```
begin Lestrade execution



                  {move 6}

                  >>> define linex14 D2 : Ug \
                      linea13

line14 is badly formed or already reserved or declared

(paused, type something to continue) >

                  >>> close


               {move 5}

               >>> define linex15 : Ug linex14

line15 is badly formed or already reserved or declared

(paused, type something to continue) >
end Lestrade execution
```

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.

```
begin Lestrade execution

               >>> close
```

```
            {move 4}

            >>> define linex17 bhyp : Fixform \
                (thetachain Cuts2, Conj (line19, Conj \
                (line21, Conj (line78, linex15))))

line17 is badly formed or already reserved or declared



(paused, type something to continue) >

            >>> save


            {move 4}

            >>> close


         {move 3}

         >>> declare bhyp10 that B E Cuts


         bhyp10 : that B E Cuts


         {move 3}

         >>> define linea17 bhyp10 : linex17 \
             bhyp10

[bhyp10 => line17 bhyp10] is not well-formed

(paused, type something to continue) >
```

```
   >>> save

   {move 3}

   >>> close

{move 2}

>>> declare B11 obj

B11 : obj

{move 2}

>>> declare bhyp11 that B11 E Cuts

bhyp11 : that B11 E Cuts

{move 2}

>>> define lineb17 bhyp11 : linea17 \
    bhyp11

[bhyp11 => linea17 bhyp11] is not well-formed

(paused, type something to continue) >

>>> save

{move 2}
```

```
      >>> close


   {move 1}

   >>> declare B12 obj


   B12 : obj


   {move 1}

   >>> declare bhyp12 that B12 E Cuts


   bhyp12 : that B12 E Cuts


   {move 1}

   >>> define linec17 bhyp12 : lineb17 bhyp12

[bhyp12 => lineb17 bhyp12] is not well-formed

(paused, type something to continue) >

   >>> open


      {move 2}

      >>> define lined17 bhyp11 : linec17 \
          bhyp11

[bhyp11 => linec17 bhyp11] is not well-formed

(paused, type something to continue) >
```

```
      >>> open

         {move 3}

         >>> declare B13 obj

         B13 : obj

         {move 3}

         >>> declare bhyp13 that B13 E Cuts

         bhyp13 : that B13 E Cuts

         {move 3}

         >>> define linee17 bhyp13 : lined17 \
             bhyp13

[bhyp13 => lined17 bhyp13] is not well-formed

(paused, type something to continue) >

         >>> open

            {move 4}

            >>> define Line17 bhyp : linee17 \
                bhyp

[bhyp => linee17 bhyp] is not well-formed
```

```
(paused, type something to continue) >

            >>> open

               {move 5}

               >>> declare K obj

               K : obj

               {move 5}

               >>> open

                  {move 6}

                  >>> declare khyp that K E Mbold

                  khyp : that K E Mbold

                  {move 6}

                  >>> define linex18 khyp \
                      : Ui Cuts2, Simp2 (Iff1 \
                      (khyp, Ui K, Separation4 \
                      Refleq Mbold))

line18 is badly formed or already reserved or declared

(paused, type something to continue) >
```

```
                  >>> define linea18 : Iff2 \
                      (Simp1 (Simp2 Line17 \
                      bhyp), Ui Cuts2, Scthm \
                      (Sc M))

Iff2 (Simp1 (Simp2 Line17 bhyp), Ui Cuts2, Scthm (Sc M)) is not well-formed

(paused, type something to continue) >

                  >>> define linex19 : Fixform \
                      (Cuts2 E Thetachain, Iff2 \
                      (Conj (linea18, Line17 \
                      bhyp), Ui Cuts2, Separation4 \
                      Refleq Thetachain))

line19 is badly formed or already reserved or declared

(paused, type something to continue) >
end Lestrade execution
```

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.

```
begin Lestrade execution

                  >>> define line110 khyp \
                      : Mp (linex19, linex18 \
                      khyp)

[khyp => Mp (line19, line18 khyp)] is not well-formed

(paused, type something to continue) >

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

```
[khyp => Iff1 (line110 khyp, Ui K, Separation4 Refleq Cuts2)] is not well-forme

(paused, type something to continue) >

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


                  line112 : [
                      ({def} (prime (B) <<=
                      B) Fixform linea14
                      (bhyp) Sepsub2 Refleq
                      (prime (B)) : that
                      prime (B) <<= B)]


                  line112 : that prime (B) <<=
                   B


                  {move 5}

                  >>> define line113 khyp \
                      : Simp2 line111 khyp

[khyp => Simp2 line111 khyp] is not well-formed

(paused, type something to continue) >

                  >>> open


                     {move 7}

                     >>> 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)) =>
    ({def} (prime (B) <<=
    K) Add1 casehyp1_1
    : that (K <<= prime
    (B)) V prime (B) <<=
    K)]


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) =>
    ({def} (K <<= prime
    (B)) Add2 line112
    Transsub casehyp2_1
    : that (K <<= prime
    (B)) V prime (B) <<=
    K)]


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


{move 6}

>>> close


{move 6}

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

```
[khyp => Cases (line113 khyp, case1, case2)] is not well-formed

(paused, type something to continue) >

                  >>> close


               {move 5}

               >>> define line115 K : Ded \
                   line114

[K => Ded line114] is not well-formed

(paused, type something to continue) >

               >>> close


            {move 4}

            >>> define line116 bhyp : Ug \
                line115

[bhyp => Ug line115] is not well-formed

(paused, type something to continue) >

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


            linea116 : [(bhyp_1 : that
                B E Cuts) =>
                ({def} line14 (bhyp_1) Mp
                B Ui Simp1 (Simp2 (Simp2
                (Mboldtheta))) : that
```

```
        prime2 ([(S'_3 : obj) =>
           ({def} thelaw (S'_3) : obj)], B) E Misset
        Mbold2 thelawchooses)]


    linea116 : [(bhyp_1 : that
        B E Cuts) => (--- : that
        prime2 ([(S'_3 : obj) =>
           ({def} thelaw (S'_3) : obj)], 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)))

[bhyp => Fixform ((prime B) E Cuts, Iff2 (Conj (linea116 bhyp, Conj (linea116 b

(paused, type something to continue) >

    >>> close


   {move 3}

   >>> define line118 B : Ded line117

[B => Ded line117] is not well-formed

(paused, type something to continue) >

   >>> close
```

```
      {move 2}

      >>> define Linea119 : Ug line118

Ug line118 is not well-formed

(paused, type something to continue) >

      >>> close


   {move 1}

   >>> define Lineb119 Misset, thelawchooses \
        : Linea119

[Misset thelawchooses => Linea119] is not well-formed

(paused, type something to continue) >

   >>> open


      {move 2}

      >>> define Line119 : Lineb119 Misset, \
          thelawchooses

Lineb119 Misset thelawchooses is not well-formed

(paused, type something to continue) >
end Lestrade execution
```

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 $\mathbf{M}$).

```
begin 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]]


      {error type}


      {move 2}

      >>> open


         {move 3}

         >>> declare D4 obj


         D4 : obj
```

```
{move 3}

>>> open


   {move 4}

   >>> declare dhyp4 that D4 <<= \
       Cuts


   dhyp4 : that D4 <<= Cuts


   {move 4}

   >>> open


      {move 5}

      >>> declare F4 obj


      F4 : obj


      {move 5}

      >>> open


         {move 6}

         >>> declare fhyp4 that \
             F4 E D4
```

```
                  fhyp4 : that F4 E D4


                  {move 6}

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

{function error}

general failure of functionsort line 3030

(paused, type something to continue) >

                  {move 6}

                  >>> goal that D4 Intersection \
                      F4 E Mbold

Failure in comparing prop to obj line 3073

(paused, type something to continue) >
Object type error in D4 Intersection F4 E Mbold

(paused, type something to continue) >
general failure of objectsort line 2989

(paused, type something to continue) >
bad proof/evidence type, body not prop line 3913

(paused, type something to continue) >

                  {error type}


                  {move 6}
```

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

{function error}

general failure of functionsort line 3030

(paused, type something to continue) >

                  {move 6}

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


                  line120 : [
                      ({def} dhyp4 Transsub
                      (Cuts <<= Mbold) Fixform
                      Separation3 (Refleq
                      (Mbold)) Sepsub2
                      Refleq (Cuts) : that
                      D4 <<= Mbold)]


                  line120 : that D4 <<= Mbold


                  {move 5}

                  >>> define line121 fhyp4 \
                      : Mpsubs fhyp4 line120


                  line121 : [(fhyp4_1 : that
```

```
    F4 E D4) =>
    ({def} fhyp4_1 Mpsubs
    line120 : that F4 E Mbold)]


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) =>
    ({def} line120 Conj
    fhyp4_1 Mp F4 Ui D4
    Ui Simp2 (Simp2 (Simp2
    (Mboldtheta))) : that
    (D4 Intersection F4) E Misset
    Mbold2 thelawchooses)]


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


{move 5}

>>> goal that cuts (D4 \
    Intersection F4)
```

```
                  that cuts (D4 Intersection
                   F4)


                  {move 6}

                  >>> declare testing that \
                      cuts (D4 Intersection \
                      F4)


                  testing : that cuts (D4
                   Intersection F4)


                  {move 6}

                  >>> test Simp1 (testing)

{function error}

general failure of functionsort line 3030

(paused, type something to continue) >

                  {move 6}

                  >>> test Simp2 (testing)

{function error}

general failure of functionsort line 3030

(paused, type something to continue) >

                  {move 6}
```

```
>>> open

   {move 7}

   >>> declare D5 obj


   D5 : obj


   {move 7}

   >>> open


      {move 8}

      >>> declare dhyp5 \
          that D5 E Mbold


      dhyp5 : that D5 E Mbold


      {move 8}

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


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

```
{move 8}

>>> declare D6 obj


D6 : obj


{move 8}

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


line123 : [
    ({def} Excmid
    (Forall ([(D6_3
       : obj) =>
       ({def} (D6_3
       E D4) -> D5
       <<= D6_3 : prop)])) : that
    Forall ([(D6_3
       : obj) =>
       ({def} (D6_3
       E D4) -> D5
       <<= D6_3 : prop)]) V ~ (Forall
    ([(D6_4 : obj) =>
       ({def} (D6_4
       E D4) -> D5
       <<= D6_4 : prop)])))]


line123 : that Forall
 ([(D6_3 : obj) =>
```

```
    ({def} (D6_3
    E D4) -> D5 <<=
    D6_3 : prop)]) V ~ (Forall
 ([(D6_4 : obj) =>
    ({def} (D6_4
    E D4) -> D5 <<=
    D6_4 : prop)]))


{move 7}

>>> open


   {move 9}

   >>> declare D7 \
       obj


   D7 : obj


   {move 9}

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


   casehyp1 : that
    Forall ([(D7_2
       : obj) =>
       ({def} (D7_2
       E D4) -> D5
       <<= D7_2 : prop)])
```

```
{move 9}

>>> open


   {move 10}

   >>> declare \
       G obj


   G : obj


   {move 10}

   >>> open


      {move 11}

      >>> declare \
          ghyp that \
          G E D5


      ghyp : that
       G E D5


      {move 11}

      >>> goal \
          that G E D4 \
          Intersection \
          F4
```

```
that G E D4
 Intersection
 F4


{move 11}

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

