Implementation of Zermelo's work of 1908 in Lestrade: Part V, working out the consequences of the main result of part IV, culminating in presentation of a well-ordering of M (with supporting proof).

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.

2 Consequences of the result of Part IV

Initially, we clear move 1 to get rid of variable clutter, and so we must recapitulate some familiar definitions.

`Lestrade execution:`

```
load whatismath4

open
clearcurrent


   define Mbold:Mbold2 Misset thelawchooses


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



   declare A1 obj

>>    A1: obj {move 2}



   declare B1 obj

>>    B1: obj {move 2}



   declare aev that A1 E Mbold

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



   declare bev that B1 E Mbold

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

```
   define Mboldstrongtotal aev bev : Mboldstrongtotal2 \
      Misset, thelawchooses, aev bev

>>    Mboldstrongtotal: [(.A1_1:obj),(aev_1:
>>         that (.A1_1 E Mbold)),(.B1_1:obj),(bev_1:
>>         that (.B1_1 E Mbold)) => (---:that
>>         ((.B1_1 <<= prime2(thelaw,.A1_1)) V
>>         (.A1_1 <<= .B1_1)))]
>>      {move 1}
```

```
   define Mboldtotal aev bev : Mboldtotal2 \
      Misset, thelawchooses, aev bev

>>    Mboldtotal: [(.A1_1:obj),(aev_1:that (.A1_1
>>         E Mbold)),(.B1_1:obj),(bev_1:that (.B1_1
>>         E Mbold)) => (---:that ((.B1_1 <<=
>>         .A1_1) V (.A1_1 <<= .B1_1)))]
>>      {move 1}
```

```
   define Mboldtheta: Mboldtheta2 Misset \
      thelawchooses

>>    Mboldtheta: [(---:that thetachain1(M,thelaw,
>>         (Misset Mbold2 thelawchooses)))]
>>      {move 1}
```

We complete the definitions we import initially. Some other imports may be made in the course of the development.

Zermelo discusses a nonempty subset P of M, the intersection P_0 of all elements of $\mathbf{M}$ containing it, and the distinguished element p_0 of P_0 (which will turn out to be an element of P, which will be the minimal element of P in the order we define on M.

```
Lestrade execution:


   declare P obj

>>    P: obj {move 2}



   define prime P: prime2 thelaw, P

>>    prime: [(P_1:obj) => (---:obj)]
>>      {move 1}



   declare Pev that P <<= M

>>    Pev: that (P <<= M) {move 2}



   declare x2 obj

>>    x2: obj {move 2}



   declare Pev2 that Exists[x2=>x2 E P] \




>>    Pev2: that Exists([(x2_1:obj) => ((x2_1
>>         E P):prop)])
```

```
>>      {move 2}


   declare x obj

>>    x: obj {move 2}


   open

      declare x1 obj

>>       x1: obj {move 3}


      define Pset: Set Mbold [x1 => P <<= \
            x1] \



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


      define P0 : Intersection(Pset,M)

>>       P0: [(---:obj)]
>>         {move 2}


      goal that P0 E Mbold
```

```
>>       Goal: that (P0 E Mbold)

      define line1: Ui M,Ui Pset,(Simp2 Simp2 \
         Simp2 Mboldtheta)

>>       line1: [(---:that (((Pset <<= (Misset
>>            Mbold2 thelawchooses)) & (M E Pset))
>>            -> ((Pset Intersection M) E (Misset
>>            Mbold2 thelawchooses))))]
>>         {move 2}



      define line2: Fixform(Pset <<= Mbold, \
         Sepsub2(Separation3 Refleq Mbold,Refleq \
         Pset))

>>       line2: [(---:that (Pset <<= Mbold))]
>>         {move 2}



      define line3: Fixform(M E Pset,Iff2(Conj \
         Simp1 Mboldtheta Pev,Ui M,Separation4 \
         Refleq Pset))

>>       line3: [(---:that (M E Pset))]
>>         {move 2}



      define line4: Fixform(P0 E Mbold,Mp \
         (Conj line2 line3, line1))

>>       line4: [(---:that (P0 E Mbold))]
>>         {move 2}
```

P_0 is in **M**.

Lestrade execution:

```
      define p0: thelaw P0

>>       p0: [(---:obj)]
>>         {move 2}



      goal that p0 E P

>>       Goal: that (p0 E P)

      open

         declare z obj

>>          z: obj {move 4}



         declare zev that z E P

>>          zev: that (z E P) {move 4}



         goal that z E P0

>>          Goal: that (z E P0)

         define line6 z: Ui z,Separation4 \
            Refleq P0

>>          line6: [(z_1:obj) => (---:that ((z_1
```

```
>>               E (M Set [(x_8:obj) => (Forall([(B_9:
>>                     obj) => (((B_9 E Pset)
>>                     -> (x_8 E B_9)):prop)])
>>                  :prop)]))
>>               == ((z_1 E M) & Forall([(B_10:
>>                  obj) => (((B_10 E Pset) ->
>>                  (z_1 E B_10)):prop)]))
>>               ))]
>>            {move 3}
```

```
         define line7 zev: Mpsubs zev Pev
```

```
>>          line7: [(.z_1:obj),(zev_1:that (.z_1
>>               E P)) => (---:that (.z_1 E M))]
>>            {move 3}
```

```
         open

            declare B obj

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

```
            open

               declare Bev that B E Pset
```

```
>>                Bev: that (B E Pset) {move
>>                  6}
```

```
                goal that z E B

>>                 Goal: that (z E B)

                define line8 Bev: Mpsubs (zev, \
                   Simp2(Iff1(Bev,Ui B,Separation4 \
                   Refleq Pset)))

>>                 line8: [(Bev_1:that (B E Pset))
>>                      => (---:that (z E B))]
>>                   {move 5}



                close

             define line9 B: Ded line8

>>              line9: [(B_1:obj) => (---:that
>>                   ((B_1 E Pset) -> (z E B_1)))]
>>                {move 4}



             close

          define line10 zev: Ug line9

>>           line10: [(.z_1:obj),(zev_1:that
>>                (.z_1 E P)) => (---:that Forall([(B_8:
>>                   obj) => (((B_8 E Pset) ->
>>                   (.z_1 E B_8)):prop)]))
>>                ]
>>             {move 3}
```

```
        define line11 zev: Fixform(z E P0, \
           Iff2(Conj line7 zev line10 zev, \
           line6 z))

>>         line11: [(.z_1:obj),(zev_1:that
>>              (.z_1 E P)) => (---:that (.z_1
>>              E P0))]
>>           {move 3}



        declare zev2 that z E P

>>         zev2: that (z E P) {move 4}



        define linea11 z: Ded [zev2 => line11 \
              zev2] \




>>         linea11: [(z_1:obj) => (---:that
>>              ((z_1 E P) -> (z_1 E P0)))]
>>           {move 3}



        declare w obj

>>         w: obj {move 4}



        define line12 zev: Fixform(Exists[w \
              => w E P0] \
           , Ei1 z line11 zev)
```

```
>>          line12: [(.z_1:obj),(zev_1:that
>>               (.z_1 E P)) => (---:that Exists([(w_4:
>>                  obj) => ((w_4 E P0):prop)]))
>>               ]
>>            {move 3}
```

```
   close

define line13: Eg Pev2 line12
```

```
>>     line13: [(---:that Exists([(w_22:obj)
>>            => ((w_22 E P0):prop)]))
>>          ]
>>       {move 2}
```

```
define linea13: Fixform(P<<= P0,Conj(Ug \
   linea11,Conj(Simp1 Simp2 Pev,Separation3 \
   Refleq P0)))
```

```
>>     linea13: [(---:that (P <<= P0))]
>>       {move 2}
```

```
define line14: Fixform(p0 E P0,thelawchooses(Sepsub2 \
   Misset Refleq P0,line13))
```

```
>>     line14: [(---:that (p0 E P0))]
>>       {move 2}
```

```
open
```

```
        declare absurdhyp that ~(p0 E P)


>>         absurdhyp: that ~((p0 E P)) {move
>>           4}



        open

           declare Q obj

>>            Q: obj {move 5}



           open

              declare Qev that Q E P

>>               Qev: that (Q E P) {move 6}



              define line15 Qev: line11 \
                 Qev

>>               line15: [(Qev_1:that (Q E
>>                    P)) => (---:that (Q E P0))]
>>                 {move 5}



              open

                 declare eqtest that Q E \
                    Usc p0
```

```
>>                     eqtest: that (Q E Usc(p0))
>>                       {move 7}



                    define line16 eqtest:Inusc1 \
                       eqtest

>>                     line16: [(eqtest_1:that
>>                          (Q E Usc(p0))) => (---:
>>                          that (Q = p0))]
>>                       {move 6}



                    define line17 eqtest: Mp(Qev, \
                       Subs1(Eqsymm line16 eqtest, \
                       absurdhyp))

>>                     line17: [(eqtest_1:that
>>                          (Q E Usc(p0))) => (---:
>>                          that ??)]
>>                       {move 6}



                    close

                 define line18 Qev : Negintro \
                    line17

>>                  line18: [(Qev_1:that (Q E
>>                       P)) => (---:that ~((Q E
>>                       Usc(p0))))]
>>                    {move 5}
```

```
            define line19 Qev: Fixform(Q \
               E prime P0,Iff2(Conj(line15 \
               Qev,line18 Qev),Ui Q,Separation4 \
               Refleq (prime P0)))

>>             line19: [(Qev_1:that (Q E
>>                  P)) => (---:that (Q E prime(P0)))]
>>               {move 5}



            close

         define line20 Q: Ded line19

>>          line20: [(Q_1:obj) => (---:that
>>               ((Q_1 E P) -> (Q_1 E prime(P0))))]
>>            {move 4}



         save

         close

      define line21 absurdhyp: Fixform(P \
         <<= prime P0,Conj(Ug line20,Conj(Add2(P=0, \
         Pev2),Separation3 Refleq prime \
         P0)))

>>       line21: [(absurdhyp_1:that ~((p0
>>            E P))) => (---:that (P <<= prime(P0)))]
>>         {move 3}



      define line22 absurdhyp: Ui prime \
```

```
            P0, Simp2 Iff1(line14,Ui p0,Separation4 \
            Refleq P0)

>>          line22: [(absurdhyp_1:that ~((p0
>>               E P))) => (---:that ((prime(P0)
>>               E Pset) -> (p0 E prime(P0))))]
>>            {move 3}
```

```
         define linea23 absurdhyp: Mp(line4, \
            Ui P0,Simp1 Simp2 Simp2 Mboldtheta)


>>          linea23: [(absurdhyp_1:that ~((p0
>>               E P))) => (---:that (prime2(thelaw,
>>               P0) E (Misset Mbold2 thelawchooses)))]
>>            {move 3}
```

```
         define line23 absurdhyp: Fixform((prime \
            P0) E Pset,Iff2(Conj(linea23 absurdhyp, \
            line21 absurdhyp),Ui prime P0, \
            Separation4 Refleq Pset))

>>          line23: [(absurdhyp_1:that ~((p0
>>               E P))) => (---:that (prime(P0)
>>               E Pset))]
>>            {move 3}
```

```
         define line24 absurdhyp: Mp line23 \
            absurdhyp line22 absurdhyp

>>          line24: [(absurdhyp_1:that ~((p0
>>               E P))) => (---:that (p0 E prime(P0)))]
```

```
>>             {move 3}



          define line25 absurdhyp: Simp2(Iff1(line24 \
             absurdhyp,Ui p0,Separation4 Refleq \
             prime P0))

>>           line25: [(absurdhyp_1:that ~((p0
>>                E P))) => (---:that ~((p0 E Usc(thelaw(P0)))))]
>>             {move 3}



          define line26 absurdhyp: Mp (Inusc2 \
             p0,line25 absurdhyp)

>>           line26: [(absurdhyp_1:that ~((p0
>>                E P))) => (---:that ??)]
>>             {move 3}



          save

          close

       define line27 : Dneg Negintro line26


>>        line27: [(---:that (p0 E P))]
>>          {move 2}
```

p_0 is in P (not merely in P_0, which is fairly obvious).

```
Lestrade execution:
```

```
      declare P1 obj

>>       P1: obj {move 3}



      goal that ~(thelaw P1) E prime P1

>>       Goal: that (~(thelaw(P1)) E prime(P1))


      open

         declare neghyp that (thelaw P1) \
            E prime P1

>>          neghyp: that (thelaw(P1) E prime(P1))
>>            {move 4}



         define line28 neghyp: Simp2(Separation5 \
            neghyp)

>>          line28: [(neghyp_1:that (thelaw(P1)
>>               E prime(P1))) => (---:that ~((thelaw(P1)
>>               E Usc(thelaw(P1)))))]
>>            {move 3}



         define line29 neghyp: Mp(Inusc2 \
            thelaw P1,line28 neghyp)

>>          line29: [(neghyp_1:that (thelaw(P1)
>>               E prime(P1))) => (---:that ??)]
```

```
>>            {move 3}



         close

      define primefact1 P1: Negintro line29


>>       primefact1: [(P1_1:obj) => (---:that
>>            ~((thelaw(P1_1) E prime(P1_1))))]
>>         {move 2}



      save

      close

   declare P2 obj

>>    P2: obj {move 2}



   define primefact2 P2:primefact1 P2

>>    primefact2: [(P2_1:obj) => (---:that ~((thelaw(P2_1)
>>         E prime(P2_1))))]
>>      {move 1}



   save

   close

declare P3 obj
```

```
>> P3: obj {move 1}



define primefact3 Misset, thelawchooses, \
   P3:primefact2 P3

>> primefact3: [(.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))]),
>>      (P3_1:obj) => (Negintro([(neghyp_5:that
>>         (.thelaw_1(P3_1) E prime2(.thelaw_1,
>>         P3_1))) => ((Inusc2(.thelaw_1(P3_1))
>>         Mp Simp2(Separation5(neghyp_5))):that
>>         ??)])
>>      :that ~((.thelaw_1(P3_1) E prime2(.thelaw_1,
>>      P3_1))))]
>>   {move 0}



open

   define primefact4 P2: primefact3 Misset, \
      thelawchooses, P2

>>    primefact4: [(P2_1:obj) => (---:that ~((thelaw(P2_1)
>>         E prime2(thelaw,P2_1))))]
>>      {move 1}



   open
```

```
      define primefact P1:primefact4 P1

>>        primefact: [(P1_1:obj) => (---:that
>>             ~((thelaw(P1_1) E prime2(thelaw,
>>             P1_1))))]
>>          {move 2}
```

This is an obvious lemma about the prime operation which should have been proved in the fourth document.

