Implementation of Zermelo's work of 1908 in Lestrade: Part IV, central impredicative argument for total ordering of $\mathbf{M}$

M. Randall Holmes

June 5, 2020

1 Introduction

This document was originally titled as an essay on the proposition that mathematics is what can be done in Automath (as opposed to what can be done in ZFC, for example). Such an essay is still in in my mind, but this particular document has transformed itself into the large project of implementing Zermelo's two important set theory papers of 1908 in Lestrade, with the further purpose of exploring the actual capabilities of Zermelo's system of 1908 as a mathematical foundation, which we think are perhaps underrated.

This is a new version of this document in modules, designed to make it possible to work more efficiently without repeated execution of slow log files when they do not need to be revisited.

This particular part is monstrously large and slow and needs some fine tuning.

In this section, we prove that $\mathbf{M}$ is totally ordered by inclusion. This involves showing that the collection of elements of $\mathbf{M}$ which either include or are included in each other element of $\mathbf{M}$ is itself a Θ-chain and so actually equal to $\mathbf{M}$. The horrible thing about this is that the proof of the third component of this result contains a proof that a further refinement of this set definition also yields a Θ-chain, with its own four parts.

```
Lestrade execution:
```

```
load whatismath3
     clearcurrent


   declare C obj

>>    C: obj {move 2}



   declare D obj

>>    D: obj {move 2}



   define cuts1 C: (C E Mbold) & Forall[D=>(D \
         E Mbold) -> (D <<= C) V (C <<= D)] \




>>    cuts1: [(C_1:obj) => (---:prop)]
>>      {move 1}



   save

   close

declare C666 obj

>> C666: obj {move 1}
```

```
define cuts2 Misset thelawchooses, C666: \
   cuts1 C666

>> cuts2: [(.M_1:obj),(Misset_1:that Isset(.M_1)),
>>      (.thelaw_1:[(S_2:obj) => (---:obj)]),
>>      (thelawchooses_1:[(.S_3:obj),(subsetev_3:
>>         that (.S_3 <<= .M_1)),(inev_3:that
>>         Exists([(x_4:obj) => ((x_4 E .S_3):
>>            prop)]))
>>         => (---:that (.thelaw_1(.S_3) E .S_3))]),
>>      (C666_1:obj) => (((C666_1 E (Misset_1
>>      Mbold2 thelawchooses_1)) & Forall([(D_5:
>>         obj) => (((D_5 E (Misset_1 Mbold2 thelawchooses_1))
>>         -> ((D_5 <<= C666_1) V (C666_1 <<=
>>         D_5))):prop)]))
>>      :prop)]
>>   {move 0}



open

   define cuts C: cuts2 Misset thelawchooses, \
      C

>>    cuts: [(C_1:obj) => (---:prop)]
>>      {move 1}



   define Cuts1: Set (Mbold, cuts)

>>    Cuts1: [(---:obj)]
>>      {move 1}



   close
```

```
define Cuts3 Misset thelawchooses: Cuts1


>> Cuts3: [(.M_1:obj),(Misset_1:that Isset(.M_1)),
>>      (.thelaw_1:[(S_2:obj) => (---:obj)]),
>>      (thelawchooses_1:[(.S_3:obj),(subsetev_3:
>>         that (.S_3 <<= .M_1)),(inev_3:that
>>         Exists([(x_4:obj) => ((x_4 E .S_3):
>>            prop)]))
>>         => (---:that (.thelaw_1(.S_3) E .S_3))])
>>      => (((Misset_1 Mbold2 thelawchooses_1)
>>      Set [(C_5:obj) => (cuts2(Misset_1,thelawchooses_1,
>>         C_5):prop)])
>>      :obj)]
>>   {move 0}
```

```
open

   define Cuts: Cuts3 Misset thelawchooses


>>    Cuts: [(---:obj)]
>>      {move 1}
```

This defines the predicate "is an element of **M** which either includes or is included in each element of `M`" and the correlated set. These things are packaged so as not to expand. The aim is to show that `Cuts` is a Θ-chain, from which we will be able to show the desired linear ordering result.

```
Lestrade execution:


   define line1:Simp1 Mboldtheta
```

```
>>    line1: [(---:that (M E (Misset Mbold2
>>         thelawchooses)))]
>>      {move 1}


   open

      declare F obj

>>       F: obj {move 3}


      open

         declare finmbold that F E Mbold


>>          finmbold: that (F E Mbold) {move
>>            4}


         define line2 finmbold: Iff1(Mp finmbold, \
            Ui F Simp1 Simp1 Simp2 Mboldtheta, \
            Ui F Scthm M)

>>          line2: [(finmbold_1:that (F E Mbold))
>>               => (---:that (F <<= M))]
>>            {move 3}


         define line3 finmbold: Add1(M <<= \
            F, line2 finmbold)
```

```
>>          line3: [(finmbold_1:that (F E Mbold))
>>               => (---:that ((F <<= M) V (M
>>               <<= F)))]
>>            {move 3}



         close

      define line4 F : Ded line3

>>       line4: [(F_1:obj) => (---:that ((F_1
>>            E Mbold) -> ((F_1 <<= M) V (M <<=
>>            F_1))))]
>>         {move 2}



      close

   define line5: Ug line4

>>    line5: [(---:that Forall([(F_12:obj) =>
>>           (((F_12 E Mbold) -> ((F_12 <<= M)
>>            V (M <<= F_12))):prop)]))
>>         ]
>>      {move 1}



   define line6: Fixform(cuts M,Conj(line1, \
      line5))

>>    line6: [(---:that cuts(M))]
>>      {move 1}
```

```
   define line7: Conj(Simp1 Mboldtheta, line6)


>>    line7: [(---:that ((M E (Misset Mbold2
>>         thelawchooses)) & cuts(M)))]
>>      {move 1}




   define line8: Ui M, Separation (Mbold, \
      cuts)

>>    line8: [(---:that ((M E (Mbold Set cuts))
>>         == ((M E Mbold) & cuts(M))))]
>>      {move 1}




   define Line9: Fixform(M E Cuts,Iff2(line7, \
      line8))

>>    Line9: [(---:that (M E Cuts))]
>>      {move 1}
```

This is the first component of the proof that `Cuts` is a Θ-chain.

```
Lestrade execution:


   define line10: Fixform(Cuts <<= (Mbold), \
      Sepsub (Mbold, cuts, Inhabited (Simp1 \
      (Mboldtheta))) )

>>    line10: [(---:that (Cuts <<= Mbold))]
>>      {move 1}
```

```
   define line11 : Fixform((Mbold)<<= Sc \
      M,Sepsub2 (Sc2 M,Refleq (Mbold)))

>>    line11: [(---:that (Mbold <<= Sc(M)))]
>>      {move 1}



   define Line12: Transsub(line10, line11)


>>    Line12: [(---:that (Cuts <<= Sc(M)))]
>>      {move 1}
```

This is the second component of the proof that `Cuts` is a Θ-chain.

```
Lestrade execution:


   open

      declare B obj

>>       B: obj {move 3}



      open

         declare bhyp that B E Cuts

>>          bhyp: that (B E Cuts) {move 4}
```

```
         define line13 bhyp: Iff1(bhyp, Ui \
            B, Separation (Mbold, cuts))

>>          line13: [(bhyp_1:that (B E Cuts))
>>               => (---:that ((B E Mbold) & cuts(B)))]
>>            {move 3}



         define line14 bhyp: Simp1 line13 \
            bhyp

>>          line14: [(bhyp_1:that (B E Cuts))
>>               => (---:that (B E Mbold))]
>>            {move 3}



         define linea14 bhyp: Setsinchains \
            Mboldtheta, line14 bhyp

>>          linea14: [(bhyp_1:that (B E Cuts))
>>               => (---:that Isset(B))]
>>            {move 3}



         define lineb14 bhyp: Iff1(Mp(line14 \
            bhyp,Ui (B, Simp1 Simp1 Simp2 Mboldtheta)), \
            Ui B, Scthm M)

>>          lineb14: [(bhyp_1:that (B E Cuts))
>>               => (---:that (B <<= M))]
>>            {move 3}
```

```
         define line15 bhyp: Simp2 Simp2 \
            line13 bhyp

>>          line15: [(bhyp_1:that (B E Cuts))
>>               => (---:that Forall([(D_3:obj)
>>                  => (((D_3 E (Misset Mbold2
>>                  thelawchooses)) -> ((D_3 <<=
>>                  B) V (B <<= D_3))):prop)]))
>>               ]
>>            {move 3}



         open

            declare F obj

>>             F: obj {move 5}



            declare fhyp that F E (Mbold)


>>             fhyp: that (F E Mbold) {move
>>               5}



            define line16 fhyp: Fixform((prime \
               F)<<=F, Sepsub2 (Setsinchains \
               Mboldtheta,fhyp, Refleq (prime \
               F)))

>>             line16: [(.F_1:obj),(fhyp_1:that
>>                  (.F_1 E Mbold)) => (---:that
>>                  (prime(.F_1) <<= .F_1))]
>>               {move 4}
```

```
            declare Y obj

>>             Y: obj {move 5}



            define cutsa2 Y: (Y <<= prime \
               B) V B <<= Y

>>             cutsa2: [(Y_1:obj) => (---:prop)]
>>               {move 4}



            save

            close

         declare Y100 obj

>>          Y100: obj {move 4}



         define cutsb2 Y100: cutsa2 Y100


>>          cutsb2: [(Y100_1:obj) => (---:prop)]
>>            {move 3}



         save

         close
```

```
      declare Y101 obj

>>       Y101: obj {move 3}



      define cutsc2 B Y101: cutsb2 Y101

>>       cutsc2: [(B_1:obj),(Y101_1:obj) =>
>>            (---:prop)]
>>         {move 2}



      save

      close

   declare Ba1 obj

>>    Ba1: obj {move 2}



   declare Y102 obj

>>    Y102: obj {move 2}



   define cutsd2 Ba1 Y102: cutsc2 Ba1 Y102


>>    cutsd2: [(Ba1_1:obj),(Y102_1:obj) => (---:
>>         prop)]
>>      {move 1}
```

```
   save

   close

declare Ba2 obj

>> Ba2: obj {move 1}



declare Y103 obj

>> Y103: obj {move 1}



define cutse2 Misset thelawchooses, Ba2 Y103: \
   cutsd2 Ba2 Y103

>> cutse2: [(.M_1:obj),(Misset_1:that Isset(.M_1)),
>>      (.thelaw_1:[(S_2:obj) => (---:obj)]),
>>      (thelawchooses_1:[(.S_3:obj),(subsetev_3:
>>         that (.S_3 <<= .M_1)),(inev_3:that
>>         Exists([(x_4:obj) => ((x_4 E .S_3):
>>            prop)]))
>>         => (---:that (.thelaw_1(.S_3) E .S_3))]),
>>      (Ba2_1:obj),(Y103_1:obj) => (((Y103_1
>>      <<= prime2(.thelaw_1,Ba2_1)) V (Ba2_1
>>      <<= Y103_1)):prop)]
>>   {move 0}



open

   define cutsf2 Ba1 Y102: cutse2 Misset, \
      thelawchooses, Ba1 Y102

>>    cutsf2: [(Ba1_1:obj),(Y102_1:obj) => (---:
>>         prop)]
>>      {move 1}



   open

      define cutsg2 B Y101: cutsf2 B Y101


>>       cutsg2: [(B_1:obj),(Y101_1:obj) =>
>>            (---:prop)]
>>         {move 2}



      open

         define cutsh2 Y100: cutsg2 B Y100


>>          cutsh2: [(Y100_1:obj) => (---:prop)]
>>            {move 3}



         open

            define cutsi2 Y: cutsh2 Y

>>             cutsi2: [(Y_1:obj) => (---:prop)]
>>               {move 4}
```

```
            define Cuts2: Set(Mbold,cutsi2)


>>             Cuts2: [(---:obj)]
>>               {move 4}
```

We are in the midst of the third component of the proof that Cuts is a Θ-chain. We have B which we assume is in Cuts and we want to show that prime(B) is in Cuts. We do this by showing that the set of all elements of M which are either included in prime(B) or include B is a Θ-chain. Thus we have four components of this proof to generate before we get to generating the third component of the proof for Cuts.

This is about the time that I defined the goal command which is used to generate helpful comments about what we are trying to prove in the rest of the files. I should probably backtrack and insert goal statements earlier!

Lestrade execution:

```
            goal that thetachain Cuts2

>>             Goal: that thetachain(Cuts2)


            test thetachain

>>             Test: thetachain

>>             [(C_1:obj) => (thetachain1(M,
>>                  thelaw,C_1):prop)]
>>



            goal that M E Cuts2
```

```
>>             Goal: that (M E Cuts2)

            define line17 : Ui M, Separation4 \
               Refleq Cuts2

>>             line17: [(---:that ((M E (Mbold
>>                  Set cutsi2)) == ((M E Mbold)
>>                  & cutsi2(M))))]
>>               {move 4}



            define line18: Conj(Simp1 \
               Mboldtheta, Add2(M<<=prime B, \
               lineb14 bhyp))

>>             line18: [(---:that ((M E (Misset
>>                  Mbold2 thelawchooses)) & ((M
>>                  <<= prime(B)) V (B <<= M))))]
>>               {move 4}



            define line19: Fixform(M E Cuts2, \
               Iff2 line18 line17)

>>             line19: [(---:that (M E Cuts2))]
>>               {move 4}
```

This is the first component of the proof that Cuts2 is a Θ-chain.

Lestrade execution:

```
            goal that Cuts2 <<= Sc M
```

```
>>             Goal: that (Cuts2 <<= Sc(M))


            declare D1 obj

>>             D1: obj {move 5}



            define line20: Fixform(Cuts2 \
               <<= Mbold, Sepsub2(Separation3 \
               Refleq Mbold,Refleq Cuts2))


>>             line20: [(---:that (Cuts2 <<=
>>                  Mbold))]
>>               {move 4}



            define line21 : Transsub line20 \
               Simp1 Simp2 Mboldtheta

>>             line21: [(---:that (Cuts2 <<=
>>                  Sc(M)))]
>>               {move 4}
```