```
{function error}

general failure of functionsort line 3030

(paused, type something to continue) >
```

```
{move 11}

>>> open


   {move
    12}

   >>> declare \
       B1 obj


   B1 : obj


   {move
    12}
```

```
>>> open

   {move
    13}

   >>> \
       declare \
       bhyp1 \
       that \
       B1 \
       E D4


   bhyp1
    : that
    B1
    E D4


   {move
    13}

   >>> \
       goal \
       that \
       G E B1


   that
    G E B1


   {move
    13}

   >>> \
       define \
```

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


line124
 : [(bhyp1_1
    : that
    B1
    E D4) =>
    ({def} ghyp
    Mpsubs
    bhyp1_1
    Mp
    B1
    Ui
    casehyp1
    : that
    G E B1)]


line124
 : [(bhyp1_1
    : that
    B1
    E D4) =>
    (---
    : that
    G E B1)]


{move
 12}
```

```
   >>> \
       close


{move
 12}

>>> define \
    line125 \
    B1 : Ded \
    line124


line125
 : [(B1_1
    : obj) =>
    ({def} Ded
    ([(bhyp1_2
       : that
       B1_1
       E D4) =>
       ({def} ghyp
       Mpsubs
       bhyp1_2
       Mp
       B1_1
       Ui
       casehyp1
       : that
       G E B1_1)]) : that
    (B1_1
    E D4) ->
    G E B1_1)]


line125
 : [(B1_1
    : obj) =>
```

```
        (---
        : that
        (B1_1
        E D4) ->
        G E B1_1)]


    {move
     11}

    >>> close


{move 11}

>>> define \
    line126 \
    ghyp : Ug \
    line125


line126
 : [(ghyp_1
    : that
    G E D5) =>
    ({def} Ug
    ([(B1_2
       : obj) =>
       ({def} Ded
       ([(bhyp1_3
          : that
          B1_2
          E D4) =>
          ({def} ghyp_1
          Mpsubs
          bhyp1_3
          Mp
          B1_2
```

```
            Ui
            casehyp1
            : that
            G E B1_2)]) : that
         (B1_2
         E D4) ->
         G E B1_2)]) : that
      Forall
      ([(x'_2
         : obj) =>
         ({def} (x'_2
         E D4) ->
         G E x'_2
         : prop)]))]


line126
 : [(ghyp_1
    : that
    G E D5) =>
    (---
    : that
    Forall
    ([(x'_2
       : obj) =>
       ({def} (x'_2
       E D4) ->
       G E x'_2
       : prop)]))]


{move 10}

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

```
    ghyp


line127
 : [(ghyp_1
    : that
    G E D5) =>
    ({def} fhyp4
    Mp F4
    Ui line126
    (ghyp_1) : that
    G E F4)]


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) =>
    ({def} line127
```

```
    (ghyp_1) Conj
    line126
    (ghyp_1) : that
    (G E F4) & Forall
    ([(x'_3
       : obj) =>
       ({def} (x'_3
       E D4) ->
       G E x'_3
       : prop)]))]


line128
 : [(ghyp_1
    : that
    G E D5) =>
    (---
    : that
    (G E F4) & Forall
    ([(x'_3
       : obj) =>
       ({def} (x'_3
       E D4) ->
       G E x'_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) =>
    ({def} (G E D4
    Intersection
    F4) Fixform
    line128
    (ghyp_1) Iff2
    G Ui
    Separation4
    (Refleq
    (D4
    Intersection
    F4)) : that
    G E D4
    Intersection
    F4)]


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


{move 10}
```

```
   >>> close


{move 10}

>>> define \
    line130 G : Ded \
    line129


line130 : [(G_1
    : obj) =>
    ({def} Ded
    ([(ghyp_2
       : that
       G_1 E D5) =>
       ({def} (G_1
       E D4
       Intersection
       F4) Fixform
       fhyp4
       Mp F4
       Ui Ug
       ([(B1_8
          : obj) =>
          ({def} Ded
          ([(bhyp1_9
             : that
             B1_8
             E D4) =>
             ({def} ghyp_2
             Mpsubs
             bhyp1_9
             Mp
             B1_8
             Ui
             casehyp1
             : that
```

```
            G_1
            E B1_8)]) : that
         (B1_8
         E D4) ->
         G_1
         E B1_8)]) Conj
      Ug ([(B1_6
         : obj) =>
         ({def} Ded
         ([(bhyp1_7
            : that
            B1_6
            E D4) =>
            ({def} ghyp_2
            Mpsubs
            bhyp1_7
            Mp
            B1_6
            Ui
            casehyp1
            : that
            G_1
            E B1_6)]) : that
         (B1_6
         E D4) ->
         G_1
         E B1_6)]) Iff2
      G_1 Ui
      Separation4
      (Refleq
      (D4
      Intersection
      F4)) : that
      G_1 E D4
      Intersection
      F4)]) : that
   (G_1 E D5) ->
   G_1 E D4
```

```
        Intersection
        F4)]


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


    {move 9}

    >>> close


{move 9}

>>> 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_3
       : obj) =>
       ({def} (D7_3
       E D4) ->
       D5 <<= D7_3
```

```
   : prop)])) =>
({def} (D5
<<= D4 Intersection
F4) Fixform
Ug ([(G_4
   : obj) =>
   ({def} Ded
   ([(ghyp_5
      : that
      G_4 E D5) =>
      ({def} (G_4
      E D4
      Intersection
      F4) Fixform
      fhyp4
      Mp F4
      Ui Ug
      ([(B1_11
         : obj) =>
         ({def} Ded
         ([(bhyp1_12
            : that
            B1_11
            E D4) =>
            ({def} ghyp_5
            Mpsubs
            bhyp1_12
            Mp
            B1_11
            Ui
            casehyp1_1
            : that
            G_4
            E B1_11)]) : that
         (B1_11
         E D4) ->
         G_4
         E B1_11)]) Conj
```

```
            Ug ([(B1_9
               : obj) =>
               ({def} Ded
               ([(bhyp1_10
                  : that
                  B1_9
                  E D4) =>
                  ({def} ghyp_5
                  Mpsubs
                  bhyp1_10
                  Mp
                  B1_9
                  Ui
                  casehyp1_1
                  : that
                  G_4
                  E B1_9)]) : that
               (B1_9
               E D4) ->
               G_4
               E B1_9)]) Iff2
            G_4 Ui
            Separation4
            (Refleq
            (D4
            Intersection
            F4)) : that
            G_4 E D4
            Intersection
            F4)]) : that
         (G_4 E D5) ->
         G_4 E D4
         Intersection
         F4)]) Conj
      Mboldtheta
      Setsinchains
      dhyp5 Conj
      Separation3
```

```
    (Refleq (D4
    Intersection
    F4)) : that
    D5 <<= D4 Intersection
    F4)]


line131 : [(casehyp1_1
    : that Forall
    ([(D7_3
       : obj) =>
       ({def} (D7_3
       E D4) ->
       D5 <<= D7_3
       : prop)])) =>
    (--- : that
    D5 <<= D4 Intersection
    F4)]


{move 8}

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


line132 : [(casehyp1_1
    : that Forall
    ([(D7_3
       : obj) =>
       ({def} (D7_3
       E D4) ->
       D5 <<= D7_3
       : prop)])) =>
    ({def} ((D4
```

```
    Intersection
    F4) <<= D5) Add1
    line131 (casehyp1_1) : that
    (D5 <<= D4
    Intersection
    F4) V (D4
    Intersection
    F4) <<= D5)]


line132 : [(casehyp1_1
    : that Forall
    ([(D7_3
       : obj) =>
       ({def} (D7_3
       E D4) ->
       D5 <<= D7_3
       : 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_3
    : obj) =>
    ({def} (D7_3
```

```
    E D4) -> D5
    <<= D7_3 : prop)]))


{move 9}

>>> open


   {move 10}

   >>> declare \
       G obj


   G : obj


   {move 10}

   >>> open


      {move 11}

      >>> declare \
          ghyp that \
          G E D4 Intersection \
          F4


      ghyp : that
       G E D4 Intersection
       F4


      {move 11}
```

```
>>> goal \
    that G E D5


that G E D5


{move 11}

>>> define \
    line133 \
    : Counterexample \
    casehyp2


line133
 : [
    ({def} Counterexample
    (casehyp2) : that
    Exists
    ([(z_2
       : obj) =>
       ({def} ~ ((z_2
       E D4) ->
       D5
       <<=
       z_2) : prop)]))]


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

```
{move 10}

>>> open


   {move
    12}

   >>> 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) =>
    ({def} Notimp1
    (hhyp_1) : that
    ~ (D5
    <<=
    .H_1))]


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) =>
    ({def} Notimp2
    (hhyp_1) : that
    .H_1
    E D4)]


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) =>
    ({def} line135
    (hhyp_1) Mp
    .H_1
    Ui
    Simp2
    (ghyp
    Iff1
    G Ui
    Separation4
    (Refleq
    (D4
    Intersection
    F4))) : that
    G E .H_1)]


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) =>
    ({def} line135
    (hhyp_1) Mpsubs
    dhyp4
    : that
    .H_1
    E Cuts)]


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) =>
    ({def} dhyp5
    Mp
```

```
    D5
    Ui
    Simp2
    (Simp2
    (line137
    (hhyp_1) Iff1
    .H_1
    Ui
    Separation4
    (Refleq
    (Cuts)))) : that
    (D5
    <<=
    .H_1) V .H_1
    <<=
    D5)]


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) =>
    ({def} line138
    (hhyp_1) Ds2
    line134
    (hhyp_1) : that
    .H_1
    <<=
    D5)]


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) =>
    ({def} line136
    (hhyp_1) Mpsubs
    line139
    (hhyp_1) : that
    G E D5)]


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

```
   {move
    11}

   >>> close


{move 11}

>>> define \
    line141 \
    ghyp : Eg \
    line133 \
    line140


line141
 : [(ghyp_1
    : that
    G E D4
    Intersection
    F4) =>
    ({def} line133
    Eg [(.H_2
       : obj), (hhyp_2
       : that
       line133
       Witnesses
       .H_2) =>
       ({def} Notimp2
       (hhyp_2) Mp
       .H_2
       Ui
       Simp2
       (ghyp_1
       Iff1
       G Ui
       Separation4
       (Refleq
```

```
            (D4
            Intersection
            F4))) Mpsubs
            dhyp5
            Mp
            D5
            Ui
            Simp2
            (Simp2
            (Notimp2
            (hhyp_2) Mpsubs
            dhyp4
            Iff1
            .H_2
            Ui
            Separation4
            (Refleq
            (Cuts)))) Ds2
            Notimp1
            (hhyp_2) : that
            G E D5)] : that
         G E D5)]


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


     {move 10}

     >>> close
```

```
{move 10}

>>> define \
    line142 G : Ded \
    line141


line142 : [(G_1
    : obj) =>
    ({def} Ded
    ([(ghyp_2
       : that
       G_1 E D4
       Intersection
       F4) =>
       ({def} Counterexample
       (casehyp2) Eg
       [(.H_3
          : obj), (hhyp_3
          : that
          Counterexample
          (casehyp2) Witnesses
          .H_3) =>
          ({def} Notimp2
          (hhyp_3) Mp
          .H_3
          Ui
          Simp2
          (ghyp_2
          Iff1
          G_1
          Ui
          Separation4
          (Refleq
          (D4
          Intersection
```

