```
:: On the Homotopy Types of Some Decomposition Spaces
::
::                    by K. Borsuk
::
:: Bull. de L'Acad. Pol., ser. math., astr. et phys., XVIII, 5, 1970
::   translated into MIZAR-4S by A. Trybulec

environ
             ::::::::::::::::::::::::::::
             ::   S e t   T h e o r y   ::
             ::::::::::::::::::::::::::::

mode Function -> set;

definition let x be Any, X be set; pred x in X; end;
definition let X be set;
 mode Subset of X -> set of Element of X;
 let Y be set; pred X c= Y;
end;
definition let x,y be Any; func <x,y> ->Any; end;
                  :: the ordered pair of x and y
definition let F be Function, x be Any;
 func F.x ->Any;
 func F"x ->set;
end;
definition let F be Function, X be set;
 func F.X -> set;
 func F"X->set;
end;
definition let F,G be Function;
 func G.F -> Function;
end;
definition let X,Y be set;
 mode Function of X,Y -> Function;
end;
definition
 let X,Y be set, F be (Function of X,Y), x be Element of X;
 redefine F.x as Element of Y;:: the image of the point x
end;
definition
 let X,Y be set, F be (Function of X,Y), y be Element of Y;
 redefine F"y as Subset of X; :: the counter-image of the point x
end;
definition
 let X,Y,Z be set,F be (Function of X,Y), G be Function of Y,Z;
 redefine G.F as Function of X,Z;
end;
definition
  let X,Y be set, F be (Function of X,Y), Z be Subset of X;
  redefine F.Z as Subset of Y; :: the image of the subset Z
end;
definition
 let X,Y be set, F be (Function of X,Y), Z be Subset of Y;
  redefine F"Z as Subset of X; :: the counter-image of the subset Z
end;
```

```
reserve X,Y,Z for set;
TM1: for X,Y,Z for x being (Element of X),
      F being (Function of X,Y), G being Function of Y,Z
      holds G.(F.x)=(G.F).x;
TM2: for X,Y,Z for A being (Subset of X),
      F being (Function of X,Y), G being Function of Y,Z
      holds G.(F.A)=(G.F).A;
TM3:
 for X,Y for A being (Subset of Y), F being (Function of X,Y)
      holds F.(F"A) c= A;
TM4:
 for X,Y,Z for F being (Function of X,Y), G being Function of X,Z
      st for x,x' being Element of X st F.x=F.x' holds G.x=G.x'
     ex H being Function of Y,Z st H.F=G;


TM5: for X,Y,Z st X c= Y & Y c= Z holds X c= Z;
TM6: for X,Y for A,B being (Subset of X),
      F being Function of X,Y holds
      A c= B implies F.A c= F.B;
TM7: for X,Y,Z for y being (Element of Y),
      F being (Function of X,Y), G being Function of Y,Z
      holds F"y c= (G.F)"(G.y);

definition let X,Y be set;
 func X#Y ->set;  :: the Cartesian product of X and Y
end;
definition let X be set, x be Any;
 func X#x -> set;
        :: the Cartesian product of X and the singleton of x
end;
definition
 let X,Y be set, A be (Subset of X), B be Subset of Y;
 redefine A#B as Subset of X#Y;
end;

TM8: for X,Y,Z st X c= Y holds X#Z c= Y#Z;

definition
 let X,Y be set, A be (Subset of X), y be Element of Y;
 redefine A#y as Subset of X#Y;
end;
definition
 let X,Y be set, x be (Element of X), y be Element of Y;
 redefine <x,y> as Element of X#Y;
end;

priority 15:;
priority 14: sp proj;
priority 13: the_carrier_of;
priority 12: Id OMEGA;

definition let X be set;
 func Id X -> Function of X,X;
```

```
end;
definition let X1,X2,Y1,Y2 be set,
 F be (Function of X1,X2), G be Function of Y1,Y2;