This is the second component of the proof that `Cuts` is a Θ-chain.

```
Lestrade execution:


            declare F1 obj

>>             F1: obj {move 5}
```

```
            goal that Forall[D1 => (D1 E \
                  Cuts2) -> (prime D1) E Cuts2] \



>>             Goal: that Forall([(D1_1297:obj)
>>                  => (((D1_1297 E Cuts2) ->
>>                  (prime(D1_1297) E Cuts2)):
>>                  prop)])
>>

            open

               declare D2 obj

>>                D2: obj {move 6}



               open

                  declare dhyp that D2 E \
                     Cuts2

>>                   dhyp: that (D2 E Cuts2)
>>                     {move 7}



                  goal that (prime D2) E \
                     Cuts2

>>                   Goal: that (prime(D2) E
>>                     Cuts2)
```

```
                  define line22 : Ui prime \
                     D2, Separation4 Refleq \
                     Cuts2

>>                   line22: [(---:that ((prime(D2)
>>                        E (Mbold Set cutsi2))
>>                        == ((prime(D2) E Mbold)
>>                        & cutsi2(prime(D2)))))]
>>                     {move 6}



                  goal that ((prime D2) E \
                     Mbold) & ((prime D2) <<= \
                     prime B) V (B <<= prime \
                     D2)

>>                   Goal: that ((prime(D2)
>>                     E Mbold) & ((prime(D2)
>>                     <<= prime(B)) V (B <<=
>>                     prime(D2))))

                  define line23 dhyp: \
                     Iff1 dhyp,Ui D2 \
                     Separation4 Refleq Cuts2


>>                   line23: [(dhyp_1:that (D2
>>                        E Cuts2)) => (---:that
>>                        ((D2 E Mbold) & cutsi2(D2)))]
>>                     {move 6}



                  define line24 dhyp: \
                     Simp1 line23 dhyp

>>                   line24: [(dhyp_1:that (D2
>>                        E Cuts2)) => (---:that
>>                        (D2 E Mbold))]
>>                     {move 6}



                  define line25 dhyp: \
                     Simp2 line23 dhyp

>>                   line25: [(dhyp_1:that (D2
>>                        E Cuts2)) => (---:that
>>                        cutsi2(D2))]
>>                     {move 6}



                  define line26: Iff1 \
                     bhyp, Ui B, Separation4 \
                     Refleq Cuts

>>                   line26: [(---:that ((B
>>                        E (Misset Mbold2 thelawchooses))
>>                        & cuts2(Misset,thelawchooses,
>>                        B)))]
>>                     {move 6}



                  define line27 dhyp: \
                     Mp line24 dhyp, \
                     Ui D2, Simp2 Simp2 line26


>>                   line27: [(dhyp_1:that (D2
>>                        E Cuts2)) => (---:that
>>                        ((D2 <<= B) V (B <<=
>>                        D2)))]
>>                     {move 6}
```

```
                  define line28 dhyp: \
                     Mp line24 dhyp, \
                     Ui D2, Simp1 Simp2 Simp2 \
                     Mboldtheta

>>                   line28: [(dhyp_1:that (D2
>>                        E Cuts2)) => (---:that
>>                        (prime2(thelaw,D2) E
>>                        (Misset Mbold2 thelawchooses)))]
>>                     {move 6}



                  define line29 dhyp: \
                     Mp line28 dhyp, \
                     Ui prime D2, Simp2 Simp2 \
                     line26

>>                   line29: [(dhyp_1:that (D2
>>                        E Cuts2)) => (---:that
>>                        ((prime(D2) <<= B) V
>>                        (B <<= prime(D2))))]
>>                     {move 6}



                  goal that ((prime D2) <<= \
                     prime B) V (B <<= prime \
                     D2)

>>                   Goal: that ((prime(D2)
>>                     <<= prime(B)) V (B <<=
>>                     prime(D2)))

                  open
```

```
                     declare U obj

>>                      U: obj {move 8}



                     declare Casehyp1 \
                        that B=0

>>                      Casehyp1: that (B =
>>                        0) {move 8}



                     define linea29 Casehyp1: \
                        Subs1(Eqsymm Casehyp1, \
                        Add2(prime D2 <<= prime \
                        B, Zeroissubset Separation3 \
                        Refleq prime D2))

>>                      linea29: [(Casehyp1_1:
>>                           that (B = 0)) =>
>>                           (---:that ((prime(D2)
>>                           <<= prime(B)) V (B
>>                           <<= (D2 Set [(x_7:
>>                              obj) => (~((x_7
>>                              E Usc(thelaw(D2)))):
>>                              prop)]))
>>                           ))]
>>                        {move 7}



                     declare Casehyp2 \
                        that Exists[U=> U E \
                           B] \
>>                        Casehyp2: that Exists([(U_1:
>>                             obj) => ((U_1 E B):
>>                             prop)])
>>                          {move 8}



                     open

                        declare casehyp1 \
                           that D2 <<= prime \
                           B

>>                         casehyp1: that (D2
>>                           <<= prime(B)) {move
>>                           9}



                        declare casehyp2 \
                           that B <<= D2

>>                         casehyp2: that (B
>>                           <<= D2) {move 9}



                        define line30 \
                           casehyp1: \
                           Transsub(line16 (line24 \
                           dhyp), casehyp1)


>>                         line30: [(casehyp1_1:
>>                              that (D2 <<= prime(B)))
>>                              => (---:that (prime(D2)
>>                              <<= prime(B)))]
>>                           {move 8}



                        define linea30 casehyp1: \
                           Add1(B <<= prime \
                           D2, line30 casehyp1)


>>                         linea30: [(casehyp1_1:
>>                              that (D2 <<= prime(B)))
>>                              => (---:that ((prime(D2)
>>                              <<= prime(B))
>>                              V (B <<= prime(D2))))]
>>                           {move 8}



                        define line31 \
                           : Excmid ((thelaw \
                           D2) = thelaw \
                           B)

>>                         line31: [(---:that
>>                              ((thelaw(D2) =
>>                              thelaw(B)) V ~((thelaw(D2)
>>                              = thelaw(B)))))]
>>                           {move 8}



                        define line32 \
                           : Separation4 \
                           Refleq prime D2

>>                         line32: [(---:that
>>                              Forall([(x_2:obj)
>>                                 => (((x_2 E
>>                                 (D2 Set [(x_3:
>>                                    obj) =>
>>                                    (~((x_3
>>                                    E Usc(thelaw(D2)))):
>>                                    prop)]))
>>                                 == ((x_2 E
>>                                 D2) & ~((x_2
>>                                 E Usc(thelaw(D2)))))):
>>                                 prop)]))
>>                              ]
>>                           {move 8}


                        open

                           declare casehypa1 \
                              that (thelaw \
                              D2 = thelaw B)


>>                            casehypa1: that
>>                              (thelaw(D2) =
>>                              thelaw(B)) {move
>>                              10}



                           declare casehypa2 \
                              that ~(thelaw \
                              D2 = thelaw B)


>>                            casehypa2: that
>>                              ~((thelaw(D2)
>>                              = thelaw(B)))
>>                              {move 10}



                           open

                              declare \
                                 G obj

>>                               G: obj {move
>>                                 11}



                              open

                                 declare \
                                    onedir \
                                    that G \
                                    E prime \
                                    D2

>>                                  onedir:
>>                                    that (G
>>                                    E prime(D2))
>>                                    {move 12}



                                 define line33 \
                                    onedir \
                                    : Iff1 \
                                    onedir, \
                                    Ui G line32


>>                                  line33:
>>                                    [(onedir_1:
```

```
>>                                         that
>>                                         (G E
>>                                         prime(D2)))
>>                                         => (---:
>>                                         that
>>                                         ((G E
>>                                         D2) &
>>                                         ~((G
>>                                         E Usc(thelaw(D2))))))]
>>                                      {move 11}
```

```
define line34 \
   onedir: \
   Simp1 line33 \
   onedir
```

```
>>                                    line34:
>>                                      [(onedir_1:
>>                                         that
>>                                         (G E
>>                                         prime(D2)))
>>                                         => (---:
>>                                         that
>>                                         (G E
>>                                         D2))]
>>                                      {move 11}
```

```
define line35 \
   onedir: \
   Simp2 line33 \
   onedir
```

```
>>                                   line35:
>>                                     [(onedir_1:
>>                                        that
>>                                        (G E
>>                                        prime(D2)))
>>                                        => (---:
>>                                        that
>>                                        ~((G
>>                                        E Usc(thelaw(D2)))))]
>>                                     {move 11}
```

```
                                open

                                   declare \
                                      eqhyp \
                                      that \
                                      G=(thelaw \
                                      D2)
```

```
>>                                      eqhyp:
>>                                        that
>>                                        (G =
>>                                        thelaw(D2))
>>                                        {move
>>                                        13}
```

```
                                   define \
                                      line36 \
                                      eqhyp: \
                                      Subs1 \
                                      Eqsymm \
                                      eqhyp \
                                      line35 \
                                       onedir

>>                                     line36:
>>                                       [(eqhyp_1:
>>                                          that (G
>>                                          = thelaw(D2)))
>>                                          => (---:
>>                                          that ~((G
>>                                          E Usc(G))))]
>>                                       {move
>>                                       12}

                                    define \
                                       line37 \
                                       eqhyp: \
                                       Mp (Inusc2 \
                                       G,line36 \
                                       eqhyp)

>>                                     line37:
>>                                       [(eqhyp_1:
>>                                          that (G
>>                                          = thelaw(D2)))
>>                                          => (---:
>>                                          that ??)]
>>                                       {move
>>                                       12}

                                    close

                                 define line38 \
                                    onedir: \
                                    Negintro \
                                    line37


>>                                  line38:
>>                                    [(onedir_1:
>>                                       that
>>                                       (G E
>>                                       prime(D2)))
>>                                       => (---:
>>                                       that
>>                                       ~((G
>>                                       = thelaw(D2))))]
>>                                    {move 11}



                                 define line39 \
                                    onedir \
                                    : Subs1 \
                                    casehypa1 \
                                    line38 \
                                    onedir


>>                                  line39:
>>                                    [(onedir_1:
>>                                       that
>>                                       (G E
>>                                       prime(D2)))
>>                                       => (---:
>>                                       that
>>                                       ~((G
>>                                       = thelaw(B))))]
>>                                    {move 11}
```

```
                                 define linea39 \
                                    onedir \
                                    : Subs1 \
                                    casehypa1 \
                                    line35 \
                                    onedir


>>                                  linea39:
>>                                    [(onedir_1:
>>                                       that
>>                                       (G E
>>                                       prime(D2)))
>>                                       => (---:
>>                                       that
>>                                       ~((G
>>                                       E Usc(thelaw(B)))))]
>>                                    {move 11}



                                 open

                                    declare \
                                       casehypb1 \
                                       that \
                                       prime \
                                       D2 <<= \
                                       B

>>                                     casehypb1:
>>                                       that
>>                                       (prime(D2)
>>                                       <<= B)
>>                                       {move
>>                                       13}
```

```
                                    define \
                                       line40 \
                                       casehypb1: \
                                       Mp(onedir, \
                                       Ui G, \
                                       Simp1 \
                                       casehypb1)


>>                                     line40:
>>                                       [(casehypb1_1:
>>                                          that (prime(D2)
>>                                          <<= B))
>>                                          => (---:
>>                                          that (G
>>                                          E B))]
>>                                       {move
>>                                       12}



                                    declare \
                                       casehypb2 \
                                       that \
                                       B <<= \
                                       prime \
                                       D2

>>                                     casehypb2:
>>                                       that
>>                                       (B <<=
>>                                       prime(D2))
>>                                       {move
>>                                       13}
```

```
                                    define \
                                       line41 \
                                       casehypb2: \
                                       Ui thelaw \
                                       B, Simp1 \
                                       casehypb2


>>                                     line41:
>>                                       [(casehypb2_1:
>>                                          that (B
>>                                          <<= prime(D2)))
>>                                          => (---:
>>                                          that ((thelaw(B)
>>                                          E B) ->
>>                                          (thelaw(B)
>>                                          E prime(D2))))]
>>                                       {move
>>                                       12}



                                    define \
                                       line42: \
                                       thelawchooses \
                                       (lineb14 \
                                       bhyp, \
                                       Casehyp2)


>>                                     line42:
>>                                       [(---:
>>                                          that (thelaw(B)
>>                                          E B))]
>>                                       {move
>>                                       12}
```

```
                                    define \
                                       line43 \
                                       casehypb2: \
                                       Mp (line42, \
                                       line41 \
                                       casehypb2)


>>                                     line43:
>>                                       [(casehypb2_1:
>>                                          that (B
>>                                          <<= prime(D2)))
>>                                          => (---:
>>                                          that (thelaw(B)
>>                                          E prime(D2)))]
>>                                       {move
>>                                       12}



                                    define \
                                       line44 \
                                       casehypb2: \
                                       Iff1(line43 \
                                       casehypb2, \
                                       Ui thelaw \
                                       B,Separation4 \
                                       Refleq \
                                       prime \
                                       D2)


>>                                     line44:
>>                                       [(casehypb2_1:
>>                                          that (B
>>                                          <<= prime(D2)))
>>                                          => (---:
>>                                          that ((thelaw(B)
>>                                          E D2) &
>>                                          ~((thelaw(B)
>>                                          E Usc(thelaw(D2))))))]
>>                                       {move
>>                                       12}



                                    define \
                                       line45 \
                                       casehypb2: \
                                       Subs1 \
                                       Eqsymm \
                                       casehypa1 \
                                       line44 \
                                       casehypb2


>>                                     line45:
>>                                       [(casehypb2_1:
>>                                          that (B
>>                                          <<= prime(D2)))
>>                                          => (---:
>>                                          that ((thelaw(D2)
>>                                          E D2) &
>>                                          ~((thelaw(D2)
>>                                          E Usc(thelaw(D2))))))]
>>                                       {move
>>                                       12}



                                    define \
                                       line46 \
                                       casehypb2: \
                                       Simp2 \
                                       line45 \
                                       casehypb2


>>                                     line46:
>>                                       [(casehypb2_1:
>>                                          that (B
>>                                          <<= prime(D2)))
>>                                          => (---:
>>                                          that ~((thelaw(D2)
>>                                          E Usc(thelaw(D2)))))]
>>                                       {move
>>                                       12}



                                    define \
                                       line47 \
                                       casehypb2: \
                                       Giveup(G \
                                       E B, \
                                       Mp(Inusc2 \
                                       thelaw \
                                       D2, \
                                       line46 \
                                       casehypb2))


>>                                     line47:
>>                                       [(casehypb2_1:
>>                                          that (B
>>                                          <<= prime(D2)))
>>                                          => (---:
>>                                          that (G
>>                                          E B))]
>>                                       {move
>>                                       12}
```

```
                                    close


                                 define line48 \
                                    onedir: \
                                    Cases( \
                                    line29 \
                                    dhyp, line40, \
                                    line47)


>>                                  line48:
>>                                    [(onedir_1:
>>                                       that
>>                                       (G E
>>                                       prime(D2)))
>>                                       => (---:
>>                                       that
>>                                       (G E
>>                                       B))]
>>                                    {move 11}



                                 define linea48 \
                                    onedir: \
                                    Fixform(G \
                                    E prime(B), \
                                    Iff2(Conj(line48 \
                                    onedir, \
                                    linea39 \
                                    onedir), \
                                    Ui G, Separation4 \
                                    Refleq \
                                    prime B))
```

```
>>                                   linea48:
>>                                     [(onedir_1:
>>                                        that
>>                                        (G E
>>                                        prime(D2)))
>>                                        => (---:
>>                                        that
>>                                        (G E
>>                                        prime(B)))]
>>                                     {move 11}
```

```
                                  declare \
                                     otherdir \
                                     that G \
                                     E B

>>                                   otherdir:
>>                                     that (G
>>                                     E B) {move
>>                                     12}
```

```
                                  define line49 \
                                     otherdir: \
                                     Mp(otherdir, \
                                     Ui G Simp1 \
                                     casehyp2)
```

```
>>                                   line49:
>>                                     [(otherdir_1:
>>                                        that
>>                                        (G E
>>                                        B)) =>
>>                                         (---:
>>                                         that
>>                                         (G E
>>                                         D2))]
>>                                      {move 11}


                                   open

                                      declare \
                                         eqhyp2 \
                                         that \
                                         G E \
                                         Usc \
                                         thelaw \
                                         D2

>>                                       eqhyp2:
>>                                         that
>>                                         (G E
>>                                         Usc(thelaw(D2)))
>>                                         {move
>>                                         13}



                                      define \
                                         eqhypa2 \
                                         eqhyp2: \
                                         Oridem(Iff1(eqhyp2, \
                                         Ui G, \
                                         Pair(thelaw \
                                         D2,thelaw \
                                         D2)))


>>                                       eqhypa2:
```

```
>>                                       [(eqhyp2_1:
>>                                          that (G
>>                                          E Usc(thelaw(D2))))
>>                                          => (---:
>>                                          that (G
>>                                          = thelaw(D2)))]
>>                                       {move
>>                                       12}
```

```
                                    define \
                                       line50 \
                                       eqhyp2: \
                                       Subs1 \
                                       eqhypa2 \
                                       eqhyp2 \
                                       otherdir
```

```
>>                                     line50:
>>                                       [(eqhyp2_1:
>>                                          that (G
>>                                          E Usc(thelaw(D2))))
>>                                          => (---:
>>                                          that (thelaw(D2)
>>                                          E B))]
>>                                       {move
>>                                       12}
```

```
                                    open
```

```
                                       declare \
                                          impossiblesub \
                                          that B \
                                          <<= prime \
                                          D2

>>                                        impossiblesub:
>>                                          that (B
>>                                          <<= prime(D2))
>>                                          {move 14}



                                       define \
                                          line51 \
                                          impossiblesub: \
                                          Mp(line50 \
                                          eqhyp2, \
                                          Ui \
                                          (thelaw \
                                          D2, Simp1 \
                                          impossiblesub))


>>                                        line51:
>>                                          [(impossiblesub_1:
>>                                             that
>>                                             (B <<=
>>                                             prime(D2)))
>>                                             => (---:
>>                                             that
>>                                             (thelaw(D2)
>>                                             E prime(D2)))]
>>                                          {move 13}



                                       define \
                                          line52 \
                                          impossiblesub: \
                                          Iff1(line51 \
                                          impossiblesub, \
                                          Ui thelaw \
                                          D2,Separation4 \
                                          Refleq \
                                          prime \
                                          D2)

>>                                        line52:
>>                                          [(impossiblesub_1:
>>                                             that
>>                                             (B <<=
>>                                             prime(D2)))
>>                                             => (---:
>>                                             that
>>                                             ((thelaw(D2)
>>                                             E D2)
>>                                             & ~((thelaw(D2)
>>                                             E Usc(thelaw(D2))))))]
>>                                          {move 13}



                                       define \
                                          line53 \
                                          impossiblesub: \
                                          Mp(Inusc2 \
                                          thelaw \
                                          D2, Simp2 \
                                          line52 \
                                          impossiblesub)


>>                                        line53:
>>                                          [(impossiblesub_1:
>>                                             that
>>                                             (B <<=
>>                                             prime(D2)))
>>                                             => (---:
```

```
>>                                             that
>>                                             ??)]
>>                                          {move 13}



                                       close

                                    define \
                                       line54 \
                                       eqhyp2 \
                                       : Negintro \
                                       line53


>>                                     line54:
>>                                       [(eqhyp2_1:
>>                                          that (G
>>                                          E Usc(thelaw(D2))))
>>                                          => (---:
>>                                          that ~((B
>>                                          <<= prime(D2))))]
>>                                        {move
>>                                        12}



                                    define \
                                       line55 \
                                       eqhyp2: \
                                       Ds1 \
                                       line29 \
                                       dhyp \
                                       line54 \
                                       eqhyp2


>>                                     line55:
```

```
>>                                        [(eqhyp2_1:
>>                                           that (G
>>                                           E Usc(thelaw(D2))))
>>                                           => (---:
>>                                           that (prime(D2)
>>                                           <<= B))]
>>                                        {move
>>                                        12}


                                     open


                                        declare \
                                           H obj


>>                                         H: obj
>>                                           {move 14}


                                        open

                                           declare \
                                              hhyp \
                                              that \
                                              H E \
                                              D2


>>                                            hhyp:
>>                                              that
>>                                              (H E
>>                                              D2)
>>                                              {move
>>                                              15}
```

```
                                          define \
                                             line56: \
                                             Excmid( \
                                             H = \
                                             thelaw \
                                             D2)


>>                                           line56: [(---:
>>                                                that ((H
>>                                                = thelaw(D2))
>>                                                V ~((H
>>                                                = thelaw(D2)))))]
>>                                             {move 14}



                                          open

                                             declare \
                                                casehhyp1 \
                                                that H \
                                                = thelaw \
                                                D2

>>                                              casehhyp1:
>>                                                that (H
>>                                                = thelaw(D2))
>>                                                {move 16}



                                             declare \
                                                casehhyp2 \
                                                that ~(H \
                                                = thelaw \
                                                D2)

>>                                              casehhyp2:
>>                                                that ~((H
>>                                                = thelaw(D2)))
>>                                                {move 16}



                                             define \
                                                line57 \
                                                casehhyp1: \
                                                Subs1(Eqsymm \
                                                casehhyp1, \
                                                line50 \
                                                eqhyp2)


>>                                              line57:
>>                                                [(casehhyp1_1:
>>                                                   that
>>                                                   (H =
>>                                                   thelaw(D2)))
>>                                                   => (---:
>>                                                   that
>>                                                   (H E
>>                                                   B))]
>>                                                {move 15}



                                             open

                                                declare \
                                                   sillyhyp \
                                                   that \
                                                   H E \
                                                   Usc \
                                                   thelaw \
                                                   D2


>>                                                 sillyhyp:
>>                                                   that
>>                                                   (H E
>>                                                   Usc(thelaw(D2)))
>>                                                   {move
>>                                                   17}