We suppose below that a set P_1 belongs to $\mathbf{M}$, includes P as a subset, and is not equal to P_0. We show that P_0 is a subset of P_1 and P_0 is a subset of P_1', so the distinguished element of P_1 is not in P_0 and so not in P. This means that P_0 is the only element of $\mathbf{M}$ which includes P and whose distinguished element is in P.

```
Lestrade execution:


      open

         declare phyp0 that P1 E Mbold

>>          phyp0: that (P1 E Mbold) {move 4}



         declare phyp1 that P <<= P1

>>          phyp1: that (P <<= P1) {move 4}



         declare phyp2 that ~(P1 = P0)

>>          phyp2: that ~((P1 = P0)) {move 4}
```

```
        goal that P0 <<= P1

>>         Goal: that (P0 <<= P1)

        open

           declare z obj

>>            z: obj {move 5}



           open

              declare zev that z E P0

>>               zev: that (z E P0) {move 6}



              goal that z E P1

>>               Goal: that (z E P1)

              define line30 zev: Ui P1 Simp2 \
                 Separation5 zev

>>               line30: [(zev_1:that (z E
>>                    P0)) => (---:that ((P1
>>                    E Pset) -> (z E P1)))]
>>                 {move 5}



              define line31 zev: Fixform(P1 \
```

```
              E Pset,Iff2(Conj phyp0 phyp1, \
              Ui P1 Separation4 Refleq \
              Pset))

>>            line31: [(zev_1:that (z E
>>                 P0)) => (---:that (P1 E
>>                 Pset))]
>>              {move 5}



           define line32 zev : Mp line31 \
              zev, line30 zev

>>            line32: [(zev_1:that (z E
>>                 P0)) => (---:that (z E
>>                 P1))]
>>              {move 5}



           close

        define line33 z: Ded line32

>>         line33: [(z_1:obj) => (---:that
>>              ((z_1 E P0) -> (z_1 E P1)))]
>>           {move 4}



        define line34: Fixform(P0 <<= \
           P1,Conj(Ug line33, Conj(Separation3 \
           Refleq P0,Simp2 Simp2 phyp1)))


>>         line34: [(---:that (P0 <<= P1))]
>>           {move 4}
```

P_0 is a subset of P_1.

Lestrade execution:

```
            goal that P0 <<= prime P1

>>             Goal: that (P0 <<= prime(P1))


            goal that ~(P1 <<= P0)

>>             Goal: that ~((P1 <<= P0))

            open

               declare sillyhyp that P1 <<= \
                  P0

>>                sillyhyp: that (P1 <<= P0)
>>                  {move 6}



               define line35 sillyhyp: Mp \
                  Antisymsub sillyhyp line34 \
                  phyp2

>>                line35: [(sillyhyp_1:that
>>                     (P1 <<= P0)) => (---:that
>>                     ??)]
>>                  {move 5}
```

```
          close

        define line36: Negintro line35


>>            line36: [(---:that ~((P1 <<=
>>                P0)))]
>>              {move 4}



        define line37: Fixform(P0 <<= \
          prime P1,Ds1 Mboldstrongtotal \
          phyp0 line4 line36)

>>            line37: [(---:that (P0 <<= prime(P1)))]
>>              {move 4}
```

and in fact a subset of P_1'

Lestrade execution:

```
        goal that ~(thelaw P1 E P)

>>            Goal: that ~((thelaw(P1) E P))


        open

          declare sillyhyp that thelaw \
            P1 E P

>>              sillyhyp: that (thelaw(P1)
>>                E P) {move 6}
```

```
              define line38 sillyhyp: Mp \
                 Mpsubs Mpsubs sillyhyp linea13 \
                 line37 primefact P1

>>               line38: [(sillyhyp_1:that
>>                    (thelaw(P1) E P)) => (---:
>>                    that ??)]
>>                 {move 5}



              close

            define line39 : Negintro line38


>>            line39: [(---:that ~((thelaw(P1)
>>                 E P)))]
>>              {move 4}
```

so the distinguished element of P_1 is not in P.

```
Lestrade execution:


            close

          define Line34 phyp0 phyp1 phyp2 \
            : line34

>>          Line34: [(phyp0_1:that (P1 E Mbold)),
>>               (phyp1_1:that (P <<= P1)),(phyp2_1:
>>               that ~((P1 = P0))) => (---:that
>>               (P0 <<= P1))]
```

```
>>            {move 3}



         define Line37 phyp0 phyp1 phyp2: \
            line37

>>          Line37: [(phyp0_1:that (P1 E Mbold)),
>>               (phyp1_1:that (P <<= P1)),(phyp2_1:
>>               that ~((P1 = P0))) => (---:that
>>               (P0 <<= prime(P1)))]
>>            {move 3}



         define Line39 phyp0 phyp1 phyp2: \
            line39

>>          Line39: [(phyp0_1:that (P1 E Mbold)),
>>               (phyp1_1:that (P <<= P1)),(phyp2_1:
>>               that ~((P1 = P0))) => (---:that
>>               ~((thelaw(P1) E P)))]
>>            {move 3}



         close

      declare phyps that (P1 E Mbold) & (P \
         <<= P1) & ~(P1=P0)

>>       phyps: that ((P1 E Mbold) & ((P <<=
>>         P1) & ~((P1 = P0)))) {move 3}



      define Lemma34 phyps: Line34 Simp1 \
         phyps Simp1 Simp2 phyps Simp2 Simp2 \
```

```
         phyps

>>       Lemma34: [(.P1_1:obj),(phyps_1:that
>>            ((.P1_1 E Mbold) & ((P <<= .P1_1)
>>            & ~((.P1_1 = P0))))) => (---:that
>>            (P0 <<= .P1_1))]
>>         {move 2}
```

```
      define Lemma37 phyps: Line37 Simp1 \
         phyps Simp1 Simp2 phyps Simp2 Simp2 \
         phyps

>>       Lemma37: [(.P1_1:obj),(phyps_1:that
>>            ((.P1_1 E Mbold) & ((P <<= .P1_1)
>>            & ~((.P1_1 = P0))))) => (---:that
>>            (P0 <<= prime(.P1_1)))]
>>         {move 2}
```

```
      define Lemma39 phyps: Line39 Simp1 \
         phyps Simp1 Simp2 phyps Simp2 Simp2 \
         phyps

>>       Lemma39: [(.P1_1:obj),(phyps_1:that
>>            ((.P1_1 E Mbold) & ((P <<= .P1_1)
>>            & ~((.P1_1 = P0))))) => (---:that
>>            ~((thelaw(.P1_1) E P)))]
>>         {move 2}
```

Some results are recapitulated at lower moves.

```
Lestrade execution:
```

```
      declare phyps2 that (P1 E Mbold) & \
         (P <<= P1) & thelaw P1 E P

>>       phyps2: that ((P1 E Mbold) & ((P <<=
>>         P1) & (thelaw(P1) E P))) {move 3}



      goal that P1 = P0

>>       Goal: that (P1 = P0)

      open

         declare sillyhyp that ~(P1 = P0)


>>          sillyhyp: that ~((P1 = P0)) {move
>>            4}



         define line40 sillyhyp:Mp(Simp2 \
            Simp2 phyps2, Lemma39 (Conj(Simp1 \
            phyps2,Conj(Simp1 Simp2 phyps2, \
            sillyhyp))))

>>          line40: [(sillyhyp_1:that ~((P1
>>               = P0))) => (---:that ??)]
>>            {move 3}



         close

      define line41 phyps2: Dneg(Negintro \
         line40)
```

```
>>        line41: [(.P1_1:obj),(phyps2_1:that
>>             ((.P1_1 E Mbold) & ((P <<= .P1_1)
>>             & (thelaw(.P1_1) E P)))) => (---:
>>             that (.P1_1 = P0))]
>>          {move 2}
```

```
      close
```

Above we show the corollary that if a set is a an element of **M**, a superset of P, and has distinguished element in P, then in fact it is P_0.

Lestrade execution:

```
   define Rcal1 P: P0

>>    Rcal1: [(P_1:obj) => (---:obj)]
>>      {move 1}
```

```
   define Rcal x: Rcal1 Usc x

>>    Rcal: [(x_1:obj) => (---:obj)]
>>      {move 1}
```

We define the function $\mathcal{R}_1$sending an arbitrary nonempty subset P of M to P_0 as defined above (the intersection of all elements of **M** containing it) and the function $\mathcal{R}$ defined by Zermelo, $\mathcal{R}(x)$ being $\mathcal{R}_1(\{x\})$, the intersection of all elements of **M** containing x.

Lestrade execution:

```
   goal that (thelaw Rcal x) = x

>>    Goal: that (thelaw(Rcal(x)) = x)

   define Linea27 Pev Pev2 : Fixform((thelaw(Rcal1 \
      P))E P,line27)

>>    Linea27: [(.P_1:obj),(Pev_1:that (.P_1
>>         <<= M)),(Pev2_1:that Exists([(x2_2:
>>            obj) => ((x2_2 E .P_1):prop)]))
>>         => (---:that (thelaw(Rcal1(.P_1)) E
>>         .P_1))]
>>      {move 1}


   save

   close

declare P77 obj

>> P77: obj {move 1}


declare Pev77 that P77 <<= M

>> Pev77: that (P77 <<= M) {move 1}


declare x77 obj

>> x77: obj {move 1}
```

```
declare Pev277 that Exists[x77 => x77 EP77] \
```

```
>> Pev277: that Exists([(x77_1:obj) => ((x77_1
>>      E P77):prop)])
>>   {move 1}
```

```
define Lineb27 Misset, thelawchooses, Pev77, \
   Pev277: Linea27 Pev77 Pev277

>> Lineb27: [(.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))]),
>>      (.P77_1:obj),(Pev77_1:that (.P77_1 <<=
>>      .M_1)),(Pev277_1:that Exists([(x77_5:obj)
>>         => ((x77_5 E .P77_1):prop)]))
>>      => (((.thelaw_1((((Misset_1 Mbold2 thelawchooses_1)
>>      Set [(x1_6:obj) => ((.P77_1 <<= x1_6):
>>         prop)])
>>      Intersection .M_1)) E .P77_1) Fixform
>>      Dneg(Negintro([(absurdhyp_9:that ~((.thelaw_1((((Misset_1
>>         Mbold2 thelawchooses_1) Set [(x1_10:
>>            obj) => ((.P77_1 <<= x1_10):prop)])
>>         Intersection .M_1)) E .P77_1))) =>
>>         ((Inusc2(.thelaw_1((((Misset_1 Mbold2
>>         thelawchooses_1) Set [(x1_14:obj) =>
>>            ((.P77_1 <<= x1_14):prop)])
>>         Intersection .M_1))) Mp Simp2(((((prime2(.thelaw_1,
>>         (((Misset_1 Mbold2 thelawchooses_1)
```

```
>>         Set [(x1_23:obj) => ((.P77_1 <<= x1_23):
>>            prop)])
>>         Intersection .M_1)) E ((Misset_1 Mbold2
>>         thelawchooses_1) Set [(x1_24:obj) =>
>>            ((.P77_1 <<= x1_24):prop)]))
>>         Fixform (((((((Misset_1 Mbold2 thelawchooses_1)
>>         Set [(x1_29:obj) => ((.P77_1 <<= x1_29):
>>            prop)])
>>         Intersection .M_1) E (Misset_1 Mbold2
>>         thelawchooses_1)) Fixform ((((((Misset_1
>>         Mbold2 thelawchooses_1) Set [(x1_33:
>>            obj) => ((.P77_1 <<= x1_33):prop)])
>>         <<= (Misset_1 Mbold2 thelawchooses_1))
>>         Fixform (Separation3(Refleq((Misset_1
>>         Mbold2 thelawchooses_1))) Sepsub2 Refleq(((Misset_1
>>         Mbold2 thelawchooses_1) Set [(x1_41:
>>            obj) => ((.P77_1 <<= x1_41):prop)]))
>>         )) Conj ((.M_1 E ((Misset_1 Mbold2
>>         thelawchooses_1) Set [(x1_43:obj) =>
>>            ((.P77_1 <<= x1_43):prop)]))
>>         Fixform ((Simp1((Misset_1 Mboldtheta2
>>         thelawchooses_1)) Conj Pev77_1) Iff2
>>         (.M_1 Ui Separation4(Refleq(((Misset_1
>>         Mbold2 thelawchooses_1) Set [(x1_52:
>>            obj) => ((.P77_1 <<= x1_52):prop)]))
>>         ))))) Mp (.M_1 Ui (((Misset_1 Mbold2
>>         thelawchooses_1) Set [(x1_58:obj) =>
>>            ((.P77_1 <<= x1_58):prop)])
>>         Ui Simp2(Simp2(Simp2((Misset_1 Mboldtheta2
>>         thelawchooses_1))))))) Mp ((((Misset_1
>>         Mbold2 thelawchooses_1) Set [(x1_71:
>>            obj) => ((.P77_1 <<= x1_71):prop)])
>>         Intersection .M_1) Ui Simp1(Simp2(Simp2((Misset_1
>>         Mboldtheta2 thelawchooses_1)))))) Conj
>>         ((.P77_1 <<= prime2(.thelaw_1,(((Misset_1
>>         Mbold2 thelawchooses_1) Set [(x1_83:
>>            obj) => ((.P77_1 <<= x1_83):prop)])
>>         Intersection .M_1))) Fixform (Ug([(Q_88:
```

```
>>              obj) => (Ded([(Qev_90:that (Q_88
>>                E .P77_1)) => (((Q_88 E prime2(.thelaw_1,
>>                (((Misset_1 Mbold2 thelawchooses_1)
>>                Set [(x1_91:obj) => ((.P77_1
>>                   <<= x1_91):prop)])
>>                Intersection .M_1))) Fixform
>>                ((((Q_88 E (((Misset_1 Mbold2
>>                thelawchooses_1) Set [(x1_95:
>>                   obj) => ((.P77_1 <<= x1_95):
>>                   prop)])
>>                Intersection .M_1)) Fixform (((Qev_90
>>                Mpsubs Pev77_1) Conj Ug([(B_102:
>>                   obj) => (Ded([(Bev_104:that
>>                      (B_102 E ((Misset_1 Mbold2
>>                      thelawchooses_1) Set [(x1_105:
>>                         obj) => ((.P77_1 <<=
>>                         x1_105):prop)]))
>>                      ) => ((Qev_90 Mpsubs Simp2((Bev_104
>>                      Iff1 (B_102 Ui Separation4(Refleq(((Misset_1
>>                      Mbold2 thelawchooses_1)
>>                      Set [(x1_111:obj) => ((.P77_1
>>                         <<= x1_111):prop)]))
>>                      ))))):that (Q_88 E B_102))])
>>                   :that ((B_102 E ((Misset_1
>>                   Mbold2 thelawchooses_1) Set
>>                   [(x1_112:obj) => ((.P77_1
>>                      <<= x1_112):prop)]))
>>                   -> (Q_88 E B_102)))]))
>>                Iff2 (Q_88 Ui Separation4(Refleq((((Misset_1
>>                Mbold2 thelawchooses_1) Set [(x1_126:
>>                   obj) => ((.P77_1 <<= x1_126):
>>                   prop)])
>>                Intersection .M_1)))))) Conj
>>                Negintro([(eqtest_129:that (Q_88
>>                   E Usc(.thelaw_1((((Misset_1
>>                   Mbold2 thelawchooses_1) Set
>>                   [(x1_130:obj) => ((.P77_1
>>                      <<= x1_130):prop)])
```

```
>>                    Intersection .M_1))))) =>
>>                    ((Qev_90 Mp (Eqsymm(Inusc1(eqtest_129))
>>                    Subs1 absurdhyp_9)):that ??)]))
>>                 Iff2 (Q_88 Ui Separation4(Refleq(prime2(.thelaw_1,
>>                 (((Misset_1 Mbold2 thelawchooses_1)
>>                 Set [(x1_148:obj) => ((.P77_1
>>                    <<= x1_148):prop)])
>>                 Intersection .M_1))))))):that
>>                 (Q_88 E prime2(.thelaw_1,(((Misset_1
>>                 Mbold2 thelawchooses_1) Set [(x1_149:
>>                    obj) => ((.P77_1 <<= x1_149):
>>                    prop)])
>>                 Intersection .M_1))))])
>>              :that ((Q_88 E .P77_1) -> (Q_88
>>              E prime2(.thelaw_1,(((Misset_1 Mbold2
>>              thelawchooses_1) Set [(x1_150:obj)
>>                 => ((.P77_1 <<= x1_150):prop)])
>>              Intersection .M_1)))))])
>>           Conj (((.P77_1 = 0) Add2 Pev277_1)
>>           Conj Separation3(Refleq(prime2(.thelaw_1,
>>           (((Misset_1 Mbold2 thelawchooses_1)
>>           Set [(x1_164:obj) => ((.P77_1 <<= x1_164):
>>              prop)])
>>           Intersection .M_1)))))))) Iff2 (prime2(.thelaw_1,
>>           (((Misset_1 Mbold2 thelawchooses_1)
>>           Set [(x1_167:obj) => ((.P77_1 <<= x1_167):
>>              prop)])
>>           Intersection .M_1)) Ui Separation4(Refleq(((Misset_1
>>           Mbold2 thelawchooses_1) Set [(x1_172:
>>              obj) => ((.P77_1 <<= x1_172):prop)]))
>>           )))) Mp (prime2(.thelaw_1,(((Misset_1
>>           Mbold2 thelawchooses_1) Set [(x1_175:
>>              obj) => ((.P77_1 <<= x1_175):prop)])
>>           Intersection .M_1)) Ui Simp2((((.thelaw_1((((Misset_1
>>           Mbold2 thelawchooses_1) Set [(x1_185:
>>              obj) => ((.P77_1 <<= x1_185):prop)])
>>           Intersection .M_1)) E (((Misset_1 Mbold2
>>           thelawchooses_1) Set [(x1_186:obj)
```

```
>>            => ((.P77_1 <<= x1_186):prop)])
>>         Intersection .M_1)) Fixform thelawchooses_1((.M_1
>>         Set [(x_187:obj) => (Forall([(B_188:
>>               obj) => (((B_188 E ((Misset_1
>>               Mbold2 thelawchooses_1) Set [(x1_189:
>>                  obj) => ((.P77_1 <<= x1_189):
>>                  prop)]))
>>               -> (x_187 E B_188)):prop)])
>>            :prop)]),
>>         (Misset_1 Sepsub2 Refleq((((Misset_1
>>         Mbold2 thelawchooses_1) Set [(x1_193:
>>            obj) => ((.P77_1 <<= x1_193):prop)])
>>         Intersection .M_1))),(Pev277_1 Eg [(.z_197:
>>            obj),(zev_197:that (.z_197 E .P77_1))
>>            => ((Exists([(w_198:obj) => ((w_198
>>               E (((Misset_1 Mbold2 thelawchooses_1)
>>               Set [(x1_199:obj) => ((.P77_1
>>                  <<= x1_199):prop)])
>>               Intersection .M_1)):prop)])
>>            Fixform (.z_197 Ei1 ((.z_197 E (((Misset_1
>>            Mbold2 thelawchooses_1) Set [(x1_202:
>>               obj) => ((.P77_1 <<= x1_202):
>>               prop)])
>>            Intersection .M_1)) Fixform (((zev_197
>>            Mpsubs Pev77_1) Conj Ug([(B_209:
>>               obj) => (Ded([(Bev_211:that (B_209
>>                  E ((Misset_1 Mbold2 thelawchooses_1)
>>                  Set [(x1_212:obj) => ((.P77_1
>>                     <<= x1_212):prop)]))
>>                  ) => ((zev_197 Mpsubs Simp2((Bev_211
>>                  Iff1 (B_209 Ui Separation4(Refleq(((Misset_1
>>                  Mbold2 thelawchooses_1) Set
>>                  [(x1_218:obj) => ((.P77_1
>>                     <<= x1_218):prop)]))
>>                  ))))):that (.z_197 E B_209))])
>>               :that ((B_209 E ((Misset_1 Mbold2
>>               thelawchooses_1) Set [(x1_219:
>>                  obj) => ((.P77_1 <<= x1_219):
```

```
>>                 prop)]))
>>               -> (.z_197 E B_209)))]))
>>             Iff2 (.z_197 Ui Separation4(Refleq((((Misset_1
>>             Mbold2 thelawchooses_1) Set [(x1_233:
>>               obj) => ((.P77_1 <<= x1_233):
>>               prop)])
>>             Intersection .M_1)))))))):that Exists([(w_234:
>>               obj) => ((w_234 E (((Misset_1
>>               Mbold2 thelawchooses_1) Set [(x1_235:
>>                 obj) => ((.P77_1 <<= x1_235):
>>                 prop)])
>>               Intersection .M_1)):prop)]))
>>             ]))
>>           ) Iff1 (.thelaw_1((((Misset_1 Mbold2
>>           thelawchooses_1) Set [(x1_240:obj)
>>             => ((.P77_1 <<= x1_240):prop)])
>>           Intersection .M_1)) Ui Separation4(Refleq((((Misset_1
>>           Mbold2 thelawchooses_1) Set [(x1_251:
>>             obj) => ((.P77_1 <<= x1_251):prop)])
>>           Intersection .M_1)))))))) Iff1 (.thelaw_1((((Misset_1
>>           Mbold2 thelawchooses_1) Set [(x1_256:
>>             obj) => ((.P77_1 <<= x1_256):prop)])
>>           Intersection .M_1)) Ui Separation4(Refleq(prime2(.thelaw_1,
>>           (((Misset_1 Mbold2 thelawchooses_1)
>>           Set [(x1_267:obj) => ((.P77_1 <<= x1_267):
>>             prop)])
>>           Intersection .M_1)))))))):that ??)]))
>>         ):that (.thelaw_1((((Misset_1 Mbold2 thelawchooses_1)
>>         Set [(x1_268:obj) => ((.P77_1 <<= x1_268):
>>           prop)])
>>         Intersection .M_1)) E .P77_1))]
>>     {move 0}
```

```
open

   define Line27 Pev Pev2: Lineb27 Misset, \
```

```
      thelawchooses, Pev, Pev2

>>    Line27: [(.P_1:obj),(Pev_1:that (.P_1
>>         <<= M)),(Pev2_1:that Exists([(x2_2:
>>            obj) => ((x2_2 E .P_1):prop)]))
>>         => (---:that (thelaw((((Misset Mbold2
>>         thelawchooses) Set [(x1_3:obj) => ((.P_1
>>            <<= x1_3):prop)])
>>         Intersection M)) E .P_1))]
>>      {move 1}



   declare xinm that x E M

>>    xinm: that (x E M) {move 2}




   open

      define line42: Iff2 xinm, Uscsubs x \
         M

>>       line42: [(---:that (Usc(x) <<= M))]
>>         {move 2}




      define line43: Pairinhabited x x

>>       line43: [(---:that Exists([(u_1:obj)
>>               => ((u_1 E (x ; x)):prop)]))
>>            ]
>>         {move 2}
```

```
      define line44: Fixform((thelaw(Rcal \
         x)= x),Inusc1 Line27 line42 line43)


>>       line44: [(---:that (thelaw(Rcal(x))
>>            = x))]
>>         {move 2}



      close

   define line45 xinm: line44

>>    line45: [(.x_1:obj),(xinm_1:that (.x_1
>>         E M)) => (---:that (thelaw(Rcal(.x_1))
>>         = .x_1))]
>>      {move 1}
```