```
            F4))) Mpsubs
            dhyp5
            Mp
            D5
            Ui
            Simp2
            (Simp2
            (Notimp2
            (hhyp_3) Mpsubs
            dhyp4
            Iff1
            .H_3
            Ui
            Separation4
            (Refleq
            (Cuts)))) Ds2
            Notimp1
            (hhyp_3) : that
            G_1
            E D5)] : that
         G_1 E D5)]) : that
      (G_1 E D4
      Intersection
      F4) ->
      G_1 E D5)]


line142 : [(G_1
      : obj) =>
      (--- : that
      (G_1 E D4
      Intersection
      F4) ->
      G_1 E D5)]


{move 9}
```

```
   >>> close


{move 9}

>>> 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_4
       : obj) =>
       ({def} (D7_4
       E D4) ->
       D5 <<= D7_4
       : prop)]))) =>
    ({def} ((D4
    Intersection
    F4) <<= D5) Fixform
    Ug ([(G_4
       : obj) =>
       ({def} Ded
       ([(ghyp_5
          : that
          G_4 E D4
          Intersection
          F4) =>
          ({def} Counterexample
          (casehyp2_1) Eg
          [(.H_6
```

```
: obj), (hhyp_6
: that
Counterexample
(casehyp2_1) Witnesses
.H_6) =>
({def} Notimp2
(hhyp_6) Mp
.H_6
Ui
Simp2
(ghyp_5
Iff1
G_4
Ui
Separation4
(Refleq
(D4
Intersection
F4))) Mpsubs
dhyp5
Mp
D5
Ui
Simp2
(Simp2
(Notimp2
(hhyp_6) Mpsubs
dhyp4
Iff1
.H_6
Ui
Separation4
(Refleq
(Cuts)))) Ds2
Notimp1
(hhyp_6) : that
G_4
E D5)] : that
```

```
          G_4 E D5)]) : that
       (G_4 E D4
       Intersection
       F4) ->
       G_4 E D5)]) Conj
    Separation3
    (Refleq (D4
    Intersection
    F4)) Conj
    Mboldtheta
    Setsinchains
    dhyp5 : that
    (D4 Intersection
    F4) <<= D5)]


line143 : [(casehyp2_1
    : that ~ (Forall
    ([(D7_4
       : obj) =>
       ({def} (D7_4
       E D4) ->
       D5 <<= D7_4
       : prop)]))) =>
    (--- : that
    (D4 Intersection
    F4) <<= D5)]


{move 8}

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


line144 : [(casehyp2_1
```

```
    : that ~ (Forall
    ([(D7_4
       : obj) =>
       ({def} (D7_4
       E D4) ->
       D5 <<= D7_4
       : prop)]))) =>
    ({def} (D5
    <<= D4 Intersection
    F4) Add2 line143
    (casehyp2_1) : that
    (D5 <<= D4
    Intersection
    F4) V (D4
    Intersection
    F4) <<= D5)]


line144 : [(casehyp2_1
    : that ~ (Forall
    ([(D7_4
       : obj) =>
       ({def} (D7_4
       E D4) ->
       D5 <<= D7_4
       : prop)]))) =>
    (--- : that
    (D5 <<= D4
    Intersection
    F4) V (D4
    Intersection
    F4) <<= D5)]


{move 8}

>>> close
```

```
{move 8}

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


line145 : [(dhyp5_1
    : that D5 E Mbold) =>
    ({def} Cases
    (line123, [(casehyp1_2
       : that Forall
       ([(D7_4
          : obj) =>
          ({def} (D7_4
          E D4) ->
          D5 <<= D7_4
          : prop)])) =>
       ({def} ((D4
       Intersection
       F4) <<= D5) Add1
       (D5 <<= D4
       Intersection
       F4) Fixform
       Ug ([(G_6
          : obj) =>
          ({def} Ded
          ([(ghyp_7
             : that
             G_6 E D5) =>
             ({def} (G_6
             E D4
             Intersection
             F4) Fixform
             fhyp4
             Mp F4
             Ui Ug
             ([(B1_13
```

```
        : obj) =>
        ({def} Ded
        ([(bhyp1_14
           : that
           B1_13
           E D4) =>
           ({def} ghyp_7
           Mpsubs
           bhyp1_14
           Mp
           B1_13
           Ui
           casehyp1_2
           : that
           G_6
           E B1_13)]) : that
        (B1_13
        E D4) ->
        G_6
        E B1_13)]) Conj
     Ug ([(B1_11
        : obj) =>
        ({def} Ded
        ([(bhyp1_12
           : that
           B1_11
           E D4) =>
           ({def} ghyp_7
           Mpsubs
           bhyp1_12
           Mp
           B1_11
           Ui
           casehyp1_2
           : that
           G_6
           E B1_11)]) : that
        (B1_11
```

```
          E D4) ->
          G_6
          E B1_11)]) Iff2
       G_6 Ui
       Separation4
       (Refleq
       (D4
       Intersection
       F4)) : that
       G_6 E D4
       Intersection
       F4)]) : that
    (G_6 E D5) ->
    G_6 E D4
    Intersection
    F4)]) Conj
Mboldtheta
Setsinchains
dhyp5_1 Conj
Separation3
(Refleq (D4
Intersection
F4)) : that
(D5 <<= D4
Intersection
F4) V (D4
Intersection
F4) <<= D5)], [(casehyp2_2
: that ~ (Forall
([(D7_5
    : obj) =>
    ({def} (D7_5
    E D4) ->
    D5 <<= D7_5
    : prop)]))) =>
({def} (D5
<<= D4 Intersection
F4) Add2 ((D4
```

```
Intersection
F4) <<= D5) Fixform
Ug ([(G_6
   : obj) =>
   ({def} Ded
   ([(ghyp_7
      : that
      G_6 E D4
      Intersection
      F4) =>
      ({def} Counterexample
      (casehyp2_2) Eg
      [(.H_8
         : obj), (hhyp_8
         : that
         Counterexample
         (casehyp2_2) Witnesses
         .H_8) =>
         ({def} Notimp2
         (hhyp_8) Mp
         .H_8
         Ui
         Simp2
         (ghyp_7
         Iff1
         G_6
         Ui
         Separation4
         (Refleq
         (D4
         Intersection
         F4))) Mpsubs
         dhyp5_1
         Mp
         D5
         Ui
         Simp2
         (Simp2
```

```
                              (Notimp2
                              (hhyp_8) Mpsubs
                              dhyp4
                              Iff1
                              .H_8
                              Ui
                              Separation4
                              (Refleq
                              (Cuts)))) Ds2
                              Notimp1
                              (hhyp_8) : that
                              G_6
                              E D5)] : that
                           G_6 E D5)]) : that
                        (G_6 E D4
                        Intersection
                        F4) ->
                        G_6 E D5)]) Conj
                     Separation3
                     (Refleq (D4
                     Intersection
                     F4)) Conj
                     Mboldtheta
                     Setsinchains
                     dhyp5_1 : that
                     (D5 <<= D4
                     Intersection
                     F4) V (D4
                     Intersection
                     F4) <<= D5)]) : that
                  (D5 <<= D4 Intersection
                  F4) V (D4 Intersection
                  F4) <<= D5)]


              line145 : [(dhyp5_1
                  : that D5 E Mbold) =>
                  (--- : that (D5
```

```
        <<= D4 Intersection
        F4) V (D4 Intersection
        F4) <<= D5)]


    {move 7}

    >>> close


{move 7}

>>> define line146 D5 \
    : Ded line145


line146 : [(D5_1 : obj) =>
    ({def} Ded ([(dhyp5_2
       : that D5_1 E Mbold) =>
       ({def} Cases
       (Excmid (Forall
       ([(D6_5 : obj) =>
          ({def} (D6_5
          E D4) -> D5_1
          <<= D6_5 : prop)])), [(casehyp1_3
          : that Forall
          ([(D7_5
             : obj) =>
             ({def} (D7_5
             E D4) ->
             D5_1 <<=
             D7_5 : prop)])) =>
          ({def} ((D4
          Intersection
          F4) <<= D5_1) Add1
          (D5_1 <<=
          D4 Intersection
          F4) Fixform
```

```
Ug ([(G_7
   : obj) =>
   ({def} Ded
   ([(ghyp_8
      : that
      G_7 E D5_1) =>
      ({def} (G_7
      E D4
      Intersection
      F4) Fixform
      fhyp4
      Mp F4
      Ui Ug
      ([(B1_14
         : obj) =>
         ({def} Ded
         ([(bhyp1_15
            : that
            B1_14
            E D4) =>
            ({def} ghyp_8
            Mpsubs
            bhyp1_15
            Mp
            B1_14
            Ui
            casehyp1_3
            : that
            G_7
            E B1_14)]) : that
         (B1_14
         E D4) ->
         G_7
         E B1_14)]) Conj
      Ug ([(B1_12
         : obj) =>
         ({def} Ded
         ([(bhyp1_13
```

```
                  : that
                  B1_12
                  E D4) =>
                  ({def} ghyp_8
                  Mpsubs
                  bhyp1_13
                  Mp
                  B1_12
                  Ui
                  casehyp1_3
                  : that
                  G_7
                  E B1_12)]) : that
               (B1_12
               E D4) ->
               G_7
               E B1_12)]) Iff2
            G_7 Ui
            Separation4
            (Refleq
            (D4
            Intersection
            F4)) : that
            G_7 E D4
            Intersection
            F4)]) : that
         (G_7 E D5_1) ->
         G_7 E D4
         Intersection
         F4)]) Conj
      Mboldtheta
      Setsinchains
      dhyp5_2 Conj
      Separation3
      (Refleq (D4
      Intersection
      F4)) : that
      (D5_1 <<=
```

```
D4 Intersection
F4) V (D4
Intersection
F4) <<= D5_1)], [(casehyp2_3
: that ~ (Forall
([(D7_6
   : obj) =>
   ({def} (D7_6
   E D4) ->
   D5_1 <<=
   D7_6 : prop)]))) =>
({def} (D5_1
<<= D4 Intersection
F4) Add2 ((D4
Intersection
F4) <<= D5_1) Fixform
Ug ([(G_7
   : obj) =>
   ({def} Ded
   ([(ghyp_8
      : that
      G_7 E D4
      Intersection
      F4) =>
      ({def} Counterexample
      (casehyp2_3) Eg
      [(.H_9
         : obj), (hhyp_9
         : that
         Counterexample
         (casehyp2_3) Witnesses
         .H_9) =>
         ({def} Notimp2
         (hhyp_9) Mp
         .H_9
         Ui
         Simp2
         (ghyp_8
```

```
                  Iff1
                  G_7
                  Ui
                  Separation4
                  (Refleq
                  (D4
                  Intersection
                  F4))) Mpsubs
                  dhyp5_2
                  Mp
                  D5_1
                  Ui
                  Simp2
                  (Simp2
                  (Notimp2
                  (hhyp_9) Mpsubs
                  dhyp4
                  Iff1
                  .H_9
                  Ui
                  Separation4
                  (Refleq
                  (Cuts)))) Ds2
                  Notimp1
                  (hhyp_9) : that
                  G_7
                  E D5_1)] : that
               G_7 E D5_1)]) : that
            (G_7 E D4
            Intersection
            F4) ->
            G_7 E D5_1)]) Conj
         Separation3
         (Refleq (D4
         Intersection
         F4)) Conj
         Mboldtheta
         Setsinchains
```