                                                define \
                                                   line58 \
                                                   sillyhyp: \
                                                   Mp(Oridem(Iff1(sillyhyp, \
                                                   Ui \
                                                   H, \
                                                   Pair(thelaw \
                                                   D2, \
                                                   thelaw \
                                                   D2))), \
                                                   casehhyp2)


>>                                                 line58: [(sillyhyp_1:
>>                                                      that (H
>>                                                      E Usc(thelaw(D2))))
>>                                                      => (---:
>>                                                      that ??)]
>>                                                   {move 16}



                                                close
```

```
                                             define line59 \
                                                casehhyp2: Negintro \
                                                line58

>>                                              line59: [(casehhyp2_1:
>>                                                   that ~((H
>>                                                   = thelaw(D2))))
>>                                                   => (---:that
>>                                                   ~((H E Usc(thelaw(D2)))))]
>>                                                {move 15}



                                             define line60 \
                                                casehhyp2: Fixform(H \
                                                E prime D2, \
                                                Iff2(Conj(hhyp, \
                                                line59 casehhyp2), \
                                                Ui H,Separation4 \
                                                Refleq prime \
                                                D2))

>>                                              line60: [(casehhyp2_1:
>>                                                   that ~((H
>>                                                   = thelaw(D2))))
>>                                                   => (---:that
>>                                                   (H E prime(D2)))]
>>                                                {move 15}



                                             define line61 \
                                                casehhyp2: Mp(line60 \
                                                casehhyp2, Ui \
                                                H,Simp1 line55 \
                                                eqhyp2)

>>                                              line61: [(casehhyp2_1:
```

```
>>                                                   that ~((H
>>                                                   = thelaw(D2))))
>>                                                   => (---:that
>>                                                   (H E B))]
>>                                                {move 15}


                                             close

                                          define line62 hhyp: \
                                             Cases line56 line57, \
                                             line61

>>                                           line62: [(hhyp_1:
>>                                                that (H E D2))
>>                                                => (---:that
>>                                                (H E B))]
>>                                             {move 14}


                                          close

                                       define line63 H: Ded \
                                          line62

>>                                        line63: [(H_1:obj)
>>                                             => (---:that ((H_1
>>                                             E D2) -> (H_1 E
>>                                             B)))]
>>                                          {move 13}


                                       close

                                    define \
                                                   line64 \
                                                   eqhyp2: \
                                                   Ug line63


>>                                                 line64:
>>                                                   [(eqhyp2_1:
>>                                                      that (G
>>                                                      E Usc(thelaw(D2))))
>>                                                      => (---:
>>                                                      that Forall([(H_16:
>>                                                         obj)
>>                                                         => (((H_16
>>                                                         E D2)
>>                                                         -> (H_16
>>                                                         E B)):
>>                                                         prop)]))
>>                                                      ]
>>                                                   {move
>>                                                   12}



                                                define \
                                                   line65 \
                                                   eqhyp2: \
                                                   Fixform(D2 \
                                                   <<= \
                                                   B, Conj(line64 \
                                                   eqhyp2, \
                                                   Conj(Simp2 \
                                                   Simp2 \
                                                   casehyp2 \
                                                   ,linea14 \
                                                   bhyp)))


>>                                                 line65:
```

```
>>                                       [(eqhyp2_1:
>>                                          that (G
>>                                          E Usc(thelaw(D2))))
>>                                          => (---:
>>                                          that (D2
>>                                          <<= B))]
>>                                       {move
>>                                       12}
```

```
                                    define \
                                       line66 \
                                       eqhyp2 \
                                       : Antisymsub(casehyp2, \
                                       line65 \
                                       eqhyp2)
```

```
>>                                     line66:
>>                                       [(eqhyp2_1:
>>                                          that (G
>>                                          E Usc(thelaw(D2))))
>>                                          => (---:
>>                                          that (B
>>                                          = D2))]
>>                                       {move
>>                                       12}
```

```
                                    define \
                                       line67 \
                                       eqhyp2 \
                                       : Mp(Refleq \
                                       thelaw \
                                       D2, \
                                       Subs1(line66 \
                                       eqhyp2, \
                                       casehypa2))


>>                                     line67:
>>                                       [(eqhyp2_1:
>>                                          that (G
>>                                          E Usc(thelaw(D2))))
>>                                          => (---:
>>                                          that ??)]
>>                                       {move
>>                                       12}



                                    close


                                 define line68 \
                                    otherdir: \
                                    Fixform(G \
                                    E prime \
                                    D2,Iff2(Conj(line49 \
                                    otherdir, \
                                    Negintro \
                                    line67), \
                                    Ui G, Separation4 \
                                    Refleq \
                                    prime D2))


>>                                  line68:
>>                                    [(otherdir_1:
>>                                       that
>>                                       (G E
>>                                       B)) =>
>>                                       (---:
>>                                       that
>>                                       (G E
>>                                       prime(D2)))]
>>                                    {move 11}


                                 close

                              define line69 \
                                 G: Ded \
                                 line68

>>                               line69: [(G_1:
>>                                    obj) =>
>>                                    (---:that
>>                                    ((G_1 E
>>                                    B) -> (G_1
>>                                    E prime(D2))))]
>>                                 {move 10}



                              define testline \
                                 G: Ded \
                                 linea48

>>                               testline: [(G_1:
>>                                    obj) =>
>>                                    (---:that
>>                                    ((G_1 E
>>                                    prime(D2))
>>                                    -> (G_1
>>                                    E prime(B))))]
>>                                 {move 10}


                              close
```

```
                           define line70 \
                              casehypa2: Ug \
                              line69

>>                            line70: [(casehypa2_1:
>>                                 that ~((thelaw(D2)
>>                                 = thelaw(B))))
>>                                 => (---:that
>>                                 Forall([(G_39:
>>                                    obj) =>
>>                                    (((G_39
>>                                    E B) ->
>>                                    (G_39 E
>>                                    prime(D2))):
>>                                    prop)]))
>>                                 ]
>>                              {move 9}



                           define line71 \
                              casehypa2: Add2((prime \
                              D2) <<= prime \
                              B, Fixform(B \
                              <<= prime D2, \
                              Conj(line70 casehypa2, \
                              Conj(linea14 \
                              bhyp, Separation3 \
                              Refleq prime \
                              D2))))

>>                            line71: [(casehypa2_1:
>>                                 that ~((thelaw(D2)
>>                                 = thelaw(B))))
>>                                 => (---:that
>>                                 ((prime(D2)
>>                                 <<= prime(B))
>>                                 V (B <<= prime(D2))))]
>>                              {move 9}



                           define testline2 \
                              casehypa1: Ug \
                              testline

>>                            testline2: [(casehypa1_1:
>>                                 that (thelaw(D2)
>>                                 = thelaw(B)))
>>                                 => (---:that
>>                                 Forall([(G_22:
>>                                    obj) =>
>>                                    (((G_22
>>                                    E prime(D2))
>>                                    -> (G_22
>>                                    E prime(B))):
>>                                    prop)]))
>>                                 ]
>>                              {move 9}



                           define line72 \
                              casehypa1: Add1(B \
                              <<= prime D2, \
                              Fixform((prime \
                              D2) <<= prime \
                              B, Conj(testline2 \
                              casehypa1, Conj(Separation3 \
                              Refleq prime \
                              D2, Separation3 \
                              Refleq prime \
                              B))))

>>                            line72: [(casehypa1_1:
```

```
>>                                 that (thelaw(D2)
>>                                 = thelaw(B)))
>>                                 => (---:that
>>                                 ((prime(D2)
>>                                 <<= prime(B))
>>                                 V (B <<= prime(D2))))]
>>                              {move 9}



                           close

                        define line73 \
                           casehyp2:Cases line31 \
                           line72, line71

>>                         line73: [(casehyp2_1:
>>                              that (B <<= D2))
>>                              => (---:that ((prime(D2)
>>                              <<= prime(B))
>>                              V (B <<= prime(D2))))]
>>                           {move 8}



                        close

                     define line74 Casehyp2: \
                        Cases (line25 dhyp, \
                        linea30, line73)

>>                      line74: [(Casehyp2_1:
>>                           that Exists([(U_2:
>>                              obj) => ((U_2
>>                              E B):prop)]))
>>                           => (---:that ((prime(D2)
>>                           <<= prime(B)) V (B
>>                           <<= prime(D2))))]
>>                        {move 7}



                       close

                    define line75 dhyp: \
                       Cases(linea14 bhyp , linea29, \
                       line74)

>>                     line75: [(dhyp_1:that (D2
>>                          E Cuts2)) => (---:that
>>                          ((prime(D2) <<= prime(B))
>>                          V (B <<= (D2 Set [(x_86:
>>                             obj) => (~((x_86
>>                             E Usc(thelaw(D2)))):
>>                             prop)]))
>>                          ))]
>>                       {move 6}



                    define line76 dhyp: \
                       Fixform((prime D2) E Cuts2, \
                       Iff2(Conj(line28 dhyp, \
                       line75 dhyp), Ui prime \
                       D2,Separation4 Refleq \
                       Cuts2))

>>                     line76: [(dhyp_1:that (D2
>>                          E Cuts2)) => (---:that
>>                          (prime(D2) E Cuts2))]
>>                       {move 6}



                    close
```

```
               define line77 D2: Ded line76


>>                line77: [(D2_1:obj) => (---:
>>                     that ((D2_1 E Cuts2) ->
>>                     (prime(D2_1) E Cuts2)))]
>>                  {move 5}



               close

            define linea78: Ug line77

>>             linea78: [(---:that Forall([(D2_135:
>>                     obj) => (((D2_135 E Cuts2)
>>                     -> (prime(D2_135) E Cuts2)):
>>                     prop)]))
>>                  ]
>>               {move 4}



            save

            close

         define lineb78 bhyp: linea78

>>          lineb78: [(bhyp_1:that (B E Cuts))
>>               => (---:that Forall([(D2_177:
>>                  obj) => (((D2_177 E (Mbold
>>                  Set [(Y_178:obj) => (cutsh2(Y_178):
>>                     prop)]))
>>                  -> (prime(D2_177) E (Mbold
>>                  Set [(Y_179:obj) => (cutsh2(Y_179):
>>                     prop)]))
>>                  ):prop)]))
>>               ]
>>            {move 3}



         save

         close

      declare bhypa1 that B E Cuts

>>       bhypa1: that (B E Cuts) {move 3}



      define linec78 bhypa1: lineb78 bhypa1


>>       linec78: [(.B_1:obj),(bhypa1_1:that
>>            (.B_1 E Cuts)) => (---:that Forall([(D2_190:
>>               obj) => (((D2_190 E (Mbold Set
>>               [(Y_191:obj) => ((.B_1 cutsg2
>>                  Y_191):prop)]))
>>               -> (prime(D2_190) E (Mbold Set
>>               [(Y_192:obj) => ((.B_1 cutsg2
>>                  Y_192):prop)]))
>>               ):prop)]))
>>            ]
>>         {move 2}



      save

      close

   declare B111 obj
```

```
>>    B111: obj {move 2}



   declare bhypa2 that B111 E Cuts

>>    bhypa2: that (B111 E Cuts) {move 2}



   define lined78 bhypa2: linec78 bhypa2


>>    lined78: [(.B111_1:obj),(bhypa2_1:that
>>         (.B111_1 E Cuts)) => (---:that Forall([(D2_190:
>>            obj) => (((D2_190 E (Mbold Set [(Y_191:
>>               obj) => ((.B111_1 cutsf2 Y_191):
>>               prop)]))
>>            -> (prime(D2_190) E (Mbold Set [(Y_192:
>>               obj) => ((.B111_1 cutsf2 Y_192):
>>               prop)]))
>>            ):prop)]))
>>         ]
>>      {move 1}



   save

   close

declare B112 obj

>> B112: obj {move 1}



declare bhypa3 that B112 E Cuts
```

```
>> bhypa3: that (B112 E Cuts) {move 1}



define linee78 Misset thelawchooses, bhypa3: \
   lined78 bhypa3

>> linee78: [(.M_1:obj),(Misset_1:that Isset(.M_1)),
>>      (.thelaw_1:[(S_2:obj) => (---:obj)]),
>>      (thelawchooses_1:[(.S_3:obj),(subsetev_3:
>>         that (.S_3 <<= .M_1)),(inev_3:that
>>         Exists([(x_4:obj) => ((x_4 E .S_3):
>>            prop)]))
>>         => (---:that (.thelaw_1(.S_3) E .S_3))]),
>>      (.B112_1:obj),(bhypa3_1:that (.B112_1
>>      E (Misset_1 Cuts3 thelawchooses_1))) =>
>>      (Ug([(D2_8:obj) => (Ded([(dhyp_11:that
>>            (D2_8 E ((Misset_1 Mbold2 thelawchooses_1)
>>            Set [(Y_12:obj) => (cutse2(Misset_1,
>>               thelawchooses_1,.B112_1,Y_12):
>>               prop)]))
>>            ) => (((prime2(.thelaw_1,D2_8) E
>>            ((Misset_1 Mbold2 thelawchooses_1)
>>            Set [(Y_13:obj) => (cutse2(Misset_1,
>>               thelawchooses_1,.B112_1,Y_13):
>>               prop)]))
>>            Fixform (((Simp1((dhyp_11 Iff1 (D2_8
>>            Ui Separation4(Refleq(((Misset_1
>>            Mbold2 thelawchooses_1) Set [(Y_20:
>>               obj) => (cutse2(Misset_1,thelawchooses_1,
>>               .B112_1,Y_20):prop)]))
>>            )))) Mp (D2_8 Ui Simp1(Simp2(Simp2((Misset_1
>>            Mboldtheta2 thelawchooses_1))))))
>>            Conj Cases(Setsinchains2(Misset_1,
>>            thelawchooses_1,(Misset_1 Mboldtheta2
>>            thelawchooses_1),Simp1((bhypa3_1
>>            Iff1 (.B112_1 Ui ((Misset_1 Mbold2
>>            thelawchooses_1) Separation [(C_35:
>>               obj) => (cuts2(Misset_1,thelawchooses_1,
>>               C_35):prop)]))
>>            ))),[(Casehyp1_37:that (.B112_1
>>               = 0)) => ((Eqsymm(Casehyp1_37)
>>               Subs1 ((prime2(.thelaw_1,D2_8)
>>               <<= prime2(.thelaw_1,.B112_1))
>>               Add2 Zeroissubset(Separation3(Refleq(prime2(.thelaw_1,
>>               D2_8)))))):that ((prime2(.thelaw_1,
>>               D2_8) <<= prime2(.thelaw_1,.B112_1))
>>               V (.B112_1 <<= (D2_8 Set [(x_43:
>>                  obj) => (~((x_43 E Usc(.thelaw_1(D2_8)))):
>>                  prop)]))
>>               ))]
>>            ,[(Casehyp2_44:that Exists([(U_45:
>>                  obj) => ((U_45 E .B112_1):
>>                  prop)]))
>>               => (Cases(Simp2((dhyp_11 Iff1
>>               (D2_8 Ui Separation4(Refleq(((Misset_1
>>               Mbold2 thelawchooses_1) Set [(Y_51:
>>                  obj) => (cutse2(Misset_1,thelawchooses_1,
>>                  .B112_1,Y_51):prop)]))
>>               )))),[(casehyp1_52:that (D2_8
>>                  <<= prime2(.thelaw_1,.B112_1)))
>>                  => (((.B112_1 <<= prime2(.thelaw_1,
>>                  D2_8)) Add1 (((prime2(.thelaw_1,
>>                  D2_8) <<= D2_8) Fixform (Setsinchains2(Misset_1,
>>                  thelawchooses_1,(Misset_1
>>                  Mboldtheta2 thelawchooses_1),
>>                  Simp1((dhyp_11 Iff1 (D2_8
>>                  Ui Separation4(Refleq(((Misset_1
>>                  Mbold2 thelawchooses_1) Set
>>                  [(Y_58:obj) => (cutse2(Misset_1,
>>                     thelawchooses_1,.B112_1,
>>                     Y_58):prop)]))
>>                  ))))) Sepsub2 Refleq(prime2(.thelaw_1,
>>                  D2_8)))) Transsub casehyp1_52)):
>>                  that ((prime2(.thelaw_1,D2_8)
>>                  <<= prime2(.thelaw_1,.B112_1))
>>                  V (.B112_1 <<= prime2(.thelaw_1,
>>                  D2_8))))]
>>               ,[(casehyp2_60:that (.B112_1
>>                  <<= D2_8)) => (Cases(Excmid((.thelaw_1(D2_8)
>>                  = .thelaw_1(.B112_1))),[(casehypa1_61:
>>                     that (.thelaw_1(D2_8) =
>>                     .thelaw_1(.B112_1))) =>
>>                     (((.B112_1 <<= prime2(.thelaw_1,
>>                     D2_8)) Add1 ((prime2(.thelaw_1,
>>                     D2_8) <<= prime2(.thelaw_1,
>>                     .B112_1)) Fixform (Ug([(G_64:
>>                        obj) => (Ded([(onedir_65:
>>                           that (G_64 E prime2(.thelaw_1,
>>                           D2_8))) => (((G_64
>>                           E prime2(.thelaw_1,
>>                           .B112_1)) Fixform
>>                           ((Cases(((Simp1((dhyp_11
>>                           Iff1 (D2_8 Ui Separation4(Refleq(((Misset_1
>>                           Mbold2 thelawchooses_1)
>>                           Set [(Y_71:obj) =>
>>                              (cutse2(Misset_1,
>>                              thelawchooses_1,
>>                              .B112_1,Y_71):
>>                              prop)]))
>>                           )))) Mp (D2_8 Ui
>>                           Simp1(Simp2(Simp2((Misset_1
>>                           Mboldtheta2 thelawchooses_1))))))
>>                           Mp (prime2(.thelaw_1,
>>                           D2_8) Ui Simp2(Simp2((bhypa3_1
>>                           Iff1 (.B112_1 Ui
>>                           Separation4(Refleq((Misset_1
>>                           Cuts3 thelawchooses_1))))))))),
>>                           [(casehypb1_87:that
>>                              (prime2(.thelaw_1,
>>                              D2_8) <<= .B112_1))
>>                              => ((onedir_65
>>                              Mp (G_64 Ui Simp1(casehypb1_87))):
```

```
>>                              that (G_64 E .B112_1))]
>>                           ,[(casehypb2_90:that
>>                              (.B112_1 <<= prime2(.thelaw_1,
>>                              D2_8))) => (((G_64
>>                              E .B112_1) Giveup
>>                              (Inusc2(.thelaw_1(D2_8))
>>                              Mp Simp2((Eqsymm(casehypa1_61)
>>                              Subs1 ((thelawchooses_1(.B112_1,
>>                              ((Simp1((bhypa3_1
>>                              Iff1 (.B112_1
>>                              Ui ((Misset_1
>>                              Mbold2 thelawchooses_1)
>>                              Separation [(C_94:
>>                                 obj) => (cuts2(Misset_1,
>>                                 thelawchooses_1,
>>                                 C_94):prop)]))
>>                              )) Mp (.B112_1
>>                              Ui Simp1(Simp1(Simp2((Misset_1
>>                              Mboldtheta2 thelawchooses_1))))))
>>                              Iff1 (.B112_1
>>                              Ui Scthm(.M_1))),
>>                              Casehyp2_44) Mp
>>                              (.thelaw_1(.B112_1)
>>                              Ui Simp1(casehypb2_90)))
>>                              Iff1 (.thelaw_1(.B112_1)
>>                              Ui Separation4(Refleq(prime2(.thelaw_1,
>>                              D2_8))))))))):
>>                              that (G_64 E .B112_1))])
>>                           Conj (casehypa1_61
>>                           Subs1 Simp2((onedir_65
>>                           Iff1 (G_64 Ui Separation4(Refleq(prime2(.thelaw_1,
>>                           D2_8)))))))) Iff2
>>                           (G_64 Ui Separation4(Refleq(prime2(.thelaw_1,
>>                           .B112_1)))))):that
>>                           (G_64 E prime2(.thelaw_1,
>>                           .B112_1)))])
>>                        :that ((G_64 E prime2(.thelaw_1,
>>                        D2_8)) -> (G_64 E prime2(.thelaw_1,
>>                        .B112_1))))])
>>                     Conj (Separation3(Refleq(prime2(.thelaw_1,
>>                     D2_8))) Conj Separation3(Refleq(prime2(.thelaw_1,
>>                     .B112_1))))))):that ((prime2(.thelaw_1,
>>                     D2_8) <<= prime2(.thelaw_1,
>>                     .B112_1)) V (.B112_1 <<=
>>                     prime2(.thelaw_1,D2_8))))]
>>                  ,[(casehypa2_123:that ~((.thelaw_1(D2_8)
>>                     = .thelaw_1(.B112_1))))
>>                     => (((prime2(.thelaw_1,
>>                     D2_8) <<= prime2(.thelaw_1,
>>                     .B112_1)) Add2 ((.B112_1
>>                     <<= prime2(.thelaw_1,D2_8))
>>                     Fixform (Ug([(G_126:obj)
>>                        => (Ded([(otherdir_127:
>>                           that (G_126 E .B112_1))
>>                           => (((G_126 E prime2(.thelaw_1,
>>                           D2_8)) Fixform (((otherdir_127
>>                           Mp (G_126 Ui Simp1(casehyp2_60)))
>>                           Conj Negintro([(eqhyp2_130:
>>                              that (G_126 E
>>                              Usc(.thelaw_1(D2_8))))
>>                              => ((Refleq(.thelaw_1(D2_8))
>>                              Mp ((casehyp2_60
>>                              Antisymsub ((D2_8
>>                              <<= .B112_1) Fixform
>>                              (Ug([(H_133:obj)
>>                                 => (Ded([(hhyp_134:
>>                                    that (H_133
>>                                    E D2_8))
>>                                    => (Cases(Excmid((H_133
>>                                    = .thelaw_1(D2_8))),
>>                                    [(casehhyp1_135:
>>                                       that
>>                                       (H_133
>>                                       = .thelaw_1(D2_8)))
>>                                       => ((Eqsymm(casehhyp1_135)
>>                                       Subs1
>>                                        (Oridem((eqhyp2_130
>>                                        Iff1
>>                                        (G_126
>>                                        Ui (.thelaw_1(D2_8)
>>                                        Pair
>>                                        .thelaw_1(D2_8)))))
>>                                        Subs1
>>                                        otherdir_127)):
>>                                        that
>>                                        (H_133
>>                                        E .B112_1))]
>>                                     ,[(casehhyp2_139:
>>                                        that
>>                                        ~((H_133
>>                                        = .thelaw_1(D2_8))))
>>                                        => ((((H_133
>>                                        E prime2(.thelaw_1,
>>                                        D2_8))
>>                                        Fixform
>>                                        ((hhyp_134
>>                                        Conj
>>                                        Negintro([(sillyhyp_140:
>>                                           that (H_133
>>                                           E Usc(.thelaw_1(D2_8))))
>>                                           => ((Oridem((sillyhyp_140
>>                                           Iff1 (H_133
>>                                           Ui (.thelaw_1(D2_8)
>>                                           Pair .thelaw_1(D2_8)))))
>>                                           Mp casehhyp2_139):
>>                                           that ??)]))
>>                                        Iff2
>>                                        (H_133
>>                                        Ui Separation4(Refleq(prime2(.thelaw_1,
>>                                        D2_8))))))
>>                                        Mp (H_133
>>                                        Ui Simp1((((Simp1((dhyp_11
>>                                        Iff1
>>                                        (D2_8
>>                                       Ui Separation4(Refleq(((Misset_1
>>                                       Mbold2
>>                                       thelawchooses_1)
>>                                       Set [(Y_153:
>>                                          obj) =>
>>                                          (cutse2(Misset_1,
>>                                          thelawchooses_1,
>>                                          .B112_1,
>>                                          Y_153):
>>                                          prop)]))
>>                                       ))))
>>                                       Mp (D2_8
>>                                       Ui Simp1(Simp2(Simp2((Misset_1
>>                                       Mboldtheta2
>>                                       thelawchooses_1))))))
>>                                       Mp (prime2(.thelaw_1,
>>                                       D2_8)
>>                                       Ui Simp2(Simp2((bhypa3_1
>>                                       Iff1
>>                                       (.B112_1
>>                                       Ui Separation4(Refleq((Misset_1
>>                                       Cuts3
>>                                       thelawchooses_1))))))))
>>                                       Ds1 Negintro([(impossiblesub_169:
>>                                          that (.B112_1
>>                                          <<= prime2(.thelaw_1,
>>                                          D2_8)))
>>                                          => ((Inusc2(.thelaw_1(D2_8))
>>                                          Mp Simp2((((Oridem((eqhyp2_130
>>                                          Iff1 (G_126
>>                                          Ui (.thelaw_1(D2_8)
>>                                          Pair .thelaw_1(D2_8)))))
>>                                          Subs1 otherdir_127)
>>                                          Mp (.thelaw_1(D2_8)
>>                                          Ui Simp1(impossiblesub_169)))
>>                                          Iff1 (.thelaw_1(D2_8)
>>                                          Ui Separation4(Refleq(prime2(.thelaw_1
>>                                          D2_8))))))):
```

```
>>                                          that ??)]))
>>                                       ))):that
>>                                       (H_133
>>                                       E .B112_1))])
>>                                    :that (H_133
>>                                    E .B112_1))])
>>                                 :that ((H_133
>>                                 E D2_8) ->
>>                                 (H_133 E .B112_1)))])
>>                              Conj (Simp2(Simp2(casehyp2_60))
>>                              Conj Setsinchains2(Misset_1,
>>                              thelawchooses_1,
>>                              (Misset_1 Mboldtheta2
>>                              thelawchooses_1),
>>                              Simp1((bhypa3_1
>>                              Iff1 (.B112_1
>>                              Ui ((Misset_1
>>                              Mbold2 thelawchooses_1)
>>                              Separation [(C_180:
>>                                 obj) => (cuts2(Misset_1,
>>                                 thelawchooses_1,
>>                                 C_180):prop)]))
>>                              ))))))) Subs1
>>                              casehypa2_123)):
>>                              that ??)]))
>>                           Iff2 (G_126 Ui Separation4(Refleq(prime2(.thelaw_1,
>>                           D2_8)))))):that (G_126
>>                           E prime2(.thelaw_1,
>>                           D2_8)))])
>>                        :that ((G_126 E .B112_1)
>>                        -> (G_126 E prime2(.thelaw_1,
>>                        D2_8))))])
>>                     Conj (Setsinchains2(Misset_1,
>>                     thelawchooses_1,(Misset_1
>>                     Mboldtheta2 thelawchooses_1),
>>                     Simp1((bhypa3_1 Iff1 (.B112_1
>>                     Ui ((Misset_1 Mbold2 thelawchooses_1)
>>                     Separation [(C_189:obj)
>>                        => (cuts2(Misset_1,thelawchooses_1,
>>                        C_189):prop)]))
>>                     ))) Conj Separation3(Refleq(prime2(.thelaw_1,
>>                     D2_8))))))):that ((prime2(.thelaw_1,
>>                     D2_8) <<= prime2(.thelaw_1,
>>                     .B112_1)) V (.B112_1 <<=
>>                     prime2(.thelaw_1,D2_8))))])
>>                  :that ((prime2(.thelaw_1,D2_8)
>>                  <<= prime2(.thelaw_1,.B112_1))
>>                  V (.B112_1 <<= prime2(.thelaw_1,
>>                  D2_8))))])
>>               :that ((prime2(.thelaw_1,D2_8)
>>               <<= prime2(.thelaw_1,.B112_1))
>>               V (.B112_1 <<= prime2(.thelaw_1,
>>               D2_8))))]))
>>            Iff2 (prime2(.thelaw_1,D2_8) Ui
>>            Separation4(Refleq(((Misset_1 Mbold2
>>            thelawchooses_1) Set [(Y_197:obj)
>>               => (cutse2(Misset_1,thelawchooses_1,
>>               .B112_1,Y_197):prop)]))
>>            )))):that (prime2(.thelaw_1,D2_8)
>>            E ((Misset_1 Mbold2 thelawchooses_1)
>>            Set [(Y_198:obj) => (cutse2(Misset_1,
>>               thelawchooses_1,.B112_1,Y_198):
>>               prop)]))
>>            )])
>>         :that ((D2_8 E ((Misset_1 Mbold2 thelawchooses_1)
>>         Set [(Y_199:obj) => (cutse2(Misset_1,
>>            thelawchooses_1,.B112_1,Y_199):prop)]))
>>         -> (prime2(.thelaw_1,D2_8) E ((Misset_1
>>         Mbold2 thelawchooses_1) Set [(Y_200:
>>            obj) => (cutse2(Misset_1,thelawchooses_1,
>>            .B112_1,Y_200):prop)]))
>>         ))])
>>      :that Forall([(D2_201:obj) => (((D2_201
>>         E ((Misset_1 Mbold2 thelawchooses_1)
>>         Set [(Y_202:obj) => (cutse2(Misset_1,
>>            thelawchooses_1,.B112_1,Y_202):prop)]))
>>         -> (prime2(.thelaw_1,D2_201) E ((Misset_1
>>         Mbold2 thelawchooses_1) Set [(Y_203:
>>            obj) => (cutse2(Misset_1,thelawchooses_1,
>>            .B112_1,Y_203):prop)]))
>>         ):prop)]))
>>      ]
>>   {move 0}
```

```
open

   define linead78 bhypa2: linee78 Misset \
      thelawchooses, bhypa2

>>    linead78: [(.B111_1:obj),(bhypa2_1:that
>>         (.B111_1 E Cuts)) => (---:that Forall([(D2_2:
>>            obj) => (((D2_2 E ((Misset Mbold2
>>            thelawchooses) Set [(Y_3:obj) =>
>>               (cutse2(Misset,thelawchooses,
>>               .B111_1,Y_3):prop)]))
>>            -> (prime2(thelaw,D2_2) E ((Misset
>>            Mbold2 thelawchooses) Set [(Y_4:
>>               obj) => (cutse2(Misset,thelawchooses,
>>               .B111_1,Y_4):prop)]))
>>            ):prop)]))
>>         ]
>>      {move 1}
```

```
   open

      define lineac78 bhypa1: linead78 bhypa1


>>       lineac78: [(.B_1:obj),(bhypa1_1:that
>>            (.B_1 E Cuts)) => (---:that Forall([(D2_2:
```

```
>>               obj) => (((D2_2 E ((Misset Mbold2
>>               thelawchooses) Set [(Y_3:obj)
>>                  => (cutse2(Misset,thelawchooses,
>>                  .B_1,Y_3):prop)]))
>>               -> (prime2(thelaw,D2_2) E ((Misset
>>               Mbold2 thelawchooses) Set [(Y_4:
>>                  obj) => (cutse2(Misset,thelawchooses,
>>                  .B_1,Y_4):prop)]))
>>               ):prop)]))
>>            ]
>>         {move 2}



      open

         define lineab78 bhyp: lineac78 bhyp


>>          lineab78: [(bhyp_1:that (B E Cuts))
>>               => (---:that Forall([(D2_2:obj)
>>                  => (((D2_2 E ((Misset Mbold2
>>                  thelawchooses) Set [(Y_3:obj)
>>                     => (cutse2(Misset,thelawchooses,
>>                     B,Y_3):prop)]))
>>                  -> (prime2(thelaw,D2_2) E
>>                  ((Misset Mbold2 thelawchooses)
>>                  Set [(Y_4:obj) => (cutse2(Misset,
>>                     thelawchooses,B,Y_4):prop)]))
>>                  ):prop)]))
>>               ]
>>            {move 3}



         open

            define line78: lineab78 bhyp
```

```
>>             line78: [(---:that Forall([(D2_1:
>>                     obj) => (((D2_1 E ((Misset
>>                     Mbold2 thelawchooses) Set
>>                     [(Y_2:obj) => (cutse2(Misset,
>>                        thelawchooses,B,Y_2):
>>                        prop)]))
>>                     -> (prime2(thelaw,D2_1)
>>                     E ((Misset Mbold2 thelawchooses)
>>                     Set [(Y_3:obj) => (cutse2(Misset,
>>                        thelawchooses,B,Y_3):
>>                        prop)]))
>>                     ):prop)]))
>>                  ]
>>               {move 4}
```