We import line 27 from above all the way to move 0, then we prove that the distinguished element of $\mathcal{R}(x)$ is x.

```
Lestrade execution:


   declare Q obj

>>    Q: obj {move 2}



   declare phypsq that (Q E Mbold) & (P <<= \
      Q) & thelaw Q E P

>>    phypsq: that ((Q E Mbold) & ((P <<= Q)
>>      & (thelaw(Q) E P))) {move 2}
```

```
   define Linea41 Pev Pev2 phypsq: line41 \
      phypsq

>>    Linea41: [(.P_1:obj),(Pev_1:that (.P_1
>>         <<= M)),(Pev2_1:that Exists([(x2_2:
>>            obj) => ((x2_2 E .P_1):prop)]))
>>         ,(.Q_1:obj),(phypsq_1:that ((.Q_1 E
>>         Mbold) & ((.P_1 <<= .Q_1) & (thelaw(.Q_1)
>>         E .P_1)))) => (---:that (.Q_1 = ((Mbold
>>         Set [(x1_169:obj) => ((.P_1 <<= x1_169):
>>            prop)])
>>         Intersection M)))]
>>      {move 1}



   save

   close

declare Q77 obj

>> Q77: obj {move 1}



declare phypsq77 that (Q77 E Mbold) & (P77 \
   <<= Q77) & thelaw Q77 E P77

>> phypsq77: that ((Q77 E Mbold) & ((P77 <<=
>>   Q77) & (thelaw(Q77) E P77))) {move 1}



define Lineb41 Misset, thelawchooses, Pev77, \
```

```
   Pev277,phypsq77: Linea41 Pev77 Pev277, phypsq77


>> Lineb41: [(.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))]),
>>      (.P77_1:obj),(Pev77_1:that (.P77_1 <<=
>>      .M_1)),(Pev277_1:that Exists([(x77_5:obj)
>>         => ((x77_5 E .P77_1):prop)]))
>>      ,(.Q77_1:obj),(phypsq77_1:that ((.Q77_1
>>      E (Misset_1 Mbold2 thelawchooses_1)) &
>>      ((.P77_1 <<= .Q77_1) & (.thelaw_1(.Q77_1)
>>      E .P77_1)))) => (Dneg(Negintro([(sillyhyp_8:
>>         that ~((.Q77_1 = (((Misset_1 Mbold2
>>         thelawchooses_1) Set [(x1_9:obj) =>
>>            ((.P77_1 <<= x1_9):prop)])
>>         Intersection .M_1)))) => ((Simp2(Simp2(phypsq77_1))
>>         Mp Negintro([(sillyhyp_10:that (.thelaw_1(.Q77_1)
>>            E .P77_1)) => ((((sillyhyp_10 Mpsubs
>>            ((.P77_1 <<= (((Misset_1 Mbold2
>>            thelawchooses_1) Set [(x1_13:obj)
>>               => ((.P77_1 <<= x1_13):prop)])
>>            Intersection .M_1)) Fixform (Ug([(z_18:
>>               obj) => (Ded([(zev2_20:that (z_18
>>                  E .P77_1)) => (((z_18 E (((Misset_1
>>                  Mbold2 thelawchooses_1) Set
>>                  [(x1_21:obj) => ((.P77_1 <<=
>>                     x1_21):prop)])
>>                  Intersection .M_1)) Fixform
>>                  (((zev2_20 Mpsubs Pev77_1)
>>                  Conj Ug([(B_28:obj) => (Ded([(Bev_30:
>>                        that (B_28 E ((Misset_1
>>                        Mbold2 thelawchooses_1)
>>                        Set [(x1_31:obj) =>
```

```
>>                           ((.P77_1 <<= x1_31):
>>                           prop)]))
>>                        ) => ((zev2_20 Mpsubs
>>                        Simp2((Bev_30 Iff1 (B_28
>>                        Ui Separation4(Refleq(((Misset_1
>>                        Mbold2 thelawchooses_1)
>>                        Set [(x1_37:obj) =>
>>                           ((.P77_1 <<= x1_37):
>>                           prop)]))
>>                        ))))):that (z_18 E B_28))])
>>                     :that ((B_28 E ((Misset_1
>>                     Mbold2 thelawchooses_1)
>>                     Set [(x1_38:obj) => ((.P77_1
>>                        <<= x1_38):prop)]))
>>                     -> (z_18 E B_28)))]))
>>                  Iff2 (z_18 Ui Separation4(Refleq((((Misset_1
>>                  Mbold2 thelawchooses_1) Set
>>                  [(x1_52:obj) => ((.P77_1 <<=
>>                     x1_52):prop)])
>>                  Intersection .M_1)))))):that
>>                  (z_18 E (((Misset_1 Mbold2
>>                  thelawchooses_1) Set [(x1_53:
>>                     obj) => ((.P77_1 <<= x1_53):
>>                     prop)])
>>                  Intersection .M_1)))])
>>               :that ((z_18 E .P77_1) -> (z_18
>>               E (((Misset_1 Mbold2 thelawchooses_1)
>>               Set [(x1_54:obj) => ((.P77_1
>>                  <<= x1_54):prop)])
>>               Intersection .M_1))))])
>>            Conj (Simp1(Simp2(Pev77_1)) Conj
>>            Separation3(Refleq((((Misset_1 Mbold2
>>            thelawchooses_1) Set [(x1_66:obj)
>>               => ((.P77_1 <<= x1_66):prop)])
>>            Intersection .M_1))))))) Mpsubs
>>            (((((Misset_1 Mbold2 thelawchooses_1)
>>            Set [(x1_67:obj) => ((.P77_1 <<=
>>               x1_67):prop)])
```

```
>>             Intersection .M_1) <<= prime2(.thelaw_1,
>>             .Q77_1)) Fixform (Mboldstrongtotal2(Misset_1,
>>             thelawchooses_1,Simp1((Simp1(phypsq77_1)
>>             Conj (Simp1(Simp2(phypsq77_1)) Conj
>>             sillyhyp_8))),(((((Misset_1 Mbold2
>>             thelawchooses_1) Set [(x1_74:obj)
>>                => ((.P77_1 <<= x1_74):prop)])
>>             Intersection .M_1) E (Misset_1 Mbold2
>>             thelawchooses_1)) Fixform ((((((Misset_1
>>             Mbold2 thelawchooses_1) Set [(x1_78:
>>                obj) => ((.P77_1 <<= x1_78):prop)])
>>             <<= (Misset_1 Mbold2 thelawchooses_1))
>>             Fixform (Separation3(Refleq((Misset_1
>>             Mbold2 thelawchooses_1))) Sepsub2
>>             Refleq(((Misset_1 Mbold2 thelawchooses_1)
>>             Set [(x1_86:obj) => ((.P77_1 <<=
>>                x1_86):prop)]))
>>             )) Conj ((.M_1 E ((Misset_1 Mbold2
>>             thelawchooses_1) Set [(x1_88:obj)
>>                => ((.P77_1 <<= x1_88):prop)]))
>>             Fixform ((Simp1((Misset_1 Mboldtheta2
>>             thelawchooses_1)) Conj Pev77_1)
>>             Iff2 (.M_1 Ui Separation4(Refleq(((Misset_1
>>             Mbold2 thelawchooses_1) Set [(x1_97:
>>                obj) => ((.P77_1 <<= x1_97):prop)]))
>>             ))))) Mp (.M_1 Ui (((Misset_1 Mbold2
>>             thelawchooses_1) Set [(x1_103:obj)
>>                => ((.P77_1 <<= x1_103):prop)])
>>             Ui Simp2(Simp2(Simp2((Misset_1 Mboldtheta2
>>             thelawchooses_1)))))))))) Ds1 Negintro([(sillyhyp_116:
>>                that (.Q77_1 <<= (((Misset_1
>>                Mbold2 thelawchooses_1) Set [(x1_117:
>>                   obj) => ((.P77_1 <<= x1_117):
>>                   prop)])
>>                Intersection .M_1))) => (((sillyhyp_116
>>                Antisymsub (((((Misset_1 Mbold2
>>                thelawchooses_1) Set [(x1_120:
>>                   obj) => ((.P77_1 <<= x1_120):
```

```
>>                     prop)])
>>                  Intersection .M_1) <<= .Q77_1)
>>                  Fixform (Ug([(z_125:obj) => (Ded([(zev_127:
>>                        that (z_125 E (((Misset_1
>>                        Mbold2 thelawchooses_1)
>>                        Set [(x1_128:obj) => ((.P77_1
>>                           <<= x1_128):prop)])
>>                        Intersection .M_1))) =>
>>                        ((((.Q77_1 E ((Misset_1
>>                        Mbold2 thelawchooses_1)
>>                        Set [(x1_130:obj) => ((.P77_1
>>                           <<= x1_130):prop)]))
>>                        Fixform ((Simp1((Simp1(phypsq77_1)
>>                        Conj (Simp1(Simp2(phypsq77_1))
>>                        Conj sillyhyp_8))) Conj
>>                        Simp1(Simp2((Simp1(phypsq77_1)
>>                        Conj (Simp1(Simp2(phypsq77_1))
>>                        Conj sillyhyp_8))))) Iff2
>>                        (.Q77_1 Ui Separation4(Refleq(((Misset_1
>>                        Mbold2 thelawchooses_1)
>>                        Set [(x1_143:obj) => ((.P77_1
>>                           <<= x1_143):prop)]))
>>                        )))) Mp (.Q77_1 Ui Simp2(Separation5(zev_127)))):
>>                        that (z_125 E .Q77_1))])
>>                     :that ((z_125 E (((Misset_1
>>                     Mbold2 thelawchooses_1) Set
>>                     [(x1_151:obj) => ((.P77_1
>>                        <<= x1_151):prop)])
>>                     Intersection .M_1)) -> (z_125
>>                     E .Q77_1)))])
>>                  Conj (Separation3(Refleq((((Misset_1
>>                  Mbold2 thelawchooses_1) Set [(x1_162:
>>                     obj) => ((.P77_1 <<= x1_162):
>>                     prop)])
>>                  Intersection .M_1))) Conj Simp2(Simp2(Simp1(Simp2((Simp1(phypsq77
>>                  Conj (Simp1(Simp2(phypsq77_1))
>>                  Conj sillyhyp_8)))))))))) Mp
>>                  Simp2(Simp2((Simp1(phypsq77_1)
```

```
>>               Conj (Simp1(Simp2(phypsq77_1))
>>               Conj sillyhyp_8))))):that ??)]))
>>            )) Mp primefact3(Misset_1,thelawchooses_1,
>>            .Q77_1)):that ??)]))
>>         :that ??)]))
>>      :that (.Q77_1 = (((Misset_1 Mbold2 thelawchooses_1)
>>      Set [(x1_172:obj) => ((.P77_1 <<= x1_172):
>>         prop)])
>>      Intersection .M_1)))]
>>   {move 0}
```

```
open

   define Line41 Pev Pev2 phypsq: Lineb41 \
      Misset, thelawchooses, Pev, Pev2,phypsq
```

```
>>    Line41: [(.P_1:obj),(Pev_1:that (.P_1
>>         <<= M)),(Pev2_1:that Exists([(x2_2:
>>            obj) => ((x2_2 E .P_1):prop)]))
>>         ,(.Q_1:obj),(phypsq_1:that ((.Q_1 E
>>         Mbold) & ((.P_1 <<= .Q_1) & (thelaw(.Q_1)
>>         E .P_1)))) => (---:that (.Q_1 = (((Misset
>>         Mbold2 thelawchooses) Set [(x1_3:obj)
>>            => ((.P_1 <<= x1_3):prop)])
>>         Intersection M)))]
>>      {move 1}
```

```
   declare Qinmbold that Q E Mbold

>>    Qinmbold: that (Q E Mbold) {move 2}
```

```
   declare y obj

>>    y: obj {move 2}



   declare Qev that y E Q

>>    Qev: that (y E Q) {move 2}



   goal that (thelaw Q = x) -> Q = Rcal x


>>    Goal: that ((thelaw(Q) = x) -> (Q = Rcal(x)))


   open

      declare thehyp that thelaw Q = x

>>       thehyp: that (thelaw(Q) = x) {move
>>         3}



      define line46: Iff1(Simp1 Separation5 \
         Qinmbold,Ui Q,Scthm M)

>>       line46: [(---:that (Q <<= M))]
>>         {move 2}



      define line47 thehyp:Iff2(Subs1 thehyp, \
         thelawchooses line46, Ei1 y Qev,Uscsubs \
         x Q)
```

```
>>       line47: [(thehyp_1:that (thelaw(Q)
>>            = x)) => (---:that (Usc(x) <<= Q))]
>>         {move 2}



      declare y1 obj

>>       y1: obj {move 3}



      define line48 thehyp: Subs Eqsymm thehyp \
         [y1 => y1 E Usc x] \
         Inusc2 x

>>       line48: [(thehyp_1:that (thelaw(Q)
>>            = x)) => (---:that (thelaw(Q) E
>>            Usc(x)))]
>>         {move 2}



      define line49 thehyp: Fixform(Q = Rcal \
         x,Line41 line42 line43 (Qinmbold Conj \
         line47 thehyp Conj line48 thehyp))


>>       line49: [(thehyp_1:that (thelaw(Q)
>>            = x)) => (---:that (Q = Rcal(x)))]
>>         {move 2}



      close

   declare thehyp2 that thelaw Q = x
```

```
>>    thehyp2: that (thelaw(Q) = x) {move 2}



   define Line49 xinm Qinmbold Qev thehyp2: \
      line49 thehyp2

>>    Line49: [(.x_1:obj),(xinm_1:that (.x_1
>>         E M)),(.Q_1:obj),(Qinmbold_1:that (.Q_1
>>         E Mbold)),(.y_1:obj),(Qev_1:that (.y_1
>>         E .Q_1)),(thehyp2_1:that (thelaw(.Q_1)
>>         = .x_1)) => (---:that (.Q_1 = Rcal(.x_1)))]
>>      {move 1}
```