 func F#G ->Function of X1#Y1,X2#Y2;
end;
TM11: for X,Y,Z for F being (Function of X,Y),
          x being (Element of X), z being Element of Z
      holds (F#Id Z).<x,z> = <F.x,z>;
TM12: for X,Y,Z for F being (Function of X,Y),
          y being (Element of Y), z being Element of Z
      holds (F#Id Z)"<y,z> = (F"y)#z;
TM13: for X,Y,Z for F being (Function of X,Y),
          A being (Subset of X), B being Subset of Z
      holds (F#Id Z).(A#B) = (F.A)#B;
TM0:
 for x,x', y,y' being Any st <x,y> =<x',y'> holds x=x' & y=y';

              ::::::::::::::::::::::::::::::
              ::     T o p o l o g y      ::
              ::::::::::::::::::::::::::::::

 mode SPACE; ::  a metrisable topological space
 reserve X,Y for SPACE;
  definition let X be SPACE;
    pred X is_compact;
    func the_carrier_of X -> set;
    mode SUBSPACE of X -> SPACE; :: a closed subspace of X
    mode Point of X -> Element of the_carrier_of X;
  end;
  definition let X,Y;
    func X#Y ->SPACE;
  end;
definition
 let X,Y be SPACE, A be (Subset of the_carrier_of X),
     y be Element of the_carrier_of Y;
 redefine A#y as Subset of the_carrier_of (X#Y);
end;
definition
  let X,Y be SPACE, A be (Subset of the_carrier_of X),
                    B be Subset of the_carrier_of Y;
 redefine A#B as Subset of the_carrier_of (X#Y);
end;
  definition let A be SPACE, B be SUBSPACE of A;
   redefine the_carrier_of B as Subset of the_carrier_of A;
  end;
  definition let X be SPACE, A be Subset of the_carrier_of X;
    pred A is_compact;
  end;
  definition let X be SPACE, W be Point of X;
    mode Neighborhood of W -> Subset of the_carrier_of X;
  end;
 definition
   let X,Y be SPACE,
       F be Function of the_carrier_of X, the_carrier_of Y;
   let x be Point of X;
```

```
   redefine F.x as Point of Y;
 end;
 definition
   let X,Y be SPACE,
       F be Function of the_carrier_of X, the_carrier_of Y;
    pred F is_continuous means
      for W being (Point of X), G being Neighborhood of F.W
           ex H being Neighborhood of W st F.H c= G;
 end;
definition let X,Y;
 mode MAP of X,Y
           -> Function of the_carrier_of X, the_carrier_of Y
  means
TP2: it is_continuous;
end;
NT5: for X,Y holds
     the_carrier_of (X#Y) = the_carrier_of X # the_carrier_of Y;
TP3: for X st X is_compact for Y being SUBSPACE of X holds
       Y is_compact;
        :: because all subspaces are supposed to be closed
definition
 let X,Y,Z be SPACE, F be (MAP of X,Y), G be MAP of Y,Z;
  redefine G.F as MAP of X,Z;
end;
  definition let X be SPACE,Y be Subset of the_carrier_of X;
    mode Neighborhood of Y -> Subset of the_carrier_of X;
  end;
definition
 let X,Y be SPACE,
   x be (Element of the_carrier_of X),
   y be Element of the_carrier_of Y;
 redefine <x,y> as Element of the_carrier_of (X # Y);
end;
definition
 let X,Y be SPACE,
   x be (Point of X), y be (Point of Y);
 redefine <x,y> as Point of X # Y;
end;
definition
 let X,Y be SPACE,
   x be (Point of X), y be (Point of Y),
    U be (Neighborhood of x), V be (Neighborhood of y);
 redefine U#V as Neighborhood of <x,y>;
end;
 definition let X,Y be SPACE, W be (Point of Y),
                  A be (MAP of X,Y), G be Neighborhood of W;
  redefine A"G as Neighborhood of A"W;