```
            dhyp5_2 : that
            (D5_1 <<=
            D4 Intersection
            F4) V (D4
            Intersection
            F4) <<= D5_1)]) : that
         (D5_1 <<= D4
         Intersection F4) V (D4
         Intersection F4) <<=
         D5_1)]) : that
      (D5_1 E Mbold) ->
      (D5_1 <<= D4 Intersection
      F4) V (D4 Intersection
      F4) <<= D5_1)]


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


  {move 6}

  >>> close


{move 6}

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


line147 : [(fhyp4_1 : that
    F4 E D4) =>
```

```
({def} line122 (fhyp4_1) Conj
line122 (fhyp4_1) Conj
Ug ([(D5_4 : obj) =>
   ({def} Ded ([(dhyp5_5
      : that D5_4 E Mbold) =>
      ({def} Cases
      (Excmid (Forall
      ([(D6_8 : obj) =>
         ({def} (D6_8
         E D4) -> D5_4
         <<= D6_8 : prop)])), [(casehyp1_6
         : that Forall
         ([(D7_8
            : obj) =>
            ({def} (D7_8
            E D4) ->
            D5_4 <<=
            D7_8 : prop)])) =>
         ({def} ((D4
         Intersection
         F4) <<= D5_4) Add1
         (D5_4 <<=
         D4 Intersection
         F4) Fixform
         Ug ([(G_10
            : obj) =>
            ({def} Ded
            ([(ghyp_11
               : that
               G_10
               E D5_4) =>
               ({def} (G_10
               E D4
               Intersection
               F4) Fixform
               fhyp4_1
               Mp F4
               Ui Ug
```

```
([(B1_17
   : obj) =>
   ({def} Ded
   ([(bhyp1_18
      : that
      B1_17
      E D4) =>
      ({def} ghyp_11
      Mpsubs
      bhyp1_18
      Mp
      B1_17
      Ui
      casehyp1_6
      : that
      G_10
      E B1_17)]) : that
   (B1_17
   E D4) ->
   G_10
   E B1_17)]) Conj
Ug ([(B1_15
   : obj) =>
   ({def} Ded
   ([(bhyp1_16
      : that
      B1_15
      E D4) =>
      ({def} ghyp_11
      Mpsubs
      bhyp1_16
      Mp
      B1_15
      Ui
      casehyp1_6
      : that
      G_10
      E B1_15)]) : that
```

```
         (B1_15
         E D4) ->
         G_10
         E B1_15)]) Iff2
      G_10
      Ui Separation4
      (Refleq
      (D4
      Intersection
      F4)) : that
      G_10
      E D4
      Intersection
      F4)]) : that
   (G_10 E D5_4) ->
   G_10 E D4
   Intersection
   F4)]) Conj
Mboldtheta
Setsinchains
dhyp5_5 Conj
Separation3
(Refleq (D4
Intersection
F4)) : that
(D5_4 <<=
D4 Intersection
F4) V (D4
Intersection
F4) <<= D5_4)], [(casehyp2_6
: that ~ (Forall
([(D7_9
   : obj) =>
   ({def} (D7_9
   E D4) ->
   D5_4 <<=
   D7_9 : prop)]))) =>
({def} (D5_4
```

```
<<= D4 Intersection
F4) Add2 ((D4
Intersection
F4) <<= D5_4) Fixform
Ug ([(G_10
   : obj) =>
   ({def} Ded
   ([(ghyp_11
      : that
      G_10
      E D4
      Intersection
      F4) =>
      ({def} Counterexample
      (casehyp2_6) Eg
      [(.H_12
         : obj), (hhyp_12
         : that
         Counterexample
         (casehyp2_6) Witnesses
         .H_12) =>
         ({def} Notimp2
         (hhyp_12) Mp
         .H_12
         Ui
         Simp2
         (ghyp_11
         Iff1
         G_10
         Ui
         Separation4
         (Refleq
         (D4
         Intersection
         F4))) Mpsubs
         dhyp5_5
         Mp
         D5_4
```

```
                Ui
                Simp2
                (Simp2
                (Notimp2
                (hhyp_12) Mpsubs
                dhyp4
                Iff1
                .H_12
                Ui
                Separation4
                (Refleq
                (Cuts)))) Ds2
                Notimp1
                (hhyp_12) : that
                G_10
                E D5_4)] : that
             G_10
             E D5_4)]) : that
          (G_10 E D4
          Intersection
          F4) ->
          G_10 E D5_4)]) Conj
       Separation3
       (Refleq (D4
       Intersection
       F4)) Conj
       Mboldtheta
       Setsinchains
       dhyp5_5 : that
       (D5_4 <<=
       D4 Intersection
       F4) V (D4
       Intersection
       F4) <<= D5_4)]) : that
    (D5_4 <<= D4
    Intersection F4) V (D4
    Intersection F4) <<=
    D5_4)]) : that
```

```
        (D5_4 E Mbold) ->
        (D5_4 <<= D4 Intersection
        F4) V (D4 Intersection
        F4) <<= D5_4)]) : that
     ((D4 Intersection
     F4) E Misset Mbold2
     thelawchooses) & ((D4
     Intersection F4) E Misset
     Mbold2 thelawchooses) & Forall
     ([(x'_4 : obj) =>
        ({def} (x'_4 E Mbold) ->
        (x'_4 <<= D4 Intersection
        F4) V (D4 Intersection
        F4) <<= x'_4 : prop)]))]


line147 : [(fhyp4_1 : that
     F4 E D4) => (--- : that
     ((D4 Intersection
     F4) E Misset Mbold2
     thelawchooses) & ((D4
     Intersection F4) E Misset
     Mbold2 thelawchooses) & Forall
     ([(x'_4 : obj) =>
        ({def} (x'_4 E Mbold) ->
        (x'_4 <<= D4 Intersection
        F4) V (D4 Intersection
        F4) <<= x'_4 : prop)]))]