We import line 41 from above, then we use it to prove that if Q is an element of $\mathbf{M}$ which is nonempty and whose distinguished element is x, then $Q = \mathcal{R}(x)$.

```
Lestrade execution:


   declare a obj

>>    a: obj {move 2}



   declare b obj

>>    b: obj {move 2}



   declare ainm that a E M
```

```
>>    ainm: that (a E M) {move 2}



   declare binm that b E M

>>    binm: that (b E M) {move 2}



   define <<~ a b: (a E M) & (b E M) & ~(a=b) \
      & b E Rcal a

>>    <<~: [(a_1:obj),(b_1:obj) => (---:prop)]
>>      {move 1}



   save

   close

declare A37 obj

>> A37: obj {move 1}



declare B37 obj

>> B37: obj {move 1}



define <<<~ Misset, thelawchooses, A37 B37: \
   A37 <<~ B37

>> <<<~: [(.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))]),
>>      (A37_1:obj),(B37_1:obj) => (((A37_1 E
>>      .M_1) & ((B37_1 E .M_1) & (~((A37_1 =
>>      B37_1)) & (B37_1 E (((Misset_1 Mbold2
>>      thelawchooses_1) Set [(x1_5:obj) => ((Usc(A37_1)
>>         <<= x1_5):prop)])
>>      Intersection .M_1))))):prop)]
>>   {move 0}
```

```
open

   define <~ a b: <<<~ Misset, thelawchooses, \
      a b

>>    <~: [(a_1:obj),(b_1:obj) => (---:prop)]
>>      {move 1}
```

We define the well-ordering of M which is the fruit of all our efforts. I prove that it is a linear order in a somewhat cleaner way than he does: I show that $b \in \mathcal{R}(a)$ $(a, b \in M)$ iff $\mathcal{R}(b) \subseteq \mathcal{R}(a)$, from which this falls out neatly. The reasoning I use is quite typical of Zermelo's approach, just not exactly the same as what he does at this point.

```
Lestrade execution:

% I am going to argue for the same result in this paragraph in a simpler (I hope)


   goal that (b E Rcal a) == (Rcal b) <<= \
```

```
      Rcal a

>>    Goal: that ((b E Rcal(a)) == (Rcal(b)
>>      <<= Rcal(a)))

   define Linea4 Pev Pev2: Fixform(P0 E Mbold, \
      line4)

>>    Linea4: [(.P_1:obj),(Pev_1:that (.P_1
>>         <<= M)),(Pev2_1:that Exists([(x2_2:
>>            obj) => ((x2_2 E .P_1):prop)]))
>>         => (---:that (((Mbold Set [(x1_45:obj)
>>            => ((.P_1 <<= x1_45):prop)])
>>         Intersection M) E Mbold))]
>>      {move 1}



   save

   close

define Lineb4 Misset, thelawchooses,Pev77, \
   Pev277: Linea4 Pev77 Pev277

>> Lineb4: [(.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))]),
>>      (.P77_1:obj),(Pev77_1:that (.P77_1 <<=
>>      .M_1)),(Pev277_1:that Exists([(x77_5:obj)
>>         => ((x77_5 E .P77_1):prop)]))
>>      => (((((Misset_1 Mbold2 thelawchooses_1)
>>      Set [(x1_6:obj) => ((.P77_1 <<= x1_6):
>>         prop)])
```

```
>>      Intersection .M_1) E (Misset_1 Mbold2
>>      thelawchooses_1)) Fixform (((((Misset_1
>>      Mbold2 thelawchooses_1) Set [(x1_7:obj)
>>         => ((.P77_1 <<= x1_7):prop)])
>>      Intersection .M_1) E (Misset_1 Mbold2
>>      thelawchooses_1)) Fixform ((((((Misset_1
>>      Mbold2 thelawchooses_1) Set [(x1_11:obj)
>>         => ((.P77_1 <<= x1_11):prop)])
>>      <<= (Misset_1 Mbold2 thelawchooses_1))
>>      Fixform (Separation3(Refleq((Misset_1
>>      Mbold2 thelawchooses_1))) Sepsub2 Refleq(((Misset_1
>>      Mbold2 thelawchooses_1) Set [(x1_19:obj)
>>         => ((.P77_1 <<= x1_19):prop)]))
>>      )) Conj ((.M_1 E ((Misset_1 Mbold2 thelawchooses_1)
>>      Set [(x1_21:obj) => ((.P77_1 <<= x1_21):
>>         prop)]))
>>      Fixform ((Simp1((Misset_1 Mboldtheta2
>>      thelawchooses_1)) Conj Pev77_1) Iff2 (.M_1
>>      Ui Separation4(Refleq(((Misset_1 Mbold2
>>      thelawchooses_1) Set [(x1_30:obj) => ((.P77_1
>>         <<= x1_30):prop)]))
>>      ))))) Mp (.M_1 Ui (((Misset_1 Mbold2 thelawchooses_1)
>>      Set [(x1_36:obj) => ((.P77_1 <<= x1_36):
>>         prop)])
>>      Ui Simp2(Simp2(Simp2((Misset_1 Mboldtheta2
>>      thelawchooses_1))))))))):that ((((Misset_1
>>      Mbold2 thelawchooses_1) Set [(x1_48:obj)
>>         => ((.P77_1 <<= x1_48):prop)])
>>      Intersection .M_1) E (Misset_1 Mbold2
>>      thelawchooses_1)))]
>>   {move 0}
```

```
open

   define Line4 Pev Pev2: Lineb4 Misset, \
      thelawchooses, Pev, Pev2
```

```
>>    Line4: [(.P_1:obj),(Pev_1:that (.P_1 <<=
>>         M)),(Pev2_1:that Exists([(x2_2:obj)
>>            => ((x2_2 E .P_1):prop)]))
>>         => (---:that ((((Misset Mbold2 thelawchooses)
>>         Set [(x1_3:obj) => ((.P_1 <<= x1_3):
>>            prop)])
>>         Intersection M) E (Misset Mbold2 thelawchooses)))]
>>      {move 1}
```

```
   define Rcalinmbold xinm: Fixform(Rcal \
      x E Mbold,Line4 line42 line43)

>>    Rcalinmbold: [(.x_1:obj),(xinm_1:that
>>         (.x_1 E M)) => (---:that (Rcal(.x_1)
>>         E Mbold))]
>>      {move 1}
```

```
   define Line44 xinm: line44

>>    Line44: [(.x_1:obj),(xinm_1:that (.x_1
>>         E M)) => (---:that (thelaw(Rcal(.x_1))
>>         = .x_1))]
>>      {move 1}
```

```
   define Lineaa13 Pev Pev2: Fixform(P <<= \
      Rcal1 P,linea13)

>>    Lineaa13: [(.P_1:obj),(Pev_1:that (.P_1
>>         <<= M)),(Pev2_1:that Exists([(x2_2:
>>            obj) => ((x2_2 E .P_1):prop)]))
>>         => (---:that (.P_1 <<= Rcal1(.P_1)))]
```

```
>>      {move 1}



   save

   close

define Lineab13 Misset, thelawchooses, Pev77, \
   Pev277: Lineaa13 Pev77 Pev277

>> Lineab13: [(.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))]),
>>      (.P77_1:obj),(Pev77_1:that (.P77_1 <<=
>>      .M_1)),(Pev277_1:that Exists([(x77_5:obj)
>>         => ((x77_5 E .P77_1):prop)]))
>>      => (((.P77_1 <<= (((Misset_1 Mbold2 thelawchooses_1)
>>      Set [(x1_6:obj) => ((.P77_1 <<= x1_6):
>>         prop)])
>>      Intersection .M_1)) Fixform ((.P77_1 <<=
>>      (((Misset_1 Mbold2 thelawchooses_1) Set
>>      [(x1_7:obj) => ((.P77_1 <<= x1_7):prop)])
>>      Intersection .M_1)) Fixform (Ug([(z_12:
>>         obj) => (Ded([(zev2_14:that (z_12 E
>>            .P77_1)) => (((z_12 E (((Misset_1
>>            Mbold2 thelawchooses_1) Set [(x1_15:
>>               obj) => ((.P77_1 <<= x1_15):prop)])
>>            Intersection .M_1)) Fixform (((zev2_14
>>            Mpsubs Pev77_1) Conj Ug([(B_22:obj)
>>               => (Ded([(Bev_24:that (B_22 E
>>                  ((Misset_1 Mbold2 thelawchooses_1)
>>                  Set [(x1_25:obj) => ((.P77_1
>>                     <<= x1_25):prop)]))
```

```
>>                  ) => ((zev2_14 Mpsubs Simp2((Bev_24
>>                  Iff1 (B_22 Ui Separation4(Refleq(((Misset_1
>>                  Mbold2 thelawchooses_1) Set
>>                  [(x1_31:obj) => ((.P77_1 <<=
>>                     x1_31):prop)]))
>>                  ))))):that (z_12 E B_22))])
>>               :that ((B_22 E ((Misset_1 Mbold2
>>               thelawchooses_1) Set [(x1_32:
>>                  obj) => ((.P77_1 <<= x1_32):
>>                  prop)]))
>>               -> (z_12 E B_22)))]))
>>            Iff2 (z_12 Ui Separation4(Refleq((((Misset_1
>>            Mbold2 thelawchooses_1) Set [(x1_46:
>>               obj) => ((.P77_1 <<= x1_46):prop)])
>>            Intersection .M_1)))))):that (z_12
>>            E (((Misset_1 Mbold2 thelawchooses_1)
>>            Set [(x1_47:obj) => ((.P77_1 <<=
>>               x1_47):prop)])
>>            Intersection .M_1)))])
>>         :that ((z_12 E .P77_1) -> (z_12 E (((Misset_1
>>         Mbold2 thelawchooses_1) Set [(x1_48:
>>            obj) => ((.P77_1 <<= x1_48):prop)])
>>         Intersection .M_1))))])
>>      Conj (Simp1(Simp2(Pev77_1)) Conj Separation3(Refleq((((Misset_1
>>      Mbold2 thelawchooses_1) Set [(x1_60:obj)
>>         => ((.P77_1 <<= x1_60):prop)])
>>      Intersection .M_1))))))):that (.P77_1
>>      <<= (((Misset_1 Mbold2 thelawchooses_1)
>>      Set [(x1_61:obj) => ((.P77_1 <<= x1_61):
>>         prop)])
>>      Intersection .M_1)))]
>>   {move 0}


open

   define Linea13 Pev Pev2: Lineab13 Misset, \
```

```
      thelawchooses, Pev, Pev2

>>    Linea13: [(.P_1:obj),(Pev_1:that (.P_1
>>         <<= M)),(Pev2_1:that Exists([(x2_2:
>>            obj) => ((x2_2 E .P_1):prop)]))
>>         => (---:that (.P_1 <<= (((Misset Mbold2
>>         thelawchooses) Set [(x1_3:obj) => ((.P_1
>>            <<= x1_3):prop)])
>>         Intersection M)))]
>>      {move 1}
```

```
   define Lineb13 xinm: Iff1(Linea13 line42 \
      line43,Uscsubs x Rcal x)

>>    Lineb13: [(.x_1:obj),(xinm_1:that (.x_1
>>         E M)) => (---:that (.x_1 E Rcal(.x_1)))]
>>      {move 1}
```