This is the third component of the proof that `Cuts2` is a Θ-chain. I want to examine the proof strategy; I also want to see if the size of the term and the slowness of generation of the term can be improved by exporting some intermediate stages to move 0.

`Lestrade execution:`

```
            goal that Forall[D1 => Forall[F1 \
                     => ((D1 <<= Cuts2) & F1 \
                     E D1) -> (D1 Intersection \
                     F1) E Cuts2] \
                  ] \
```

```
>>             Goal: that Forall([(D1_677:obj)
>>                  => (Forall([(F1_678:obj) =>
>>                     ((((D1_677 <<= Cuts2) &
>>                     (F1_678 E D1_677)) -> ((D1_677
>>                     Intersection F1_678) E
>>                     Cuts2)):prop)])
>>                  :prop)])
>>

            open

               declare D2 obj

>>                D2: obj {move 6}



               open

                  declare F2 obj

>>                   F2: obj {move 7}



                  open

                     declare intev that (D2 \
                        <<= Cuts2) & F2 E D2


>>                      intev: that ((D2 <<=
>>                        Cuts2) & (F2 E D2))
>>                        {move 8}



                     goal that (D2 Intersection \
                        F2) E Cuts2
```

```
>>                      Goal: that ((D2 Intersection
>>                        F2) E Cuts2)

                     define line79 : Ui D2 \
                        Intersection F2, Separation4 \
                        Refleq Cuts2

>>                      line79: [(---:that (((D2
>>                           Intersection F2)
>>                           E (Mbold Set cutsi2))
>>                           == (((D2 Intersection
>>                           F2) E Mbold) & cutsi2((D2
>>                           Intersection F2)))))]
>>                        {move 7}



                     goal that (D2 Intersection \
                        F2) E Mbold

>>                      Goal: that ((D2 Intersection
>>                        F2) E Mbold)

                     define line80 :Ui F2, \
                        Ui D2, Simp2(Simp2(Simp2 \
                        Mboldtheta))

>>                      line80: [(---:that (((D2
>>                           <<= (Misset Mbold2
>>                           thelawchooses)) &
>>                           (F2 E D2)) -> ((D2
>>                           Intersection F2)
>>                           E (Misset Mbold2
>>                           thelawchooses))))]
>>                        {move 7}
```

```
                    define line81 intev: \
                       Mp(Conj(Transsub(Simp1 \
                       intev,line20), Simp2 \
                       intev), line80)

>>                     line81: [(intev_1:that
>>                          ((D2 <<= Cuts2) &
>>                          (F2 E D2))) => (---:
>>                          that ((D2 Intersection
>>                          F2) E (Misset Mbold2
>>                          thelawchooses)))]
>>                       {move 7}



                    goal that ((D2 Intersection \
                       F2) <<= prime B) V \
                       B <<= D2 Intersection \
                       F2

>>                     Goal: that (((D2 Intersection
>>                       F2) <<= prime(B)) V
>>                       (B <<= (D2 Intersection
>>                       F2)))

                    declare K obj

>>                     K: obj {move 8}



                    define line82: Excmid \
                       Forall [K=> (K E D2) \
                          -> B<<=K] \
>>                      line82: [(---:that (Forall([(K_1:
>>                              obj) => (((K_1
>>                              E D2) -> (B <<=
>>                              K_1)):prop)])
>>                           V ~(Forall([(K_1:
>>                              obj) => (((K_1
>>                              E D2) -> (B <<=
>>                              K_1)):prop)]))
>>                           ))]
>>                        {move 7}



                     open

                        goal that ((D2 Intersection \
                           F2) <<= prime B) \
                           V B <<= D2 Intersection \
                           F2

>>                         Goal: that (((D2
>>                           Intersection F2)
>>                           <<= prime(B)) V (B
>>                           <<= (D2 Intersection
>>                           F2)))

                        declare K1 obj

>>                         K1: obj {move 9}



                        declare casehyp1 \
                           that Forall [K1=> \
                              (K1 E D2) \
                              -> B<<=K1] \
>>                         casehyp1: that Forall([(K1_1:
>>                              obj) => (((K1_1
>>                              E D2) -> (B <<=
>>                              K1_1)):prop)])
>>                           {move 9}


                        goal that B <<= D2 \
                           Intersection F2


>>                         Goal: that (B <<=
>>                           (D2 Intersection
>>                           F2))

                        open

                           declare K2 obj


>>                            K2: obj {move
>>                              10}



                           open

                              declare \
                                 khyp that K2 \
                                 E B

>>                               khyp: that
>>                                 (K2 E B) {move
>>                                 11}
```

```
                              open

                                 declare \
                                    B2 obj


>>                                  B2: obj
>>                                    {move 12}



                                 open

                                    declare \
                                       bhyp2 \
                                       that \
                                       B2 E \
                                       D2

>>                                     bhyp2:
>>                                       that
>>                                       (B2 E
>>                                       D2) {move
>>                                       13}



                                    define \
                                       line83 \
                                       bhyp2: \
                                       Mpsubs \
                                       (khyp, \
                                       Mp(bhyp2, \
                                       Ui B2, \
                                       casehyp1))
```

```
>>                                     line83:
>>                                       [(bhyp2_1:
>>                                          that (B2
>>                                          E D2))
>>                                          => (---:
>>                                          that (K2
>>                                          E B2))]
>>                                       {move
>>                                       12}
```

```
                                    close
```

```
                                 define line84 \
                                    B2: Ded \
                                    line83
```

```
>>                                  line84:
>>                                    [(B2_1:obj)
>>                                       => (---:
>>                                       that
>>                                       ((B2_1
>>                                       E D2)
>>                                       -> (K2
>>                                       E B2_1)))]
>>                                    {move 11}
```

```
                                 close
```

```
                              define line85 \
                                 khyp: Ug line84
```

```
>>                               line85: [(khyp_1:
>>                                    that (K2
>>                                    E B)) =>
>>                                    (---:that
>>                                    Forall([(B2_6:
>>                                       obj)
>>                                       => (((B2_6
>>                                       E D2)
>>                                       -> (K2
>>                                       E B2_6)):
>>                                       prop)]))
>>                                    ]
>>                                 {move 10}
```

```
define line86 \
   khyp: Mp(Simp2 \
   intev, \
   Ui F2, line85 \
   khyp)
```

```
>>                               line86: [(khyp_1:
>>                                    that (K2
>>                                    E B)) =>
>>                                    (---:that
>>                                    (K2 E F2))]
>>                                 {move 10}
```

```
define line87 \
   khyp: Fixform(K2 \
   E D2 Intersection \
   F2,Iff2(Conj(line86 \
   khyp, line85 \
   khyp), \
                                 Ui K2, Separation4 \
                                 Refleq (D2 \
                                 Intersection \
                                 F2)))

>>                               line87: [(khyp_1:
>>                                    that (K2
>>                                    E B)) =>
>>                                    (---:that
>>                                    (K2 E (D2
>>                                    Intersection
>>                                    F2)))]
>>                                 {move 10}



                              close

                           define line88 \
                              K2: Ded line87


>>                            line88: [(K2_1:
>>                                 obj) => (---:
>>                                 that ((K2_1
>>                                 E B) -> (K2_1
>>                                 E (D2 Intersection
>>                                 F2))))]
>>                              {move 9}



                           close

                        define line89 \
                           casehyp1: \
                           Fixform(B <<= \
                           D2 Intersection F2, \
                           Conj(Ug line88, \
                           Conj(linea14 bhyp, \
                           Separation3 Refleq \
                           (D2 Intersection \
                           F2))))
```

```
>>                         line89: [(casehyp1_1:
>>                              that Forall([(K1_2:
>>                                 obj) => (((K1_2
>>                                 E D2) -> (B
>>                                 <<= K1_2)):
>>                                 prop)]))
>>                              => (---:that (B
>>                              <<= (D2 Intersection
>>                              F2)))]
>>                           {move 8}
```

```
                        define line90 \
                           casehyp1: \
                           Add2((D2 Intersection \
                           F2) <<= prime B, \
                           line89 casehyp1)
```

```
>>                         line90: [(casehyp1_1:
>>                              that Forall([(K1_2:
>>                                 obj) => (((K1_2
>>                                 E D2) -> (B
>>                                 <<= K1_2)):
>>                                 prop)]))
>>                              => (---:that (((D2
>>                              Intersection F2)
>>                              <<= prime(B))
>>                              V (B <<= (D2 Intersection
>>                              F2))))]
>>                           {move 8}
```

```
                        declare casehyp2 \
                           that ~(Forall [K1=> \
                              (K1 E D2) -> \
                              B<<=K1]) \


>>                         casehyp2: that ~(Forall([(K1_1:
>>                              obj) => (((K1_1
>>                              E D2) -> (B <<=
>>                              K1_1)):prop)]))
>>                           {move 9}


                        goal that ((D2 Intersection \
                           F2) <<= prime B)


>>                         Goal: that ((D2 Intersection
>>                           F2) <<= prime(B))


                        open

                           declare K2 obj


>>                            K2: obj {move
>>                              10}


                           open
```

```
                              declare \
                                 khyp2 that \
                                 K2 E D2 Intersection \
                                 F2

>>                               khyp2: that
>>                                 (K2 E (D2 Intersection
>>                                 F2)) {move
>>                                 11}



                              define line91: \
                                 Counterexample \
                                 casehyp2

>>                               line91: [(---:
>>                                    that Exists([(z_2:
>>                                       obj)
>>                                       => (~(((z_2
>>                                       E D2)
>>                                       -> (B
>>                                       <<= z_2))):
>>                                       prop)]))
>>                                    ]
>>                                 {move 10}



                              open

                                 declare \
                                    F3 obj


>>                                  F3: obj
>>                                    {move 12}
```

```
                                 declare \
                                    fhyp3 that \
                                    Witnesses \
                                    line91 \
                                    F3

>>                                  fhyp3: that
>>                                    (line91
>>                                    Witnesses
>>                                    F3) {move
>>                                    12}



                                 define line92 \
                                    fhyp3: \
                                    Notimp2 \
                                    fhyp3

>>                                  line92:
>>                                    [(.F3_1:
>>                                       obj),
>>                                       (fhyp3_1:
>>                                       that
>>                                       (line91
>>                                       Witnesses
>>                                       .F3_1))
>>                                       => (---:
>>                                       that
>>                                       (.F3_1
>>                                       E D2))]
>>                                    {move 11}
```

```
                                 define line93 \
                                    fhyp3 : \
                                    Notimp1 \
                                    fhyp3

>>                                  line93:
>>                                    [(.F3_1:
>>                                       obj),
>>                                       (fhyp3_1:
>>                                       that
>>                                       (line91
>>                                       Witnesses
>>                                       .F3_1))
>>                                       => (---:
>>                                       that
>>                                       ~((B
>>                                       <<= .F3_1)))]
>>                                    {move 11}



                                 define line94 \
                                    fhyp3: \
                                    Simp2(Iff1(Mpsubs(line92 \
                                    fhyp3, \
                                    Simp1 intev), \
                                    Ui F3, \
                                    Separation4 \
                                    Refleq \
                                    Cuts2))


>>                                  line94:
>>                                    [(.F3_1:
>>                                       obj),
>>                                       (fhyp3_1:
>>                                       that
>>                                       (line91
>>                                        Witnesses
>>                                        .F3_1))
>>                                        => (---:
>>                                        that
>>                                        cutsi2(.F3_1))]
>>                                     {move 11}
```

```
define line95 \
   fhyp3: \
   Ds1(line94 \
   fhyp3, \
   line93 \
   fhyp3)
```

```
>>                                   line95:
>>                                     [(.F3_1:
>>                                        obj),
>>                                        (fhyp3_1:
>>                                        that
>>                                        (line91
>>                                        Witnesses
>>                                        .F3_1))
>>                                        => (---:
>>                                        that
>>                                        (.F3_1
>>                                        <<= prime2(thelaw,
>>                                        B)))]
>>                                     {move 11}
```

```
define line96 \
   fhyp3: \
   Mp line92 \
   fhyp3, \
                                    Ui F3, \
                                    Simp2(Iff1 \
                                    khyp2, \
                                    Ui K2, \
                                    Separation4 \
                                    Refleq \
                                    (D2 Intersection \
                                    F2))

>>                                  line96:
>>                                    [(.F3_1:
>>                                       obj),
>>                                       (fhyp3_1:
>>                                       that
>>                                       (line91
>>                                       Witnesses
>>                                       .F3_1))
>>                                       => (---:
>>                                       that
>>                                       (K2 E
>>                                       .F3_1))]
>>                                    {move 11}



                                 define line97 \
                                    fhyp3: \
                                    Mpsubs \
                                    line96 \
                                    fhyp3 line95 \
                                    fhyp3

>>                                  line97:
>>                                    [(.F3_1:
>>                                       obj),
>>                                       (fhyp3_1:
>>                                       that
>>                                       (line91
>>                                       Witnesses
>>                                       .F3_1))
>>                                       => (---:
>>                                       that
>>                                       (K2 E
>>                                       prime2(thelaw,
>>                                       B)))]
>>                                    {move 11}


                                 close

                              define line98 \
                                 khyp2 : Eg \
                                 line91 line97


>>                               line98: [(khyp2_1:
>>                                    that (K2
>>                                    E (D2 Intersection
>>                                    F2))) =>
>>                                    (---:that
>>                                    (K2 E prime2(thelaw,
>>                                    B)))]
>>                                 {move 10}


                              close

                           define line99 \
                              K2: Ded line98


>>                            line99: [(K2_1:
>>                                 obj) => (---:
>>                                 that ((K2_1
>>                                 E (D2 Intersection
>>                                 F2)) -> (K2_1
>>                                 E prime2(thelaw,
>>                                 B))))]
>>                              {move 9}



                           close

                        define line100 casehyp2: \
                           Fixform((D2 Intersection \
                           F2) <<= prime B, \
                           Conj(Ug line99, \
                           Conj(Separation3 \
                           Refleq(D2 Intersection \
                           F2),Separation3 \
                           Refleq (prime B))))


>>                      line100: [(casehyp2_1:
>>                           that ~(Forall([(K1_2:
>>                              obj) => (((K1_2
>>                              E D2) -> (B
>>                              <<= K1_2)):
>>                              prop)]))
>>                           ) => (---:that
>>                           ((D2 Intersection
>>                           F2) <<= prime(B)))]
>>                        {move 8}



                        define line101 casehyp2: \
                           Add1(B <<= D2 Intersection \
                           F2,line100 casehyp2)
```

```
>>                         line101: [(casehyp2_1:
>>                              that ~(Forall([(K1_2:
>>                                 obj) => (((K1_2
>>                                 E D2) -> (B
>>                                 <<= K1_2)):
>>                                 prop)]))
>>                              ) => (---:that
>>                              (((D2 Intersection
>>                              F2) <<= prime(B))
>>                              V (B <<= (D2 Intersection
>>                              F2))))]
>>                           {move 8}



                        close

                     define line102 intev: \
                        Cases line82 line90, \
                        line101

>>                      line102: [(intev_1:that
>>                           ((D2 <<= Cuts2) &
>>                           (F2 E D2))) => (---:
>>                           that (((D2 Intersection
>>                           F2) <<= prime(B))
>>                           V (B <<= (D2 Intersection
>>                           F2))))]
>>                        {move 7}



                     define linea102 intev: \
                        Conj(line81 intev, \
                        line102 intev)

>>                      linea102: [(intev_1:
>>                           that ((D2 <<= Cuts2)
>>                           & (F2 E D2))) =>
>>                           (---:that (((D2 Intersection
>>                           F2) E (Misset Mbold2
>>                           thelawchooses)) &
>>                           (((D2 Intersection
>>                           F2) <<= prime(B))
>>                           V (B <<= (D2 Intersection
>>                           F2)))))]
>>                        {move 7}



                     define lineb102 intev: \
                        Fixform((D2 IntersectionF2) \
                        E Cuts2, Iff2(linea102 \
                        intev,Ui (D2 Intersection \
                        F2,Separation4 Refleq \
                        Cuts2)))

>>                      lineb102: [(intev_1:
>>                           that ((D2 <<= Cuts2)
>>                           & (F2 E D2))) =>
>>                           (---:that ((D2 Intersection
>>                           F2) E Cuts2))]
>>                        {move 7}



                     close

                  define line103 F2: Ded \
                     lineb102

>>                   line103: [(F2_1:obj) =>
>>                        (---:that (((D2 <<=
>>                        Cuts2) & (F2_1 E D2))
>>                        -> ((D2 Intersection
>>                        F2_1) E Cuts2)))]
>>                     {move 6}
```

```
                  close

               define line104 D2: Ug line103
```

```
>>                line104: [(D2_1:obj) => (---:
>>                     that Forall([(F2_82:obj)
>>                        => ((((D2_1 <<= Cuts2)
>>                        & (F2_82 E D2_1)) ->
>>                        ((D2_1 Intersection
>>                        F2_82) E Cuts2)):prop)]))
>>                     ]
>>                  {move 5}
```

```
               close

            define line105: Ug line104
```

```
>>             line105: [(---:that Forall([(D2_83:
>>                     obj) => (Forall([(F2_84:
>>                        obj) => ((((D2_83 <<=
>>                        Cuts2) & (F2_84 E D2_83))
>>                        -> ((D2_83 Intersection
>>                        F2_84) E Cuts2)):prop)])
>>                     :prop)]))
>>                  ]
>>               {move 4}
```