{move 5}

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

```
   linea147 : [(fhyp4_1
       : that F4 E D4) =>
       ({def} line147 (fhyp4_1) Iff2
       (D4 Intersection F4) Ui
       Separation4 (Refleq
       (Cuts)) : that (D4
       Intersection F4) E Misset
       Mbold2 thelawchooses
       Set [(C_3 : obj) =>
          ({def} cuts2 (Misset, thelawchooses, C_3) : prop)])]


   linea147 : [(fhyp4_1
       : that F4 E D4) =>
       (--- : that (D4 Intersection
       F4) E Misset Mbold2
       thelawchooses Set [(C_3
          : obj) =>
          ({def} cuts2 (Misset, thelawchooses, C_3) : prop)])]


   {move 5}

   >>> close


{move 5}

>>> define line148 F4 : Ded \
    linea147


line148 : [(F4_1 : obj) =>
    ({def} Ded ([(fhyp4_2
       : that F4_1 E D4) =>
       ({def} dhyp4 Transsub
       (Cuts <<= Mbold) Fixform
       Separation3 (Refleq
```

```
(Mbold)) Sepsub2
Refleq (Cuts) Conj
fhyp4_2 Mp F4_1 Ui D4
Ui Simp2 (Simp2 (Simp2
(Mboldtheta))) Conj
dhyp4 Transsub (Cuts
<<= Mbold) Fixform
Separation3 (Refleq
(Mbold)) Sepsub2
Refleq (Cuts) Conj
fhyp4_2 Mp F4_1 Ui D4
Ui Simp2 (Simp2 (Simp2
(Mboldtheta))) Conj
Ug ([(D5_6 : obj) =>
   ({def} Ded ([(dhyp5_7
      : that D5_6 E Mbold) =>
      ({def} Cases
      (Excmid (Forall
      ([(D6_10 : obj) =>
         ({def} (D6_10
         E D4) -> D5_6
         <<= D6_10 : prop)])), [(casehyp1_8
         : that Forall
         ([(D7_10
            : obj) =>
            ({def} (D7_10
            E D4) ->
            D5_6 <<=
            D7_10 : prop)])) =>
         ({def} ((D4
         Intersection
         F4_1) <<=
         D5_6) Add1
         (D5_6 <<=
         D4 Intersection
         F4_1) Fixform
         Ug ([(G_12
            : obj) =>
```

```
({def} Ded
([(ghyp_13
   : that
   G_12
   E D5_6) =>
   ({def} (G_12
   E D4
   Intersection
   F4_1) Fixform
   fhyp4_2
   Mp F4_1
   Ui Ug
   ([(B1_19
      : obj) =>
      ({def} Ded
      ([(bhyp1_20
         : that
         B1_19
         E D4) =>
         ({def} ghyp_13
         Mpsubs
         bhyp1_20
         Mp
         B1_19
         Ui
         casehyp1_8
         : that
         G_12
         E B1_19)]) : that
      (B1_19
      E D4) ->
      G_12
      E B1_19)]) Conj
   Ug ([(B1_17
      : obj) =>
      ({def} Ded
      ([(bhyp1_18
         : that
```

```
            B1_17
            E D4) =>
            ({def} ghyp_13
            Mpsubs
            bhyp1_18
            Mp
            B1_17
            Ui
            casehyp1_8
            : that
            G_12
            E B1_17)]) : that
         (B1_17
         E D4) ->
         G_12
         E B1_17)]) Iff2
      G_12
      Ui Separation4
      (Refleq
      (D4
      Intersection
      F4_1)) : that
      G_12
      E D4
      Intersection
      F4_1)]) : that
   (G_12 E D5_6) ->
   G_12 E D4
   Intersection
   F4_1)]) Conj
Mboldtheta
Setsinchains
dhyp5_7 Conj
Separation3
(Refleq (D4
Intersection
F4_1)) : that
(D5_6 <<=
```

```
D4 Intersection
F4_1) V (D4
Intersection
F4_1) <<=
D5_6)], [(casehyp2_8
: that ~ (Forall
([(D7_11
   : obj) =>
   ({def} (D7_11
   E D4) ->
   D5_6 <<=
   D7_11 : prop)]))) =>
({def} (D5_6
<<= D4 Intersection
F4_1) Add2
((D4 Intersection
F4_1) <<=
D5_6) Fixform
Ug ([(G_12
   : obj) =>
   ({def} Ded
   ([(ghyp_13
      : that
      G_12
      E D4
      Intersection
      F4_1) =>
      ({def} Counterexample
      (casehyp2_8) Eg
      [(.H_14
         : obj), (hhyp_14
         : that
         Counterexample
         (casehyp2_8) Witnesses
         .H_14) =>
         ({def} Notimp2
         (hhyp_14) Mp
         .H_14
```

```
                    Ui
                    Simp2
                    (ghyp_13
                    Iff1
                    G_12
                    Ui
                    Separation4
                    (Refleq
                    (D4
                    Intersection
                    F4_1))) Mpsubs
                    dhyp5_7
                    Mp
                    D5_6
                    Ui
                    Simp2
                    (Simp2
                    (Notimp2
                    (hhyp_14) Mpsubs
                    dhyp4
                    Iff1
                    .H_14
                    Ui
                    Separation4
                    (Refleq
                    (Cuts)))) Ds2
                    Notimp1
                    (hhyp_14) : that
                    G_12
                    E D5_6)] : that
                 G_12
                 E D5_6)]) : that
              (G_12 E D4
              Intersection
              F4_1) ->
              G_12 E D5_6)]) Conj
           Separation3
           (Refleq (D4
```

```
                Intersection
                F4_1)) Conj
                Mboldtheta
                Setsinchains
                dhyp5_7 : that
                (D5_6 <<=
                D4 Intersection
                F4_1) V (D4
                Intersection
                F4_1) <<=
                D5_6)]) : that
             (D5_6 <<= D4
             Intersection F4_1) V (D4
             Intersection F4_1) <<=
             D5_6)]) : that
          (D5_6 E Mbold) ->
          (D5_6 <<= D4 Intersection
          F4_1) V (D4 Intersection
          F4_1) <<= D5_6)]) Iff2
       (D4 Intersection F4_1) Ui
       Separation4 (Refleq
       (Cuts)) : that (D4
       Intersection F4_1) E Misset
       Mbold2 thelawchooses
       Set [(C_4 : obj) =>
          ({def} cuts2 (Misset, thelawchooses, C_4) : prop)])])
    (F4_1 E D4) -> (D4 Intersection
    F4_1) E Misset Mbold2
    thelawchooses Set [(C_4
       : obj) =>
       ({def} cuts2 (Misset, thelawchooses, C_4) : prop)])]


line148 : [(F4_1 : obj) =>
    (--- : that (F4_1 E D4) ->
    (D4 Intersection F4_1) E Misset
    Mbold2 thelawchooses Set
    [(C_4 : obj) =>
```

```
          ({def} cuts2 (Misset, thelawchooses, C_4) : prop)])]


   {move 4}

   >>> close


{move 4}

>>> define line149 dhyp4 : Ug \
    line148


line149 : [(dhyp4_1 : that
    D4 <<= Cuts) =>
    ({def} Ug ([(F4_2 : obj) =>
       ({def} Ded ([(fhyp4_3
          : that F4_2 E D4) =>
          ({def} dhyp4_1 Transsub
          (Cuts <<= Mbold) Fixform
          Separation3 (Refleq
          (Mbold)) Sepsub2
          Refleq (Cuts) Conj
          fhyp4_3 Mp F4_2 Ui D4
          Ui Simp2 (Simp2 (Simp2
          (Mboldtheta))) Conj
          dhyp4_1 Transsub (Cuts
          <<= Mbold) Fixform
          Separation3 (Refleq
          (Mbold)) Sepsub2
          Refleq (Cuts) Conj
          fhyp4_3 Mp F4_2 Ui D4
          Ui Simp2 (Simp2 (Simp2
          (Mboldtheta))) Conj
          Ug ([(D5_7 : obj) =>
             ({def} Ded ([(dhyp5_8
                : that D5_7 E Mbold) =>
```

```
({def} Cases
(Excmid (Forall
([(D6_11 : obj) =>
   ({def} (D6_11
   E D4) -> D5_7
   <<= D6_11 : prop)])), [(casehyp1_9
   : that Forall
   ([(D7_11
      : obj) =>
      ({def} (D7_11
      E D4) ->
      D5_7 <<=
      D7_11 : prop)])) =>
   ({def} ((D4
   Intersection
   F4_2) <<=
   D5_7) Add1
   (D5_7 <<=
   D4 Intersection
   F4_2) Fixform
   Ug ([(G_13
      : obj) =>
      ({def} Ded
      ([(ghyp_14
         : that
         G_13
         E D5_7) =>
         ({def} (G_13
         E D4
         Intersection
         F4_2) Fixform
         fhyp4_3
         Mp F4_2
         Ui Ug
         ([(B1_20
            : obj) =>
            ({def} Ded
            ([(bhyp1_21
```

```
                  : that
                  B1_20
                  E D4) =>
                  ({def} ghyp_14
                  Mpsubs
                  bhyp1_21
                  Mp
                  B1_20
                  Ui
                  casehyp1_9
                  : that
                  G_13
                  E B1_20)]) : that
               (B1_20
               E D4) ->
               G_13
               E B1_20)]) Conj
            Ug ([(B1_18
               : obj) =>
               ({def} Ded
               ([(bhyp1_19
                  : that
                  B1_18
                  E D4) =>
                  ({def} ghyp_14
                  Mpsubs
                  bhyp1_19
                  Mp
                  B1_18
                  Ui
                  casehyp1_9
                  : that
                  G_13
                  E B1_18)]) : that
               (B1_18
               E D4) ->
               G_13
               E B1_18)]) Iff2
```

```
      G_13
      Ui Separation4
      (Refleq
      (D4
      Intersection
      F4_2)) : that
      G_13
      E D4
      Intersection
      F4_2)]) : that
   (G_13 E D5_7) ->
   G_13 E D4
   Intersection
   F4_2)]) Conj
Mboldtheta
Setsinchains
dhyp5_8 Conj
Separation3
(Refleq (D4
Intersection
F4_2)) : that
(D5_7 <<=
D4 Intersection
F4_2) V (D4
Intersection
F4_2) <<=
D5_7)], [(casehyp2_9
: that ~ (Forall
([(D7_12
   : obj) =>
   ({def} (D7_12
   E D4) ->
   D5_7 <<=
   D7_12 : prop)]))) =>
({def} (D5_7
<<= D4 Intersection
F4_2) Add2
((D4 Intersection
```

```
F4_2) <<=
D5_7) Fixform
Ug ([(G_13
   : obj) =>
   ({def} Ded
   ([(ghyp_14
      : that
      G_13
      E D4
      Intersection
      F4_2) =>
      ({def} Counterexample
      (casehyp2_9) Eg
      [(.H_15
         : obj), (hhyp_15
         : that
         Counterexample
         (casehyp2_9) Witnesses
         .H_15) =>
         ({def} Notimp2
         (hhyp_15) Mp
         .H_15
         Ui
         Simp2
         (ghyp_14
         Iff1
         G_13
         Ui
         Separation4
         (Refleq
         (D4
         Intersection
         F4_2))) Mpsubs
         dhyp5_8
         Mp
         D5_7
         Ui
         Simp2
```

```
                    (Simp2
                    (Notimp2
                    (hhyp_15) Mpsubs
                    dhyp4_1
                    Iff1
                    .H_15
                    Ui
                    Separation4
                    (Refleq
                    (Cuts)))) Ds2
                    Notimp1
                    (hhyp_15) : that
                    G_13
                    E D5_7)] : that
                 G_13
                 E D5_7)]) : that
              (G_13 E D4
              Intersection
              F4_2) ->
              G_13 E D5_7)]) Conj
           Separation3
           (Refleq (D4
           Intersection
           F4_2)) Conj
           Mboldtheta
           Setsinchains
           dhyp5_8 : that
           (D5_7 <<=
           D4 Intersection
           F4_2) V (D4
           Intersection
           F4_2) <<=
           D5_7)]) : that
        (D5_7 <<= D4
        Intersection F4_2) V (D4
        Intersection F4_2) <<=
        D5_7)]) : that
     (D5_7 E Mbold) ->
```

```
               (D5_7 <<= D4 Intersection
               F4_2) V (D4 Intersection
               F4_2) <<= D5_7)]) Iff2
            (D4 Intersection F4_2) Ui
            Separation4 (Refleq
            (Cuts)) : that (D4
            Intersection F4_2) E Misset
            Mbold2 thelawchooses
            Set [(C_5 : obj) =>
               ({def} cuts2 (Misset, thelawchooses, C_5) : prop)])])
         (F4_2 E D4) -> (D4 Intersection
         F4_2) E Misset Mbold2
         thelawchooses Set [(C_5
            : obj) =>
            ({def} cuts2 (Misset, thelawchooses, C_5) : prop)])]) : t
      Forall ([(x'_2 : obj) =>
         ({def} (x'_2 E D4) ->
         (D4 Intersection x'_2) E Misset
         Mbold2 thelawchooses Set
         [(C_5 : obj) =>
            ({def} cuts2 (Misset, thelawchooses, C_5) : prop)] : prop


line149 : [(dhyp4_1 : that
      D4 <<= Cuts) => (--- : that
      Forall ([(x'_2 : obj) =>
         ({def} (x'_2 E D4) ->
         (D4 Intersection x'_2) E Misset
         Mbold2 thelawchooses Set
         [(C_5 : obj) =>
            ({def} cuts2 (Misset, thelawchooses, C_5) : prop)] : prop


{move 3}

>>> close
```

```
{move 3}

>>> define line150 D4 : Ded line149


line150 : [(D4_1 : obj) =>
    ({def} Ded ([(dhyp4_2 : that
       D4_1 <<= Cuts) =>
       ({def} Ug ([(F4_3 : obj) =>
          ({def} Ded ([(fhyp4_4
             : that F4_3 E D4_1) =>
             ({def} dhyp4_2 Transsub
             (Cuts <<= Mbold) Fixform
             Separation3 (Refleq
             (Mbold)) Sepsub2
             Refleq (Cuts) Conj
             fhyp4_4 Mp F4_3 Ui D4_1
             Ui Simp2 (Simp2 (Simp2
             (Mboldtheta))) Conj
             dhyp4_2 Transsub (Cuts
             <<= Mbold) Fixform
             Separation3 (Refleq