I import some lines from above to support the following results.

```
Lestrade execution:


   open

      declare dir1 that b E Rcal a

>>       dir1: that (b E Rcal(a)) {move 3}



      declare dir2 that (Rcal b) <<= Rcal \
         a
```

```
>>      dir2: that (Rcal(b) <<= Rcal(a)) {move
>>        3}
```

```
define line50: Mboldstrongtotal Rcalinmbold \
   binm Rcalinmbold ainm
```

```
>>      line50: [(---:that ((Rcal(a) <<= prime2(thelaw,
>>           Rcal(b))) V (Rcal(b) <<= Rcal(a))))]
>>        {move 2}
```

```
open
```

```
   declare case1 that Rcal b <<= Rcal \
      a
```

```
>>         case1: that (Rcal(b) <<= Rcal(a))
>>           {move 4}
```

```
   define line51 case1: case1
```

```
>>         line51: [(case1_1:that (Rcal(b)
>>              <<= Rcal(a))) => (---:that (Rcal(b)
>>              <<= Rcal(a)))]
>>           {move 3}
```

```
   declare case2 that Rcal a <<= prime \
      Rcal b
```

```
>>         case2: that (Rcal(a) <<= prime(Rcal(b)))
>>           {move 4}
```

```
         define line52 case2: Mpsubs dir1 \
            case2

>>          line52: [(case2_1:that (Rcal(a)
>>               <<= prime(Rcal(b)))) => (---:
>>               that (b E prime(Rcal(b))))]
>>            {move 3}



         declare z1 obj

>>          z1: obj {move 4}



         define line53 case2: Subs(Eqsymm \
            Line44 binm,[z1=>z1 E prime(Rcal \
               b)] \
            ,line52 case2)

>>          line53: [(case2_1:that (Rcal(a)
>>               <<= prime(Rcal(b)))) => (---:
>>               that (thelaw(Rcal(b)) E prime(Rcal(b))))]
>>            {move 3}



         define line54 case2: Mp line53 case2, \
            primefact Rcal b

>>          line54: [(case2_1:that (Rcal(a)
>>               <<= prime(Rcal(b)))) => (---:
>>               that ??)]
>>            {move 3}
```

```
         declare testobj obj

>>          testobj: obj {move 4}



         define line55 case2: Giveup(Rcal \
            b <<= Rcal a,line54 case2)

>>          line55: [(case2_1:that (Rcal(a)
>>               <<= prime(Rcal(b)))) => (---:
>>               that (Rcal(b) <<= Rcal(a)))]
>>            {move 3}



         close

      define line56 dir1: Cases line50, line55, \
         line51

>>       line56: [(dir1_1:that (b E Rcal(a)))
>>            => (---:that (Rcal(b) <<= Rcal(a)))]
>>         {move 2}



      define line57 dir2: Mpsubs(Lineb13 \
         binm,dir2)

>>       line57: [(dir2_1:that (Rcal(b) <<=
>>            Rcal(a))) => (---:that (b E Rcal(a)))]
>>         {move 2}
```

```
      close

   define line58 ainm binm: Dediff line56, \
      line57

>>    line58: [(.a_1:obj),(ainm_1:that (.a_1
>>         E M)),(.b_1:obj),(binm_1:that (.b_1
>>         E M)) => (---:that ((.b_1 E Rcal(.a_1))
>>         == (Rcal(.b_1) <<= Rcal(.a_1))))]
>>      {move 1}
```

I prove that for $a, b \in M$, $b \in \mathcal{R}(a) \leftrightarrow \mathcal{R}(b) \subseteq \mathcal{R}(a)$. This makes it straightforward to establish that we have a linear order.

```
Lestrade execution:


   goal that (a = b) V (a <~ b) V (b <~ a)


>>    Goal: that ((a = b) V ((a <~ b) V (b <~
>>      a)))

   define line59 a b: Excmid (a=b)

>>    line59: [(a_1:obj),(b_1:obj) => (---:that
>>         ((a_1 = b_1) V ~((a_1 = b_1))))]
>>      {move 1}



   open

      declare case1 that a=b
```

```
>>       case1: that (a = b) {move 3}



      define line60 case1: Add1((a<~b) V \
         b <~ a,case1)

>>       line60: [(case1_1:that (a = b)) =>
>>            (---:that ((a = b) V ((a <~ b) V
>>            (b <~ a))))]
>>         {move 2}



      declare case2 that ~(a=b)

>>       case2: that ~((a = b)) {move 3}



      define line61: Mboldtotal Rcalinmbold \
         ainm Rcalinmbold binm

>>       line61: [(---:that ((Rcal(b) <<= Rcal(a))
>>            V (Rcal(a) <<= Rcal(b))))]
>>         {move 2}



      open

         declare casea1 that Rcal b <<= Rcal \
            a

>>          casea1: that (Rcal(b) <<= Rcal(a))
>>            {move 4}
```

```
        define line62 casea1: Iff2(casea1, \
           line58 ainm binm)

>>         line62: [(casea1_1:that (Rcal(b)
>>              <<= Rcal(a))) => (---:that (b
>>              E Rcal(a)))]
>>           {move 3}



        define line63 casea1: Fixform(a \
           <~ b,ainm Conj binm Conj case2 \
           Conj line62 casea1)

>>         line63: [(casea1_1:that (Rcal(b)
>>              <<= Rcal(a))) => (---:that (a
>>              <~ b))]
>>           {move 3}



        define linea63 casea1: Add2(a=b, \
           Add1(b<~ a,line63 casea1))

>>         linea63: [(casea1_1:that (Rcal(b)
>>              <<= Rcal(a))) => (---:that ((a
>>              = b) V ((a <~ b) V (b <~ a))))]
>>           {move 3}



        declare casea2 that Rcal a <<= Rcal \
           b

>>         casea2: that (Rcal(a) <<= Rcal(b))
>>           {move 4}
```

```
        define line64 casea2: Iff2(casea2, \
           line58 binm ainm)

>>         line64: [(casea2_1:that (Rcal(a)
>>              <<= Rcal(b))) => (---:that (a
>>              E Rcal(b)))]
>>           {move 3}



        define line65 casea2:Fixform(b <~a, \
           binm Conj ainm Conj Negeqsymm case2 \
           Conj line64 casea2)

>>         line65: [(casea2_1:that (Rcal(a)
>>              <<= Rcal(b))) => (---:that (b
>>              <~ a))]
>>           {move 3}



        define linea65 casea2: Add2 a=b, \
           Add2 a <~ b,line65 casea2

>>         linea65: [(casea2_1:that (Rcal(a)
>>              <<= Rcal(b))) => (---:that ((a
>>              = b) V ((a <~ b) V (b <~ a))))]
>>           {move 3}



        close

     define line66 case2: Cases line61 linea63, \
        linea65
```

```
>>       line66: [(case2_1:that ~((a = b)))
>>            => (---:that ((a = b) V ((a <~ b)
>>            V (b <~ a))))]
>>         {move 2}



      close

   define linea67 ainm binm: Cases line59 \
      a b line60, line66

>>    linea67: [(.a_1:obj),(ainm_1:that (.a_1
>>         E M)),(.b_1:obj),(binm_1:that (.b_1
>>         E M)) => (---:that ((.a_1 = .b_1) V
>>         ((.a_1 <~ .b_1) V (.b_1 <~ .a_1))))]
>>      {move 1}



   save

   close

declare A77 obj

>> A77: obj {move 1}



declare B77 obj

>> B77: obj {move 1}



declare ainm77 that A77 E M
```

```
>> ainm77: that (A77 E M) {move 1}



declare binm77 that B77 E M

>> binm77: that (B77 E M) {move 1}



define lineb67 Misset, thelawchooses, ainm77 \
   binm77: linea67 ainm77 binm77

>> lineb67: [(.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))]),
>>      (.A77_1:obj),(ainm77_1:that (.A77_1 E
>>      .M_1)),(.B77_1:obj),(binm77_1:that (.B77_1
>>      E .M_1)) => (Cases(Excmid((.A77_1 = .B77_1)),
>>      [(case1_5:that (.A77_1 = .B77_1)) => (((<<<~(Misset_1,
>>         thelawchooses_1,.A77_1,.B77_1) V <<<~(Misset_1,
>>         thelawchooses_1,.B77_1,.A77_1)) Add1
>>         case1_5):that ((.A77_1 = .B77_1) V
>>         (<<<~(Misset_1,thelawchooses_1,.A77_1,
>>         .B77_1) V <<<~(Misset_1,thelawchooses_1,
>>         .B77_1,.A77_1))))]
>>      ,[(case2_6:that ~((.A77_1 = .B77_1)))
>>         => (Cases(Mboldtotal2(Misset_1,thelawchooses_1,
>>         (((((Misset_1 Mbold2 thelawchooses_1)
>>         Set [(x1_12:obj) => ((Usc(.A77_1) <<=
>>            x1_12):prop)])
>>         Intersection .M_1) E (Misset_1 Mbold2
>>         thelawchooses_1)) Fixform Lineb4(Misset_1,
>>         thelawchooses_1,(ainm77_1 Iff2 (.A77_1
```

```
>>          Uscsubs .M_1)),(.A77_1 Pairinhabited
>>          .A77_1))),(((((Misset_1 Mbold2 thelawchooses_1)
>>          Set [(x1_14:obj) => ((Usc(.B77_1) <<=
>>             x1_14):prop)])
>>          Intersection .M_1) E (Misset_1 Mbold2
>>          thelawchooses_1)) Fixform Lineb4(Misset_1,
>>          thelawchooses_1,(binm77_1 Iff2 (.B77_1
>>          Uscsubs .M_1)),(.B77_1 Pairinhabited
>>          .B77_1)))),[(casea1_15:that ((((Misset_1
>>             Mbold2 thelawchooses_1) Set [(x1_16:
>>                obj) => ((Usc(.B77_1) <<= x1_16):
>>                prop)])
>>             Intersection .M_1) <<= (((Misset_1
>>             Mbold2 thelawchooses_1) Set [(x1_17:
>>                obj) => ((Usc(.A77_1) <<= x1_17):
>>                prop)])
>>             Intersection .M_1))) => (((.A77_1
>>             = .B77_1) Add2 (<<<~(Misset_1,thelawchooses_1,
>>             .B77_1,.A77_1) Add1 (<<<~(Misset_1,
>>             thelawchooses_1,.A77_1,.B77_1) Fixform
>>             (ainm77_1 Conj (binm77_1 Conj (case2_6
>>             Conj (casea1_15 Iff2 Dediff([(dir1_27:
>>                that (.B77_1 E (((Misset_1 Mbold2
>>                thelawchooses_1) Set [(x1_28:
>>                   obj) => ((Usc(.A77_1) <<=
>>                   x1_28):prop)])
>>                Intersection .M_1))) => (Cases(Mboldstrongtotal2(Misset_1,
>>                thelawchooses_1,(((((Misset_1
>>                Mbold2 thelawchooses_1) Set [(x1_34:
>>                   obj) => ((Usc(.B77_1) <<=
>>                   x1_34):prop)])
>>                Intersection .M_1) E (Misset_1
>>                Mbold2 thelawchooses_1)) Fixform
>>                Lineb4(Misset_1,thelawchooses_1,
>>                (binm77_1 Iff2 (.B77_1 Uscsubs
>>                .M_1)),(.B77_1 Pairinhabited
>>                .B77_1))),(((((Misset_1 Mbold2
>>                thelawchooses_1) Set [(x1_36:
```

```
>>                  obj) => ((Usc(.A77_1) <<=
>>                  x1_36):prop)])
>>               Intersection .M_1) E (Misset_1
>>               Mbold2 thelawchooses_1)) Fixform
>>               Lineb4(Misset_1,thelawchooses_1,
>>               (ainm77_1 Iff2 (.A77_1 Uscsubs
>>               .M_1)),(.A77_1 Pairinhabited
>>               .A77_1)))),[(case2_39:that ((((Misset_1
>>                  Mbold2 thelawchooses_1) Set
>>                  [(x1_40:obj) => ((Usc(.A77_1)
>>                     <<= x1_40):prop)])
>>                  Intersection .M_1) <<= prime2(.thelaw_1,
>>                  (((Misset_1 Mbold2 thelawchooses_1)
>>                  Set [(x1_41:obj) => ((Usc(.B77_1)
>>                     <<= x1_41):prop)])
>>                  Intersection .M_1)))) => ((((((Misset_1
>>                  Mbold2 thelawchooses_1) Set
>>                  [(x1_42:obj) => ((Usc(.B77_1)
>>                     <<= x1_42):prop)])
>>                  Intersection .M_1) <<= (((Misset_1
>>                  Mbold2 thelawchooses_1) Set
>>                  [(x1_43:obj) => ((Usc(.A77_1)
>>                     <<= x1_43):prop)])
>>                  Intersection .M_1)) Giveup
>>                  (Subs(Eqsymm(((.thelaw_1((((Misset_1
>>                  Mbold2 thelawchooses_1) Set
>>                  [(x1_48:obj) => ((Usc(.B77_1)
