short names to editor. Return an alist with entries ((Shorten atom) .atom)
          for all atoms in e with extensions. Upperalist contains entries we wont duplicate)


    (PROG (alist short)
          (RETURN (if (ATOM e)
                      then short←(Shorten e)
                           (if short~=e and ~(FASSOC short upperalist)
                               then <<short ! e>>
                             else NIL)
                    elseif (LISTP e)
                      then (for l in e do alist← < !!(CreateExpansions l
                                                                        < ! alist ! upperalist>)
                                                    ! alist>)
                           alist
                    else NIL])
```

3

```
(EditTheorem
  [LAMBDA (editorCommands)                                    (* R.Erickson "18-Apr-80 13:55")

          (* * Allow user to edit a theorem, using edit macros we provide. Return the transformation for possible annotation.)


    (PROG ((edited (getPrettyNorm))
           trans)
          (editorCommands←(if editorCommands
                              then                            (* commands given: don't ask for more)
                                   <'CollectAnds !(MKLIST editorCommands) >
                          else                                (* no commands given. Read from the terminal)
                               '(CollectAnds TTY:)))          (* STOP will pop us up to master UNDONLSETQ)
          (EDITE edited editorCommands)
          [Descend (Transform trans←(create Transformation
                                            command ←('@)
                                            children ←(<edited>]
          (RETURN trans])
```

4

```
(ExpandDefs
  [LAMBDA (defns expression option)                           (* D.Thompson " 9-Mar-80 18:32")
                                                              (* called from editor macro)
    (SELECTQ option
             [(first NIL)
               (RemoveIfs (EVAL (ExpandVisibleDefs defns expression T]
             [all (RemoveIfs (EVAL (ExpandVisibleDefs defns expression]
             NIL])
```

5

```
(ExpandVisibleDefs
  [LAMBDA (defns expr firstonly)                              (* R.Erickson "18-DEC-78 13:22")

          (* expand all instances of these defns in the current expression and return the result. Note that simply using
          InvokedDefinitions and ExpandAllDefns would cause looping on recursive definitions. Does MANY conses.
          If firstonly, expand 1st of each in preorder traversal.)


    (ExpandVisibleDefs1 expr])
```

6

```
(ExpandVisibleDefs1
  [LAMBDA (xpr)                                               (* R.Bates "14-Dec-79 14:42")
                                                              (* recursively run thru xpr. Upon finding an instance
                                                              (defn args) recurse for args, and expand defn)
    (if (NLISTP xpr) or ~defns
        then xpr
      elseif xpr:Operator ~MEMB defns
        then <xpr:Operator !(for e in xpr:Arguments collect (ExpandVisibleDefs1 e)) >
      else                                                    (* xpr:Operator memb defns)
           (RESETVARS ((InvokedDefinitions defns)
                       (ExpandAllDefns NIL))
                      (if firstonly
                          then defns←(REMOVE xpr:Operator defns))
                                                              (* stop expansion here)
                      (RETURN (APPLY xpr:Operator (for e in xpr:Arguments
                                                     collect (ExpandVisibleDefs1 e])
```

7

```
(ShortenExpr
  [LAMBDA (e)                                                 (* RodErickson " 5-OCT-78 17:30")
    (if e=NIL
        then NIL
      elseif (LITATOM e)
        then (Shorten e)
      elseif (LISTP e)
        then (if e:1=QOP
                 then <'given (ShortenExpr e:given)
                              'find
                              (ShortenExpr e:find)
                              (ShortenExpr e:expr) >
               else (for l in e collect (ShortenExpr l)))
```

```
      else e])
)

(RPAQQ TEDITPRINTFNS (TeditPrint TeditPrint0 TeditLVLPRIN TeditLVLPRIN0))
(DEFINEQ
```

8

```
(TeditPrint
  [LAMBDA (X)                                          (* R.Bates "23-Jan-80 13:40")
                                                       (* like BPNT,except calls TeditLVLPRINT)
    (PROG (Y N Z)
          [if X:1=0
              then Y←L:1
                   Z←L:2
            else Y←(CAR (if (MINUSP X:1)
                            then (NLEFT L:1-X:1)
                          else (FNTH L:1 X:1]
          (if X::1=NIL
              then N←1
            elseif ~(NUMBERP N←X:2)
              then (ERROR!)
            elseif (MINUSP N)
              then N←N+1
            else                                       (* Makes (P 0 N) have same effect as it did in old
                                                       system.)
                 N←N-1)
          (RETURN (TeditPrint0 Y T N (OR X:3 20)
                               Z])
```

9

