```
(FILECREATED "22-Sep-80 19:14:22" <AFFIRM>BOOLEAN..20 13012

      changes to:  SOME\Boolean ALL\Boolean

      previous date: "11-Aug-80 21:32:48" <AFFIRM>BOOLEAN..19)


(PRETTYCOMPRINT BOOLEANCOMS)

(RPAQQ BOOLEANCOMS [(P (CheckLoad (QUOTE TYPE)
                                  (QUOTE (110 . <AFFIRM>BASE-AFFIRM.EXE.54))
                                  (QUOTE Boolean)))
                    (FNS * BooleanFNS)
                    (FNS * Boolean\InterfaceFNS)
                    (VARS * BooleanConstants)
                    (VARS * Boolean\InterfaceConstants)
                    (IFPROP ALL * BooleanConstants)
                    (IFPROP (PrimaryLHSides EqualOp EQOP)
                            * BooleanFNS)
                    (IFPROP (PrimaryLHSides EqualOp EQOP)
                            * Boolean\InterfaceFNS)
                    [P (InitializeLoad TYPE Boolean 110 ((NoteInterfaces Boolean\InterfaceFNS)
                                       (initInfix (QUOTE Boolean))
                                       (initNeeds (QUOTE Boolean))
                                       (NoteDeclarations (QUOTE Boolean))
                                       (NoteLeftHandSides BooleanFNS]
                    (DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS
                              (ADDVARS (NLAMA)
                                       (NLAML ALL\Boolean SOME\Boolean)
                                       (LAMA])
(CheckLoad (QUOTE TYPE)
           (QUOTE (110 . <AFFIRM>BASE-AFFIRM.EXE.54))
           (QUOTE Boolean))

(RPAQQ BooleanFNS (SOME\Boolean ALL\Boolean EQV\Boolean NOT\Boolean IMP\Boolean OR\Boolean AND\Boolean
                   IH\Boolean))
(DEFINEQ

                                                                                              1

(SOME\Boolean
  [NLAMBDA (var%  ex% )                                       (* R.Erickson "22-Sep-80 15:32")

          (* * We rebind Assumed/Denied when evaluating ex. See All\Boolean, Qexpression.)


    var% ←(EVAL var% )
    (if (LISTP var% )
        then                                                  (* user slipped in a nilary constant.)
             (AffirmError <"Not a variable:" (Shorten var% :Operator)
                           >))
    ex% ←(PROG (Assumed Denied)
               (RETURN (EVAL ex% )))
    (if ex% :Operator=QOP
        then (if (FASSOC var%  ex% :given) or (FASSOC var%  ex% :find) or var%  ~MEMB ex% :free
                 then ex%
               else (create Qexpression
                            find ←(<<var% > ! ex% :find>)
                            given ←(for v in ex% :given collect (AddArg var%  v))
                            free ←(REMOVE var%  ex% :free)
                            expr ← ex% :expr))
      else (create Qexpression
                   find ←(<<var% >>)
                   given ← NIL
                   free ←(REMOVE var%  (Frees ex% ))
                   expr ← ex% ])

                                                                                              2

(ALL\Boolean
  (NLAMBDA (var% ex% )                                     (* R.Erickson "22-Sep-80 15:30")

         (* * We have to be an NLAMBDA and rebind Assumed/Denied when evaluating ex, since otherwise var might conflict
         with A/D)


    var% ←(EVAL var% )
    (if (LISTP var% )
        then                                                    (* user slipped in a nilary constant.)
             (AffirmError <"Not a variable:" (Shorten var% :Operator)
                         >))
    ex% ←(PROG (Assumed Denied)
              (RETURN (EVAL ex% )))
    (if ex% :Operator=QOP
        then (if (FASSOC var% ex% :given) or (FASSOC var% ex% :find) or var% ~MEMB ex% :free
                 then ex%
               else (create Qexpression
                            given ←(<<var% > ! ex% :given>)
                            find ←(for v in ex% :find collect (AddArg var% v))
                            free ←(REMOVE var% ex% :free)
                            expr ← ex% :expr))
      else (create Qexpression
                   given ←(<<var% >>)
                   find ← NIL
                   free ←(REMOVE var% (Frees ex% ))
                   expr ← ex% ))
```