This is the fourth component of the proof that Cuts is a Θ-chain. I wonder whether this has common features with the fourth component of the larger proof which can be used to shorten the file. This also might be worth exporting to move 0.

```
Lestrade execution:



            close

         define line107 bhyp: Fixform(thetachain \
            Cuts2,Conj(line19,Conj(line21, \
            Conj(line78,line105))))

>>          line107: [(bhyp_1:that (B E Cuts))
>>               => (---:that thetachain((Mbold
>>               Set [(Y_175:obj) => (cutsh2(Y_175):
>>                  prop)]))
>>               )]
>>            {move 3}



         save

         close

      declare bhyp100 that B E Cuts

>>       bhyp100: that (B E Cuts) {move 3}



      define linea107 bhyp100 : line107 bhyp100


>>       linea107: [(.B_1:obj),(bhyp100_1:that
>>            (.B_1 E Cuts)) => (---:that thetachain((Mbold
>>            Set [(Y_186:obj) => ((.B_1 cutsg2
>>               Y_186):prop)]))
>>            )]
>>         {move 2}



      save

      close

   declare B101 obj

>>    B101: obj {move 2}



   declare bhyp101 that B101 E Cuts

>>    bhyp101: that (B101 E Cuts) {move 2}



   define lineb107 bhyp101: linea107 bhyp101


>>    lineb107: [(.B101_1:obj),(bhyp101_1:that
>>         (.B101_1 E Cuts)) => (---:that thetachain((Mbold
>>         Set [(Y_186:obj) => ((.B101_1 cutsf2
>>            Y_186):prop)]))
>>         )]
>>      {move 1}



   save

   close

declare B102 obj
```

```
>> B102: obj {move 1}



declare bhyp102 that B102 E Cuts

>> bhyp102: that (B102 E Cuts) {move 1}



define linec107 bhyp102 :lineb107 bhyp102


>> linec107: [(.B102_1:obj),(.M_1:obj),(.Misset_1:
>>      that Isset(.M_1)),(.thelaw_1:[(S_2:obj)
>>         => (---:obj)]),
>>      (.thelawchooses_1:[(.S_3:obj),(subsetev_3:
>>         that (.S_3 <<= .M_1)),(inev_3:that
>>         Exists([(x_4:obj) => ((x_4 E .S_3):
>>            prop)]))
>>         => (---:that (.thelaw_1(.S_3) E .S_3))]),
>>      (bhyp102_1:that (.B102_1 E (.Misset_1
>>      Cuts3 .thelawchooses_1))) => ((thetachain1(.M_1,
>>      .thelaw_1,((.Misset_1 Mbold2 .thelawchooses_1)
>>      Set [(Y_5:obj) => (cutse2(.Misset_1,.thelawchooses_1,
>>         .B102_1,Y_5):prop)]))
>>      Fixform (((.M_1 E ((.Misset_1 Mbold2 .thelawchooses_1)
>>      Set [(Y_7:obj) => (cutse2(.Misset_1,.thelawchooses_1,
>>         .B102_1,Y_7):prop)]))
>>      Fixform ((Simp1((.Misset_1 Mboldtheta2
>>      .thelawchooses_1)) Conj ((.M_1 <<= prime2(.thelaw_1,
>>      .B102_1)) Add2 ((Simp1((bhyp102_1 Iff1
>>      (.B102_1 Ui ((.Misset_1 Mbold2 .thelawchooses_1)
>>      Separation [(C_13:obj) => (cuts2(.Misset_1,
>>         .thelawchooses_1,C_13):prop)]))
>>      )) Mp (.B102_1 Ui Simp1(Simp1(Simp2((.Misset_1
>>      Mboldtheta2 .thelawchooses_1)))))) Iff1
>>      (.B102_1 Ui Scthm(.M_1))))) Iff2 (.M_1
>>      Ui Separation4(Refleq(((.Misset_1 Mbold2
>>      .thelawchooses_1) Set [(Y_28:obj) => (cutse2(.Misset_1,
>>         .thelawchooses_1,.B102_1,Y_28):prop)]))
>>      )))) Conj ((((((.Misset_1 Mbold2 .thelawchooses_1)
>>      Set [(Y_39:obj) => (cutse2(.Misset_1,.thelawchooses_1,
>>         .B102_1,Y_39):prop)])
>>      <<= (.Misset_1 Mbold2 .thelawchooses_1))
>>      Fixform (Separation3(Refleq((.Misset_1
>>      Mbold2 .thelawchooses_1))) Sepsub2 Refleq(((.Misset_1
>>      Mbold2 .thelawchooses_1) Set [(Y_47:obj)
>>         => (cutse2(.Misset_1,.thelawchooses_1,
>>         .B102_1,Y_47):prop)]))
>>      )) Transsub Simp1(Simp2((.Misset_1 Mboldtheta2
>>      .thelawchooses_1)))) Conj (linee78(.Misset_1,
>>      .thelawchooses_1,bhyp102_1) Conj Ug([(D2_72:
>>         obj) => (Ug([(F2_76:obj) => (Ded([(intev_79:
>>               that ((D2_72 <<= ((.Misset_1
>>               Mbold2 .thelawchooses_1) Set
>>               [(Y_80:obj) => (cutse2(.Misset_1,
>>                  .thelawchooses_1,.B102_1,Y_80):
>>                  prop)]))
>>               & (F2_76 E D2_72))) => (((D2_72
>>               Intersection F2_76) E ((.Misset_1
>>               Mbold2 .thelawchooses_1) Set
>>               [(Y_81:obj) => (cutse2(.Misset_1,
>>                  .thelawchooses_1,.B102_1,Y_81):
>>                  prop)]))
>>               Fixform (((((Simp1(intev_79)
>>               Transsub ((((.Misset_1 Mbold2
>>               .thelawchooses_1) Set [(Y_84:
>>                  obj) => (cutse2(.Misset_1,
>>                  .thelawchooses_1,.B102_1,Y_84):
>>                  prop)])
>>               <<= (.Misset_1 Mbold2 .thelawchooses_1))
>>               Fixform (Separation3(Refleq((.Misset_1
>>               Mbold2 .thelawchooses_1))) Sepsub2
>>               Refleq(((.Misset_1 Mbold2 .thelawchooses_1)
>>               Set [(Y_92:obj) => (cutse2(.Misset_1,
>>                  .thelawchooses_1,.B102_1,Y_92):
>>                  prop)]))
>>               ))) Conj Simp2(intev_79)) Mp
>>               (F2_76 Ui (D2_72 Ui Simp2(Simp2(Simp2((.Misset_1
>>               Mboldtheta2 .thelawchooses_1)))))))
>>               Conj Cases(Excmid(Forall([(K_108:
>>                  obj) => (((K_108 E D2_72)
>>                  -> (.B102_1 <<= K_108)):prop)]))
>>               ,[(casehyp1_109:that Forall([(K1_110:
>>                     obj) => (((K1_110 E D2_72)
>>                     -> (.B102_1 <<= K1_110)):
>>                     prop)]))
>>                  => ((((D2_72 Intersection
>>                  F2_76) <<= prime2(.thelaw_1,
>>                  .B102_1)) Add2 ((.B102_1 <<=
>>                  (D2_72 Intersection F2_76))
>>                  Fixform (Ug([(K2_113:obj)
>>                     => (Ded([(khyp_114:that
>>                        (K2_113 E .B102_1))
>>                        => (((K2_113 E (D2_72
>>                        Intersection F2_76))
>>                        Fixform (((Simp2(intev_79)
>>                        Mp (F2_76 Ui Ug([(B2_119:
>>                           obj) => (Ded([(bhyp2_120:
>>                              that (B2_119 E
>>                              D2_72)) => ((khyp_114
>>                              Mpsubs (bhyp2_120
>>                              Mp (B2_119 Ui
>>                              casehyp1_109))):
>>                              that (K2_113 E
>>                              B2_119))])
>>                           :that ((B2_119 E
>>                           D2_72) -> (K2_113
>>                           E B2_119)))]))
>>                        ) Conj Ug([(B2_124:obj)
>>                           => (Ded([(bhyp2_125:
>>                              that (B2_124 E
>>                              D2_72)) => ((khyp_114
>>                              Mpsubs (bhyp2_125
>>                              Mp (B2_124 Ui
>>                              casehyp1_109))):
>>                              that (K2_113 E
>>                              B2_124))])
>>                           :that ((B2_124 E
>>                           D2_72) -> (K2_113
>>                           E B2_124)))]))
>>                        Iff2 (K2_113 Ui Separation4(Refleq((D2_72
>>                        Intersection F2_76)))))):
>>                        that (K2_113 E (D2_72
>>                        Intersection F2_76)))])
>>                     :that ((K2_113 E .B102_1)
>>                     -> (K2_113 E (D2_72 Intersection
>>                     F2_76))))])
>>                  Conj (Setsinchains2(.Misset_1,
>>                  .thelawchooses_1,(.Misset_1
>>                  Mboldtheta2 .thelawchooses_1),
>>                  Simp1((bhyp102_1 Iff1 (.B102_1
>>                  Ui ((.Misset_1 Mbold2 .thelawchooses_1)
>>                  Separation [(C_139:obj) =>
>>                     (cuts2(.Misset_1,.thelawchooses_1,
>>                     C_139):prop)]))
>>                  ))) Conj Separation3(Refleq((D2_72
>>                  Intersection F2_76))))))):
>>                  that (((D2_72 Intersection
>>                  F2_76) <<= prime2(.thelaw_1,
>>                  .B102_1)) V (.B102_1 <<= (D2_72
>>                  Intersection F2_76))))]
>>               ,[(casehyp2_144:that ~(Forall([(K1_145:
>>                     obj) => (((K1_145 E D2_72)
>>                     -> (.B102_1 <<= K1_145)):
>>                     prop)]))
>>                  ) => (((.B102_1 <<= (D2_72
>>                  Intersection F2_76)) Add1
>>                  (((D2_72 Intersection F2_76)
>>                  <<= prime2(.thelaw_1,.B102_1))
>>                   Fixform (Ug([[(K2_148:obj)
>>                      => (Ded([[(khyp2_149:that
>>                         (K2_148 E (D2_72 Intersection
>>                         F2_76))) => ((Counterexample(casehyp2_144)
>>                         Eg [(.F3_152:obj),(fhyp3_152:
>>                            that (Counterexample(casehyp2_144)
>>                            Witnesses .F3_152))
>>                            => (((Notimp2(fhyp3_152)
>>                            Mp (.F3_152 Ui Simp2((khyp2_149
>>                            Iff1 (K2_148 Ui Separation4(Refleq((D2_72
>>                            Intersection F2_76))))))))
>>                            Mpsubs (Simp2(((Notimp2(fhyp3_152)
>>                            Mpsubs Simp1(intev_79))
>>                            Iff1 (.F3_152 Ui
>>                            Separation4(Refleq(((.Misset_1
>>                            Mbold2 .thelawchooses_1)
>>                            Set [(Y_171:obj)
>>                               => (cutse2(.Misset_1,
>>                               .thelawchooses_1,
>>                               .B102_1,Y_171):
>>                               prop)]))
>>                            )))) Ds1 Notimp1(fhyp3_152))):
>>                            that (K2_148 E prime2(.thelaw_1,
>>                            .B102_1)))])
>>                         :that (K2_148 E prime2(.thelaw_1,
>>                         .B102_1)))])
>>                      :that ((K2_148 E (D2_72
>>                      Intersection F2_76)) ->
>>                      (K2_148 E prime2(.thelaw_1,
>>                      .B102_1))))])
>>                   Conj (Separation3(Refleq((D2_72
>>                   Intersection F2_76))) Conj
>>                   Separation3(Refleq(prime2(.thelaw_1,
>>                   .B102_1))))))):that (((D2_72
>>                   Intersection F2_76) <<= prime2(.thelaw_1,
>>                   .B102_1)) V (.B102_1 <<= (D2_72
>>                   Intersection F2_76))))]))
>>                Iff2 ((D2_72 Intersection F2_76)
>>               Ui Separation4(Refleq(((.Misset_1
>>               Mbold2 .thelawchooses_1) Set
>>               [(Y_186:obj) => (cutse2(.Misset_1,
>>                  .thelawchooses_1,.B102_1,Y_186):
>>                  prop)]))
>>               )))):that ((D2_72 Intersection
>>               F2_76) E ((.Misset_1 Mbold2 .thelawchooses_1)
>>               Set [(Y_187:obj) => (cutse2(.Misset_1,
>>                  .thelawchooses_1,.B102_1,Y_187):
>>                  prop)]))
>>               )])
>>            :that (((D2_72 <<= ((.Misset_1 Mbold2
>>            .thelawchooses_1) Set [(Y_188:obj)
>>               => (cutse2(.Misset_1,.thelawchooses_1,
>>               .B102_1,Y_188):prop)]))
>>            & (F2_76 E D2_72)) -> ((D2_72 Intersection
>>            F2_76) E ((.Misset_1 Mbold2 .thelawchooses_1)
>>            Set [(Y_189:obj) => (cutse2(.Misset_1,
>>               .thelawchooses_1,.B102_1,Y_189):
>>               prop)]))
>>            ))])
>>         :that Forall([(F2_190:obj) => ((((D2_72
>>            <<= ((.Misset_1 Mbold2 .thelawchooses_1)
>>            Set [(Y_191:obj) => (cutse2(.Misset_1,
>>               .thelawchooses_1,.B102_1,Y_191):
>>               prop)]))
>>            & (F2_190 E D2_72)) -> ((D2_72 Intersection
>>            F2_190) E ((.Misset_1 Mbold2 .thelawchooses_1)
>>            Set [(Y_192:obj) => (cutse2(.Misset_1,
>>               .thelawchooses_1,.B102_1,Y_192):
>>               prop)]))
>>            ):prop)]))
>>         ]))
>>      ))):that thetachain1(.M_1,.thelaw_1,((.Misset_1
>>      Mbold2 .thelawchooses_1) Set [(Y_193:obj)
>>         => (cutse2(.Misset_1,.thelawchooses_1,
>>         .B102_1,Y_193):prop)]))
>>      )]
>>   {move 0}



open

   define lined107 bhyp101: linec107 bhyp101


>>    lined107: [(.B101_1:obj),(bhyp101_1:that
>>         (.B101_1 E Cuts)) => (---:that thetachain1(M,
>>         thelaw,((Misset Mbold2 thelawchooses)
>>         Set [(Y_2:obj) => (cutse2(Misset,thelawchooses,
>>            .B101_1,Y_2):prop)]))
>>         )]
>>      {move 1}



   open

      declare B103 obj

>>       B103: obj {move 3}



      declare bhyp103 that B103 E Cuts

>>       bhyp103: that (B103 E Cuts) {move 3}



      define linee107 bhyp103: lined107 bhyp103


>>       linee107: [(.B103_1:obj),(bhyp103_1:
>>            that (.B103_1 E Cuts)) => (---:that
>>            thetachain1(M,thelaw,((Misset Mbold2
>>            thelawchooses) Set [(Y_2:obj) =>
>>               (cutse2(Misset,thelawchooses,
>>               .B103_1,Y_2):prop)]))
>>            )]
>>         {move 2}



      open

         define Line107 bhyp: linee107 bhyp


>>          Line107: [(bhyp_1:that (B E Cuts))
>>               => (---:that thetachain1(M,thelaw,
>>               ((Misset Mbold2 thelawchooses)
>>               Set [(Y_2:obj) => (cutse2(Misset,
>>                  thelawchooses,B,Y_2):prop)]))
>>               )]
>>            {move 3}



         open

            declare K obj

>>             K: obj {move 5}



            open

               declare khyp that K E Mbold


>>                khyp: that (K E Mbold) {move
>>                    6}



               define line108 khyp: Ui Cuts2, \
                  Simp2(Iff1(khyp, Ui K,Separation4 \
                  Refleq Mbold))

>>                  line108: [(khyp_1:that (K
>>                       E Mbold)) => (---:that
>>                       ((Cuts2 E (Sc(Sc(M)) Set
>>                       [(C_17:obj) => (thetachain1(M,
>>                          thelaw,C_17):prop)]))
>>                       -> (K E Cuts2)))]
>>                    {move 5}



               define linea108: Iff2(Simp1(Simp2 \
                  Line107 bhyp),Ui Cuts2, Scthm(Sc \
                  M))

>>                  linea108: [(---:that (Cuts2
>>                       E Sc(Sc(M))))]
>>                    {move 5}



               define line109: Fixform(Cuts2 \
                  E Thetachain,Iff2(Conj(linea108, \
                  Line107 bhyp),Ui Cuts2, Separation4 \
                  Refleq Thetachain))

>>                  line109: [(---:that (Cuts2
>>                       E Thetachain))]
>>                    {move 5}
```