```
(TeditPrint0
  [LAMBDA (X FILE CARLVL CDRLVL TAIL)                  (* edited: " 5-OCT-78 13:46")
                                                       (* combines LVLPRINT.LVLPRIN2 but suited to Affirm
                                                       theorems)
    (PROG ((PRIN2FLG T))
          (TeditLVLPRIN X CARLVL CDRLVL TAIL))
    (TERPRI FILE)
    X])
```

10

```
(TeditLVLPRIN
  [LAMBDA (X CARLVL CDRLVL TAIL)                       (* R.Bates "23-Jan-80 12:58")
    (if (NLISTP X)
        then (if TAIL and X=TAIL::-1::1 and X ~MEMB TAIL
                 then (PRIN1 '"... . " FILE)
                      (if PRIN2FLG
                          then (PRIN2 (Shorten X)
                                      FILE T)
                        else (PRIN1 (Shorten X)
                                    FILE))
```

(* We use standard system read table for printing on grounds that even if this is going to a file, user is only dumping it with bpnt to look at it, not to read it back in.)

```
                 (PRIN1 '%) FILE)
            elseif PRIN2FLG
              then (PRIN2 (Shorten X)
                          FILE T)
            else (PRIN1 (Shorten X)
                        FILE))
      else (PRIN1 (if TAIL and (TAILP X TAIL)
                      then                             (* Tail)
                           '"... "
                    else '%()
                  FILE)
           (TeditLVLPRIN0 X CARLVL CDRLVL)
           (PRIN1 '%) FILE])
```

11

```
(TeditLVLPRIN0
  [LAMBDA (X CARLVL CDRLVL)                            (* edited: " 5-OCT-78 14:55")
                                                       (* like LVLPRIN0, except shortens atoms when printing,
```

```
                                                    treats QOP specially, ignores COMMENTFLG &nd
                                                    CLISPTRANFLG)
                                                    (* LVLPRIN0 is like subprint %.
                                                    It prints the interior segment of a list)
    (PROG ((CDRLVL0 CDRLVL)
           X0)
          (GO LP1)
      LP  (if X←X::1=NIL
              then (RETURN)
            elseif (NLISTP X)
              then (PRIN1 '" . " FILE)
                   (if PRIN2FLG
                       then (PRIN2 (Shorten X)
                                   FILE T)
                     else (PRIN1 (Shorten X)
                                 FILE))
                   (RETURN)
            elseif CARLVL and (MINUSP CARLVL) and (LISTP X0) and (LISTP X:1)
              then (TERPRI FILE)
            else (SPACES 1 FILE))
      LP1 (if CDRLVL=0
              then (PRIN1 '-- FILE)
                   (RETURN)
            elseif X:1=QOP
              then                                  (* X = (Qexpression (given) (find)
                                                    (free) expr))
                   (PRIN1 'given FILE)
                   (SPACES 1 FILE)
                   (TeditLVLPRIN X:given CARLVL (AND CDRLVL0 CDRLVL0-1))
                   (SPACES 1 FILE)
                   (PRIN1 'find FILE)
                   (SPACES 1 FILE)
                   (TeditLVLPRIN X:find CARLVL (AND CDRLVL0 CDRLVL0-1))
                   (SPACES 1 FILE)
                   (TeditLVLPRIN X:expr (AND CARLVL CARLVL+(if (MINUSP CARLVL)
                                                               then 1
                                                             else -1))
                                 (AND CDRLVL0 CDRLVL0-1))   (* note that we skip the free list)
                   (RETURN)
            elseif (NLISTP X:1)
              then (if PRIN2FLG
                       then (PRIN2 (Shorten X:1)
                                   FILE T)
                     else (PRIN1 (Shorten X:1)
                                 FILE))
            elseif CARLVL=0 or CDRLVL0 and CDRLVL0-1=0
              then                                  (* the reason for the second check is that why bother to
                                                    recurse only to print (--). & is better)
                   (PRIN1 '& FILE)
            else (PRIN1 '%( FILE)
                 (TeditLVLPRINO X:1 (AND CARLVL CARLVL+(if (MINUSP CARLVL)
                                                           then 1
                                                         else -1))
                                    (AND CDRLVL0 CDRLVL0-1))
                 (PRIN1 '%) FILE))
          (if CDRLVL
              then CDRLVL←CDRLVL-1)
          (GO LP])
)
[DECLARE: DONTEVAL@LOAD DONTCOPY
```