3

```
(EQV\Boolean
  (LAMBDA (b1 b2)
    (if (Report EQV\Boolean 1 axiom)
        then (IfThenElse b2 (IfThenElse b1 TRUE FALSE)
                         (IfThenElse b1 FALSE TRUE))
      elseif <'EQV\Boolean b1 b2>]))
```

4

```
(NOT\Boolean
  (LAMBDA (b1)
    (if (Report NOT\Boolean 1 axiom)
        then (IfThenElse b1 FALSE TRUE)
      elseif <'NOT\Boolean b1>]))
```

5

```
(IMP\Boolean
  (LAMBDA (b1 b2)
    (if (Report IMP\Boolean 1 axiom)
        then (IfThenElse b1 (IfThenElse b2 TRUE FALSE)
                         TRUE)
      elseif <'IMP\Boolean b1 b2>]))
```

6

```
(OR\Boolean
  (LAMBDA (b1 b2)
    (if (Report OR\Boolean 1 axiom)
        then (IfThenElse b1 TRUE (IfThenElse b2 TRUE FALSE))
      elseif <'OR\Boolean b1 b2>]))
```

7

```
(AND\Boolean
  (LAMBDA (b1 b2)
    (if (Report AND\Boolean 1 axiom)
        then (IfThenElse b1 (IfThenElse b2 TRUE FALSE)
                                FALSE)
          elseif <'AND\Boolean b1 b2>])
```

8

```
(IH\Boolean
  [LAMBDA (val target)                                            (* R.Bates " 1-Jul-80 14:33")

            (* * created by IHOP. appears in propositions. Handcoded. takes the place of a dummy Boolean rule)


    (if (Report IH\Boolean 1 defn.) and ((FIXP target) or (LITATOM target) and ~(Extension target))
        then

            (* test is to make sure we have valid nodeid, not just a formal parameter from LHS (which happens when print
            type Boolean))


              (PROG (node var)
                    (node←(GetNode target))
                    (if node and (L-CASE node:trans:command)='employ
                        then var←node:trans:parameters:1:Arg1
                             (RETURN (ComputeInductionExpression (ActualExprAt target)
                                                                 var val))
                      else (AffirmError <"IH no longer current for" target>)))
      elseif <IH2OP val target>])
)

(RPAQQ Boolean\InterfaceFNS (SOME\Boolean\Interface ALL\Boolean\Interface EQV\Boolean\Interface
                             NOT\Boolean\Interface IMP\Boolean\Interface
                             OR\Boolean\Interface AND\Boolean\Interface
                             IH\Boolean\Interface))
(DEFINEQ
```

9

```
(SOME\Boolean\Interface
  [LAMBDA (a1 b1 TooManyArguments)
    (if a1:1='ExpressionWithType and b1:1='ExpressionWithType and b1:3=Boolean and TooManyArguments=NIL
          and (Report SOME\Boolean\Interface 1 interface)
        then (ExpressionWithType <'SOME\Boolean a1:2 b1:2> b1:3)
      elseif NIL])
```

10

```
(ALL\Boolean\Interface
  [LAMBDA (a1 b1 TooManyArguments)
    (if a1:1='ExpressionWithType and b1:1='ExpressionWithType and b1:3=Boolean and TooManyArguments=NIL
          and (Report ALL\Boolean\Interface 1 interface)
        then (ExpressionWithType <'ALL\Boolean a1:2 b1:2> b1:3)
      elseif NIL])
```

11

```
(EQV\Boolean\Interface
  [LAMBDA (b1 b2 TooManyArguments)
    (if b1:1='ExpressionWithType and b2:3=Boolean and b2:1='ExpressionWithType and b2:3=Boolean
          and TooManyArguments=NIL and (EQUAL b1:3 b2:3) and (Report EQV\Boolean\Interface 1 interface)
        then (ExpressionWithType <'EQV\Boolean b1:2 b2:2> b2:3)
      elseif NIL])
```