             (Mbold)) Sepsub2
             Refleq (Cuts) Conj
             fhyp4_4 Mp F4_3 Ui D4_1
             Ui Simp2 (Simp2 (Simp2
             (Mboldtheta))) Conj
             Ug ([(D5_8 : obj) =>
                ({def} Ded ([(dhyp5_9
                   : that D5_8 E Mbold) =>
                   ({def} Cases
                   (Excmid (Forall
                   ([(D6_12 : obj) =>
                      ({def} (D6_12
                      E D4_1) ->
                      D5_8 <<= D6_12
                      : prop)])), [(casehyp1_10
                      : that Forall
```

```
([(D7_12
   : obj) =>
   ({def} (D7_12
   E D4_1) ->
   D5_8 <<=
   D7_12 : prop)])) =>
({def} ((D4_1
Intersection
F4_3) <<=
D5_8) Add1
(D5_8 <<=
D4_1 Intersection
F4_3) Fixform
Ug ([(G_14
   : obj) =>
   ({def} Ded
   ([(ghyp_15
      : that
      G_14
      E D5_8) =>
      ({def} (G_14
      E D4_1
      Intersection
      F4_3) Fixform
      fhyp4_4
      Mp F4_3
      Ui Ug
      ([(B1_21
         : obj) =>
         ({def} Ded
         ([(bhyp1_22
            : that
            B1_21
            E D4_1) =>
            ({def} ghyp_15
            Mpsubs
            bhyp1_22
            Mp
```

```
      B1_21
      Ui
      casehyp1_10
      : that
      G_14
      E B1_21)]) : that
   (B1_21
   E D4_1) ->
   G_14
   E B1_21)]) Conj
Ug ([(B1_19
   : obj) =>
   ({def} Ded
   ([(bhyp1_20
      : that
      B1_19
      E D4_1) =>
      ({def} ghyp_15
      Mpsubs
      bhyp1_20
      Mp
      B1_19
      Ui
      casehyp1_10
      : that
      G_14
      E B1_19)]) : that
   (B1_19
   E D4_1) ->
   G_14
   E B1_19)]) Iff2
G_14
Ui Separation4
(Refleq
(D4_1
Intersection
F4_3)) : that
G_14
```

```
      E D4_1
      Intersection
      F4_3)]) : that
   (G_14 E D5_8) ->
   G_14 E D4_1
   Intersection
   F4_3)]) Conj
Mboldtheta
Setsinchains
dhyp5_9 Conj
Separation3
(Refleq (D4_1
Intersection
F4_3)) : that
(D5_8 <<=
D4_1 Intersection
F4_3) V (D4_1
Intersection
F4_3) <<=
D5_8)], [(casehyp2_10
: that ~ (Forall
([(D7_13
   : obj) =>
   ({def} (D7_13
   E D4_1) ->
   D5_8 <<=
   D7_13 : prop)]))) =>
({def} (D5_8
<<= D4_1 Intersection
F4_3) Add2
((D4_1 Intersection
F4_3) <<=
D5_8) Fixform
Ug ([(G_14
   : obj) =>
   ({def} Ded
   ([(ghyp_15
      : that
```

```
G_14
E D4_1
Intersection
F4_3) =>
({def} Counterexample
(casehyp2_10) Eg
[(.H_16
   : obj), (hhyp_16
   : that
   Counterexample
   (casehyp2_10) Witnesses
   .H_16) =>
   ({def} Notimp2
   (hhyp_16) Mp
   .H_16
   Ui
   Simp2
   (ghyp_15
   Iff1
   G_14
   Ui
   Separation4
   (Refleq
   (D4_1
   Intersection
   F4_3))) Mpsubs
   dhyp5_9
   Mp
   D5_8
   Ui
   Simp2
   (Simp2
   (Notimp2
   (hhyp_16) Mpsubs
   dhyp4_2
   Iff1
   .H_16
   Ui
```

```
                        Separation4
                        (Refleq
                        (Cuts)))) Ds2
                        Notimp1
                        (hhyp_16) : that
                        G_14
                        E D5_8)] : that
                     G_14
                     E D5_8)]) : that
                  (G_14 E D4_1
                  Intersection
                  F4_3) ->
                  G_14 E D5_8)]) Conj
               Separation3
               (Refleq (D4_1
               Intersection
               F4_3)) Conj
               Mboldtheta
               Setsinchains
               dhyp5_9 : that
               (D5_8 <<=
               D4_1 Intersection
               F4_3) V (D4_1
               Intersection
               F4_3) <<=
               D5_8)]) : that
            (D5_8 <<= D4_1
            Intersection F4_3) V (D4_1
            Intersection F4_3) <<=
            D5_8)]) : that
         (D5_8 E Mbold) ->
         (D5_8 <<= D4_1 Intersection
         F4_3) V (D4_1 Intersection
         F4_3) <<= D5_8)]) Iff2
      (D4_1 Intersection
      F4_3) Ui Separation4
      (Refleq (Cuts)) : that
      (D4_1 Intersection
```

```
              F4_3) E Misset Mbold2
              thelawchooses Set [(C_6
                 : obj) =>
                 ({def} cuts2 (Misset, thelawchooses, C_6) : prop)])])
           (F4_3 E D4_1) -> (D4_1
           Intersection F4_3) E Misset
           Mbold2 thelawchooses Set
           [(C_6 : obj) =>
              ({def} cuts2 (Misset, thelawchooses, C_6) : prop)])]) : t
        Forall ([(x'_3 : obj) =>
           ({def} (x'_3 E D4_1) ->
           (D4_1 Intersection x'_3) E Misset
           Mbold2 thelawchooses Set
           [(C_6 : obj) =>
              ({def} cuts2 (Misset, thelawchooses, C_6) : prop)] : prop
     (D4_1 <<= Cuts) -> Forall ([(x'_3
        : obj) =>
        ({def} (x'_3 E D4_1) ->
        (D4_1 Intersection x'_3) E Misset
        Mbold2 thelawchooses Set [(C_6
           : obj) =>
           ({def} cuts2 (Misset, thelawchooses, C_6) : prop)] : prop)])


line150 : [(D4_1 : obj) => (---
     : that (D4_1 <<= Cuts) -> Forall
     ([(x'_3 : obj) =>
        ({def} (x'_3 E D4_1) ->
        (D4_1 Intersection x'_3) E Misset
        Mbold2 thelawchooses Set [(C_6
           : obj) =>
           ({def} cuts2 (Misset, thelawchooses, C_6) : prop)] : prop)])


{move 2}

>>> close
```

```
{move 2}

>>> define line151 : Ug line150


line151 : Ug ([(D4_2 : obj) =>
    ({def} Ded ([(dhyp4_3 : that
       D4_2 <<= Cuts) =>
       ({def} Ug ([(F4_4 : obj) =>
          ({def} Ded ([(fhyp4_5
             : that F4_4 E D4_2) =>
             ({def} dhyp4_3 Transsub
             (Cuts <<= Mbold) Fixform
             Separation3 (Refleq (Mbold)) Sepsub2
             Refleq (Cuts) Conj fhyp4_5
             Mp F4_4 Ui D4_2 Ui Simp2
             (Simp2 (Simp2 (Mboldtheta))) Conj
             dhyp4_3 Transsub (Cuts
             <<= Mbold) Fixform Separation3
             (Refleq (Mbold)) Sepsub2
             Refleq (Cuts) Conj fhyp4_5
             Mp F4_4 Ui D4_2 Ui Simp2
             (Simp2 (Simp2 (Mboldtheta))) Conj
             Ug ([(D5_9 : obj) =>
                ({def} Ded ([(dhyp5_10
                   : that D5_9 E Mbold) =>
                   ({def} Cases (Excmid
                   (Forall ([(D6_13
                      : obj) =>
                      ({def} (D6_13
                      E D4_2) -> D5_9
                      <<= D6_13 : prop)])), [(casehyp1_11
                      : that Forall
                      ([(D7_13 : obj) =>
                         ({def} (D7_13
                         E D4_2) ->
                         D5_9 <<= D7_13
```

```
   : prop)])) =>
({def} ((D4_2
Intersection F4_4) <<=
D5_9) Add1 (D5_9
<<= D4_2 Intersection
F4_4) Fixform
Ug ([(G_15
   : obj) =>
   ({def} Ded
   ([(ghyp_16
      : that G_15
      E D5_9) =>
      ({def} (G_15
      E D4_2 Intersection
      F4_4) Fixform
      fhyp4_5
      Mp F4_4
      Ui Ug ([(B1_22
         : obj) =>
         ({def} Ded
         ([(bhyp1_23
            : that
            B1_22
            E D4_2) =>
            ({def} ghyp_16
            Mpsubs
            bhyp1_23
            Mp
            B1_22
            Ui
            casehyp1_11
            : that
            G_15
            E B1_22)]) : that
         (B1_22
         E D4_2) ->
         G_15
         E B1_22)]) Conj
```

```
      Ug ([(B1_20
         : obj) =>
         ({def} Ded
         ([(bhyp1_21
            : that
            B1_20
            E D4_2) =>
            ({def} ghyp_16
            Mpsubs
            bhyp1_21
            Mp
            B1_20
            Ui
            casehyp1_11
            : that
            G_15
            E B1_20)]) : that
         (B1_20
         E D4_2) ->
         G_15
         E B1_20)]) Iff2
      G_15 Ui
      Separation4
      (Refleq
      (D4_2 Intersection
      F4_4)) : that
      G_15 E D4_2
      Intersection
      F4_4)]) : that
   (G_15 E D5_9) ->
   G_15 E D4_2
   Intersection
   F4_4)]) Conj
Mboldtheta Setsinchains
dhyp5_10 Conj
Separation3 (Refleq
(D4_2 Intersection
F4_4)) : that
```

```
(D5_9 <<= D4_2
Intersection F4_4) V (D4_2
Intersection F4_4) <<=
D5_9)], [(casehyp2_11
: that ~ (Forall
([(D7_14 : obj) =>
   ({def} (D7_14
   E D4_2) ->
   D5_9 <<= D7_14
   : prop)]))) =>
({def} (D5_9
<<= D4_2 Intersection
F4_4) Add2 ((D4_2
Intersection F4_4) <<=
D5_9) Fixform
Ug ([(G_15
   : obj) =>
   ({def} Ded
   ([(ghyp_16
      : that G_15
      E D4_2 Intersection
      F4_4) =>
      ({def} Counterexample
      (casehyp2_11) Eg
      [(.H_17
         : obj), (hhyp_17
         : that
         Counterexample
         (casehyp2_11) Witnesses
         .H_17) =>
         ({def} Notimp2
         (hhyp_17) Mp
         .H_17
         Ui Simp2
         (ghyp_16
         Iff1
         G_15
         Ui Separation4
```

```
                    (Refleq
                    (D4_2
                    Intersection
                    F4_4))) Mpsubs
                    dhyp5_10
                    Mp D5_9
                    Ui Simp2
                    (Simp2
                    (Notimp2
                    (hhyp_17) Mpsubs
                    dhyp4_3
                    Iff1
                    .H_17
                    Ui Separation4
                    (Refleq
                    (Cuts)))) Ds2
                    Notimp1
                    (hhyp_17) : that
                    G_15
                    E D5_9)] : that
                 G_15 E D5_9)]) : that
              (G_15 E D4_2
              Intersection
              F4_4) -> G_15
              E D5_9)]) Conj
           Separation3 (Refleq
           (D4_2 Intersection
           F4_4)) Conj
           Mboldtheta Setsinchains
           dhyp5_10 : that
           (D5_9 <<= D4_2
           Intersection F4_4) V (D4_2
           Intersection F4_4) <<=
           D5_9)]) : that
        (D5_9 <<= D4_2 Intersection
        F4_4) V (D4_2 Intersection
        F4_4) <<= D5_9)]) : that
     (D5_9 E Mbold) ->
```

```
               (D5_9 <<= D4_2 Intersection
               F4_4) V (D4_2 Intersection
               F4_4) <<= D5_9)]) Iff2
            (D4_2 Intersection F4_4) Ui
            Separation4 (Refleq (Cuts)) : that
            (D4_2 Intersection F4_4) E Misset
            Mbold2 thelawchooses Set
            [(C_7 : obj) =>
               ({def} cuts2 (Misset, thelawchooses, C_7) : prop)])]) : t
         (F4_4 E D4_2) -> (D4_2
         Intersection F4_4) E Misset
         Mbold2 thelawchooses Set [(C_7
            : obj) =>
            ({def} cuts2 (Misset, thelawchooses, C_7) : prop)])]) : that
      Forall ([(x'_4 : obj) =>
         ({def} (x'_4 E D4_2) ->
         (D4_2 Intersection x'_4) E Misset
         Mbold2 thelawchooses Set [(C_7
            : obj) =>
            ({def} cuts2 (Misset, thelawchooses, C_7) : prop)] : prop)])
   (D4_2 <<= Cuts) -> Forall ([(x'_4
      : obj) =>
      ({def} (x'_4 E D4_2) -> (D4_2
      Intersection x'_4) E Misset
      Mbold2 thelawchooses Set [(C_7
         : obj) =>
         ({def} cuts2 (Misset, thelawchooses, C_7) : prop)] : prop)]))])


line151 : that Forall ([(x'_2 : obj) =>
   ({def} (x'_2 <<= Cuts) -> Forall
   ([(x'_4 : obj) =>
      ({def} (x'_4 E x'_2) -> (x'_2
      Intersection x'_4) E Misset
      Mbold2 thelawchooses Set [(C_7
         : obj) =>
         ({def} cuts2 (Misset, thelawchooses, C_7) : prop)] : prop)]) :
```

```
{move 1}

>>> open


   {move 3}

   >>> declare D9 obj


   D9 : obj


   {move 3}

   >>> open


      {move 4}

      >>> declare F9 obj


      F9 : obj


      {move 4}

      >>> open


         {move 5}

         >>> 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) =>
    ({def} Simp1 (conjhyp_1) : that
    D9 <<= Cuts)]


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) =>
    ({def} Simp2 (conjhyp_1) : that
    F9 E D9)]


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) =>
       ({def} secondhyp (conjhyp_1) Mp
       F9 Ui firsthyp (conjhyp_1) Mp
       D9 Ui line151 : that (D9
       Intersection F9) E Misset
       Mbold2 thelawchooses Set
       [(C_3 : obj) =>
          ({def} cuts2 (Misset, thelawchooses, C_3) : prop)])]


   line152 : [(conjhyp_1 : that
       (D9 <<= Cuts) & F9 E D9) =>
       (--- : that (D9 Intersection
       F9) E Misset Mbold2 thelawchooses
       Set [(C_3 : obj) =>
          ({def} cuts2 (Misset, thelawchooses, C_3) : prop)])]


   {move 4}

   >>> close


{move 4}

>>> define line153 F9 : Ded line152
```

```
   line153 : [(F9_1 : obj) =>
       ({def} Ded ([(conjhyp_2
          : that (D9 <<= Cuts) & F9_1
          E D9) =>