```
(* ↑ macros for editor: PPa calls PP with current expression made with short names. Note: current
   is clobbered,reset. We first try to place the short-named stuff in place of the first
   element,descend into it, and print. If this fails, we replace the current expression with the
   shortened thing. If that fails, we punt and print with long names.) ]

(ADDTOVAR USERMACROS [PPa NIL (BIND (ORR ((E (SETQ #1 (## 1))
                                             T)
                                          (I 1 (ShortenExpr (##)))
                                          1
                                          (ORIGINAL PP)
                                          0
                                          (I 1 #1))
                                         ((E (SETQ #1 (##))
                                             T)
                                          (I : (ShortenExpr (##)))
                                          (ORIGINAL PP)
                                          (I : #1))
                                         (ORIGINAL PP])

(ADDTOVAR EDITCOMSA PPa)
[DECLARE: DONTEVAL@LOAD DONTCOPY
```

(* Pa macro does the housekeeping P would do, calls our custom version of LVLPRINT)]

```
(ADDTOVAR USERMACROS (Pa NIL (E (COND ((NEQ LASTP1 L)
                                       (SETQ LASTP2 LASTP1)
                                       (SETQ LASTP1 L)))
                                  T)
                             (E (TeditPrint0 (CAR L)
                                             T 2 20 (CADR L))
                                T))
                       (Pa args (E (COND ((NEQ LASTP1 L)
                                          (SETQ LASTP2 LASTP1)
                                          (SETQ LASTP1 L)))
                                   T)
                             (E (TeditPrint (QUOTE args))
                                T)))

(ADDTOVAR EDITCOMSA Pa)

(ADDTOVAR EDITCOMSL Pa)
[DECLARE: DONTEVAL@LOAD DONTCOPY
```

```
(* CollectAnds converts expressions of the form (AND (AND & &)
                                                    &)
   to calls to n-ary AND. Note that this depends on the way RemoveIfs nests them; when the fix is
   made, this must be switched.) ]

(ADDTOVAR USERMACROS (CollectAnds NIL (BIND (E (SETQ #1 (LIST ANDOP (LIST ANDOP (QUOTE --))
                                                                   (QUOTE &)))
                                               T)
                                           (E (SETQ #2 (LIST ANDOP (QUOTE --)))
                                               T)
                                           (MARK #3)
                                           (LPQ (I F #1 (QUOTE N))
                                                MARK
                                                (I 1 NANDOP)
                                                (BO 2)
                                                (2)
                                                (LPQ (I F #2 NIL)
                                                     UP
                                                     (BO 1)
                                                     (1))
                                                ←)
                                           (\ #3))))

(ADDTOVAR EDITCOMSA CollectAnds)
[DECLARE: DONTEVAL@LOAD DONTCOPY
```

**(* (invoke ...list of defns...)
expands all instances of these definitions in the current expression.)]**

```
(ADDTOVAR USERMACROS (invoke defs (BIND (E [COND [(MEMB (CAR (QUOTE defs))
                                                        (QUOTE (all first)))
                                                  (SETQ #1 (CDR (QUOTE defs)))
                                                  (SETQ #2 (CAR (QUOTE defs]
                                                 (T (SETQ #1 (QUOTE defs))
                                                    (SETQ #2 (QUOTE first]
                                              T)
                                             (E (SETQ #1 (CheckDefs #1 (##)))
                                              T)
                                             (IF #1)
                                             [IF (CDR L)
                                                 ((IF (TAILP (##)
                                                             (## 0))
                                                      (MARK 1 (LPQ (I : (ExpandDefs #1 (##)
                                                                                    #2))
                                                                   NX)
                                                            ←)
                                                      ((I : (ExpandDefs #1 (##)
                                                                           #2))
                                                       1)))
                                                  ((IF (EQ (## 1)
                                                           QOP)
                                                      (5 (I : (ExpandDefs #1 (##)
                                                                          #2))
                                                         0)
                                                      (MARK 2 (LPQ (I : (ExpandDefs #1 (##)
                                                                                    #2))
                                                                   NX)
                                                            ←]
                                             CollectAnds)))

(ADDTOVAR EDITCOMSL invoke)
[DECLARE: DONTEVAL@LOAD DONTCOPY
```

(* eval replaces current with evaluated version. Tries to do a :. If that fails, clobbers the first list cell.)]