>>                     <<= x1_48):prop)])
>>                  Intersection .M_1)) = .B77_1)
>>                  Fixform Inusc1(Lineb27(Misset_1,
>>                  thelawchooses_1,(binm77_1
>>                  Iff2 (.B77_1 Uscsubs .M_1)),
>>                  (.B77_1 Pairinhabited .B77_1))))),
>>                  [(z1_50:obj) => ((z1_50 E
>>                     prime2(.thelaw_1,(((Misset_1
>>                     Mbold2 thelawchooses_1)
>>                     Set [(x1_51:obj) => ((Usc(.B77_1)
>>                        <<= x1_51):prop)])
```

```
>>                        Intersection .M_1))):prop)]
>>                     ,(dir1_27 Mpsubs case2_39))
>>                     Mp primefact3(Misset_1,thelawchooses_1,
>>                     (((Misset_1 Mbold2 thelawchooses_1)
>>                     Set [(x1_54:obj) => ((Usc(.B77_1)
>>                        <<= x1_54):prop)])
>>                     Intersection .M_1)))):that
>>                     ((((Misset_1 Mbold2 thelawchooses_1)
>>                     Set [(x1_55:obj) => ((Usc(.B77_1)
>>                        <<= x1_55):prop)])
>>                     Intersection .M_1) <<= (((Misset_1
>>                     Mbold2 thelawchooses_1) Set
>>                     [(x1_56:obj) => ((Usc(.A77_1)
>>                        <<= x1_56):prop)])
>>                     Intersection .M_1)))]
>>                  ,[(case1_57:that ((((Misset_1
>>                     Mbold2 thelawchooses_1) Set
>>                     [(x1_58:obj) => ((Usc(.B77_1)
>>                        <<= x1_58):prop)])
>>                     Intersection .M_1) <<= (((Misset_1
>>                     Mbold2 thelawchooses_1) Set
>>                     [(x1_59:obj) => ((Usc(.A77_1)
>>                        <<= x1_59):prop)])
>>                     Intersection .M_1))) => (case1_57:
>>                     that ((((Misset_1 Mbold2 thelawchooses_1)
>>                     Set [(x1_60:obj) => ((Usc(.B77_1)
>>                        <<= x1_60):prop)])
>>                     Intersection .M_1) <<= (((Misset_1
>>                     Mbold2 thelawchooses_1) Set
>>                     [(x1_61:obj) => ((Usc(.A77_1)
>>                        <<= x1_61):prop)])
>>                     Intersection .M_1)))])
>>                  :that ((((Misset_1 Mbold2 thelawchooses_1)
>>                  Set [(x1_62:obj) => ((Usc(.B77_1)
>>                     <<= x1_62):prop)])
>>                  Intersection .M_1) <<= (((Misset_1
>>                  Mbold2 thelawchooses_1) Set [(x1_63:
>>                     obj) => ((Usc(.A77_1) <<=
```

```
>>                x1_63):prop)])
>>             Intersection .M_1)))]
>>          ,[(dir2_64:that ((((Misset_1 Mbold2
>>             thelawchooses_1) Set [(x1_65:
>>                obj) => ((Usc(.B77_1) <<=
>>                x1_65):prop)])
>>             Intersection .M_1) <<= (((Misset_1
>>             Mbold2 thelawchooses_1) Set [(x1_66:
>>                obj) => ((Usc(.A77_1) <<=
>>                x1_66):prop)])
>>             Intersection .M_1))) => (((Lineab13(Misset_1,
>>             thelawchooses_1,(binm77_1 Iff2
>>             (.B77_1 Uscsubs .M_1)),(.B77_1
>>             Pairinhabited .B77_1)) Iff1 (.B77_1
>>             Uscsubs (((Misset_1 Mbold2 thelawchooses_1)
>>             Set [(x1_70:obj) => ((Usc(.B77_1)
>>                <<= x1_70):prop)])
>>             Intersection .M_1))) Mpsubs dir2_64):
>>             that (.B77_1 E (((Misset_1 Mbold2
>>             thelawchooses_1) Set [(x1_72:
>>                obj) => ((Usc(.A77_1) <<=
>>                x1_72):prop)])
>>             Intersection .M_1)))]))
>>          )))))):that ((.A77_1 = .B77_1) V
>>          (<<<~(Misset_1,thelawchooses_1,.A77_1,
>>          .B77_1) V <<<~(Misset_1,thelawchooses_1,
>>          .B77_1,.A77_1))))]
>>       ,[(casea2_73:that ((((Misset_1 Mbold2
>>          thelawchooses_1) Set [(x1_74:obj)
>>             => ((Usc(.A77_1) <<= x1_74):prop)])
>>          Intersection .M_1) <<= (((Misset_1
>>          Mbold2 thelawchooses_1) Set [(x1_75:
>>             obj) => ((Usc(.B77_1) <<= x1_75):
>>             prop)])
>>          Intersection .M_1))) => (((.A77_1
>>          = .B77_1) Add2 (<<<~(Misset_1,thelawchooses_1,
>>          .A77_1,.B77_1) Add2 (<<<~(Misset_1,
>>          thelawchooses_1,.B77_1,.A77_1) Fixform
```

```
>>             (binm77_1 Conj (ainm77_1 Conj (Negeqsymm(case2_6)
>>             Conj (casea2_73 Iff2 Dediff([(dir1_85:
>>                that (.A77_1 E (((Misset_1 Mbold2
>>                thelawchooses_1) Set [(x1_86:
>>                   obj) => ((Usc(.B77_1) <<=
>>                   x1_86):prop)])
>>                Intersection .M_1))) => (Cases(Mboldstrongtotal2(Misset_1,
>>                thelawchooses_1,(((((Misset_1
>>                Mbold2 thelawchooses_1) Set [(x1_92:
>>                   obj) => ((Usc(.A77_1) <<=
>>                   x1_92):prop)])
>>                Intersection .M_1) E (Misset_1
>>                Mbold2 thelawchooses_1)) Fixform
>>                Lineb4(Misset_1,thelawchooses_1,
>>                (ainm77_1 Iff2 (.A77_1 Uscsubs
>>                .M_1)),(.A77_1 Pairinhabited
>>                .A77_1))),(((((Misset_1 Mbold2
>>                thelawchooses_1) Set [(x1_94:
>>                   obj) => ((Usc(.B77_1) <<=
>>                   x1_94):prop)])
>>                Intersection .M_1) E (Misset_1
>>                Mbold2 thelawchooses_1)) Fixform
>>                Lineb4(Misset_1,thelawchooses_1,
>>                (binm77_1 Iff2 (.B77_1 Uscsubs
>>                .M_1)),(.B77_1 Pairinhabited
>>                .B77_1)))),[(case2_97:that ((((Misset_1
>>                   Mbold2 thelawchooses_1) Set
>>                   [(x1_98:obj) => ((Usc(.B77_1)
>>                      <<= x1_98):prop)])
>>                   Intersection .M_1) <<= prime2(.thelaw_1,
>>                   (((Misset_1 Mbold2 thelawchooses_1)
>>                   Set [(x1_99:obj) => ((Usc(.A77_1)
>>                      <<= x1_99):prop)])
>>                   Intersection .M_1)))) => ((((((Misset_1
>>                   Mbold2 thelawchooses_1) Set
>>                   [(x1_100:obj) => ((Usc(.A77_1)
>>                      <<= x1_100):prop)])
>>                   Intersection .M_1) <<= (((Misset_1
```

```
>>                    Mbold2 thelawchooses_1) Set
>>                    [(x1_101:obj) => ((Usc(.B77_1)
>>                       <<= x1_101):prop)])
>>                    Intersection .M_1)) Giveup
>>                    (Subs(Eqsymm(((.thelaw_1((((Misset_1
>>                    Mbold2 thelawchooses_1) Set
>>                    [(x1_106:obj) => ((Usc(.A77_1)
>>                       <<= x1_106):prop)])
>>                    Intersection .M_1)) = .A77_1)
>>                    Fixform Inusc1(Lineb27(Misset_1,
>>                    thelawchooses_1,(ainm77_1
>>                    Iff2 (.A77_1 Uscsubs .M_1)),
>>                    (.A77_1 Pairinhabited .A77_1))))),
>>                    [(z1_108:obj) => ((z1_108
>>                       E prime2(.thelaw_1,(((Misset_1
>>                       Mbold2 thelawchooses_1)
>>                       Set [(x1_109:obj) => ((Usc(.A77_1)
>>                          <<= x1_109):prop)])
>>                       Intersection .M_1))):prop)]
>>                    ,(dir1_85 Mpsubs case2_97))
>>                    Mp primefact3(Misset_1,thelawchooses_1,
>>                    (((Misset_1 Mbold2 thelawchooses_1)
>>                    Set [(x1_112:obj) => ((Usc(.A77_1)
>>                       <<= x1_112):prop)])
>>                    Intersection .M_1)))):that
>>                    ((((Misset_1 Mbold2 thelawchooses_1)
>>                    Set [(x1_113:obj) => ((Usc(.A77_1)
>>                       <<= x1_113):prop)])
>>                    Intersection .M_1) <<= (((Misset_1
>>                    Mbold2 thelawchooses_1) Set
>>                    [(x1_114:obj) => ((Usc(.B77_1)
>>                       <<= x1_114):prop)])
>>                    Intersection .M_1)))]
>>                 ,[(case1_115:that ((((Misset_1
>>                    Mbold2 thelawchooses_1) Set
>>                    [(x1_116:obj) => ((Usc(.A77_1)
>>                       <<= x1_116):prop)])
>>                    Intersection .M_1) <<= (((Misset_1
```

```
>>                Mbold2 thelawchooses_1) Set
>>                [(x1_117:obj) => ((Usc(.B77_1)
>>                   <<= x1_117):prop)])
>>                Intersection .M_1))) => (case1_115:
>>                that ((((Misset_1 Mbold2 thelawchooses_1)
>>                Set [(x1_118:obj) => ((Usc(.A77_1)
>>                   <<= x1_118):prop)])
>>                Intersection .M_1) <<= (((Misset_1
>>                Mbold2 thelawchooses_1) Set
>>                [(x1_119:obj) => ((Usc(.B77_1)
>>                   <<= x1_119):prop)])
>>                Intersection .M_1)))])
>>             :that ((((Misset_1 Mbold2 thelawchooses_1)
>>             Set [(x1_120:obj) => ((Usc(.A77_1)
>>                <<= x1_120):prop)])
>>             Intersection .M_1) <<= (((Misset_1
>>             Mbold2 thelawchooses_1) Set [(x1_121:
>>                obj) => ((Usc(.B77_1) <<=
>>                x1_121):prop)])
>>             Intersection .M_1)))]
>>          ,[(dir2_122:that ((((Misset_1 Mbold2
>>             thelawchooses_1) Set [(x1_123:
>>                obj) => ((Usc(.A77_1) <<=
>>                x1_123):prop)])
>>             Intersection .M_1) <<= (((Misset_1
>>             Mbold2 thelawchooses_1) Set [(x1_124:
>>                obj) => ((Usc(.B77_1) <<=
>>                x1_124):prop)])
>>             Intersection .M_1))) => (((Lineab13(Misset_1,
>>             thelawchooses_1,(ainm77_1 Iff2
>>             (.A77_1 Uscsubs .M_1)),(.A77_1
>>             Pairinhabited .A77_1)) Iff1 (.A77_1
>>             Uscsubs (((Misset_1 Mbold2 thelawchooses_1)
>>             Set [(x1_128:obj) => ((Usc(.A77_1)
>>                <<= x1_128):prop)])
>>             Intersection .M_1))) Mpsubs dir2_122):
>>             that (.A77_1 E (((Misset_1 Mbold2
>>             thelawchooses_1) Set [(x1_130:
```

```
>>                  obj) => ((Usc(.B77_1) <<=
>>                  x1_130):prop)])
>>               Intersection .M_1))]]))
>>            )))))):that ((.A77_1 = .B77_1) V
>>            (<<<~(Misset_1,thelawchooses_1,.A77_1,
>>            .B77_1) V <<<~(Misset_1,thelawchooses_1,
>>            .B77_1,.A77_1))))])
>>         :that ((.A77_1 = .B77_1) V (<<<~(Misset_1,
>>         thelawchooses_1,.A77_1,.B77_1) V <<<~(Misset_1,
>>         thelawchooses_1,.B77_1,.A77_1))))])
>>      :that ((.A77_1 = .B77_1) V (<<<~(Misset_1,
>>      thelawchooses_1,.A77_1,.B77_1) V <<<~(Misset_1,
>>      thelawchooses_1,.B77_1,.A77_1))))]
>>   {move 0}
```

