

File created: 6-Feb-87 10:18:07 {DSK}<LISPFIL2>H.ALFA>BLOCKS.HKB;2

changes to: (VARS *functions1* *predicates1* *variables1* *temp-foo* *temp-pred*)

previous date: 3-Nov-86 11:06:40 {DSK}<LISPFIL2>H>BLOCKS.HKB;9

Read Table: OLD-INTERLISP-FILE

Package: INTERLISP

Format: XCCS

(RPAQQ **BLOCKSCOMS** (*functions1* *predicates1* *variables1* *temp-foo* *temp-pred*))

(RPAQQ ***functions1*** (HRPRINT LISTMEMB MEMBER fail true noteq))

(RPAQQ ***predicates1*** (color-of showworld SPLIT putdown pickup please ART PREP GoOnNp PARTIC OPTPARTIC VP VERB NP1
NP NOM BLOCK on clear puton))

(RPAQQ ***variables1***

(:d :c :color :bl :a4 :a3 :a2 :a1 :e :q :s :m :string :e4 :e3 :e2 :e1 :rest1 :block2 :q4 :q3 :q2 :q1
:block1 :rest :block :oper :vf :b :a :w2 :w1 :x1 :y1 :u :r :v :z :i :h :j :l :w :y :x :k :p))

(RPAQQ ***temp-foo***

```
[[LAMBDA (y)
  (PRINTOUT T y T)
[LAMBDA (x y)
  (PROG ((temp x)
    loop
      (COND ((NULL temp)
        (RETURN T))
      (T (COND ([OR (MEMBER (CAR temp)
        y)
        (EQ (CAR temp (QUOTE one]
        (SETQ temp (CDR temp))
        (GO loop))
        (T (RETURN NIL]
[LAMBDA (x y)
  (MEMBER x y]
[LAMBDA NIL NIL]
[LAMBDA NIL T]
[LAMBDA (x y)
  (NOT (EQ x y])
```

(RPAQQ ***temp-pred***

```
[(((color-of :block :color)
  <
  (BLOCK :block :color :a :b :c :d)))
((showworld)
  <
  (on :x :y)
  (HRPRINT (on :x :y))
  (fail)))
((SPLIT (:a . :b)
  :a :b))
((putdown :x)
  <
  (puton :x table))
((pickup :x)
  <
  (puton :x hand))
((please :string)
  <
  (VP :string))
((ART the)
  (ART a)
  (ART an))
((PREP on on))
((GoOnNp (:x . :y)
  :v :rest)
  <
  (PREP :x :x1)
  (NP :y :v :rest))
((PARTIC down)
  (PARTIC up)
  (PARTIC to))
((OPTPARTIC NIL :x)
  (OPTPARTIC (:x . :y)
  :z)
  <
  (PARTIC :x))
((VP (:x :y . :z)
  <
  (VERB :x :vf :oper)
  (PARTIC :y)
  (MEMBER :y :vf)
  (NP :z :block NIL)
  (:oper :block))
```

```

((VP (:x . :y))
 <
 (VERB :x :vf :oper)
 (MEMBER one :vf)
 (NP :y :block :rest)
 (OPTPARTIC :rest :vf)
 (:oper :block))
((VP (:x . :y))
 <
 (VERB :x :vf :oper)
 (MEMBER two :vf)
 (NP :y :block1 :rest)
 (BLOCK :block1 :q1 :q2 :q3 :q4 stackable)
 (GoOnNp :rest :block2 :rest1)
 (BLOCK :block2 :e1 :e2 :e3 supportive :e4)
 (:oper :block1 :block2)))
((VERB pickup (one)
 pickup))
((VERB pick (up one)
 pickup))
((VERB put (two)
 puton))
((VERB stack (two)
 puton))
((VERB put (down one)
 putdown))
((NP1 (:x :y . :z)
 :w :u :r)
 <
 (PREP :y :y1)
 (NOM :x :x1)
 (NP :z :v :r)
 (:y1 :w :v)
 (BLOCK . :w1)
 (LISTMEMB (:x1 . :u)
 :w1)
 (SPLIT :w1 :w :w2))
((NP1 (:x . :y)
 :v :u :r)
 <
 (NOM :x :x1)
 (NP1 :y :v (:x1 . :u)
 :r))
((NP1 (:x . :y)
 :w :u :y)
 <
 (NOM :x :x1)
 (BLOCK . :w1)
 (LISTMEMB (:x1 . :u)
 :w1)
 (SPLIT :w1 :w :w2)))
((NP (:x . :y)
 :v :r)
 <
 (ART :x)
 (NP1 :y :v NIL :r))
((NP :x :v :r)
 <
 (NP1 :x :v NIL :r)))
((NOM red red))
((NOM block cube))
((NOM cube cube))
((NOM cube1 cube1))
((NOM cube2 cube2))
((NOM cube3 cube3))
((NOM big large))
((NOM small small))
((NOM blue blue))
((NOM white white))
((NOM green green))
((NOM pyramid1 pyramid))
((NOM pyramid pyramid))
((NOM sphere sphere)))
((BLOCK pyramid1 white pyramid 3 NIL stackable))
((BLOCK cube2 blue cube 5 supportive stackable))
((BLOCK cube3 green cube 1 supportive stackable))
((BLOCK cube1 red cube 10 supportive stackable))
((BLOCK sphere black sphere 3 NIL stackable))
((BLOCK table NIL NIL NIL supportive NIL))
((BLOCK hand NIL NIL NIL supportive NIL))
((on cube3 hand))
((on sphere table))
((on cube1 table))
((on cube2 table))
((on pyramid1 table)))
((clear table))
((clear :x)

```

```
<
(on :y :x)
(puton :y table)
((clear :x))
((puton :x :y)
<
(noteq :x table)
(clear :x)
(noteq :y pyramid)
(noteq :y sphere)
(clear :y)
(on :x :w)
(delete (on :x :w))
(assert (on :x :y])
```

VARIABLE INDEX

functions1 ..1 *predicates1* .1 *temp-foo*1 *temp-pred* ...1 *variables1* ..1 BLOCKSCOMS1