          ({def} Simp2 (conjhyp_2) Mp
          F9_1 Ui Simp1 (conjhyp_2) Mp
          D9 Ui line151 : that (D9
          Intersection F9_1) E Misset
          Mbold2 thelawchooses Set
          [(C_4 : obj) =>
             ({def} cuts2 (Misset, thelawchooses, C_4) : prop)])]) : t
       ((D9 <<= Cuts) & F9_1 E D9) ->
       (D9 Intersection F9_1) E Misset
       Mbold2 thelawchooses Set [(C_4
          : obj) =>
          ({def} cuts2 (Misset, thelawchooses, C_4) : prop)])]


   line153 : [(F9_1 : obj) =>
       (--- : that ((D9 <<= Cuts) & F9_1
       E D9) -> (D9 Intersection
       F9_1) E Misset Mbold2 thelawchooses
       Set [(C_4 : obj) =>
          ({def} cuts2 (Misset, thelawchooses, C_4) : prop)])]


   {move 3}

   >>> close


{move 3}

>>> define line154 D9 : Ug line153


line154 : [(D9_1 : obj) =>
```

```
    ({def} Ug ([(F9_2 : obj) =>
       ({def} Ded ([(conjhyp_3
          : that (D9_1 <<= Cuts) & F9_2
          E D9_1) =>
          ({def} Simp2 (conjhyp_3) Mp
          F9_2 Ui Simp1 (conjhyp_3) Mp
          D9_1 Ui line151 : that
          (D9_1 Intersection F9_2) E Misset
          Mbold2 thelawchooses Set
          [(C_5 : obj) =>
             ({def} cuts2 (Misset, thelawchooses, C_5) : prop)])]) : t
       ((D9_1 <<= Cuts) & F9_2
       E D9_1) -> (D9_1 Intersection
       F9_2) E Misset Mbold2 thelawchooses
       Set [(C_5 : obj) =>
          ({def} cuts2 (Misset, thelawchooses, C_5) : prop)])]) : that
    Forall ([(x'_2 : obj) =>
       ({def} ((D9_1 <<= Cuts) & x'_2
       E D9_1) -> (D9_1 Intersection
       x'_2) E Misset Mbold2 thelawchooses
       Set [(C_5 : obj) =>
          ({def} cuts2 (Misset, thelawchooses, C_5) : prop)] : prop)])


line154 : [(D9_1 : obj) => (---
    : that Forall ([(x'_2 : obj) =>
       ({def} ((D9_1 <<= Cuts) & x'_2
       E D9_1) -> (D9_1 Intersection
       x'_2) E Misset Mbold2 thelawchooses
       Set [(C_5 : obj) =>
          ({def} cuts2 (Misset, thelawchooses, C_5) : prop)] : prop)])


{move 2}

>>> close
```

```
{move 2}

>>> define linea155 : Ug line154


linea155 : Ug ([(D9_2 : obj) =>
    ({def} Ug ([(F9_3 : obj) =>
       ({def} Ded ([(conjhyp_4 : that
          (D9_2 <<= Cuts) & F9_3 E D9_2) =>
          ({def} Simp2 (conjhyp_4) Mp
          F9_3 Ui Simp1 (conjhyp_4) Mp
          D9_2 Ui line151 : that (D9_2
          Intersection F9_3) E Misset
          Mbold2 thelawchooses Set [(C_6
             : obj) =>
             ({def} cuts2 (Misset, thelawchooses, C_6) : prop)])]) : that
       ((D9_2 <<= Cuts) & F9_3 E D9_2) ->
       (D9_2 Intersection F9_3) E Misset
       Mbold2 thelawchooses Set [(C_6
          : obj) =>
          ({def} cuts2 (Misset, thelawchooses, C_6) : prop)])]) : that
    Forall ([(x'_3 : obj) =>
       ({def} ((D9_2 <<= Cuts) & x'_3
       E D9_2) -> (D9_2 Intersection
       x'_3) E Misset Mbold2 thelawchooses
       Set [(C_6 : obj) =>
          ({def} cuts2 (Misset, thelawchooses, C_6) : prop)] : prop)]))])


linea155 : that Forall ([(x'_2 : obj) =>
    ({def} Forall ([(x'_3 : obj) =>
       ({def} ((x'_2 <<= Cuts) & x'_3
       E x'_2) -> (x'_2 Intersection
       x'_3) E Misset Mbold2 thelawchooses
       Set [(C_6 : obj) =>
          ({def} cuts2 (Misset, thelawchooses, C_6) : prop)] : prop)]) :
```

```
   {move 1}

   >>> save


   {move 2}

   >>> close


{move 1}

>>> define lineb155 Misset, thelawchooses \
    : linea155


lineb155 : [(.M_1 : obj), (Misset_1
    : that Isset (.M_1)), (.thelaw_1
    : [(S_2 : obj) => (--- : obj)]), (thelawchooses_1
    : [(.S_2 : obj), (subsetev_2 : that
       .S_2 <<= .M_1), (inev_2 : that
       Exists ([(x_4 : obj) =>
          ({def} x_4 E .S_2 : prop)])) =>
       (--- : that .thelaw_1 (.S_2) E .S_2)]) =>
    ({def} Ug ([(D9_2 : obj) =>
       ({def} Ug ([(F9_3 : obj) =>
          ({def} Ded ([(conjhyp_4 : that
             (D9_2 <<= Misset_1 Cuts3
             thelawchooses_1) & F9_3 E D9_2) =>
             ({def} Simp2 (conjhyp_4) Mp
             F9_3 Ui Simp1 (conjhyp_4) Mp
             D9_2 Ui Ug ([(D4_9 : obj) =>
                ({def} Ded ([(dhyp4_10
                   : that D4_9 <<= Misset_1
                   Cuts3 thelawchooses_1) =>
                   ({def} Ug ([(F4_11
                      : obj) =>
                      ({def} Ded ([(fhyp4_12
```

```
: that F4_11 E D4_9) =>
({def} dhyp4_10
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_12 Mp F4_11
Ui D4_9 Ui Simp2
(Simp2 (Simp2
(Misset_1 Mboldtheta2
thelawchooses_1))) Conj
dhyp4_10 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_12 Mp F4_11
Ui D4_9 Ui Simp2
(Simp2 (Simp2
(Misset_1 Mboldtheta2
thelawchooses_1))) Conj
Ug ([(D5_16
   : obj) =>
   ({def} Ded
   ([(dhyp5_17
      : that D5_16
      E Misset_1
      Mbold2 thelawchooses_1) =>
```

```
({def} Cases
(Excmid
(Forall
([(D6_20
   : obj) =>
   ({def} (D6_20
   E D4_9) ->
   D5_16
   <<= D6_20
   : prop)])), [(casehyp1_18
   : that
   Forall
   ([(D7_20
      : obj) =>
      ({def} (D7_20
      E D4_9) ->
      D5_16
      <<=
      D7_20
      : prop)])) =>
   ({def} ((D4_9
   Intersection
   F4_11) <<=
   D5_16) Add1
   (D5_16
   <<= D4_9
   Intersection
   F4_11) Fixform
   Ug ([(G_22
      : obj) =>
      ({def} Ded
      ([(ghyp_23
         : that
         G_22
         E D5_16) =>
         ({def} (G_22
         E D4_9
         Intersection
```

```
F4_11) Fixform
fhyp4_12
Mp
F4_11
Ui
Ug
([(B1_29
   : obj) =>
   ({def} Ded
   ([(bhyp1_30
      : that
      B1_29
      E D4_9) =>
      ({def} ghyp_23
      Mpsubs
      bhyp1_30
      Mp
      B1_29
      Ui
      casehyp1_18
      : that
      G_22
      E B1_29)]) : that
   (B1_29
   E D4_9) ->
   G_22
   E B1_29)]) Conj
Ug ([(B1_27
   : obj) =>
   ({def} Ded
   ([(bhyp1_28
      : that
      B1_27
      E D4_9) =>
      ({def} ghyp_23
      Mpsubs
      bhyp1_28
      Mp
```

```
            B1_27
            Ui
            casehyp1_18
            : that
            G_22
            E B1_27)]) : that
         (B1_27
         E D4_9) ->
         G_22
         E B1_27)]) Iff2
      G_22
      Ui Separation4
      (Refleq
      (D4_9
      Intersection
      F4_11)) : that
      G_22
      E D4_9
      Intersection
      F4_11)]) : that
   (G_22
   E D5_16) ->
   G_22 E D4_9
   Intersection
   F4_11)]) Conj
Setsinchains2
(Misset_1, thelawchooses_1, Misset_1
Mboldtheta2
thelawchooses_1, dhyp5_17) Conj
Separation3
(Refleq (D4_9
Intersection
F4_11)) : that
(D5_16 <<=
D4_9 Intersection
F4_11) V (D4_9
Intersection
F4_11) <<=
```

```
D5_16)], [(casehyp2_18
: that
~ (Forall
([(D7_21
   : obj) =>
   ({def} (D7_21
   E D4_9) ->
   D5_16
   <<=
   D7_21
   : prop)]))) =>
({def} (D5_16
<<= D4_9
Intersection
F4_11) Add2
((D4_9
Intersection
F4_11) <<=
D5_16) Fixform
Ug ([(G_22
   : obj) =>
   ({def} Ded
   ([(ghyp_23
      : that
      G_22
      E D4_9
      Intersection
      F4_11) =>
      ({def} Counterexample
      (casehyp2_18) Eg
      [(.H_24
         : obj), (hhyp_24
         : that
         Counterexample
         (casehyp2_18) Witnesses
         .H_24) =>
         ({def} Notimp2
         (hhyp_24) Mp
```

```
      .H_24
      Ui
      Simp2
      (ghyp_23
      Iff1
      G_22
      Ui
      Separation4
      (Refleq
      (D4_9
      Intersection
      F4_11))) Mpsubs
      dhyp5_17
      Mp
      D5_16
      Ui
      Simp2
      (Simp2
      (Notimp2
      (hhyp_24) Mpsubs
      dhyp4_10
      Iff1
      .H_24
      Ui
      Separation4
      (Refleq
      (Misset_1
      Cuts3
      thelawchooses_1)))) Ds2
      Notimp1
      (hhyp_24) : that
      G_22
      E D5_16)] : that
   G_22
   E D5_16)]) : that
(G_22
E D4_9
Intersection
```

```
            F4_11) ->
            G_22
            E D5_16)]) Conj
          Separation3
          (Refleq
          (D4_9
          Intersection
          F4_11)) Conj
          Setsinchains2
          (Misset_1, thelawchooses_1, Misset_1
          Mboldtheta2
          thelawchooses_1, dhyp5_17) : that
          (D5_16
          <<= D4_9
          Intersection
          F4_11) V (D4_9
          Intersection
          F4_11) <<=
          D5_16)]) : that
        (D5_16
        <<= D4_9
        Intersection
        F4_11) V (D4_9
        Intersection
        F4_11) <<=
        D5_16)]) : that
      (D5_16 E Misset_1
      Mbold2 thelawchooses_1) ->
      (D5_16 <<=
      D4_9 Intersection
      F4_11) V (D4_9
      Intersection
      F4_11) <<=
      D5_16)]) Iff2
    (D4_9 Intersection
    F4_11) Ui Separation4
    (Refleq (Misset_1
    Cuts3 thelawchooses_1)) : that
```

```
                (D4_9 Intersection
                F4_11) E Misset_1
                Mbold2 thelawchooses_1
                Set [(C_14 : obj) =>
                   ({def} cuts2
                   (Misset_1, thelawchooses_1, C_14) : prop)])]) :
             (F4_11 E D4_9) ->
             (D4_9 Intersection
             F4_11) E Misset_1
             Mbold2 thelawchooses_1
             Set [(C_14 : obj) =>
                ({def} cuts2
                (Misset_1, thelawchooses_1, C_14) : prop)])]) : tha
          Forall ([(x'_11 : obj) =>
             ({def} (x'_11 E D4_9) ->
             (D4_9 Intersection
             x'_11) E Misset_1
             Mbold2 thelawchooses_1
             Set [(C_14 : obj) =>
                ({def} cuts2
                (Misset_1, thelawchooses_1, C_14) : prop)] : prop)]
       (D4_9 <<= Misset_1 Cuts3
       thelawchooses_1) -> Forall
       ([(x'_11 : obj) =>
          ({def} (x'_11 E D4_9) ->
          (D4_9 Intersection
          x'_11) E Misset_1 Mbold2
          thelawchooses_1 Set
          [(C_14 : obj) =>
             ({def} cuts2 (Misset_1, thelawchooses_1, C_14) : prop)
   (D9_2 Intersection F9_3) E Misset_1
   Mbold2 thelawchooses_1 Set
   [(C_6 : obj) =>
      ({def} cuts2 (Misset_1, thelawchooses_1, C_6) : prop)])]) :
((D9_2 <<= Misset_1 Cuts3 thelawchooses_1) & F9_3
E D9_2) -> (D9_2 Intersection
F9_3) E Misset_1 Mbold2 thelawchooses_1
Set [(C_6 : obj) =>
```

```
            ({def} cuts2 (Misset_1, thelawchooses_1, C_6) : prop)])]) : tha
         Forall ([(x'_3 : obj) =>
            ({def} ((D9_2 <<= Misset_1
            Cuts3 thelawchooses_1) & x'_3
            E D9_2) -> (D9_2 Intersection
            x'_3) E Misset_1 Mbold2 thelawchooses_1
            Set [(C_6 : obj) =>
               ({def} cuts2 (Misset_1, thelawchooses_1, C_6) : prop)] : prop)]
      Forall ([(x'_2 : obj) =>
         ({def} Forall ([(x'_3 : obj) =>
            ({def} ((x'_2 <<= Misset_1
            Cuts3 thelawchooses_1) & x'_3
            E x'_2) -> (x'_2 Intersection
            x'_3) E Misset_1 Mbold2 thelawchooses_1
            Set [(C_6 : obj) =>
               ({def} cuts2 (Misset_1, thelawchooses_1, C_6) : prop)] : prop)]


lineb155 : [(.M_1 : obj), (Misset_1
    : that Isset (.M_1)), (.thelaw_1
    : [(S_2 : obj) => (--- : obj)]), (thelawchooses_1
    : [(.S_2 : obj), (subsetev_2 : that
       .S_2 <<= .M_1), (inev_2 : that
       Exists ([(x_4 : obj) =>
          ({def} x_4 E .S_2 : prop)])) =>
       (--- : that .thelaw_1 (.S_2) E .S_2)]) =>
    (--- : that Forall ([(x'_2 : obj) =>
       ({def} Forall ([(x'_3 : obj) =>
          ({def} ((x'_2 <<= Misset_1
          Cuts3 thelawchooses_1) & x'_3
          E x'_2) -> (x'_2 Intersection
          x'_3) E Misset_1 Mbold2 thelawchooses_1
          Set [(C_6 : obj) =>
             ({def} cuts2 (Misset_1, thelawchooses_1, C_6) : prop)] : prop)]


{move 0}
```

```
   >>> open


      {move 2}

      >>> define line155 : lineb155 Misset, thelawchooses


      line155 : [
          ({def} Misset lineb155 thelawchooses
          : that Forall ([(x'_2 : obj) =>
             ({def} Forall ([(x'_3 : obj) =>
                ({def} ((x'_2 <<= Misset
                Cuts3 thelawchooses) & x'_3
                E x'_2) -> (x'_2 Intersection
                x'_3) E Misset Mbold2 thelawchooses
                Set [(C_6 : obj) =>
                   ({def} cuts2 (Misset, thelawchooses, C_6) : prop)] : prop)])


      line155 : that Forall ([(x'_2 : obj) =>
          ({def} Forall ([(x'_3 : obj) =>
             ({def} ((x'_2 <<= Misset Cuts3
             thelawchooses) & x'_3 E x'_2) ->
             (x'_2 Intersection x'_3) E Misset
             Mbold2 thelawchooses Set [(C_6
                : obj) =>
                ({def} cuts2 (Misset, thelawchooses, C_6) : prop)] : prop)]) :


      {move 1}
end Lestrade execution
```