Here we have line 107 to the effect that Cuts2 is a Θ-chain and line 109 to the effect that it belongs to the set of Θ-chains.

```
Lestrade execution:


               define line110 khyp: Mp(line109, \
                  line108 khyp)

>>                line110: [(khyp_1:that (K
>>                     E Mbold)) => (---:that
>>                     (K E Cuts2))]
>>                  {move 5}




               define line111 khyp: Iff1(line110 \
                  khyp,Ui K,Separation4 \
                  Refleq Cuts2)

>>                line111: [(khyp_1:that (K
>>                     E Mbold)) => (---:that
>>                     ((K E Mbold) & cutsi2(K)))]
>>                  {move 5}




               define line112: Fixform((prime \
                  B) <<= B,Sepsub2 (linea14 \
                  bhyp,Refleq prime B))

>>                line112: [(---:that (prime(B)
>>                     <<= B))]
>>                  {move 5}




               define line113 khyp: Simp2 \
                  line111 khyp

>>                line113: [(khyp_1:that (K
>>                     E Mbold)) => (---:that
>>                     cutsi2(K))]
>>                  {move 5}



               open

                  declare casehyp1 that K \
                     <<= prime B

>>                   casehyp1: that (K <<= prime(B))
>>                     {move 7}



                  declare casehyp2 that B \
                     <<= K

>>                   casehyp2: that (B <<= K)
>>                     {move 7}



                  define case1 casehyp1: \
                     Add1((prime B)<<=K, casehyp1)


>>                   case1: [(casehyp1_1:that
>>                        (K <<= prime(B))) =>
>>                        (---:that ((K <<= prime(B))
>>                        V (prime(B) <<= K)))]
>>                     {move 6}
```

```
                  define case2 casehyp2: \
                     Add2(K <<= prime B, Transsub \
                     line112,casehyp2)

>>                   case2: [(casehyp2_1:that
>>                        (B <<= K)) => (---:that
>>                        ((K <<= prime(B)) V
>>                        (prime(B) <<= K)))]
>>                     {move 6}



                  close

               define line114 khyp: Cases \
                  (line113 khyp, case1, \
                  case2)

>>                line114: [(khyp_1:that (K
>>                     E Mbold)) => (---:that
>>                     ((K <<= prime(B)) V (prime(B)
>>                     <<= K)))]
>>                  {move 5}



               close

            define line115 K: Ded line114


>>             line115: [(K_1:obj) => (---:that
>>                  ((K_1 E Mbold) -> ((K_1 <<=
>>                  prime(B)) V (prime(B) <<=
>>                  K_1))))]
>>               {move 4}
```

```
               close

            define line116 bhyp: Ug line115


>>             line116: [(bhyp_1:that (B E Cuts))
>>                  => (---:that Forall([(K_61:obj)
>>                     => (((K_61 E Mbold) -> ((K_61
>>                     <<= prime(B)) V (prime(B)
>>                     <<= K_61))):prop)]))
>>                  ]
>>               {move 3}



            define linea116 bhyp: Mp(line14 \
               bhyp,Ui B,Simp1 Simp2 Simp2 \
               Mboldtheta)

>>             linea116: [(bhyp_1:that (B E Cuts))
>>                  => (---:that (prime2(thelaw,B)
>>                  E (Misset Mbold2 thelawchooses)))]
>>               {move 3}



            define line117 bhyp: Fixform((prime \
               B) E Cuts,Iff2(Conj(linea116 bhyp, \
               Conj(linea116 bhyp,line116 bhyp)), \
               Ui(prime B,Separation4 Refleq Cuts)))


>>             line117: [(bhyp_1:that (B E Cuts))
>>                  => (---:that (prime(B) E Cuts))]
>>               {move 3}
```

```
         close

      define line118 B: Ded line117

>>       line118: [(B_1:obj) => (---:that ((B_1
>>            E Cuts) -> (prime(B_1) E Cuts)))]
>>         {move 2}



      close

   define Linea119: Ug line118

>>    Linea119: [(---:that Forall([(B_92:obj)
>>            => (((B_92 E Cuts) -> (prime(B_92)
>>            E Cuts)):prop)]))
>>         ]
>>      {move 1}



   close

define Lineb119 Misset thelawchooses: Linea119


>> Lineb119: [(.M_1:obj),(Misset_1:that Isset(.M_1)),
>>      (.thelaw_1:[(S_2:obj) => (---:obj)]),
>>      (thelawchooses_1:[(.S_3:obj),(subsetev_3:
>>         that (.S_3 <<= .M_1)),(inev_3:that
>>         Exists([(x_4:obj) => ((x_4 E .S_3):
>>            prop)]))
>>         => (---:that (.thelaw_1(.S_3) E .S_3))])
>>      => (Ug([(B_6:obj) => (Ded([(bhyp_7:that
>>            (B_6 E (Misset_1 Cuts3 thelawchooses_1)))
>>            => (((prime2(.thelaw_1,B_6) E (Misset_1
>>            Cuts3 thelawchooses_1)) Fixform
>>            (((Simp1((bhyp_7 Iff1 (B_6 Ui ((Misset_1
>>            Mbold2 thelawchooses_1) Separation
>>            [(C_11:obj) => (cuts2(Misset_1,thelawchooses_1,
>>               C_11):prop)]))
>>            )) Mp (B_6 Ui Simp1(Simp2(Simp2((Misset_1
>>            Mboldtheta2 thelawchooses_1))))))
>>            Conj ((Simp1((bhyp_7 Iff1 (B_6 Ui
>>            ((Misset_1 Mbold2 thelawchooses_1)
>>            Separation [(C_25:obj) => (cuts2(Misset_1,
>>               thelawchooses_1,C_25):prop)]))
>>            )) Mp (B_6 Ui Simp1(Simp2(Simp2((Misset_1
>>            Mboldtheta2 thelawchooses_1))))))
>>            Conj Ug([(K_38:obj) => (Ded([(khyp_39:
>>                  that (K_38 E (Misset_1 Mbold2
>>                  thelawchooses_1))) => (Cases(Simp2(((((((Misset_1
>>                  Mbold2 thelawchooses_1) Set
>>                  [(Y_43:obj) => (cutse2(Misset_1,
>>                     thelawchooses_1,B_6,Y_43):
>>                     prop)])
>>                  E (Sc(Sc(.M_1)) Set [(C_44:
>>                     obj) => (thetachain1(.M_1,
>>                     .thelaw_1,C_44):prop)]))
>>                  Fixform (((Simp1(Simp2(linec107(bhyp_7)))
>>                  Iff2 (((Misset_1 Mbold2 thelawchooses_1)
>>                  Set [(Y_67:obj) => (cutse2(Misset_1,
>>                     thelawchooses_1,B_6,Y_67):
>>                     prop)])
>>                  Ui Scthm(Sc(.M_1)))) Conj
>>                  linec107(bhyp_7)) Iff2 (((Misset_1
>>                  Mbold2 thelawchooses_1) Set
>>                  [(Y_72:obj) => (cutse2(Misset_1,
>>                     thelawchooses_1,B_6,Y_72):
>>                     prop)])
>>                  Ui Separation4(Refleq((Sc(Sc(.M_1))
>>                  Set [(C_77:obj) => (thetachain1(.M_1,
>>                     .thelaw_1,C_77):prop)]))
>>                  )))) Mp (((Misset_1 Mbold2
>>                  thelawchooses_1) Set [(Y_79:
>>                     obj) => (cutse2(Misset_1,
>>                     thelawchooses_1,B_6,Y_79):
>>                     prop)])
>>                  Ui Simp2((khyp_39 Iff1 (K_38
>>                  Ui Separation4(Refleq((Misset_1
>>                  Mbold2 thelawchooses_1)))))))
>>                  Iff1 (K_38 Ui Separation4(Refleq(((Misset_1
>>                  Mbold2 thelawchooses_1) Set
>>                  [(Y_99:obj) => (cutse2(Misset_1,
>>                     thelawchooses_1,B_6,Y_99):
>>                     prop)]))
>>                  )))),[(casehyp1_100:that (K_38
>>                     <<= prime2(.thelaw_1,B_6)))
>>                     => (((prime2(.thelaw_1,
>>                     B_6) <<= K_38) Add1 casehyp1_100):
>>                     that ((K_38 <<= prime2(.thelaw_1,
>>                     B_6)) V (prime2(.thelaw_1,
>>                     B_6) <<= K_38)))]
>>                  ,[(casehyp2_101:that (B_6
>>                     <<= K_38)) => (((K_38 <<=
>>                     prime2(.thelaw_1,B_6))
>>                     Add2 (((prime2(.thelaw_1,
>>                     B_6) <<= B_6) Fixform (Setsinchains2(Misset_1,
>>                     thelawchooses_1,(Misset_1
>>                     Mboldtheta2 thelawchooses_1),
>>                     Simp1((bhyp_7 Iff1 (B_6
>>                     Ui ((Misset_1 Mbold2 thelawchooses_1)
>>                     Separation [(C_104:obj)
>>                        => (cuts2(Misset_1,thelawchooses_1,
>>                        C_104):prop)]))
>>                     ))) Sepsub2 Refleq(prime2(.thelaw_1,
>>                     B_6)))) Transsub casehyp2_101)):
>>                     that ((K_38 <<= prime2(.thelaw_1,
>>                     B_6)) V (prime2(.thelaw_1,
>>                     B_6) <<= K_38)))])
>>                  :that ((K_38 <<= prime2(.thelaw_1,
>>                  B_6)) V (prime2(.thelaw_1,
>>                  B_6) <<= K_38)))])
>>               :that ((K_38 E (Misset_1 Mbold2
>>               thelawchooses_1)) -> ((K_38 <<=
>>               prime2(.thelaw_1,B_6)) V (prime2(.thelaw_1,
>>               B_6) <<= K_38))))]))
>>            ) Iff2 (prime2(.thelaw_1,B_6) Ui
>>            Separation4(Refleq((Misset_1 Cuts3
>>            thelawchooses_1)))))):that (prime2(.thelaw_1,
>>            B_6) E (Misset_1 Cuts3 thelawchooses_1)))])
>>         :that ((B_6 E (Misset_1 Cuts3 thelawchooses_1))
>>         -> (prime2(.thelaw_1,B_6) E (Misset_1
>>         Cuts3 thelawchooses_1))))])
>>      :that Forall([(B_110:obj) => (((B_110
>>         E (Misset_1 Cuts3 thelawchooses_1))
>>         -> (prime2(.thelaw_1,B_110) E (Misset_1
>>         Cuts3 thelawchooses_1))):prop)]))
>>      ]
>>   {move 0}
```

```
open

   define Line119: Lineb119 Misset thelawchooses
```

```
>>    Line119: [(---:that Forall([(B_1:obj)
>>            => (((B_1 E (Misset Cuts3 thelawchooses))
>>            -> (prime2(thelaw,B_1) E (Misset
>>            Cuts3 thelawchooses))):prop)]))
>>         ]
>>      {move 1}
```