12

```
(NOT\Boolean\Interface
  [LAMBDA (b1 TooManyArguments)
    (if b1:1='ExpressionWithType and b1:3=Boolean and TooManyArguments=NIL
          and (Report NOT\Boolean\Interface 1 interface)
```

```
        then (ExpressionWithType <'NOT\Boolean b1:2> b1:3)
      elseif NIL))
```

13

```
(IMP\Boolean\Interface
  (LAMBDA (b1 b2 TooManyArguments)
    (if b1:1='ExpressionWithType and b2:3=Boolean and b2:1='ExpressionWithType and b2:3=Boolean
          and TooManyArguments=NIL and (EQUAL b1:3 b2:3) and (Report IMP\Boolean\Interface 1 interface)
        then (ExpressionWithType <'IMP\Boolean b1:2 b2:2> b2:3)
      elseif NIL))
```

14

```
(OR\Boolean\Interface
  (LAMBDA (b1 b2 TooManyArguments)
    (if b1:1='ExpressionWithType and b2:3=Boolean and b2:1='ExpressionWithType and b2:3=Boolean
          and TooManyArguments=NIL and (EQUAL b1:3 b2:3) and (Report OR\Boolean\Interface 1 interface)
        then (ExpressionWithType <'OR\Boolean b1:2 b2:2> b2:3)
      elseif NIL))
```

15

```
(AND\Boolean\Interface
  (LAMBDA (b1 b2 TooManyArguments)
    (if b1:1='ExpressionWithType and b2:3=Boolean and b2:1='ExpressionWithType and b2:3=Boolean
          and TooManyArguments=NIL and (EQUAL b1:3 b2:3) and (Report AND\Boolean\Interface 1 interface)
        then (ExpressionWithType <'AND\Boolean b1:2 b2:2> b2:3)
      elseif NIL))
```