 end;
  FC3: for X for A,B being (Subset of the_carrier_of X),
           U being Neighborhood of B st A c= B holds
         U is Neighborhood of A;
  TC1: for X,Y for XT being Point of X#Y
         ex W being (Point of X), T being Point of Y st XT=<W,T>;
  TC1': for X,Y for XT being Element of the_carrier_of (X#Y)
         ex W being (Element of the_carrier_of X),
```

```
          T being Element of the_carrier_of Y st XT=<W,T>;
 definition let X,Y be SPACE,
          W be (Subset of the_carrier_of X),
          T be (Point of Y),
          U be (Neighborhood of W), V being Neighborhood of T ;
     redefine U#V as Neighborhood of W#T;
 end;
 TC5: for X,Y for T being (Point of Y),
          A being (Subset of the_carrier_of X),
          G being Neighborhood of A#T
         st  A is_compact
    ex U being (Neighborhood of A) ,V being Neighborhood of T st
           U#V c= G;
 definition let X,Y being SPACE;
  mode Homeomorphism of X,Y -> MAP of X,Y;
  pred  X,Y are_homeomorphic;
 end;

                :: D e c o m p o s i t i o n s ::

definition let X be SPACE;
   mode DECOMPOSITION of X;
          :: upper semicontinuous decomposition into compacta
end;
 definition let X be SPACE,D be DECOMPOSITION of X;
   func sp D ->SPACE;
   func proj D ->MAP of X,sp D;
 end;
 definition
   let XX be SPACE,
    X be (SUBSPACE of XX),D be DECOMPOSITION of X ;
   func ^D -> DECOMPOSITION of XX;
   :: ^D trivial extension (i.e. by singletons) of D onto XX
 end;
definition let X be SPACE, Y be (SUBSPACE of X),
             D be DECOMPOSITION of Y;
 redefine sp D as SUBSPACE of sp^D;
end;
 DC1: for X for D being (DECOMPOSITION of X),
           W being Element of the_carrier_of sp(D)
         holds (proj D)"W is_compact;
 DC2: for X for D being (DECOMPOSITION of X),
          W being (Point of sp D),
          G being Neighborhood of proj D " W
        ex U being Neighborhood of proj D " W st
          U c= G & proj(D).U is Neighborhood of W;
 DC3: for X for D being DECOMPOSITION of X
        for W being Element of the_carrier_of sp(D)
         ex W' being Element of the_carrier_of X st proj(D).W'=W;
 DC4: for XX being SPACE , X being (SUBSPACE of XX)
        for D being (DECOMPOSITION of X)
         for W being Element of the_carrier_of XX holds
      proj^D.W in the_carrier_of sp D iff W in the_carrier_of X;
 DC5: for XX being SPACE,
         X being (SUBSPACE of XX), D being (DECOMPOSITION of X),
```

```
           W,W' being Element of the_carrier_of XX
           st not W in the_carrier_of X & proj(^D).W=proj(^D).W'
        holds W=W';
  DC6: for XX being SPACE, X being (SUBSPACE of XX),
           D being (DECOMPOSITION of X),
           W being Element of the_carrier_of XX
           st W in the_carrier_of X
        holds proj(^D).W=proj(D).W;

definition let X,Y be SPACE, h be (Homeomorphism of X,Y),
   D be (DECOMPOSITION of X),
   D' be DECOMPOSITION of Y;
 pred h preserves D,D';
end;
DC7: for X,Y for X' being (SUBSPACE of X),
                 Y' being (SUBSPACE of Y),
        DX being (DECOMPOSITION of X'),
        DY being (DECOMPOSITION of Y'),
        h  being (Homeomorphism of X,Y),
        h' being Homeomorphism of X',Y' st h' c= h holds
     h' preserves DX,DY implies h preserves ^DX,^DY;