The purported order is trichotomous (so total).

```
Lestrade execution:


open

   define line67 ainm binm: lineb67 Misset, \
      thelawchooses, ainm binm

>>    line67: [(.a_1:obj),(ainm_1:that (.a_1
>>         E M)),(.b_1:obj),(binm_1:that (.b_1
>>         E M)) => (---:that ((.a_1 = .b_1) V
>>         (<<<~(Misset,thelawchooses,.a_1,.b_1)
>>         V <<<~(Misset,thelawchooses,.b_1,.a_1))))]
>>      {move 1}



   goal that ~(a <~ a)
```

```
>>    Goal: that ~((a <~ a))

   open

      declare sillyhyp that a <~ a

>>       sillyhyp: that (a <~ a) {move 3}



      define line68 sillyhyp: Mp Refleq a, \
         Simp1 Simp2 Simp2 sillyhyp

>>       line68: [(sillyhyp_1:that (a <~ a))
>>            => (---:that ??)]
>>         {move 2}



      close

   define line69 ainm: Negintro line68

>>    line69: [(.a_1:obj),(ainm_1:that (.a_1
>>         E M)) => (---:that ~((.a_1 <~ .a_1)))]
>>      {move 1}
```

The purported order is irreflexive.

```
Lestrade execution:


   goal that (a <~ b) -> ~(b <~ a)

>>    Goal: that ((a <~ b) -> ~((b <~ a)))
```

```
   open

      declare thehyp that a <~ b

>>       thehyp: that (a <~ b) {move 3}



      define line70 thehyp: Iff1 Simp2 Simp2 \
         Simp2 thehyp, line58 ainm binm

>>       line70: [(thehyp_1:that (a <~ b)) =>
>>            (---:that (Rcal(b) <<= Rcal(a)))]
>>         {move 2}



      open

         declare sillyhyp that b <~ a

>>          sillyhyp: that (b <~ a) {move 4}



         define line71 sillyhyp: Iff1 Simp2 \
            Simp2 Simp2 sillyhyp, line58 binm \
            ainm

>>          line71: [(sillyhyp_1:that (b <~
>>               a)) => (---:that (Rcal(a) <<=
>>               Rcal(b)))]
>>            {move 3}



         define line72 sillyhyp: Antisymsub \
            line70 thehyp, line71 sillyhyp
```

```
>>          line72: [(sillyhyp_1:that (b <~
>>               a)) => (---:that (Rcal(b) = Rcal(a)))]
>>            {move 3}
```

```
        define line73 sillyhyp: Subs1 Line44 \
           ainm, Subs1 Line44 binm,bothsides \
           thelaw, line72 sillyhyp
```

```
>>          line73: [(sillyhyp_1:that (b <~
>>               a)) => (---:that (b = a))]
>>            {move 3}
```

```
        define line74 sillyhyp: Mp line73 \
           sillyhyp, Simp1 Simp2 Simp2 sillyhyp
```

```
>>          line74: [(sillyhyp_1:that (b <~
>>               a)) => (---:that ??)]
>>            {move 3}
```

```
        close

     define line75 thehyp: Negintro line74
```

```
>>       line75: [(thehyp_1:that (a <~ b)) =>
>>            (---:that ~((b <~ a)))]
>>         {move 2}
```

```
      close

   define linea76 ainm binm: Ded line75

>>    linea76: [(.a_1:obj),(ainm_1:that (.a_1
>>         E M)),(.b_1:obj),(binm_1:that (.b_1
>>         E M)) => (---:that ((.a_1 <~ .b_1)
>>         -> ~((.b_1 <~ .a_1))))]
>>      {move 1}



   save

   close

define lineb76 Misset, thelawchooses, ainm77, \
   binm77: linea76 ainm77 binm77

>> lineb76: [(.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))]),
>>      (.A77_1:obj),(ainm77_1:that (.A77_1 E
>>      .M_1)),(.B77_1:obj),(binm77_1:that (.B77_1
>>      E .M_1)) => (Ded([(thehyp_5:that <<<~(Misset_1,
>>         thelawchooses_1,.A77_1,.B77_1)) =>
>>         (Negintro([(sillyhyp_6:that <<<~(Misset_1,
>>            thelawchooses_1,.B77_1,.A77_1))
>>            => (((((.thelaw_1((((Misset_1 Mbold2
>>            thelawchooses_1) Set [(x1_8:obj)
>>               => ((Usc(.A77_1) <<= x1_8):prop)])
>>            Intersection .M_1)) = .A77_1) Fixform
>>            Inusc1(Lineb27(Misset_1,thelawchooses_1,
```

```
>>              (ainm77_1 Iff2 (.A77_1 Uscsubs .M_1)),
>>              (.A77_1 Pairinhabited .A77_1))))
>>              Subs1 (((.thelaw_1((((Misset_1 Mbold2
>>              thelawchooses_1) Set [(x1_12:obj)
>>                 => ((Usc(.B77_1) <<= x1_12):prop)])
>>              Intersection .M_1)) = .B77_1) Fixform
>>              Inusc1(Lineb27(Misset_1,thelawchooses_1,
>>              (binm77_1 Iff2 (.B77_1 Uscsubs .M_1)),
>>              (.B77_1 Pairinhabited .B77_1))))
>>              Subs1 bothsides(.thelaw_1,((Simp2(Simp2(Simp2(thehyp_5)))
>>              Iff1 Dediff([(dir1_29:that (.B77_1
>>                 E (((Misset_1 Mbold2 thelawchooses_1)
>>                 Set [(x1_30:obj) => ((Usc(.A77_1)
>>                    <<= x1_30):prop)])
>>                 Intersection .M_1))) => (Cases(Mboldstrongtotal2(Misset_1,
>>                 thelawchooses_1,(((((Misset_1
>>                 Mbold2 thelawchooses_1) Set [(x1_36:
>>                    obj) => ((Usc(.B77_1) <<=
>>                    x1_36):prop)])
>>                 Intersection .M_1) E (Misset_1
>>                 Mbold2 thelawchooses_1)) Fixform
>>                 Lineb4(Misset_1,thelawchooses_1,
>>                 (binm77_1 Iff2 (.B77_1 Uscsubs
>>                 .M_1)),(.B77_1 Pairinhabited
>>                 .B77_1))),(((((Misset_1 Mbold2
>>                 thelawchooses_1) Set [(x1_38:
>>                    obj) => ((Usc(.A77_1) <<=
>>                    x1_38):prop)])
>>                 Intersection .M_1) E (Misset_1
>>                 Mbold2 thelawchooses_1)) Fixform
>>                 Lineb4(Misset_1,thelawchooses_1,
>>                 (ainm77_1 Iff2 (.A77_1 Uscsubs
>>                 .M_1)),(.A77_1 Pairinhabited
>>                 .A77_1)))),[(case2_41:that ((((Misset_1
>>                    Mbold2 thelawchooses_1) Set
>>                    [(x1_42:obj) => ((Usc(.A77_1)
>>                       <<= x1_42):prop)])
>>                    Intersection .M_1) <<= prime2(.thelaw_1,
```

```
>>                  (((Misset_1 Mbold2 thelawchooses_1)
>>                  Set [(x1_43:obj) => ((Usc(.B77_1)
>>                     <<= x1_43):prop)])
>>                  Intersection .M_1)))) => ((((((Misset_1
>>                  Mbold2 thelawchooses_1) Set
>>                  [(x1_44:obj) => ((Usc(.B77_1)
>>                     <<= x1_44):prop)])
>>                  Intersection .M_1) <<= (((Misset_1
>>                  Mbold2 thelawchooses_1) Set
>>                  [(x1_45:obj) => ((Usc(.A77_1)
>>                     <<= x1_45):prop)])
>>                  Intersection .M_1)) Giveup
>>                  (Subs(Eqsymm(((.thelaw_1((((Misset_1
>>                  Mbold2 thelawchooses_1) Set
>>                  [(x1_50:obj) => ((Usc(.B77_1)
>>                     <<= x1_50):prop)])
>>                  Intersection .M_1)) = .B77_1)
>>                  Fixform Inusc1(Lineb27(Misset_1,
>>                  thelawchooses_1,(binm77_1
>>                  Iff2 (.B77_1 Uscsubs .M_1)),
>>                  (.B77_1 Pairinhabited .B77_1))))),
>>                  [(z1_52:obj) => ((z1_52 E
>>                     prime2(.thelaw_1,(((Misset_1
>>                     Mbold2 thelawchooses_1)
>>                     Set [(x1_53:obj) => ((Usc(.B77_1)
>>                        <<= x1_53):prop)])
>>                     Intersection .M_1))):prop)]
>>                  ,(dir1_29 Mpsubs case2_41))
>>                  Mp primefact3(Misset_1,thelawchooses_1,
>>                  (((Misset_1 Mbold2 thelawchooses_1)
>>                  Set [(x1_56:obj) => ((Usc(.B77_1)
>>                     <<= x1_56):prop)])
>>                  Intersection .M_1)))):that
>>                  ((((Misset_1 Mbold2 thelawchooses_1)
>>                  Set [(x1_57:obj) => ((Usc(.B77_1)
>>                     <<= x1_57):prop)])
>>                  Intersection .M_1) <<= (((Misset_1
>>                  Mbold2 thelawchooses_1) Set
```

```
>>                    [(x1_58:obj) => ((Usc(.A77_1)
>>                       <<= x1_58):prop)])
>>                    Intersection .M_1)))]
>>                 ,[(case1_59:that ((((Misset_1
>>                    Mbold2 thelawchooses_1) Set
>>                    [(x1_60:obj) => ((Usc(.B77_1)
>>                       <<= x1_60):prop)])
>>                    Intersection .M_1) <<= (((Misset_1
>>                    Mbold2 thelawchooses_1) Set
>>                    [(x1_61:obj) => ((Usc(.A77_1)
>>                       <<= x1_61):prop)])
>>                    Intersection .M_1))) => (case1_59:
>>                    that ((((Misset_1 Mbold2 thelawchooses_1)
>>                    Set [(x1_62:obj) => ((Usc(.B77_1)
>>                       <<= x1_62):prop)])
>>                    Intersection .M_1) <<= (((Misset_1
>>                    Mbold2 thelawchooses_1) Set
>>                    [(x1_63:obj) => ((Usc(.A77_1)
>>                       <<= x1_63):prop)])
>>                    Intersection .M_1)))])
>>                 :that ((((Misset_1 Mbold2 thelawchooses_1)
>>                 Set [(x1_64:obj) => ((Usc(.B77_1)
>>                    <<= x1_64):prop)])
>>                 Intersection .M_1) <<= (((Misset_1
>>                 Mbold2 thelawchooses_1) Set [(x1_65:
>>                    obj) => ((Usc(.A77_1) <<=
>>                    x1_65):prop)])
>>                 Intersection .M_1)))]
>>              ,[(dir2_66:that ((((Misset_1 Mbold2
>>                 thelawchooses_1) Set [(x1_67:
>>                    obj) => ((Usc(.B77_1) <<=
>>                    x1_67):prop)])
>>                 Intersection .M_1) <<= (((Misset_1
>>                 Mbold2 thelawchooses_1) Set [(x1_68:
>>                    obj) => ((Usc(.A77_1) <<=
>>                    x1_68):prop)])
>>                 Intersection .M_1))) => (((Lineab13(Misset_1,
>>                 thelawchooses_1,(binm77_1 Iff2
```

```
>>               (.B77_1 Uscsubs .M_1)),(.B77_1
>>               Pairinhabited .B77_1)) Iff1 (.B77_1
>>               Uscsubs (((Misset_1 Mbold2 thelawchooses_1)
>>               Set [(x1_72:obj) => ((Usc(.B77_1)
>>                  <<= x1_72):prop)])
>>               Intersection .M_1))) Mpsubs dir2_66):
>>               that (.B77_1 E (((Misset_1 Mbold2
>>               thelawchooses_1) Set [(x1_74:
>>                  obj) => ((Usc(.A77_1) <<=
>>                  x1_74):prop)])
>>               Intersection .M_1)))]))
>>            Antisymsub (Simp2(Simp2(Simp2(sillyhyp_6)))
>>            Iff1 Dediff([(dir1_84:that (.A77_1
>>               E (((Misset_1 Mbold2 thelawchooses_1)
>>               Set [(x1_85:obj) => ((Usc(.B77_1)
>>                  <<= x1_85):prop)])
>>               Intersection .M_1))) => (Cases(Mboldstrongtotal2(Misset_1,
>>               thelawchooses_1,(((((Misset_1
>>               Mbold2 thelawchooses_1) Set [(x1_91:
>>                  obj) => ((Usc(.A77_1) <<=
>>                  x1_91):prop)])
>>               Intersection .M_1) E (Misset_1
>>               Mbold2 thelawchooses_1)) Fixform
>>               Lineb4(Misset_1,thelawchooses_1,
>>               (ainm77_1 Iff2 (.A77_1 Uscsubs
>>               .M_1)),(.A77_1 Pairinhabited
>>               .A77_1))),(((((Misset_1 Mbold2
>>               thelawchooses_1) Set [(x1_93:
>>                  obj) => ((Usc(.B77_1) <<=
>>                  x1_93):prop)])
>>               Intersection .M_1) E (Misset_1
>>               Mbold2 thelawchooses_1)) Fixform
>>               Lineb4(Misset_1,thelawchooses_1,
>>               (binm77_1 Iff2 (.B77_1 Uscsubs
>>               .M_1)),(.B77_1 Pairinhabited
>>               .B77_1)))),[(case2_96:that ((((Misset_1
>>                  Mbold2 thelawchooses_1) Set
>>                  [(x1_97:obj) => ((Usc(.B77_1)
```

```
>>                      <<= x1_97):prop)])
>>                   Intersection .M_1) <<= prime2(.thelaw_1,
>>                   (((Misset_1 Mbold2 thelawchooses_1)
>>                   Set [(x1_98:obj) => ((Usc(.A77_1)
>>                      <<= x1_98):prop)])
>>                   Intersection .M_1)))) => ((((((Misset_1
>>                   Mbold2 thelawchooses_1) Set
>>                   [(x1_99:obj) => ((Usc(.A77_1)
>>                      <<= x1_99):prop)])
>>                   Intersection .M_1) <<= (((Misset_1
>>                   Mbold2 thelawchooses_1) Set
>>                   [(x1_100:obj) => ((Usc(.B77_1)
>>                      <<= x1_100):prop)])
>>                   Intersection .M_1)) Giveup
>>                   (Subs(Eqsymm(((.thelaw_1((((Misset_1
>>                   Mbold2 thelawchooses_1) Set
>>                   [(x1_105:obj) => ((Usc(.A77_1)
>>                      <<= x1_105):prop)])
>>                   Intersection .M_1)) = .A77_1)
>>                   Fixform Inusc1(Lineb27(Misset_1,
>>                   thelawchooses_1,(ainm77_1
>>                   Iff2 (.A77_1 Uscsubs .M_1)),
>>                   (.A77_1 Pairinhabited .A77_1))))),
>>                   [(z1_107:obj) => ((z1_107
>>                      E prime2(.thelaw_1,(((Misset_1
>>                      Mbold2 thelawchooses_1)
>>                      Set [(x1_108:obj) => ((Usc(.A77_1)
>>                         <<= x1_108):prop)])
>>                      Intersection .M_1))):prop)]
>>                   ,(dir1_84 Mpsubs case2_96))
>>                   Mp primefact3(Misset_1,thelawchooses_1,
>>                   (((Misset_1 Mbold2 thelawchooses_1)
>>                   Set [(x1_111:obj) => ((Usc(.A77_1)
>>                      <<= x1_111):prop)])
>>                   Intersection .M_1)))):that
>>                   ((((Misset_1 Mbold2 thelawchooses_1)
>>                   Set [(x1_112:obj) => ((Usc(.A77_1)
>>                      <<= x1_112):prop)])
```

```
>>                    Intersection .M_1) <<= (((Misset_1
>>                    Mbold2 thelawchooses_1) Set
>>                    [(x1_113:obj) => ((Usc(.B77_1)
>>                       <<= x1_113):prop)])
>>                    Intersection .M_1)))]
>>                 ,[(case1_114:that ((((Misset_1
>>                    Mbold2 thelawchooses_1) Set
>>                    [(x1_115:obj) => ((Usc(.A77_1)
>>                       <<= x1_115):prop)])
>>                    Intersection .M_1) <<= (((Misset_1
>>                    Mbold2 thelawchooses_1) Set
>>                    [(x1_116:obj) => ((Usc(.B77_1)
>>                       <<= x1_116):prop)])
>>                    Intersection .M_1))) => (case1_114:
>>                    that ((((Misset_1 Mbold2 thelawchooses_1)
>>                    Set [(x1_117:obj) => ((Usc(.A77_1)
>>                       <<= x1_117):prop)])
>>                    Intersection .M_1) <<= (((Misset_1
>>                    Mbold2 thelawchooses_1) Set
>>                    [(x1_118:obj) => ((Usc(.B77_1)
>>                       <<= x1_118):prop)])
>>                    Intersection .M_1)))])
>>                 :that ((((Misset_1 Mbold2 thelawchooses_1)
>>                 Set [(x1_119:obj) => ((Usc(.A77_1)
>>                    <<= x1_119):prop)])
>>                 Intersection .M_1) <<= (((Misset_1
>>                 Mbold2 thelawchooses_1) Set [(x1_120:
>>                    obj) => ((Usc(.B77_1) <<=
>>                    x1_120):prop)])
>>                 Intersection .M_1)))]
>>              ,[(dir2_121:that ((((Misset_1 Mbold2
>>                 thelawchooses_1) Set [(x1_122:
>>                    obj) => ((Usc(.A77_1) <<=
>>                    x1_122):prop)])
>>                 Intersection .M_1) <<= (((Misset_1
>>                 Mbold2 thelawchooses_1) Set [(x1_123:
>>                    obj) => ((Usc(.B77_1) <<=
>>                    x1_123):prop)])
```

```
>>              Intersection .M_1))) => (((Lineab13(Misset_1,
>>              thelawchooses_1,(ainm77_1 Iff2
>>              (.A77_1 Uscsubs .M_1)),(.A77_1
>>              Pairinhabited .A77_1)) Iff1 (.A77_1
>>              Uscsubs (((Misset_1 Mbold2 thelawchooses_1)
>>              Set [(x1_127:obj) => ((Usc(.A77_1)
>>                 <<= x1_127):prop)])
>>              Intersection .M_1))) Mpsubs dir2_121):
>>              that (.A77_1 E (((Misset_1 Mbold2
>>              thelawchooses_1) Set [(x1_129:
>>                 obj) => ((Usc(.B77_1) <<=
>>                 x1_129):prop)])
>>              Intersection .M_1)))]))
>>           )))) Mp Simp1(Simp2(Simp2(sillyhyp_6)))):
>>           that ??)])
>>        :that ~(<<<~(Misset_1,thelawchooses_1,
>>        .B77_1,.A77_1)))])
>>     :that (<<<~(Misset_1,thelawchooses_1,.A77_1,
>>     .B77_1) -> ~(<<<~(Misset_1,thelawchooses_1,
>>     .B77_1,.A77_1))))]
>>  {move 0}
```

```
open

   define line76 ainm binm: lineb76 Misset, \
      thelawchooses, ainm binm

>>    line76: [(.a_1:obj),(ainm_1:that (.a_1
>>         E M)),(.b_1:obj),(binm_1:that (.b_1
>>         E M)) => (---:that (<<<~(Misset,thelawchooses,
>>         .a_1,.b_1) -> ~(<<<~(Misset,thelawchooses,
>>         .b_1,.a_1))))]
>>     {move 1}
```