16

```
(IH\Boolean\Interface
  (LAMBDA (a1 int TooManyArguments)
    (if a1:1='ExpressionWithType and int:1='ExpressionWithType and TooManyArguments=NIL
          and (Report IH\Boolean\Interface 1 interface)
        then (ExpressionWithType <'IH\Boolean a1:2 int:2> Boolean)
      elseif NIL))
)

(RPAQQ BooleanConstants (Boolean))

(RPAQQ Boolean Boolean)

(RPAQQ Boolean\InterfaceConstants NIL)
  (PUTPROPS Boolean DeclaredType Boolean
                    LocalDeclarations ((a1\Interface ExpressionWithType a1\Boolean any)
                                       (b1\Interface ExpressionWithType b1\Boolean Boolean)
                                       (b2\Interface ExpressionWithType b2\Boolean Boolean)
                                       (b3\Interface ExpressionWithType b3\Boolean Boolean)
                                       (int\Interface ExpressionWithType int\Boolean Integer))
                    Infix NIL
                    Needs NIL
                    EqualOp EQV\Boolean
                    IsConstant T)

(RPAQQ BooleanFNS (SOME\Boolean ALL\Boolean EQV\Boolean NOT\Boolean IMP\Boolean OR\Boolean AND\Boolean
                   IH\Boolean))

(PUTPROPS EQV\Boolean PrimaryLHSides (1 (1 EQV\Boolean b2\Boolean b1\Boolean)))

(PUTPROPS NOT\Boolean PrimaryLHSides (1 (1 NOT\Boolean b1\Boolean)))

(PUTPROPS IMP\Boolean PrimaryLHSides (1 (1 IMP\Boolean b1\Boolean b2\Boolean)))

(PUTPROPS OR\Boolean PrimaryLHSides (1 (1 OR\Boolean b1\Boolean b2\Boolean)))

(PUTPROPS AND\Boolean PrimaryLHSides (1 (1 AND\Boolean b1\Boolean b2\Boolean)))
```

```
(PUTPROPS IH\Boolean PrimaryLHSides (1 (1 IH\Boolean a1\Boolean int\Boolean)))

(PUTPROPS SOME\Boolean EqualOp EQV\Boolean)

(PUTPROPS ALL\Boolean EqualOp EQV\Boolean)

(PUTPROPS EQV\Boolean EqualOp EQV\Boolean)

(PUTPROPS NOT\Boolean EqualOp EQV\Boolean)

(PUTPROPS IMP\Boolean EqualOp EQV\Boolean)

(PUTPROPS OR\Boolean EqualOp EQV\Boolean)

(PUTPROPS AND\Boolean EqualOp EQV\Boolean)

(PUTPROPS IH\Boolean EqualOp EQV\Boolean)

(PUTPROPS EQV\Boolean EQOP T)

(RPAQQ Boolean\InterfaceFNS (SOME\Boolean\Interface ALL\Boolean\Interface EQV\Boolean\Interface
                             NOT\Boolean\Interface IMP\Boolean\Interface
                             OR\Boolean\Interface AND\Boolean\Interface
                             IH\Boolean\Interface))

(PUTPROPS SOME\Boolean\Interface PrimaryLHSides (1 (1 SOME\Boolean\Interface (ExpressionWithType a1\Boolean
                                                                                                  any)
                                                      (ExpressionWithType b1\Boolean Boolean)
                                                      NIL)))

(PUTPROPS ALL\Boolean\Interface PrimaryLHSides (1 (1 ALL\Boolean\Interface (ExpressionWithType a1\Boolean any)
                                                     (ExpressionWithType b1\Boolean Boolean)
                                                     NIL)))

(PUTPROPS EQV\Boolean\Interface PrimaryLHSides (1 (1 EQV\Boolean\Interface (ExpressionWithType b1\Boolean
                                                                                                Boolean)
                                                     (ExpressionWithType b2\Boolean Boolean)
                                                     NIL)))

(PUTPROPS NOT\Boolean\Interface PrimaryLHSides (1 (1 NOT\Boolean\Interface (ExpressionWithType b1\Boolean
                                                                                                Boolean)
                                                     NIL)))

(PUTPROPS IMP\Boolean\Interface PrimaryLHSides (1 (1 IMP\Boolean\Interface (ExpressionWithType b1\Boolean
                                                                                                Boolean)
                                                     (ExpressionWithType b2\Boolean Boolean)
                                                     NIL)))

(PUTPROPS OR\Boolean\Interface PrimaryLHSides (1 (1 OR\Boolean\Interface (ExpressionWithType b1\Boolean
                                                                                             Boolean)
                                                    (ExpressionWithType b2\Boolean Boolean)
                                                    NIL)))

(PUTPROPS AND\Boolean\Interface PrimaryLHSides (1 (1 AND\Boolean\Interface (ExpressionWithType b1\Boolean
                                                                                                Boolean)
                                                     (ExpressionWithType b2\Boolean Boolean)
                                                     NIL)))

(PUTPROPS IH\Boolean\Interface PrimaryLHSides (1 (1 IH\Boolean\Interface (ExpressionWithType a1\Boolean any)
                                                    (ExpressionWithType int\Boolean Integer)
                                                    NIL)))
(InitializeLoad TYPE Boolean 110 ((NoteInterfaces Boolean\InterfaceFNS)
              (InitInfix (QUOTE Boolean))
              (InitNeeds (QUOTE Boolean))
              (NoteDeclarations (QUOTE Boolean))
              (NoteLeftHandSides BooleanFNS)))
(DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML ALL\Boolean SOME\Boolean)
(ADDTOVAR LAMA )
)
(DECLARE: DONTCOPY
  (FILEMAP (NIL (1280 5709 (SOME\Boolean 1292 . 2454) (ALL\Boolean 2458 . 3669) (EQV\Boolean 3673 . 3909) (
NOT\Boolean 3913 . 4083) (IMP\Boolean 4087 . 4295) (OR\Boolean 4299 . 4499) (AND\Boolean 4503 . 4712) (
IH\Boolean 4716 . 5706)) (5962 8986 (SOME\Boolean\Interface 5974 . 6329) (ALL\Boolean\Interface 6333 . 6685) (
EQV\Boolean\Interface 6689 . 7098) (NOT\Boolean\Interface 7102 . 7411) (IMP\Boolean\Interface 7415 . 7824) (
OR\Boolean\Interface 7828 . 8234) (AND\Boolean\Interface 8238 . 8647) (IH\Boolean\Interface 8651 . 8983)))))
STOP
```