This is the third component of the proof that `Cuts` is a Θ-chain, proved with the aid of the result that `Cuts2` is a Θ-chain (and so coincides with $\mathbf{M}$).

```
Lestrade execution:


   declare D3 obj

>>    D3: obj {move 2}



   declare F3 obj

>>    F3: obj {move 2}



   goal that Forall[D3 =>[F3 =>((D3 <<= Cuts) \
            & F3 E D3) -> (D3 Intersection \
            F3) E Cuts] \
         ] \




>>    Goal: that Forall([(D3_24863:obj) => ([(F3_24864:
>>            obj) => ((((D3_24863 <<= Cuts) &
>>            (F3_24864 E D3_24863)) -> ((D3_24863
>>            Intersection F3_24864) E Cuts)):
>>            prop)]
>>         :[(F3_24865:obj) => (---:prop)])
>>         ])
>>

   open

      declare D4 obj

>>       D4: obj {move 3}
```

```
      open

         declare dhyp4 that D4 <<= Cuts

>>          dhyp4: that (D4 <<= Cuts) {move
>>            4}



         open

            declare F4 obj

>>             F4: obj {move 5}



            open

               declare fhyp4 that F4 E D4


>>                fhyp4: that (F4 E D4) {move
>>                  6}



               test Ui (D4 Intersection F4, \
                  Separation4 Refleq Cuts)


>>                Test: ((D4 Intersection F4)
>>                  Ui Separation4(Refleq(Cuts)))


>>                that (((D4 Intersection F4)
>>                  E ((Misset Mbold2 thelawchooses)
>>                  Set [(C_449:obj) => (cuts2(Misset,
>>                     thelawchooses,C_449):prop)]))
>>                  == (((D4 Intersection F4)
>>                  E (Misset Mbold2 thelawchooses))
>>                  & cuts2(Misset,thelawchooses,
>>                  (D4 Intersection F4))))



               goal that D4 Intersection \
                  F4 E Mbold

>>                Goal: that (D4 Intersection
>>                  (F4 E Mbold))

               test Fixform(Cuts <<= Mbold, \
                  Sepsub2(Separation3 Refleq \
                  Mbold,Refleq Cuts))

>>                Test: ((Cuts <<= Mbold) Fixform
>>                  (Separation3(Refleq(Mbold))
>>                  Sepsub2 Refleq(Cuts)))

>>                that (Cuts <<= Mbold)



               define line120: Transsub(dhyp4, \
                  Fixform(Cuts <<= Mbold, Sepsub2(Separation3 \
                  Refleq Mbold,Refleq Cuts)))


>>                line120: [(---:that (D4 <<=
>>                     Mbold))]
>>                  {move 5}
```

```
               define line121 fhyp4: Mpsubs \
                  fhyp4 line120

>>                line121: [(fhyp4_1:that (F4
>>                     E D4)) => (---:that (F4
>>                     E Mbold))]
>>                  {move 5}



               define line122 fhyp4: Mp(line120 \
                  Conj fhyp4,Ui F4, Ui D4, \
                  Simp2 Simp2 Simp2 Mboldtheta)


>>                line122: [(fhyp4_1:that (F4
>>                     E D4)) => (---:that ((D4
>>                     Intersection F4) E (Misset
>>                     Mbold2 thelawchooses)))]
>>                  {move 5}



               goal that cuts (D4 Intersection \
                  F4)

>>                Goal: that cuts((D4 Intersection
>>                  F4))

               declare testing that cuts(D4 \
                  Intersection F4)

>>                testing: that cuts((D4 Intersection
>>                  F4)) {move 6}
```

```
               test Simp1(testing)

>>                Test: Simp1(testing)

>>                that ((D4 Intersection F4)
>>                  E (Misset Mbold2 thelawchooses))



               test Simp2(testing)

>>                Test: Simp2(testing)

>>                that Forall([(D_1692:obj)
>>                     => (((D_1692 E (Misset
>>                     Mbold2 thelawchooses))
>>                     -> ((D_1692 <<= (D4 Intersection
>>                     F4)) V ((D4 Intersection
>>                     F4) <<= D_1692))):prop)])
>>



               open

                  declare D5 obj

>>                   D5: obj {move 7}



                  open

                     declare dhyp5 that D5 \
                        E Mbold

>>                      dhyp5: that (D5 E Mbold)
>>                        {move 8}
```

```
                     goal that (D5 <<= D4 \
                        Intersection F4) V \
                        (D4 Intersection F4) \
                        <<= D5

>>                      Goal: that ((D5 <<=
>>                        (D4 Intersection F4))
>>                        V ((D4 Intersection
>>                        F4) <<= D5))

                     declare D6 obj

>>                      D6: obj {move 8}



                     define line123 : \
                        Excmid(Forall[D6 \
                           => (D6 E D4) -> D5 \
                           <<= D6]) \




>>                      line123: [(---:that
>>                           (Forall([(D6_1:obj)
>>                              => (((D6_1 E D4)
>>                              -> (D5 <<= D6_1)):
>>                              prop)])
>>                           V ~(Forall([(D6_1:
>>                              obj) => (((D6_1
>>                              E D4) -> (D5 <<=
>>                              D6_1)):prop)]))
>>                           ))]
>>                        {move 7}
```

```
                     open

                        declare D7 obj

>>                         D7: obj {move 9}



                        declare casehyp1 \
                           that Forall[D7 => \
                              (D7 E D4) -> \
                              D5 <<= D7] \




>>                         casehyp1: that Forall([(D7_1:
>>                              obj) => (((D7_1
>>                              E D4) -> (D5 <<=
>>                              D7_1)):prop)])
>>                           {move 9}



                        open

                           declare G obj


>>                            G: obj {move 10}



                           open
```

```
                              declare \
                                 ghyp that G \
                                 E D5

>>                               ghyp: that
>>                                 (G E D5) {move
>>                                 11}



                              goal that G \
                                 E D4 Intersection \
                                 F4

>>                               Goal: that
>>                                 (G E (D4 Intersection
>>                                 F4))

                              test Ui \
                                 G,Separation4 \
                                 Refleq \
                                 (D4 Intersection \
                                 F4)

>>                               Test: (G Ui
>>                                 Separation4(Refleq((D4
>>                                 Intersection
>>                                 F4))))

>>                               that ((G E
>>                                 (F4 Set [(x_419:
>>                                    obj) =>
>>                                    (Forall([(B_421:
>>                                       obj)
>>                                       => (((B_421
>>                                       E D4)
>>                                       -> (x_419
>>                                       E B_421)):
```

```
>>                                       prop)])
>>                                    :prop)]))
>>                                 == ((G E F4)
>>                                 & Forall([(B_422:
>>                                    obj) =>
>>                                    (((B_422
>>                                    E D4) ->
>>                                    (G E B_422)):
>>                                    prop)]))
>>                                 )


                              open

                                 declare \
                                    B1 obj


>>                                  B1: obj
>>                                    {move 12}



                                 open

                                    declare \
                                       bhyp1 \
                                       that \
                                       B1 E \
                                       D4

>>                                     bhyp1:
>>                                       that
>>                                       (B1 E
>>                                       D4) {move
>>                                       13}
```

```
                                      goal \
                                         that \
                                         G E \
                                         B1

>>                                       Goal:
>>                                         that
>>                                         (G E
>>                                         B1)

                                      define \
                                         line124 \
                                         bhyp1: \
                                         Mpsubs \
                                         ghyp, \
                                         Mp bhyp1, \
                                         Ui B1 \
                                         casehyp1


>>                                       line124:
>>                                         [(bhyp1_1:
>>                                            that (B1
>>                                            E D4))
>>                                            => (---:
>>                                            that (G
>>                                            E B1))]
>>                                         {move
>>                                         12}



                                      close


                                   define line125 \
                                     B1: Ded \
                                     line124


>>                                   line125:
>>                                     [(B1_1:obj)
>>                                        => (---:
>>                                        that
>>                                        ((B1_1
>>                                        E D4)
>>                                        -> (G
>>                                        E B1_1)))]
>>                                     {move 11}



                                  close

                               define line126 \
                                  ghyp: Ug line125


>>                                line126: [(ghyp_1:
>>                                     that (G
>>                                     E D5)) =>
>>                                     (---:that
>>                                     Forall([(B1_6:
>>                                        obj)
>>                                        => (((B1_6
>>                                        E D4)
>>                                        -> (G
>>                                        E B1_6)):
>>                                        prop)]))
>>                                     ]
>>                                  {move 10}
```

```
                              define line127 \
                                 ghyp: Mp fhyp4, \
                                 Ui F4, line126 \
                                 ghyp

>>                               line127: [(ghyp_1:
>>                                    that (G
>>                                    E D5)) =>
>>                                    (---:that
>>                                    (G E F4))]
>>                                 {move 10}



                              define line128 \
                                 ghyp: Conj(line127 \
                                 ghyp, line126 \
                                 ghyp)

>>                               line128: [(ghyp_1:
>>                                    that (G
>>                                    E D5)) =>
>>                                    (---:that
>>                                    ((G E F4)
>>                                    & Forall([(B1_3:
>>                                       obj)
>>                                       => (((B1_3
>>                                       E D4)
>>                                       -> (G
>>                                       E B1_3)):
>>                                       prop)]))
>>                                    )]
>>                                 {move 10}



                              define line129 \
                                 ghyp: Fixform(G \
                                 E D4 Intersection \
                                 F4,Iff2(line128 \
                                 ghyp, Ui G, \
                                 Separation4 \
                                 Refleq (D4 \
                                 Intersection \
                                 F4)))

>>                               line129: [(ghyp_1:
>>                                    that (G
>>                                    E D5)) =>
>>                                    (---:that
>>                                    (G E (D4
>>                                    Intersection
>>                                    F4)))]
>>                                 {move 10}



                              close

                           define line130 \
                              G : Ded line129


>>                            line130: [(G_1:
>>                                 obj) => (---:
>>                                 that ((G_1
>>                                 E D5) -> (G_1
>>                                 E (D4 Intersection
>>                                 F4))))]
>>                              {move 9}



                           close

                        define line131 casehyp1: \
                           Fixform(D5 <<= D4 \
                           Intersection F4, \
                           Conj(Ug line130, \
                           Conj(Setsinchains \
                           Mboldtheta, dhyp5, \
                           Separation3 Refleq \
                           (D4 Intersection \
                           F4))))

>>                         line131: [(casehyp1_1:
>>                              that Forall([(D7_2:
>>                                 obj) => (((D7_2
>>                                 E D4) -> (D5
>>                                 <<= D7_2)):
>>                                 prop)]))
>>                              => (---:that (D5
>>                              <<= (D4 Intersection
>>                              F4)))]
>>                           {move 8}



                        define line132 casehyp1: \
                           Add1((D4 Intersection \
                           F4) <<= D5,line131 \
                           casehyp1)

>>                         line132: [(casehyp1_1:
>>                              that Forall([(D7_2:
>>                                 obj) => (((D7_2
>>                                 E D4) -> (D5
>>                                 <<= D7_2)):
>>                                 prop)]))
>>                              => (---:that ((D5
>>                              <<= (D4 Intersection
>>                              F4)) V ((D4 Intersection
>>                              F4) <<= D5)))]
>>                           {move 8}
```

```
                        declare casehyp2 \
                           that ~(Forall[D7 \
                              => (D7 E D4) \
                              -> D5 <<= \
                              D7]) \



>>                         casehyp2: that ~(Forall([(D7_1:
>>                              obj) => (((D7_1
>>                              E D4) -> (D5 <<=
>>                              D7_1)):prop)]))
>>                           {move 9}



                        open

                           declare G obj


>>                            G: obj {move 10}



                           open

                              declare \
                                 ghyp that G \
                                 E D4 Intersection \
                                 F4

>>                               ghyp: that
>>                                 (G E (D4 Intersection
>>                                 F4)) {move
>>                                 11}

                              goal that G \
                                 E D5

>>                               Goal: that
>>                                 (G E D5)

                              define line133 \
                                 : Counterexample \
                                 casehyp2

>>                               line133: [(---:
>>                                    that Exists([(z_2:
>>                                       obj)
>>                                       => (~(((z_2
>>                                       E D4)
>>                                       -> (D5
>>                                       <<= z_2))):
>>                                       prop)]))
>>                                    ]
>>                                 {move 10}

                              open

                                 declare \
                                    H obj

>>                                  H: obj {move
>>                                    12}
```

```
                                 declare \
                                    hhyp that \
                                    Witnesses \
                                    line133 \
                                    H

>>                                  hhyp: that
>>                                    (line133
>>                                    Witnesses
>>                                    H) {move
>>                                    12}



                                 define line134 \
                                    hhyp: Notimp1 \
                                    hhyp

>>                                  line134:
>>                                    [(.H_1:obj),
>>                                       (hhyp_1:
>>                                       that
>>                                       (line133
>>                                       Witnesses
>>                                       .H_1))
>>                                       => (---:
>>                                       that
>>                                       ~((D5
>>                                       <<= .H_1)))]
>>                                    {move 11}



                                 define line135 \
                                    hhyp: Notimp2 \
                                    hhyp

>>                                  line135:
```

```
>>                                    [(.H_1:obj),
>>                                       (hhyp_1:
>>                                       that
>>                                       (line133
>>                                       Witnesses
>>                                       .H_1))
>>                                       => (---:
>>                                       that
>>                                       (.H_1
>>                                       E D4))]
>>                                    {move 11}
```

```
                               define line136 \
                                  hhyp: Mp \
                                  line135 \
                                  hhyp, Ui \
                                  H, Simp2 \
                                  (Iff1(ghyp, \
                                  Ui G, Separation4 \
                                  Refleq \
                                  (D4 Intersection \
                                  F4)))
```

```
>>                                  line136:
>>                                    [(.H_1:obj),
>>                                       (hhyp_1:
>>                                       that
>>                                       (line133
>>                                       Witnesses
>>                                       .H_1))
>>                                       => (---:
>>                                       that
>>                                       (G E
>>                                       .H_1))]
>>                                    {move 11}
```

```
                                 define line137 \
                                    hhyp: Mpsubs \
                                    line135 \
                                    hhyp, dhyp4


>>                                  line137:
>>                                    [(.H_1:obj),
>>                                       (hhyp_1:
>>                                       that
>>                                       (line133
>>                                       Witnesses
>>                                       .H_1))
>>                                       => (---:
>>                                       that
>>                                       (.H_1
>>                                       E Cuts))]
>>                                    {move 11}



                                 define line138 \
                                    hhyp: Mp \
                                    dhyp5, \
                                    Ui D5, \
                                    Simp2(Simp2(Iff1(line137 \
                                    hhyp, Ui \
                                    H, Separation4 \
                                    Refleq \
                                    Cuts)))


>>                                  line138:
>>                                    [(.H_1:obj),
>>                                       (hhyp_1:
>>                                       that
>>                                       (line133
>>                                       Witnesses
>>                                       .H_1))
>>                                       => (---:
>>                                       that
>>                                       ((D5
>>                                       <<= .H_1)
>>                                       V (.H_1
>>                                       <<= D5)))]
>>                                    {move 11}



                                 define line139 \
                                    hhyp: Ds2(line138 \
                                    hhyp, line134 \
                                    hhyp)

>>                                  line139:
>>                                    [(.H_1:obj),
>>                                       (hhyp_1:
>>                                       that
>>                                       (line133
>>                                       Witnesses
>>                                       .H_1))
>>                                       => (---:
>>                                       that
>>                                       (.H_1
>>                                       <<= D5))]
>>                                    {move 11}



                                 define line140 \
                                    hhyp: Mpsubs(line136 \
                                    hhyp, line139 \
                                    hhyp)
```

```
>>                                  line140:
>>                                    [(.H_1:obj),
>>                                       (hhyp_1:
>>                                       that
>>                                       (line133
>>                                       Witnesses
>>                                       .H_1))
>>                                       => (---:
>>                                       that
>>                                       (G E
>>                                       D5))]
>>                                    {move 11}



                                 close

                              define line141 \
                                 ghyp: Eg line133 \
                                 line140

>>                               line141: [(ghyp_1:
>>                                    that (G
>>                                    E (D4 Intersection
>>                                    F4))) =>
>>                                    (---:that
>>                                    (G E D5))]
>>                                 {move 10}



                              close

                           define line142 \
                              G: Ded line141


>>                            line142: [(G_1:
```

```
>>                                 obj) => (---:
>>                                 that ((G_1
>>                                 E (D4 Intersection
>>                                 F4)) -> (G_1
>>                                 E D5)))]
>>                              {move 9}
```

```
                           close

                        define line143 casehyp2: \
                           Fixform((D4 Intersection \
                           F4)<<= D5,Conj(Ug \
                           line142, Conj(Separation3 \
                           Refleq (D4 Intersection \
                           F4),Setsinchains \
                           Mboldtheta, dhyp5)))
```

```
>>                         line143: [(casehyp2_1:
>>                              that ~(Forall([(D7_2:
>>                                 obj) => (((D7_2
>>                                 E D4) -> (D5
>>                                 <<= D7_2)):
>>                                 prop)]))
>>                              ) => (---:that
>>                              ((D4 Intersection
>>                              F4) <<= D5))]
>>                           {move 8}
```

```
                        define line144 casehyp2: \
                           Add2(D5 <<= D4 Intersection \
                           F4,line143 casehyp2)
```

```
>>                         line144: [(casehyp2_1:
>>                              that ~(Forall([(D7_2:
>>                                 obj) => (((D7_2
>>                                 E D4) -> (D5
>>                                 <<= D7_2)):
>>                                 prop)]))
>>                              ) => (---:that
>>                              ((D5 <<= (D4 Intersection
>>                              F4)) V ((D4 Intersection
>>                              F4) <<= D5)))]
>>                           {move 8}


                        close

                     define line145 dhyp5: \
                        Cases line123, line132, \
                        line144

>>                      line145: [(dhyp5_1:that
>>                           (D5 E Mbold)) =>
>>                           (---:that ((D5 <<=
>>                           (D4 Intersection
>>                           F4)) V ((D4 Intersection
>>                           F4) <<= D5)))]
>>                        {move 7}



                     close

                  define line146 D5: Ded \
                     line145

>>                   line146: [(D5_1:obj) =>
>>                        (---:that ((D5_1 E Mbold)
>>                        -> ((D5_1 <<= (D4 Intersection
>>                        F4)) V ((D4 Intersection
>>                        F4) <<= D5_1))))]
>>                     {move 6}



                  close

               define line147 fhyp4: Conj(line122 \
                  fhyp4,Conj(line122 fhyp4, \
                  Ug line146))

>>                line147: [(fhyp4_1:that (F4
>>                     E D4)) => (---:that (((D4
>>                     Intersection F4) E (Misset
>>                     Mbold2 thelawchooses))
>>                     & (((D4 Intersection F4)
>>                     E (Misset Mbold2 thelawchooses))
>>                     & Forall([(D5_72:obj) =>
>>                        (((D5_72 E Mbold) ->
>>                        ((D5_72 <<= (D4 Intersection
>>                        F4)) V ((D4 Intersection
>>                        F4) <<= D5_72))):prop)]))
>>                     ))]
>>                  {move 5}



               define linea147 fhyp4: \
                  Iff2(line147 fhyp4,Ui(D4 Intersection \
                  F4,Separation4 Refleq Cuts))


>>                linea147: [(fhyp4_1:that (F4
>>                     E D4)) => (---:that ((D4
>>                     Intersection F4) E ((Misset
>>                     Mbold2 thelawchooses) Set
>>                     [(C_7:obj) => (cuts2(Misset,
>>                        thelawchooses,C_7):prop)]))
>>                     )]
>>                  {move 5}



               close

            define line148 F4: Ded linea147


>>             line148: [(F4_1:obj) => (---:
>>                  that ((F4_1 E D4) -> ((D4
>>                  Intersection F4_1) E ((Misset
>>                  Mbold2 thelawchooses) Set
>>                  [(C_118:obj) => (cuts2(Misset,
>>                     thelawchooses,C_118):prop)]))
>>                  ))]
>>               {move 4}



            close

         define line149 dhyp4: Ug line148


>>          line149: [(dhyp4_1:that (D4 <<=
>>               Cuts)) => (---:that Forall([(F4_122:
>>                  obj) => (((F4_122 E D4) ->
>>                  ((D4 Intersection F4_122)
>>                  E ((Misset Mbold2 thelawchooses)
>>                  Set [(C_123:obj) => (cuts2(Misset,
>>                     thelawchooses,C_123):prop)]))
>>                  ):prop)]))
>>               ]
>>            {move 3}
```

```
         close

      define line150 D4: Ded line149

>>       line150: [(D4_1:obj) => (---:that ((D4_1
>>            <<= Cuts) -> Forall([(F4_127:obj)
>>               => (((F4_127 E D4_1) -> ((D4_1
>>               Intersection F4_127) E ((Misset
>>               Mbold2 thelawchooses) Set [(C_128:
>>                  obj) => (cuts2(Misset,thelawchooses,
>>                  C_128):prop)]))
>>               ):prop)]))
>>            )]
>>         {move 2}



      close

   define line151 : Ug line150

>>    line151: [(---:that Forall([(D4_129:obj)
>>            => (((D4_129 <<= Cuts) -> Forall([(F4_130:
>>               obj) => (((F4_130 E D4_129) ->
>>               ((D4_129 Intersection F4_130)
>>               E ((Misset Mbold2 thelawchooses)
>>               Set [(C_131:obj) => (cuts2(Misset,
>>                  thelawchooses,C_131):prop)]))
>>               ):prop)]))
>>            :prop)]))
>>         ]
>>      {move 1}



   open
```

```
      declare D9 obj

>>       D9: obj {move 3}



      open

         declare F9 obj

>>          F9: obj {move 4}



         open

            declare conjhyp that (D9 <<= \
               Cuts) & F9 E D9

>>             conjhyp: that ((D9 <<= Cuts)
>>               & (F9 E D9)) {move 5}



            define firsthyp conjhyp: Simp1 \
               conjhyp

>>             firsthyp: [(conjhyp_1:that ((D9
>>                  <<= Cuts) & (F9 E D9))) =>
>>                  (---:that (D9 <<= Cuts))]
>>               {move 4}



            define secondhyp conjhyp: \
               Simp2 conjhyp
```

```
>>             secondhyp: [(conjhyp_1:that ((D9
>>                  <<= Cuts) & (F9 E D9))) =>
>>                  (---:that (F9 E D9))]
>>               {move 4}



            define line152 conjhyp: Mp secondhyp \
               conjhyp, Ui F9, Mp (firsthyp \
               conjhyp, Ui D9 line151)

>>             line152: [(conjhyp_1:that ((D9
>>                  <<= Cuts) & (F9 E D9))) =>
>>                  (---:that ((D9 Intersection
>>                  F9) E ((Misset Mbold2 thelawchooses)
>>                  Set [(C_10:obj) => (cuts2(Misset,
>>                     thelawchooses,C_10):prop)]))
>>                  )]
>>               {move 4}



            close

         define line153 F9: Ded line152

>>          line153: [(F9_1:obj) => (---:that
>>               (((D9 <<= Cuts) & (F9_1 E D9))
>>               -> ((D9 Intersection F9_1) E
>>               ((Misset Mbold2 thelawchooses)
>>               Set [(C_13:obj) => (cuts2(Misset,
>>                  thelawchooses,C_13):prop)]))
>>               ))]
>>            {move 3}



         close
```

```
      define line154 D9: Ug line153

>>       line154: [(D9_1:obj) => (---:that Forall([(F9_17:
>>               obj) => ((((D9_1 <<= Cuts) &
>>               (F9_17 E D9_1)) -> ((D9_1 Intersection
>>               F9_17) E ((Misset Mbold2 thelawchooses)
>>               Set [(C_18:obj) => (cuts2(Misset,
>>                  thelawchooses,C_18):prop)]))
>>               ):prop)]))
>>            ]
>>         {move 2}



      close

   define linea155: Ug line154

>>    linea155: [(---:that Forall([(D9_19:obj)
>>            => (Forall([(F9_20:obj) => ((((D9_19
>>               <<= Cuts) & (F9_20 E D9_19))
>>               -> ((D9_19 Intersection F9_20)
>>               E ((Misset Mbold2 thelawchooses)
>>               Set [(C_21:obj) => (cuts2(Misset,
>>                  thelawchooses,C_21):prop)]))
>>               ):prop)])
>>            :prop)]))
>>         ]
>>      {move 1}



   save

   close

define lineb155 Misset, thelawchooses: linea155
```

```
>> lineb155: [(.M_1:obj),(Misset_1:that Isset(.M_1)),
>>      (.thelaw_1:[(S_2:obj) => (---:obj)]),
>>      (thelawchooses_1:[(.S_3:obj),(subsetev_3:
>>         that (.S_3 <<= .M_1)),(inev_3:that
>>         Exists([(x_4:obj) => ((x_4 E .S_3):
>>            prop)]))
>>         => (---:that (.thelaw_1(.S_3) E .S_3))])
>>      => (Ug([(D9_8:obj) => (Ug([(F9_11:obj)
>>            => (Ded([(conjhyp_13:that ((D9_8
>>               <<= (Misset_1 Cuts3 thelawchooses_1))
>>               & (F9_11 E D9_8))) => ((Simp2(conjhyp_13)
>>               Mp (F9_11 Ui (Simp1(conjhyp_13)
>>               Mp (D9_8 Ui Ug([(D4_25:obj) =>
>>                  (Ded([(dhyp4_28:that (D4_25
>>                     <<= (Misset_1 Cuts3 thelawchooses_1)))
>>                     => (Ug([(F4_31:obj) =>
>>                        (Ded([(fhyp4_33:that
>>                           (F4_31 E D4_25))
>>                           => ((((((dhyp4_28
>>                           Transsub (((Misset_1
>>                           Cuts3 thelawchooses_1)
>>                           <<= (Misset_1 Mbold2
>>                           thelawchooses_1))
>>                           Fixform (Separation3(Refleq((Misset_1
>>                           Mbold2 thelawchooses_1)))
>>                           Sepsub2 Refleq((Misset_1
>>                           Cuts3 thelawchooses_1)))))
>>                           Conj fhyp4_33) Mp
>>                           (F4_31 Ui (D4_25
>>                           Ui Simp2(Simp2(Simp2((Misset_1
>>                           Mboldtheta2 thelawchooses_1)))))))
>>                           Conj ((((dhyp4_28
>>                           Transsub (((Misset_1
>>                           Cuts3 thelawchooses_1)
>>                           <<= (Misset_1 Mbold2
>>                           thelawchooses_1))
>>                            Fixform (Separation3(Refleq((Misset_1
>>                            Mbold2 thelawchooses_1)))
>>                            Sepsub2 Refleq((Misset_1
>>                            Cuts3 thelawchooses_1)))))
>>                            Conj fhyp4_33) Mp
>>                            (F4_31 Ui (D4_25
>>                            Ui Simp2(Simp2(Simp2((Misset_1
>>                            Mboldtheta2 thelawchooses_1)))))))
>>                            Conj Ug([(D5_76:obj)
>>                               => (Ded([(dhyp5_77:
>>                                  that (D5_76
>>                                  E (Misset_1
>>                                  Mbold2 thelawchooses_1)))
>>                                  => (Cases(Excmid(Forall([(D6_80:
>>                                     obj) =>
>>                                     (((D6_80
>>                                     E D4_25)
>>                                     -> (D5_76
>>                                     <<= D6_80)):
>>                                     prop)]))
>>                                  ,[(casehyp1_81:
>>                                     that Forall([(D7_82:
>>                                        obj)
>>                                        => (((D7_82
>>                                        E D4_25)
>>                                        -> (D5_76
>>                                        <<= D7_82)):
>>                                        prop)]))
>>                                     => ((((D4_25
>>                                     Intersection
>>                                     F4_31) <<=
>>                                     D5_76) Add1
>>                                     ((D5_76
>>                                     <<= (D4_25
>>                                     Intersection
>>                                     F4_31))
>>                                     Fixform
>>                                     (Ug([(G_85:
```

```
>>                                       obj)
>>                                       => (Ded([(ghyp_86:
>>                                          that (G_85
>>                                          E D5_76))
>>                                          => (((G_85
>>                                          E (D4_25
>>                                          Intersection
>>                                          F4_31))
>>                                          Fixform
>>                                          (((fhyp4_33
>>                                          Mp (F4_31
>>                                          Ui Ug([(B1_90:
>>                                             obj)
>>                                             => (Ded([(bhyp1_91:
>>                                                that (B1_90
>>                                                E D4_25))
>>                                                => ((ghyp_86
>>                                                Mpsubs
>>                                                (bhyp1_91
>>                                                Mp (B1_90
>>                                                Ui casehyp1_81))):
>>                                                that (G_85
>>                                                E B1_90))])
>>                                             :that ((B1_90
>>                                             E D4_25) ->
>>                                             (G_85 E B1_90)))]))
>>                                          ) Conj Ug([(B1_95:
>>                                             obj) => (Ded([(bhyp1_96:
>>                                                that (B1_95
>>                                                E D4_25))
>>                                                => ((ghyp_86
>>                                                Mpsubs
>>                                                (bhyp1_96
>>                                                Mp (B1_95
>>                                                Ui casehyp1_81))):
>>                                                that (G_85
>>                                                E B1_95))])
>>                                             :that ((B1_95
>>                                             E D4_25) ->
>>                                             (G_85 E B1_95)))]))
>>                                          Iff2 (G_85 Ui
>>                                          Separation4(Refleq((D4_25
>>                                          Intersection
>>                                          F4_31)))))):that
>>                                          (G_85 E (D4_25
>>                                          Intersection
>>                                          F4_31)))])
>>                                       :that
>>                                       ((G_85
>>                                       E D5_76)
>>                                       -> (G_85
>>                                       E (D4_25
>>                                       Intersection
>>                                       F4_31))))])
>>                                    Conj (Setsinchains2(Misset_1,
>>                                    thelawchooses_1,
>>                                    (Misset_1
>>                                    Mboldtheta2
>>                                    thelawchooses_1),
>>                                    dhyp5_77)
>>                                    Conj Separation3(Refleq((D4_25
>>                                    Intersection
>>                                    F4_31))))))):
>>                                    that ((D5_76
>>                                    <<= (D4_25
>>                                    Intersection
>>                                    F4_31))
>>                                    V ((D4_25
>>                                    Intersection
>>                                    F4_31) <<=
>>                                    D5_76)))]
>>                                 ,[(casehyp2_112:
>>                                    that ~(Forall([(D7_113:
>>                                       obj)
>>                                       => (((D7_113
>>                                       E D4_25)
>>                                         -> (D5_76
>>                                         <<= D7_113)):
>>                                         prop)]))
>>                                      ) => (((D5_76
>>                                      <<= (D4_25
>>                                      Intersection
>>                                      F4_31))
>>                                      Add2 (((D4_25
>>                                      Intersection
>>                                      F4_31) <<=
>>                                      D5_76) Fixform
>>                                      (Ug([(G_116:
>>                                         obj)
>>                                         => (Ded([(ghyp_117:
>>                                            that (G_116
>>                                            E (D4_25
>>                                            Intersection
>>                                            F4_31)))
>>                                            => ((Counterexample(casehyp2_112)
>>                                            Eg [(.H_120:
>>                                               obj),
>>                                               (hhyp_120:
>>                                               that
>>                                               (Counterexample(casehyp2_112)
>>                                               Witnesses
>>                                               .H_120))
>>                                               => (((Notimp2(hhyp_120)
>>                                               Mp (.H_120
>>                                               Ui Simp2((ghyp_117
>>                                               Iff1
>>                                               (G_116
>>                                               Ui Separation4(Refleq((D4_25
>>                                               Intersection
>>                                               F4_31)))))))
>>                                               Mpsubs
>>                                               ((dhyp5_77
>>                                               Mp (D5_76
>>                                               Ui Simp2(Simp2(((Notimp2(hhyp_120)
>>                                             Mpsubs
>>                                             dhyp4_28)
>>                                             Iff1
>>                                             (.H_120
>>                                             Ui Separation4(Refleq((Misset_1
>>                                             Cuts3
>>                                             thelawchooses_1)))))))))
>>                                             Ds2
>>                                             Notimp1(hhyp_120))):
>>                                             that
>>                                             (G_116
>>                                             E D5_76))])
>>                                          :that (G_116
>>                                          E D5_76))])
>>                                       :that
>>                                       ((G_116
>>                                       E (D4_25
>>                                       Intersection
>>                                       F4_31))
>>                                       -> (G_116
>>                                       E D5_76)))])
>>                                    Conj (Separation3(Refleq((D4_25
>>                                    Intersection
>>                                    F4_31)))
>>                                    Conj Setsinchains2(Misset_1,
>>                                    thelawchooses_1,
>>                                    (Misset_1
>>                                    Mboldtheta2
>>                                    thelawchooses_1),
>>                                    dhyp5_77))))):
>>                                    that ((D5_76
>>                                    <<= (D4_25
>>                                    Intersection
>>                                    F4_31))
>>                                    V ((D4_25
>>                                    Intersection
>>                                    F4_31) <<=
>>                                    D5_76)))])
>>                                 :that ((D5_76
>>                                 <<= (D4_25
>>                                 Intersection
>>                                 F4_31)) V ((D4_25
>>                                 Intersection
>>                                 F4_31) <<=
>>                                 D5_76)))])
>>                              :that ((D5_76
>>                              E (Misset_1 Mbold2
>>                              thelawchooses_1))
>>                              -> ((D5_76 <<=
>>                              (D4_25 Intersection
>>                              F4_31)) V ((D4_25
>>                              Intersection F4_31)
>>                              <<= D5_76))))]))
>>                           ) Iff2 ((D4_25 Intersection
>>                           F4_31) Ui Separation4(Refleq((Misset_1
>>                           Cuts3 thelawchooses_1))))):
>>                           that ((D4_25 Intersection
>>                           F4_31) E ((Misset_1
>>                           Mbold2 thelawchooses_1)
>>                           Set [(C_147:obj)
>>                              => (cuts2(Misset_1,
>>                              thelawchooses_1,
>>                              C_147):prop)]))
>>                           )])
>>                        :that ((F4_31 E D4_25)
>>                        -> ((D4_25 Intersection
>>                        F4_31) E ((Misset_1
>>                        Mbold2 thelawchooses_1)
>>                        Set [(C_148:obj) =>
>>                           (cuts2(Misset_1,thelawchooses_1,
>>                           C_148):prop)]))
>>                        ))])
>>                     :that Forall([(F4_149:obj)
>>                        => (((F4_149 E D4_25)
>>                        -> ((D4_25 Intersection
>>                        F4_149) E ((Misset_1
>>                        Mbold2 thelawchooses_1)
>>                        Set [(C_150:obj) =>
>>                           (cuts2(Misset_1,thelawchooses_1,
>>                           C_150):prop)]))
>>                        ):prop)]))
>>                     ])
>>                  :that ((D4_25 <<= (Misset_1
>>                  Cuts3 thelawchooses_1)) ->
>>                  Forall([(F4_151:obj) => (((F4_151
>>                     E D4_25) -> ((D4_25 Intersection
>>                     F4_151) E ((Misset_1 Mbold2
>>                     thelawchooses_1) Set [(C_152:
>>                        obj) => (cuts2(Misset_1,
>>                        thelawchooses_1,C_152):
>>                        prop)]))
>>                     ):prop)]))
>>                  )]))
>>               ))):that ((D9_8 Intersection
>>               F9_11) E ((Misset_1 Mbold2 thelawchooses_1)
>>               Set [(C_153:obj) => (cuts2(Misset_1,
>>                  thelawchooses_1,C_153):prop)]))
>>               )])
>>            :that (((D9_8 <<= (Misset_1 Cuts3
>>            thelawchooses_1)) & (F9_11 E D9_8))
>>            -> ((D9_8 Intersection F9_11) E
>>            ((Misset_1 Mbold2 thelawchooses_1)
>>            Set [(C_154:obj) => (cuts2(Misset_1,
>>               thelawchooses_1,C_154):prop)]))
>>            ))])
>>         :that Forall([(F9_155:obj) => ((((D9_8
>>            <<= (Misset_1 Cuts3 thelawchooses_1))
>>            & (F9_155 E D9_8)) -> ((D9_8 Intersection
>>            F9_155) E ((Misset_1 Mbold2 thelawchooses_1)
>>            Set [(C_156:obj) => (cuts2(Misset_1,
>>               thelawchooses_1,C_156):prop)]))
>>            ):prop)]))
>>         ])
>>      :that Forall([(D9_157:obj) => (Forall([(F9_158:
```

```
>>             obj) => ((((D9_157 <<= (Misset_1
>>             Cuts3 thelawchooses_1)) & (F9_158
>>             E D9_157)) -> ((D9_157 Intersection
>>             F9_158) E ((Misset_1 Mbold2 thelawchooses_1)
>>             Set [(C_159:obj) => (cuts2(Misset_1,
>>                thelawchooses_1,C_159):prop)]))
>>             ):prop)])
>>          :prop)]))
>>       ]
>>    {move 0}
```

```
open

   define line155: lineb155 Misset, thelawchooses


>>    line155: [(---:that Forall([(D9_1:obj)
>>             => (Forall([(F9_2:obj) => ((((D9_1
>>                <<= (Misset Cuts3 thelawchooses))
>>                & (F9_2 E D9_1)) -> ((D9_1 Intersection
>>                F9_2) E ((Misset Mbold2 thelawchooses)
>>                Set [(C_3:obj) => (cuts2(Misset,
>>                   thelawchooses,C_3):prop)]))
>>                ):prop)])
>>             :prop)]))
>>          ]
>>       {move 1}
```