DC8: for X,Y for h being (Homeomorphism of X,Y),
      D being (DECOMPOSITION of X),
      D' being DECOMPOSITION of Y st h preserves D,D'
    holds sp D, sp D' are_homeomorphic;

definition let X be SPACE, Y be (SUBSPACE of X),
             D be DECOMPOSITION of X;
 func D/Y -> DECOMPOSITION of Y;
end;


             :: Omega and Similarity  ::

  definition let X be SPACE, D be DECOMPOSITION of X;
    func OMEGA D  -> SUBSPACE of X;
      :: the closure of the union of classes of D
      :: consisting of at least two points
  end;
definition
 let X,X' be SPACE,
    D be (DECOMPOSITION of X), D' be DECOMPOSITION of X';
  pred  D,D' are_similar means
    ex h being Homeomorphism of OMEGA D, OMEGA D' st
     h preserves D/OMEGA D, D'/OMEGA D';
end;

  OM1: for X for Y being (SUBSPACE of X),
          D,D' being DECOMPOSITION of Y
           st D,D' are_similar
        holds ^D,^D' are_similar;
  OM2: for X being SPACE, D being DECOMPOSITION of X holds
           (D/OMEGA D) = D;
  OM3: for X for Y being (SUBSPACE of X),
           D being DECOMPOSITION of Y
```

```
         holds OMEGA ^D = OMEGA D;

               :: Theory of Retracts ::

const I->SPACE; :: the unit segment of real numbers
const IO->Element of the_carrier_of I; :: the number 0
const I1->Element of the_carrier_of I; :: the number 1

definition let A be SPACE, B be (SUBSPACE of A), F be MAP of A,B;
    pred F is_a_retraction means
RT1:
 for W being Element of the_carrier_of A st W in the_carrier_of B
         holds F.W=W;
end;
definition let X be SPACE,Y be SUBSPACE of X;
  pred Y is_a_retract_of X means
   ex F being MAP of X,Y st F is_a_retraction;
  pred  Y is_an_SDR_of X means
   ex H being MAP of X#I,X st
    for A being Element of the_carrier_of X holds
     H.<A,IO> =A & H.<A,I1> in the_carrier_of Y &
     (A in the_carrier_of Y implies
        for T being Element of the_carrier_of I holds H.<A,T> =A);
end;

                :: Homotopy Theory ::

  definition let X,Y being SPACE;
   pred X,Y are_homotopically_equivalent;
   pred X,Y have_the_same_shape;
  end;
  HT1: for A,B being SPACE st
           A,B are_homeomorphic
        holds A,B are_homotopically_equivalent;
  HT2: for X,Y st X,Y are_homotopically_equivalent
        holds Y,X are_homotopically_equivalent;
  HT3: for X,Y,Z being SPACE
        st X,Y are_homotopically_equivalent
           & Y,Z are_homotopically_equivalent
        holds X,Z are_homotopically_equivalent;
  HT4: for X,Y st  X,Y are_homotopically_equivalent
       holds X,Y have_the_same_shape;
  HT5: for Y for X being SUBSPACE of Y holds
     X is_an_SDR_of Y implies X,Y are_homotopically_equivalent;

               :: Euclidean Spaces ::

const the_Hilbert_space->SPACE;
mode Euclidean_ball -> SPACE;

 definition let N be Nat;
  func E N -> SUBSPACE of the_Hilbert_space;
          :: the N-dimensional Euclidean space
  func Q N -> SUBSPACE of E N;
          :: the N-dimensional Euclidean ball
```

```
 end;
  reserve N for Nat;
  EU1: for N holds Q N is_compact;
  EU2: for N holds E N is_an_SDR_of the_Hilbert_space;
  EU3: for N holds Q N is_an_SDR_of E N;
  EU5: for X being Euclidean_ball ex N st X=Q(N);

Victor_Klee:       ::  Some topological properties of convex sets,