```
(ADDTOVAR USERMACROS [eval NIL (BIND (IF (COND ((TAILP (CAR L)
                                                       (CADR L))
                                                  (PRINTLINES "  !0 is implicit for tail" T)
                                                  T))
                                             (!0)
                                             NIL)
                                       (E [SETQ #1 (RemoveIfsQ (EVAL (##]
                                          T)
                                       (ORR ((I : #1)
                                             1 CollectAnds Pa)
                                            ((IF (LISTP #1))
                                             (E (SETQ #2 (##))
                                                T)
                                             (E (/RPLACA #2 (CAR #1))
                                                T)
                                             (E (/RPLACD #2 (CDR #1))
                                                T)
                                             CollectAnds Pa)
                                            ((E "can't - at top. Result would be:")
                                             (E #1)
                                             (E (ERROR!])

(ADDTOVAR EDITCOMSA eval)

(ADDTOVAR USERMACROS (infix NIL (E (INFIX\PRINT (##))
                                   T)))

(ADDTOVAR EDITCOMSA infix)

(RPAQQ NANDOP AND\Nary)

(RPAQQ ExpandAllDefns NIL)
(DEFINEQ
```

12

```
(AND\Nary
  [NLAMBDA n%                                    (* edited: " 5-OCT-78 11:50")
                                                 (* reconstruct the IfThenElse form from nary AND)
    (EVAL (AND1\Nary n% ])
```

13

```
(AND1\Nary
  [LAMBDA (l)                                    (* R.Erickson "23-Sep-81 17:47")
                                                 (* l is a list of unevaluated elements to be ANDed
                                                 together. Construct IfThenElse form)
    (if l
        then <IFOP l:1 (if l::1
                           then (AND1\Nary l::1)
                           else TRUE)
                FALSE>
      else TRUE])
)
(DEFINEQ
```

14

```
(ParityP
  [LAMBDA (coms)                                 (* R.Erickson "20-Mar-80 12:50")
                                                 (* is the global s of the expression designated by "coms"

                                                 (possibly NIL) positive (T) or negative
                                                 (NIL) ?)
    (PROG (senses gs prev)
          (if coms and (NLISTP coms)
              then coms← <coms>)
          (prev←(APPLY '## coms))

          (* we might start out with a tail as (# # coms) Since this presumably will involve the (:) command, we really are
          interested in the 1st element.)

          (NLSETQ (if (TAILP prev (APPLY '## < ! coms '!0 >))
```

```
                then coms← < !! coms '!0 > prev←(APPLY '## coms)))
        (senses←(for (zer ←(QUOTE (!0))) by <'!0 ! zer> bind this s
                   while (NLSETQ (PROGN this←(APPLY '## < ! coms ! zer>)
                                        s←(sense prev this)
                                        prev←this))
                   collect s))
        (gs←T)
        (for s in senses until gs='undefined
           do gs←(if s='undefined
                     then 'undefined
                   elseif s
                     then gs
                   else ~gs))
        (RETURN gs])
```

15

(sense

```
  [LAMBDA (expr containing)                                    (* R.Bates "14-Dec-79 14:43")

        (* expr is a member of containing:Arguments. We wish to know if containing:Operator makes exprs sense positive or
        negative. (Return T if positive) OPERATORSENSE is an alist with elements "(op.T)" if the sense is pos.
        for all args, (like ANDOP) "(op.(T NIL T))" where it varies among args (as in IMPOP) Operators not on the list are
        considered to have undefined sense.)


    (PROG (assoc slist count)
          (assoc←(FASSOC containing:Operator OPERATORSENSE))
          (if assoc
              then slist←assoc::1
                   (if (LISTP slist)
                       then                                    (* check arg posn)
                            count←(for i from 1 as j in containing:Arguments when j=expr
                                     do (RETURN i))
                            (if count
                                then (RETURN (FNTH slist count):1)
                              else (ERROR "sense has bad args" <expr containing>))
                     else (RETURN slist))
            else (RETURN 'undefined])
)
(DEFINEQ
```

16

(DeletableP

```
  [LAMBDA (coms)                                               (* R.Bates "14-Dec-79 14:43")

        (* May the expression denoted by coms be deleted (with its !0 context) or XTRacted from that context? If "(Op.s)" is
        in OPERANDTRIM, we may delete operands, making the theorem stronger, if the sense of the (OP --) expression is s.)