This is the fourth component of the proof that `Cuts` is a Θ-chain.

```
Lestrade execution:


   define Cutstheta2: Fixform(thetachain(Cuts), \
      Line9 Conj Line12 Conj Line119 Conj line155)


>>    Cutstheta2: [(---:that thetachain(Cuts))]
>>      {move 1}



   close

define Cutstheta Misset thelawchooses: Cutstheta2


>> Cutstheta: [(.M_1:obj),(Misset_1:that Isset(.M_1)),
>>      (.thelaw_1:[(S_2:obj) => (---:obj)]),
>>      (thelawchooses_1:[(.S_3:obj),(subsetev_3:
>>         that (.S_3 <<= .M_1)),(inev_3:that
>>         Exists([(x_4:obj) => ((x_4 E .S_3):
>>            prop)]))
>>         => (---:that (.thelaw_1(.S_3) E .S_3))])
>>      => ((thetachain1(.M_1,.thelaw_1,(Misset_1
>>      Cuts3 thelawchooses_1)) Fixform (((.M_1
>>      E (Misset_1 Cuts3 thelawchooses_1)) Fixform
>>      ((Simp1((Misset_1 Mboldtheta2 thelawchooses_1))
>>      Conj (cuts2(Misset_1,thelawchooses_1,.M_1)
>>      Fixform (Simp1((Misset_1 Mboldtheta2 thelawchooses_1))
>>      Conj Ug([(F_13:obj) => (Ded([(finmbold_14:
>>            that (F_13 E (Misset_1 Mbold2 thelawchooses_1)))
>>            => (((.M_1 <<= F_13) Add1 ((finmbold_14
>>            Mp (F_13 Ui Simp1(Simp1(Simp2((Misset_1
>>            Mboldtheta2 thelawchooses_1))))))
>>            Iff1 (F_13 Ui Scthm(.M_1)))):that
>>            ((F_13 <<= .M_1) V (.M_1 <<= F_13)))])
>>         :that ((F_13 E (Misset_1 Mbold2 thelawchooses_1))
>>         -> ((F_13 <<= .M_1) V (.M_1 <<= F_13))))]))
>>      )) Iff2 (.M_1 Ui ((Misset_1 Mbold2 thelawchooses_1)
>>      Separation [(C_27:obj) => (cuts2(Misset_1,
>>         thelawchooses_1,C_27):prop)]))
>>      )) Conj (((((Misset_1 Cuts3 thelawchooses_1)
>>      <<= (Misset_1 Mbold2 thelawchooses_1))
>>      Fixform Sepsub((Misset_1 Mbold2 thelawchooses_1),
>>      [(C_32:obj) => (cuts2(Misset_1,thelawchooses_1,
>>         C_32):prop)]
>>      ,Inhabited(Simp1((Misset_1 Mboldtheta2
>>      thelawchooses_1))))) Transsub (((Misset_1
>>      Mbold2 thelawchooses_1) <<= Sc(.M_1))
>>      Fixform (Sc2(.M_1) Sepsub2 Refleq((Misset_1
>>      Mbold2 thelawchooses_1))))) Conj ((Misset_1
>>      Lineb119 thelawchooses_1) Conj (Misset_1
>>      lineb155 thelawchooses_1))))):that thetachain1(.M_1,
>>      .thelaw_1,(Misset_1 Cuts3 thelawchooses_1)))]
>>   {move 0}


clearcurrent
```

This is the proof that `Cuts` is a Θ-chain. Suppressing definitional expansion of its four components has made it somewhat manageable in size.

Since I clear move 1 above, a number of convenient definitions are restated.

```
Lestrade execution:



save

declare M obj

>> M: obj {move 1}




declare Misset that Isset M

>> Misset: that Isset(M) {move 1}
```

```
open

   declare S obj

>>    S: obj {move 2}



   declare x obj

>>    x: obj {move 2}



   declare subsetev that S <<= M

>>    subsetev: that (S <<= M) {move 2}



   declare inev that Exists [x => x E S] \




>>    inev: that Exists([(x_1:obj) => ((x_1
>>         E S):prop)])
>>      {move 2}



   postulate thelaw S : obj

>>    thelaw: [(S_1:obj) => (---:obj)]
>>      {move 1}
```

```
   postulate thelawchooses subsetev inev \
      : that (thelaw S) E S

>>    thelawchooses: [(.S_1:obj),(subsetev_1:
>>         that (.S_1 <<= M)),(inev_1:that Exists([(x_2:
>>            obj) => ((x_2 E .S_1):prop)]))
>>         => (---:that (thelaw(.S_1) E .S_1))]
>>      {move 1}



   open

      define Mbold: Mbold2 Misset thelawchooses


>>       Mbold: [(---:obj)]
>>         {move 2}



      declare X obj

>>       X: obj {move 3}



      define thetachain X: thetachain1 M \
         thelaw, X

>>       thetachain: [(X_1:obj) => (---:prop)]
>>         {move 2}
```

```
      define Thetachain: Set (Sc(Sc M),thetachain)


>>       Thetachain: [(---:obj)]
>>         {move 2}



      open

         declare Y obj

>>          Y: obj {move 4}



         declare theta1 that thetachain Y


>>          theta1: that thetachain(Y) {move
>>            4}



         declare theta2 that Y E Thetachain


>>          theta2: that (Y E Thetachain) {move
>>            4}



         define thetaa1 theta1: Iff2(Simp1 \
            Simp2 theta1,Ui Y, Scthm Sc M)


>>          thetaa1: [(.Y_1:obj),(theta1_1:that
>>               thetachain(.Y_1)) => (---:that
>>               (.Y_1 E Sc(Sc(M))))]
>>            {move 3}



         define Theta1 theta1: Iff2(Conj(thetaa1 \
            theta1,theta1),Ui Y,Separation4 \
            Refleq Thetachain)

>>          Theta1: [(.Y_1:obj),(theta1_1:that
>>               thetachain(.Y_1)) => (---:that
>>               (.Y_1 E (Sc(Sc(M)) Set thetachain)))]
>>            {move 3}



         define Theta2 theta2: Simp2(Iff1(theta2, \
            Ui Y,Separation4 Refleq Thetachain))


>>          Theta2: [(.Y_1:obj),(theta2_1:that
>>               (.Y_1 E Thetachain)) => (---:
>>               that thetachain(.Y_1))]
>>            {move 3}



         close

      define Cutstheta1: Cutstheta Misset \
         thelawchooses

>>       Cutstheta1: [(---:that thetachain1(M,
>>            thelaw,(Misset Cuts3 thelawchooses)))]
>>         {move 2}
```

```
      define Cuts: Misset Cuts3 thelawchooses


>>       Cuts: [(---:obj)]
>>         {move 2}



      declare A obj

>>       A: obj {move 3}



      declare B obj

>>       B: obj {move 3}



      declare aev that A E Mbold

>>       aev: that (A E Mbold) {move 3}



      declare bev that B E Mbold

>>       bev: that (B E Mbold) {move 3}



      goal that (A <<= B) V B <<= A

>>       Goal: that ((A <<= B) V (B <<= A))


      define line1 aev: Fixform(Forall[X=>(X \
            E Thetachain)->A E X] \
         ,Simp2(Iff1(aev,Ui A, Separation4 \
         Refleq Mbold)))

>>       line1: [(.A_1:obj),(aev_1:that (.A_1
>>            E Mbold)) => (---:that Forall([(X_16:
>>               obj) => (((X_16 E Thetachain)
>>               -> (.A_1 E X_16)):prop)]))
>>            ]
>>         {move 2}



      define Mboldtotal aev bev: Mp bev, \
         Ui B,Simp2(Simp2(Iff1(Mp(Theta1 Cutstheta1, \
         Ui Cuts, line1 aev),Ui A,Separation4 \
         Refleq Cuts)))

>>       Mboldtotal: [(.A_1:obj),(aev_1:that
>>            (.A_1 E Mbold)),(.B_1:obj),(bev_1:
>>            that (.B_1 E Mbold)) => (---:that
>>            ((.B_1 <<= .A_1) V (.A_1 <<= .B_1)))]
>>         {move 2}



      define prime A: prime2 thelaw,A

>>       prime: [(A_1:obj) => (---:obj)]
>>         {move 2}



      define Mboldstrongtotal aev bev: Fixform((B \
         <<= prime A) V A <<= B,Simp2( Separation5 \
         Univcheat ( Theta1 linec107 Mp(Theta1 \
         Cutstheta1, Ui Cuts, line1 aev),line1 \
         bev)))
```

```
>>       Mboldstrongtotal: [(.A_1:obj),(aev_1:
>>            that (.A_1 E Mbold)),(.B_1:obj),
>>            (bev_1:that (.B_1 E Mbold)) => (---:
>>            that ((.B_1 <<= prime(.A_1)) V (.A_1
>>            <<= .B_1)))]
>>         {move 2}



      save

      close

   declare A1 obj

>>    A1: obj {move 2}



   declare B1 obj

>>    B1: obj {move 2}



   declare aev1 that A1 E Mbold

>>    aev1: that (A1 E Mbold) {move 2}



   declare bev1 that B1 E Mbold

>>    bev1: that (B1 E Mbold) {move 2}
```

```
   define Mboldtotal1 aev1 bev1:Mboldtotal \
      aev1 bev1

>>    Mboldtotal1: [(.A1_1:obj),(aev1_1:that
>>         (.A1_1 E (Misset Mbold2 thelawchooses))),
>>         (.B1_1:obj),(bev1_1:that (.B1_1 E (Misset
>>         Mbold2 thelawchooses))) => (---:that
>>         ((.B1_1 <<= .A1_1) V (.A1_1 <<= .B1_1)))]
>>      {move 1}



   define Mboldstrongtotal1 aev1 bev1:Mboldstrongtotal \
      aev1 bev1

>>    Mboldstrongtotal1: [(.A1_1:obj),(aev1_1:
>>         that (.A1_1 E (Misset Mbold2 thelawchooses))),
>>         (.B1_1:obj),(bev1_1:that (.B1_1 E (Misset
>>         Mbold2 thelawchooses))) => (---:that
>>         ((.B1_1 <<= prime2(thelaw,.A1_1)) V
>>         (.A1_1 <<= .B1_1)))]
>>      {move 1}



   save

   close

declare A2 obj

>> A2: obj {move 1}



declare B2 obj

>> B2: obj {move 1}
```

```
declare aev2 that A2 E (Mbold2 Misset thelawchooses)


>> aev2: that (A2 E (Misset Mbold2 thelawchooses))
>>   {move 1}



declare bev2 that B2 E (Mbold2 Misset thelawchooses)


>> bev2: that (B2 E (Misset Mbold2 thelawchooses))
>>   {move 1}




define Mboldtotal2 Misset thelawchooses, \
   aev2 bev2: Mboldtotal1 aev2 bev2

>> Mboldtotal2: [(.M_1:obj),(Misset_1:that Isset(.M_1)),
>>      (.thelaw_1:[(S_2:obj) => (---:obj)]),
>>      (thelawchooses_1:[(.S_3:obj),(subsetev_3:
>>         that (.S_3 <<= .M_1)),(inev_3:that
>>         Exists([(x_4:obj) => ((x_4 E .S_3):
>>            prop)]))
>>         => (---:that (.thelaw_1(.S_3) E .S_3))]),
>>      (.A2_1:obj),(aev2_1:that (.A2_1 E (Misset_1
>>      Mbold2 thelawchooses_1))),(.B2_1:obj),
>>      (bev2_1:that (.B2_1 E (Misset_1 Mbold2
>>      thelawchooses_1))) => ((bev2_1 Mp (.B2_1
>>      Ui Simp2(Simp2((((((Simp1(Simp2((Misset_1
>>      Cutstheta thelawchooses_1))) Iff2 ((Misset_1
>>      Cuts3 thelawchooses_1) Ui Scthm(Sc(.M_1))))
>>      Conj (Misset_1 Cutstheta thelawchooses_1))
>>      Iff2 ((Misset_1 Cuts3 thelawchooses_1)
>>      Ui Separation4(Refleq((Sc(Sc(.M_1)) Set
>>      [(X_20:obj) => (thetachain1(.M_1,.thelaw_1,
>>         X_20):prop)]))
>>      ))) Mp ((Misset_1 Cuts3 thelawchooses_1)
>>      Ui (Forall([(X_23:obj) => (((X_23 E (Sc(Sc(.M_1))
>>         Set [(X_24:obj) => (thetachain1(.M_1,
>>            .thelaw_1,X_24):prop)]))
>>         -> (.A2_1 E X_23)):prop)])
>>      Fixform Simp2((aev2_1 Iff1 (.A2_1 Ui Separation4(Refleq((Misset_1
>>      Mbold2 thelawchooses_1)))))))) Iff1 (.A2_1
>>      Ui Separation4(Refleq((Misset_1 Cuts3
>>      thelawchooses_1))))))))):that ((.B2_1
>>      <<= .A2_1) V (.A2_1 <<= .B2_1)))]
>>   {move 0}
```

```
define Mboldstrongtotal2 Misset thelawchooses, \
   aev2 bev2: Mboldstrongtotal1 aev2 bev2

>> Mboldstrongtotal2: [(.M_1:obj),(Misset_1:
>>      that Isset(.M_1)),(.thelaw_1:[(S_2:obj)
>>         => (---:obj)]),
>>      (thelawchooses_1:[(.S_3:obj),(subsetev_3:
>>         that (.S_3 <<= .M_1)),(inev_3:that
>>         Exists([(x_4:obj) => ((x_4 E .S_3):
>>            prop)]))
>>         => (---:that (.thelaw_1(.S_3) E .S_3))]),
>>      (.A2_1:obj),(aev2_1:that (.A2_1 E (Misset_1
>>      Mbold2 thelawchooses_1))),(.B2_1:obj),
>>      (bev2_1:that (.B2_1 E (Misset_1 Mbold2
>>      thelawchooses_1))) => ((((.B2_1 <<= prime2(.thelaw_1,
>>      .A2_1)) V (.A2_1 <<= .B2_1)) Fixform Simp2(Separation5(((((Simp1(Simp2(lin
>>      Cutstheta thelawchooses_1))) Iff2 ((Misset_1
>>      Cuts3 thelawchooses_1) Ui Scthm(Sc(.M_1))))
>>      Conj (Misset_1 Cutstheta thelawchooses_1))
>>      Iff2 ((Misset_1 Cuts3 thelawchooses_1)
>>      Ui Separation4(Refleq((Sc(Sc(.M_1)) Set
>>      [(X_42:obj) => (thetachain1(.M_1,.thelaw_1,
>>         X_42):prop)]))
>>      ))) Mp ((Misset_1 Cuts3 thelawchooses_1)
>>      Ui (Forall([(X_45:obj) => (((X_45 E (Sc(Sc(.M_1))
>>         Set [(X_46:obj) => (thetachain1(.M_1,
>>            .thelaw_1,X_46):prop)]))
>>         -> (.A2_1 E X_45)):prop)])
>>      Fixform Simp2((aev2_1 Iff1 (.A2_1 Ui Separation4(Refleq((Misset_1
>>      Mbold2 thelawchooses_1)))))))))))) Iff2
>>      (((Misset_1 Mbold2 thelawchooses_1) Set
>>      [(Y_61:obj) => (cutse2(Misset_1,thelawchooses_1,
>>         .A2_1,Y_61):prop)])
>>      Ui Scthm(Sc(.M_1)))) Conj linec107(((((Simp1(Simp2((Misset_1
>>      Cutstheta thelawchooses_1))) Iff2 ((Misset_1
>>      Cuts3 thelawchooses_1) Ui Scthm(Sc(.M_1))))
>>      Conj (Misset_1 Cutstheta thelawchooses_1))
>>      Iff2 ((Misset_1 Cuts3 thelawchooses_1)
>>      Ui Separation4(Refleq((Sc(Sc(.M_1)) Set
>>      [(X_77:obj) => (thetachain1(.M_1,.thelaw_1,
>>         X_77):prop)]))
>>      ))) Mp ((Misset_1 Cuts3 thelawchooses_1)
>>      Ui (Forall([(X_80:obj) => (((X_80 E (Sc(Sc(.M_1))
>>         Set [(X_81:obj) => (thetachain1(.M_1,
>>            .thelaw_1,X_81):prop)]))
>>         -> (.A2_1 E X_80)):prop)])
>>      Fixform Simp2((aev2_1 Iff1 (.A2_1 Ui Separation4(Refleq((Misset_1
>>      Mbold2 thelawchooses_1))))))))))) Iff2
>>      (((Misset_1 Mbold2 thelawchooses_1) Set
>>      [(Y_97:obj) => (cutse2(Misset_1,thelawchooses_1,
>>         .A2_1,Y_97):prop)])
>>      Ui Separation4(Refleq((Sc(Sc(.M_1)) Set
>>      [(X_102:obj) => (thetachain1(.M_1,.thelaw_1,
>>         X_102):prop)]))
>>      ))) Univcheat (Forall([(X_104:obj) =>
>>         (((X_104 E (Sc(Sc(.M_1)) Set [(X_105:
>>            obj) => (thetachain1(.M_1,.thelaw_1,
>>            X_105):prop)]))
>>         -> (.B2_1 E X_104)):prop)])
>>      Fixform Simp2((bev2_1 Iff1 (.B2_1 Ui Separation4(Refleq((Misset_1
>>      Mbold2 thelawchooses_1))))))))))):that
>>      ((.B2_1 <<= prime2(.thelaw_1,.A2_1)) V
>>      (.A2_1 <<= .B2_1)))]
>>   {move 0}
```

We deliver results on the total linear ordering of **M** by the inclusion relation. Notice that we also prove the stronger result embodied in `Cuts2`.