                   ::  Trans. Amer. Math. Soc. 78(1958), PP.30-45

for X,Y being SUBSPACE of the_Hilbert_space st X is_compact
 for h being Homeomorphism of X,Y
 ex h' being Homeomorphism of the_Hilbert_space,the_Hilbert_space
  st h c= h';
begin
T23: for XX being SPACE, X being (SUBSPACE of XX),
      D being DECOMPOSITION of X holds
      (X is_a_retract_of XX implies sp(D) is_a_retract_of sp(^D))
      & (X is_an_SDR_of XX implies sp(D) is_an_SDR_of sp(^D))
 proof
    let XX be SPACE, X be (SUBSPACE of XX),
      D be DECOMPOSITION of X;
    thus X is_a_retract_of XX implies sp(D) is_a_retract_of sp(^D)
      proof given R being MAP of XX,X such that
A:        R is_a_retraction;
       now
        given xx,xx' being Element of the_carrier_of XX such that
C1:       proj^D.xx=proj^D.xx' and
C2:       (proj D.R).xx<>(proj D.R).xx';
          xx<>xx' by C2; then
C3:       xx in the_carrier_of X & xx' in the_carrier_of X
                                                       by DC5,C1;
          then R.xx=xx & R.xx'=xx' by A,RT1; then
C6:       proj D.xx = (proj D.R).xx & proj D.xx'= (proj D.R).xx'
                                                       by TM1;
          proj^D.xx=proj D.xx & proj^D.xx'=proj D.xx' by DC6,C3;
         hence contradiction by C1,C2,C6;
       end;
     then
     consider F
      being Function of the_carrier_of sp^D, the_carrier_of sp D
     such that
         K': F.(proj^D)=(proj D).R by TM4;
     F is_continuous
      proof let W be Point of sp(^D);
       let G be Neighborhood of F.W;
        proj^D"W c= (proj D.R)"(F.W) by TM7,K';
        then (proj D.R)"G is Neighborhood of proj^D"W by FC3;
        then reconsider GG=(proj D.R)"G
                                    as Neighborhood of proj^D"W;
        proj^D"W c= (proj D.R)"(F.W) by K',TM7;
        then consider U being Neighborhood of proj^D"W such that
U1:      U c= GG and
U2:      proj^D.U is Neighborhood of W by DC2;
        reconsider V'=proj^D.U as Neighborhood of W by U2;
```

```
       take V=V';
        F.V=(proj D.R).U by K',TM2,U2;
        then F.V c= (proj D.R).GG & (proj D.R).GG c= G
                                                        by TM3,TM6,U1;
       hence F.V c= G by TM5;
      end;
        then reconsider F'=F as MAP of sp^D,sp D by TP2;
      take F''=F';
      let W be Element of the_carrier_of sp^D;
       consider W' being Element of the_carrier_of XX such that
W2:     proj^D.W'=W by DC3;
      assume W in the_carrier_of sp D;
       then W' in the_carrier_of X by DC4,W2; then
       W'=R.W' & proj D.W'=proj^D.W' by A,RT1,DC6;
       then F'.W = (F.proj^D).W' & W = (proj D.R).W' by TM1,W2;
      hence F''.W=W by K';
     end;
 the_carrier_of (XX#I) = the_carrier_of XX # the_carrier_of I &
 the_carrier_of (sp^D#I) = the_carrier_of sp^D # the_carrier_of I
                                                                 by NT5;
 then reconsider II=proj^D # Id the_carrier_of I as
  Function of the_carrier_of (XX#I), the_carrier_of (sp^D#I);
    given CH1 being MAP of XX#I,XX such that
W2:    for A being Element of the_carrier_of XX holds
        CH1.<A,I0> =A & CH1.<A,I1> in the_carrier_of X &
        (A in the_carrier_of X implies
    for T being Element of the_carrier_of I holds CH1.<A,T> =A);