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

```
begin Lestrade execution
```

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

Fixform (thetachain (Cuts), Line9 Conj Line12 Conj Line119 Conj line155) is not

(paused, type something to continue) >

      >>> close


   {move 1}

   >>> define Cutstheta Misset, thelawchooses \
       : Cutstheta2

[Misset thelawchooses => Cutstheta2] is not well-formed

(paused, type something to continue) >

   >>> clearcurrent

{move 1}
end Lestrade execution
```

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.

```
begin Lestrade execution

   >>> save


   {move 1}
```

```
>>> declare M obj

M : obj

{move 1}

>>> declare Misset that Isset M

Misset : that Isset (M)

{move 1}

>>> open

   {move 2}

   >>> 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_2 : obj) =>
    ({def} x_2 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_3 : obj) =>
       ({def} x_3 E .S_1 : prop)])) =>
    (--- : that thelaw (.S_1) E .S_1)]


{move 1}
```

```
    >>> open


       {move 3}

       >>> define Mbold : Mbold2 Misset, \
           thelawchooses

Mbold2 Misset thelawchooses is not well-formed

(paused, type something to continue) >

       >>> declare X obj


       X : obj


       {move 3}

       >>> define thetachain X : thetachain1 \
           M, thelaw, X


       thetachain : [(X_1 : obj) =>
           ({def} thetachain1 (M, thelaw, X_1) : prop)]


       thetachain : [(X_1 : obj) =>
           (--- : prop)]


       {move 2}

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

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


Thetachain : obj


{move 2}

>>> open


   {move 4}

   >>> 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)) =>
    ({def} Simp1 (Simp2 (theta1_1)) Iff2
    .Y_1 Ui Scthm (Sc (M)) : that
    .Y_1 E Sc (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)) =>
    ({def} thetaa1 (theta1_1) Conj
    theta1_1 Iff2 .Y_1 Ui Separation4
    (Refleq (Thetachain)) : that
    .Y_1 E Sc (Sc (M)) Set
    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) =>
            ({def} Simp2 (theta2_1 Iff1
            .Y_1 Ui Separation4 (Refleq
            (Thetachain))) : that
            thetachain (.Y_1))]


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


        {move 3}

        >>> close


     {move 3}

     >>> define Cutstheta1 : Cutstheta \
         Misset, thelawchooses

Cutstheta Misset thelawchooses is not well-formed

(paused, type something to continue) >

     >>> define Cuts : Misset Cuts3 thelawchooses
```

```
Cuts : [
    ({def} Misset Cuts3 thelawchooses
    : obj)]


Cuts : obj


{move 2}

>>> declare A obj


A : obj


{move 3}

>>> declare B obj
```

B is badly formed or already reserved or declared

(paused, type something to continue) >

```
>>> declare aev that A E Mbold
```

{declare command error}

(paused, type something to continue) >

```
>>> declare bev that B E Mbold
```

{declare command error}

(paused, type something to continue) >

```
>>> goal that (A <<= B) V B <<= \
    A


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


{move 3}

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

```
aev : Fixform (Forall [X => (X E Thetachain) -> A E X], Simp2 (Iff1 (aev, Ui A,

(paused, type something to continue) >
```

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

```
aev bev : Mp bev, Ui B, Simp2 (Simp2 (Iff1 (Mp (Theta1 Cutstheta1, Ui Cuts, lin

(paused, type something to continue) >
```

```
>>> define prime A : prime2 thelaw, A


prime : [(A_1 : obj) =>
    ({def} prime2 (thelaw, A_1) : obj)]


prime : [(A_1 : obj) => (---
    : obj)]
```

```
      {move 2}

      >>> define Mboldstrongtotal aev \
          bev : Fixform ((B <<= prime A) V A <<= \
          B, Simp2 (Separation5 Univcheat \
          (Theta1 linec17 Mp (Theta1 Cutstheta1, Ui \
          Cuts, line1 aev), line1 bev)))

aev bev : Fixform ((B <<= prime A) V A <<= B, Simp2 (Separation5 Univcheat (The

(paused, type something to continue) >

      >>> save


      {move 3}

      >>> close


   {move 2}

   >>> declare A1 obj


   A1 : obj


   {move 2}

   >>> declare B1 obj


   B1 : obj


   {move 2}
```

```
      >>> declare aev1 that A1 E Mbold

{declare command error}

(paused, type something to continue) >

      >>> declare bev1 that B1 E Mbold

{declare command error}

(paused, type something to continue) >

      >>> define Mboldtotal1 aev1 bev1 : Mboldtotal \
          aev1 bev1

aev1 bev1 : Mboldtotal aev1 bev1 is not well-formed

(paused, type something to continue) >

      >>> define Mboldstrongtotal1 aev1 bev1 \
          : Mboldstrongtotal aev1 bev1

aev1 bev1 : Mboldstrongtotal aev1 bev1 is not well-formed

(paused, type something to continue) >

      >>> save


      {move 2}

      >>> close


   {move 1}

   >>> declare A2 obj
```

```
   A2 : obj


   {move 1}

   >>> declare B2 obj


   B2 : obj


   {move 1}

   >>> declare aev2 that A2 E (Mbold2 Misset, \
       thelawchooses)

{declare command error}

(paused, type something to continue) >

   >>> declare bev2 that B2 E (Mbold2 Misset, \
       thelawchooses)

{declare command error}

(paused, type something to continue) >

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

[Misset thelawchooses, => aev2 bev2 : Mboldtotal1 aev2 bev2] is not well-formed

(paused, type something to continue) >

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

```
[Misset thelawchooses, => aev2 bev2 : Mboldstrongtotal1 aev2 bev2] is not well-

(paused, type something to continue) >
end Lestrade execution
```

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`.