    (PROG (n1 opt)
          (if (LISTP coms)
              and n1←(NLSETQ (PROGN opt←(FASSOC (APPLY '## < ! coms '!0 '1 >)
                                                OPERANDTRIM)
                                    (if opt
                                        then (ParityP < ! coms '!0 >)=opt::1
                                             and (coms:-1~=1
                                                    or (TAILP (APPLY '## coms)
                                                              (APPLY '## < ! coms '!0 >)))
                                      else NIL)))
            then (RETURN n1:1)
            else (RETURN NIL])
```

17

(fixdeleteordinals

```
  [LAMBDA (ords)                                               (* R.Erickson "12-OCT-78 11:25")

        (* ords is a list of ordinal #'s, eg (1 3 4 5) Since we delete one at a time, we have to shift them.)


    (PROG (sords)
          (sords←(SORT < ! ords> 'ILESSP))
          (RETURN (for i from 0 as j in sords collect j-i])
```

18

```
(extractinstead
  [LAMBDA (dels)                                    (* R.Bates "24-Jan-80 13:15")
                                                    (* were about to delete the operands inpositions "dels".
                                                    Some binary operators wont like this. so XTR instead)

    (if dels::1
        then (if (## 1) MEMB OPERATOR2ARY
                 then (PRINTLINES T "cant delete >1 arg of binary operator! " T)
                      (ERROR!)
             else NIL)
      else (## 1) MEMB OPERATOR2ARY])
)
[DECLARE: DONTEVAL@LOAD DONTCOPY
```

```
(* OPERANDTRIM determines when operands may be deleted. OPERATOR2ARY shows when these op
erators
   demand 2 arguments (hence we XTR instead)) ]


(RPAQQ OPERATORSENSE ((Qexpression undefined undefined undefined T)
                      (AND\Nary . T)
                      (NE\Boolean . undefined)
                      (EQV\Boolean . undefined)
                      (NOT\Boolean NIL)
                      (IMP\Boolean NIL T)
                      (OR\Boolean . T)
                      (AND\Boolean . T)
                      (IfThenElse . undefined)))

(RPAQQ OPERANDTRIM ((AND\Nary)
                    (AND\Boolean)
                    (OR\Boolean . T)))

(RPAQQ OPERATOR2ARY (AND\Boolean OR\Boolean))

(ADDTOVAR USERMACROS [extract this (IF (DeletableP (CAR (QUOTE this)))
                                       [(I XTR (CAR (QUOTE this]
                                       ((E "may not extract"]
                     [delete which (IF (for i in (QUOTE which)
                                          always
                                          (DeletableP <i>))
                                       [(IF (extractinstead (QUOTE which))
                                            ((I XTR (if (CAR (QUOTE which))
                                                        =2 then 3 else 2)))
                                            ((BIND (E (SETQ #1 (fixdeleteordinals (QUOTE which)))
                                                      T)
                                                   (LPQ (IF #1)
                                                        (COMS (LIST (CAR #1)))
                                                        (E (SETQ #1 (CDR #1))
                                                           T]
                                       ((E "may not delete"]
                     (con NIL 5 3)
                     (hyp NIL 5 2)
                     (extract NIL (E "syntax is (extract n)")
                              (E (ERROR!)))
                     (delete NIL (E "syntax is (delete 2 4 5 etc) ")
                             (E (ERROR!)))
                     [trim NIL (BIND (IF L::1 NIL (↑ hyp))
                                     1
                                     (LPQ Pa (E [SETQ #1
                                                      (ASKUSER NIL NIL "delete it?"
                                                               (QUOTE ((Y . "es [confirm]")
                                                                       (N "o
")
                                                                       ("" NIL EXPLAINSTRING
                                                                  " CR or space -- No is default"
                                                                       RETURN
                                                                       (QUOTE N))
                                                                       (E "nd normally
")
                                                                       (A .
                                                                  "bort and discard all edits [confirm]"]
                                                  T)
                                                (IF (EQ #1 (QUOTE Y))
                                                    ((:)
                                                     1)
                                                    ((IF (EQ #1 (QUOTE N))
                                                         (NX)
                                                         ((IF (EQ #1 (QUOTE A))
                                                              (STOP])

(ADDTOVAR EDITCOMSA con hyp extract delete trim)

(ADDTOVAR EDITCOMSL extract delete)
(DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR NLAMA AND\Nary)

(ADDTOVAR NLAML )

(ADDTOVAR LAMA )
)
(DECLARE: DONTCOPY
  (FILEMAP (NIL (1952 6335 (CheckDefs 1964 . 2405) (CreateExpansions 2409 . 3169) (EditTheorem 3173 .
4023) (ExpandDefs 4027 . 4431) (ExpandVisibleDefs 4435 . 4899) (ExpandVisibleDefs1 4903 . 5862) (
```