       now given xx,xx' being Element of the_carrier_of (XX#I)
         such that
C1:        II.xx=II.xx'  and
C2:        (proj^D.CH1).xx<>(proj^D.CH1).xx';
           consider x being (Element of the_carrier_of XX),
                    t being Element of the_carrier_of I such that
C3:         xx=<x,t> by TC1';
           consider x' being (Element of the_carrier_of XX),
                    t' being Element of the_carrier_of I such that
C4:         xx'=<x',t'> by TC1';
           II.xx=<proj^D.x,t> & II.xx'=<proj^D.x',t'>
                                                  by TM11,C3,C4; then
C5:        t=t' & proj^D.x=proj^D.x' by TM0,C1,C3,C4; then
           x<>x' by C2,C3,C4; then
           x in the_carrier_of X & x' in the_carrier_of X
                                                  by DC5,C5; then
C6:        CH1.xx=x & CH1.xx'=x' by W2,C3,C4;
           (proj^D.CH1).xx = proj^D.(CH1.xx) &
           (proj^D.CH1).xx' = proj^D.(CH1.xx') by TM1;
         hence contradiction by C2,C6,C5;
        end;
         then consider THETA being
          Function of the_carrier_of(sp^D#I), the_carrier_of sp^D
         such that T'': proj^D.CH1 = THETA.II by TM4;
 THETA is_continuous
  proof let WT be Point of sp^D # I;
   the_carrier_of sp^D # the_carrier_of I
                      = the_carrier_of (sp^D # I) by NT5; then
```

```
    reconsider xt'=WT
     as Element of the_carrier_of sp^D # the_carrier_of I;
   let G be Neighborhood of THETA.WT;
    consider W being (Point of sp(^D)), T being Point of I
     such that T3: WT=<W,T> by TC1;
    II"xt' = (proj^D"W)#T by TM12,T3;
    then (proj^D"W)#T c= (proj^D.CH1)"(THETA.WT) by T'',TM7;
    then reconsider GG=(proj^D.CH1)"G
                      as Neighborhood of (proj^D"W)#T by FC3;
    proj^D"W is_compact by DC1;
    then consider U  being (Neighborhood of proj^D"W),
                   V being Neighborhood of T such
          that U3: U#V c= GG by TC5;
    consider U' being Neighborhood of proj^D"W such
          that U1': U' c= U
           and U2': proj(^D).U' is Neighborhood of W by DC2;
    reconsider H'=proj(^D).U' as Neighborhood of W by U2';
    reconsider H''=H'#V as Neighborhood of WT by T3;
   take H=H'';
T9:  (proj^D.CH1).GG c= G by TM3;
    II.(U'#V)=(proj^D.U')#V by TM13; then
S9:  THETA.H=(proj^D.CH1).(U'#V) by TM2,T'';
    U'#V c= U#V by TM8,U1';
    then T10: THETA.H c= (proj^D.CH1).(U#V) by TM6,S9;
    (proj^D.CH1).(U#V) c= (proj^D.CH1).GG by TM6,U3;
    then (proj^D.CH1).(U#V) c= G by TM5,T9;
   hence THETA.H c= G by TM5,T10;
  end;
  then reconsider THETA'=THETA as MAP of sp^D#I,sp^D by TP2;
 take THETA''=THETA';
 let W be Element of the_carrier_of sp(^D);
  consider W' being Element of the_carrier_of XX such that
K11: proj(^D).W'=W by DC3;
  CH1.<W',I0> =W' & II.<W',I0> = <W,I0> by W2,TM11,K11;
  then (THETA'.II).<W',I0> = THETA'.<W,I0> &
       (THETA'.II).<W',I0> = W by T'',TM1,K11;
 hence THETA''.<W,I0> =W;
K33: II.<W',I1> =<W,I1>  by TM11,K11; then
K34: (THETA'.II).<W',I1> = THETA'.<W,I1> &
  (THETA'.II).<W',I1> = proj^D.(CH1.<W',I1>) by T'',TM1,K11;
  CH1.<W',I1> in the_carrier_of X by W2;
 hence THETA''.<W,I1> in the_carrier_of sp(D) by DC4,K34,K33;
 assume
ZAL: W in the_carrier_of sp(D);
 let T be Element of the_carrier_of I;
  W' in the_carrier_of X by DC4,K11,ZAL;
  then CH1.<W',T> =W' & II.<W',T> = <W,T> by W2,TM11,K11;
  then (THETA'.II).<W',T> = THETA'.<W,T> &
       (THETA'.II).<W',T> = W by T'',TM1,K11;
 hence THETA''.<W,T> =W;
end;
 EU4: for D,D' being DECOMPOSITION of the_Hilbert_space
       st D,D' are_similar
        & OMEGA(D) is_compact
       holds sp D, sp D' are_homeomorphic
```

```
 proof let D,D' be DECOMPOSITION of the_Hilbert_space;
  given h being Homeomorphism of OMEGA D, OMEGA D' such that
Z1:  h preserves D/OMEGA D, D'/OMEGA D';
  assume OMEGA D is_compact; then consider h' being
   Homeomorphism of the_Hilbert_space, the_Hilbert_space
    such that
Z2:  h c= h' by Victor_Klee;
   (D/OMEGA D) = D &(D'/OMEGA D') = D' by OM2;
  then h' preserves D,D' by DC7,Z1,Z2;
 hence thesis by DC8;
end;
Aux: now let A,B,C,D be SPACE;
      assume A, C are_homotopically_equivalent &
             C, D are_homotopically_equivalent;
       then AD:A, D are_homotopically_equivalent by HT3;
      assume B, D are_homotopically_equivalent;
       then D, B are_homotopically_equivalent by HT2;
      hence A, B are_homotopically_equivalent by HT3,AD;
    end;
  C31: for N being Nat,
          D,D' being DECOMPOSITION of E(N)
          st D,D' are_similar&
            OMEGA(D) is_compact
          holds sp D, sp D' are_homotopically_equivalent
    proof let N be Nat;
      let D,D' be DECOMPOSITION of E(N);
      E(N) is_an_SDR_of the_Hilbert_space by EU2;
      then sp(D) is_an_SDR_of sp(^D)
        & sp(D') is_an_SDR_of sp(^D') by T23;
      then W: sp D, sp^D are_homotopically_equivalent
          & sp D', sp^D' are_homotopically_equivalent by HT5;
      assume D,D' are_similar ; then
      U: ^D,^D' are_similar by OM1;
      U': OMEGA D = OMEGA ^D by OM3;
      assume OMEGA(D) is_compact;
      then sp^D, sp^D' are_homeomorphic by U,U',EU4;
       then sp^D, sp^D' are_homotopically_equivalent by HT1;
      hence thesis by W,Aux;
    end;
  C32: for N being Nat,
         D,D' being DECOMPOSITION of Q(N)
           st D,D' are_similar
         holds sp D, sp D' are_homotopically_equivalent
    proof let N be Nat;
      let D,D' be DECOMPOSITION of Q(N);
      Q(N) is_an_SDR_of E(N) by EU3;
      then sp D is_an_SDR_of sp^D &
        sp D' is_an_SDR_of sp^D' by T23;
      then W: sp D, sp^D are_homotopically_equivalent &
               sp D', sp^D' are_homotopically_equivalent by HT5;
     assume D,D' are_similar; then
      U: ^D,^D' are_similar by OM1;
      Q(N) is_compact by EU1;
      then OMEGA D is_compact & OMEGA D = OMEGA ^D by OM3,TP3;
      then sp^D, sp^D' are_homotopically_equivalent by C31,U;
```

```
    hence thesis by Aux,W;
   end;
 C33: for EB being Euclidean_ball,
       D,D' being DECOMPOSITION of EB
        st D,D' are_similar
       holds sp D, sp D' have_the_same_shape
   proof
    let EB be Euclidean_ball, D,D' be DECOMPOSITION of EB;
     consider N being Nat such
       that W: EB=Q(N) by EU5;
     reconsider DD=D as DECOMPOSITION of Q N by W;
     reconsider DD'=D' as DECOMPOSITION of Q N by W;
    assume D,D' are_similar;
     then sp DD, sp DD' are_homotopically_equivalent by C32,W;
    hence thesis by HT4,W;
   end;
```