The purported order is asymmetric.

```
Lestrade execution:


   declare c obj

>>    c: obj {move 2}



   declare cinm that c E M

>>    cinm: that (c E M) {move 2}



   goal that ((a <~b) & (b <~ c))-> \
      a <~ c

>>    Goal: that (((a <~ b) & (b <~ c)) -> (a
>>      <~ c))

   open

      declare thehyp that (a <~ b) & b <~ \
         c

>>       thehyp: that ((a <~ b) & (b <~ c))
>>         {move 3}



      define line77 thehyp: Iff1(Simp2 Simp2 \
         Simp2 Simp1 thehyp, line58 \
         ainm binm)

>>       line77: [(thehyp_1:that ((a <~ b) &
```

```
>>            (b <~ c))) => (---:that (Rcal(b)
>>            <<= Rcal(a)))]
>>         {move 2}
```

```
      define line78 thehyp: Iff1 (Simp2 Simp2 \
         Simp2 Simp2 thehyp,line58 binm cinm)
```

```
>>       line78: [(thehyp_1:that ((a <~ b) &
>>            (b <~ c))) => (---:that (Rcal(c)
>>            <<= Rcal(b)))]
>>         {move 2}
```

```
      define line79 thehyp: Iff2(Transsub \
         line78 thehyp, line77 thehyp, line58 \
         ainm cinm)
```

```
>>       line79: [(thehyp_1:that ((a <~ b) &
>>            (b <~ c))) => (---:that (c E Rcal(a)))]
>>         {move 2}
```

```
      open

         declare sillyhyp that a=c
```

```
>>          sillyhyp: that (a = c) {move 4}
```

```
         define line80 sillyhyp: Subs1 Eqsymm \
            sillyhyp Simp2 thehyp
```

```
>>          line80: [(sillyhyp_1:that (a = c))
>>               => (---:that (b <~ a))]
>>            {move 3}



         define line81 sillyhyp: Mp line80 \
            sillyhyp, Mp Simp1 thehyp, line76 \
            ainm binm

>>          line81: [(sillyhyp_1:that (a = c))
>>               => (---:that ??)]
>>            {move 3}




         close

      define line82 thehyp: Negintro line81


>>       line82: [(thehyp_1:that ((a <~ b) &
>>            (b <~ c))) => (---:that ~((a = c)))]
>>         {move 2}



      define line83 thehyp: Fixform(a <~ \
         c,ainm Conj cinm Conj line82 thehyp \
         Conj line79 thehyp)

>>       line83: [(thehyp_1:that ((a <~ b) &
>>            (b <~ c))) => (---:that (a <~ c))]
>>         {move 2}



      close
```

```
   define linea84 ainm binm cinm: Ded line83


>>    linea84: [(.a_1:obj),(ainm_1:that (.a_1
>>         E M)),(.b_1:obj),(binm_1:that (.b_1
>>         E M)),(.c_1:obj),(cinm_1:that (.c_1
>>         E M)) => (---:that (((.a_1 <~ .b_1)
>>         & (.b_1 <~ .c_1)) -> (.a_1 <~ .c_1)))]
>>      {move 1}



   save

   close

declare C77 obj

>> C77: obj {move 1}



declare cinm77 that C77 E M

>> cinm77: that (C77 E M) {move 1}



define lineb84 Misset, thelawchooses, ainm77 \
   binm77 cinm77: linea84 ainm77 binm77 cinm77


>> lineb84: [(.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))]),
>>     (.A77_1:obj),(ainm77_1:that (.A77_1 E
>>     .M_1)),(.B77_1:obj),(binm77_1:that (.B77_1
>>     E .M_1)),(.C77_1:obj),(cinm77_1:that (.C77_1
>>     E .M_1)) => (Ded([(thehyp_5:that (<<<~(Misset_1,
>>        thelawchooses_1,.A77_1,.B77_1) & <<<~(Misset_1,
>>        thelawchooses_1,.B77_1,.C77_1))) =>
>>        ((<<<~(Misset_1,thelawchooses_1,.A77_1,
>>        .C77_1) Fixform (ainm77_1 Conj (cinm77_1
>>        Conj (Negintro([(sillyhyp_8:that (.A77_1
>>           = .C77_1)) => (((Eqsymm(sillyhyp_8)
>>           Subs1 Simp2(thehyp_5)) Mp (Simp1(thehyp_5)
>>           Mp lineb76(Misset_1,thelawchooses_1,
>>           ainm77_1,binm77_1))):that ??)])
>>        Conj (((Simp2(Simp2(Simp2(Simp2(thehyp_5))))
>>        Iff1 Dediff([(dir1_24:that (.C77_1
>>           E (((Misset_1 Mbold2 thelawchooses_1)
>>           Set [(x1_25:obj) => ((Usc(.B77_1)
>>              <<= x1_25):prop)])
>>           Intersection .M_1))) => (Cases(Mboldstrongtotal2(Misset_1,
>>           thelawchooses_1,(((((Misset_1 Mbold2
>>           thelawchooses_1) Set [(x1_31:obj)
>>              => ((Usc(.C77_1) <<= x1_31):prop)])
>>           Intersection .M_1) E (Misset_1 Mbold2
>>           thelawchooses_1)) Fixform Lineb4(Misset_1,
>>           thelawchooses_1,(cinm77_1 Iff2 (.C77_1
>>           Uscsubs .M_1)),(.C77_1 Pairinhabited
>>           .C77_1))),(((((Misset_1 Mbold2 thelawchooses_1)
>>           Set [(x1_33:obj) => ((Usc(.B77_1)
>>              <<= x1_33):prop)])
>>           Intersection .M_1) E (Misset_1 Mbold2
>>           thelawchooses_1)) Fixform Lineb4(Misset_1,
>>           thelawchooses_1,(binm77_1 Iff2 (.B77_1
>>           Uscsubs .M_1)),(.B77_1 Pairinhabited
>>           .B77_1)))),[(case2_36:that ((((Misset_1
>>              Mbold2 thelawchooses_1) Set [(x1_37:
>>                 obj) => ((Usc(.B77_1) <<=
```

```
>>                  x1_37):prop)])
>>               Intersection .M_1) <<= prime2(.thelaw_1,
>>               (((Misset_1 Mbold2 thelawchooses_1)
>>               Set [(x1_38:obj) => ((Usc(.C77_1)
>>                  <<= x1_38):prop)])
>>               Intersection .M_1)))) => ((((((Misset_1
>>               Mbold2 thelawchooses_1) Set [(x1_39:
>>                  obj) => ((Usc(.C77_1) <<=
>>                  x1_39):prop)])
>>               Intersection .M_1) <<= (((Misset_1
>>               Mbold2 thelawchooses_1) Set [(x1_40:
>>                  obj) => ((Usc(.B77_1) <<=
>>                  x1_40):prop)])
>>               Intersection .M_1)) Giveup (Subs(Eqsymm(((.thelaw_1((((Misset_1
>>               Mbold2 thelawchooses_1) Set [(x1_45:
>>                  obj) => ((Usc(.C77_1) <<=
>>                  x1_45):prop)])
>>               Intersection .M_1)) = .C77_1)
>>               Fixform Inusc1(Lineb27(Misset_1,
>>               thelawchooses_1,(cinm77_1 Iff2
>>               (.C77_1 Uscsubs .M_1)),(.C77_1
>>               Pairinhabited .C77_1))))),[(z1_47:
>>                  obj) => ((z1_47 E prime2(.thelaw_1,
>>                  (((Misset_1 Mbold2 thelawchooses_1)
>>                  Set [(x1_48:obj) => ((Usc(.C77_1)
>>                     <<= x1_48):prop)])
>>                  Intersection .M_1))):prop)]
>>               ,(dir1_24 Mpsubs case2_36)) Mp
>>               primefact3(Misset_1,thelawchooses_1,
>>               (((Misset_1 Mbold2 thelawchooses_1)
>>               Set [(x1_51:obj) => ((Usc(.C77_1)
>>                  <<= x1_51):prop)])
>>               Intersection .M_1)))):that ((((Misset_1
>>               Mbold2 thelawchooses_1) Set [(x1_52:
>>                  obj) => ((Usc(.C77_1) <<=
>>                  x1_52):prop)])
>>               Intersection .M_1) <<= (((Misset_1
>>               Mbold2 thelawchooses_1) Set [(x1_53:
```

```
>>                  obj) => ((Usc(.B77_1) <<=
>>                  x1_53):prop)])
>>               Intersection .M_1)))]
>>            ,[(case1_54:that ((((Misset_1 Mbold2
>>               thelawchooses_1) Set [(x1_55:
>>                  obj) => ((Usc(.C77_1) <<=
>>                  x1_55):prop)])
>>               Intersection .M_1) <<= (((Misset_1
>>               Mbold2 thelawchooses_1) Set [(x1_56:
>>                  obj) => ((Usc(.B77_1) <<=
>>                  x1_56):prop)])
>>               Intersection .M_1))) => (case1_54:
>>               that ((((Misset_1 Mbold2 thelawchooses_1)
>>               Set [(x1_57:obj) => ((Usc(.C77_1)
>>                  <<= x1_57):prop)])
>>               Intersection .M_1) <<= (((Misset_1
>>               Mbold2 thelawchooses_1) Set [(x1_58:
>>                  obj) => ((Usc(.B77_1) <<=
>>                  x1_58):prop)])
>>               Intersection .M_1)))])
>>            :that ((((Misset_1 Mbold2 thelawchooses_1)
>>            Set [(x1_59:obj) => ((Usc(.C77_1)
>>               <<= x1_59):prop)])
>>            Intersection .M_1) <<= (((Misset_1
>>            Mbold2 thelawchooses_1) Set [(x1_60:
>>               obj) => ((Usc(.B77_1) <<= x1_60):
>>               prop)])
>>            Intersection .M_1)))]
>>         ,[(dir2_61:that ((((Misset_1 Mbold2
>>            thelawchooses_1) Set [(x1_62:obj)
>>               => ((Usc(.C77_1) <<= x1_62):prop)])
>>            Intersection .M_1) <<= (((Misset_1
>>            Mbold2 thelawchooses_1) Set [(x1_63:
>>               obj) => ((Usc(.B77_1) <<= x1_63):
>>               prop)])
>>            Intersection .M_1))) => (((Lineab13(Misset_1,
>>            thelawchooses_1,(cinm77_1 Iff2 (.C77_1
>>            Uscsubs .M_1)),(.C77_1 Pairinhabited
```

```
>>              .C77_1)) Iff1 (.C77_1 Uscsubs (((Misset_1
>>              Mbold2 thelawchooses_1) Set [(x1_67:
>>                 obj) => ((Usc(.C77_1) <<= x1_67):
>>                 prop)])
>>              Intersection .M_1))) Mpsubs dir2_61):
>>              that (.C77_1 E (((Misset_1 Mbold2
>>              thelawchooses_1) Set [(x1_69:obj)
>>                 => ((Usc(.B77_1) <<= x1_69):prop)])
>>              Intersection .M_1)))]))
>>           Transsub (Simp2(Simp2(Simp2(Simp1(thehyp_5))))
>>           Iff1 Dediff([(dir1_80:that (.B77_1
>>              E (((Misset_1 Mbold2 thelawchooses_1)
>>              Set [(x1_81:obj) => ((Usc(.A77_1)
>>                 <<= x1_81):prop)])
>>              Intersection .M_1))) => (Cases(Mboldstrongtotal2(Misset_1,
>>              thelawchooses_1,(((((Misset_1 Mbold2
>>              thelawchooses_1) Set [(x1_87:obj)
>>                 => ((Usc(.B77_1) <<= x1_87):prop)])
>>              Intersection .M_1) E (Misset_1 Mbold2
>>              thelawchooses_1)) Fixform Lineb4(Misset_1,
>>              thelawchooses_1,(binm77_1 Iff2 (.B77_1
>>              Uscsubs .M_1)),(.B77_1 Pairinhabited
>>              .B77_1))),(((((Misset_1 Mbold2 thelawchooses_1)
>>              Set [(x1_89:obj) => ((Usc(.A77_1)
>>                 <<= x1_89):prop)])
>>              Intersection .M_1) E (Misset_1 Mbold2
>>              thelawchooses_1)) Fixform Lineb4(Misset_1,
>>              thelawchooses_1,(ainm77_1 Iff2 (.A77_1
>>              Uscsubs .M_1)),(.A77_1 Pairinhabited
>>              .A77_1)))),[(case2_92:that ((((Misset_1
>>                 Mbold2 thelawchooses_1) Set [(x1_93:
>>                    obj) => ((Usc(.A77_1) <<=
>>                    x1_93):prop)])
>>                 Intersection .M_1) <<= prime2(.thelaw_1,
>>                 (((Misset_1 Mbold2 thelawchooses_1)
>>                 Set [(x1_94:obj) => ((Usc(.B77_1)
>>                    <<= x1_94):prop)])
>>                 Intersection .M_1)))) => ((((((Misset_1
```

```
>>                 Mbold2 thelawchooses_1) Set [(x1_95:
>>                    obj) => ((Usc(.B77_1) <<=
>>                    x1_95):prop)])
>>                 Intersection .M_1) <<= (((Misset_1
>>                 Mbold2 thelawchooses_1) Set [(x1_96:
>>                    obj) => ((Usc(.A77_1) <<=
>>                    x1_96):prop)])
>>                 Intersection .M_1)) Giveup (Subs(Eqsymm(((.thelaw_1((((Misset_1
>>                 Mbold2 thelawchooses_1) Set [(x1_101:
>>                    obj) => ((Usc(.B77_1) <<=
>>                    x1_101):prop)])
>>                 Intersection .M_1)) = .B77_1)
>>                 Fixform Inusc1(Lineb27(Misset_1,
>>                 thelawchooses_1,(binm77_1 Iff2
>>                 (.B77_1 Uscsubs .M_1)),(.B77_1
>>                 Pairinhabited .B77_1))))),[(z1_103:
>>                    obj) => ((z1_103 E prime2(.thelaw_1,
>>                    (((Misset_1 Mbold2 thelawchooses_1)
>>                    Set [(x1_104:obj) => ((Usc(.B77_1)
>>                       <<= x1_104):prop)])
>>                    Intersection .M_1))):prop)]
>>                 ,(dir1_80 Mpsubs case2_92)) Mp
>>                 primefact3(Misset_1,thelawchooses_1,
>>                 (((Misset_1 Mbold2 thelawchooses_1)
>>                 Set [(x1_107:obj) => ((Usc(.B77_1)
>>                    <<= x1_107):prop)])
>>                 Intersection .M_1)))):that ((((Misset_1
>>                 Mbold2 thelawchooses_1) Set [(x1_108:
>>                    obj) => ((Usc(.B77_1) <<=
>>                    x1_108):prop)])
>>                 Intersection .M_1) <<= (((Misset_1
>>                 Mbold2 thelawchooses_1) Set [(x1_109:
>>                    obj) => ((Usc(.A77_1) <<=
>>                    x1_109):prop)])
>>                 Intersection .M_1)))]
>>              ,[(case1_110:that ((((Misset_1 Mbold2
>>                 thelawchooses_1) Set [(x1_111:
>>                    obj) => ((Usc(.B77_1) <<=
```

```
>>                  x1_111):prop)])
>>               Intersection .M_1) <<= (((Misset_1
>>               Mbold2 thelawchooses_1) Set [(x1_112:
>>                  obj) => ((Usc(.A77_1) <<=
>>                  x1_112):prop)])
>>               Intersection .M_1))) => (case1_110:
>>               that ((((Misset_1 Mbold2 thelawchooses_1)
>>               Set [(x1_113:obj) => ((Usc(.B77_1)
>>                  <<= x1_113):prop)])
>>               Intersection .M_1) <<= (((Misset_1
>>               Mbold2 thelawchooses_1) Set [(x1_114:
>>                  obj) => ((Usc(.A77_1) <<=
>>                  x1_114):prop)])
>>               Intersection .M_1)))])
>>            :that ((((Misset_1 Mbold2 thelawchooses_1)
>>            Set [(x1_115:obj) => ((Usc(.B77_1)
>>               <<= x1_115):prop)])
>>            Intersection .M_1) <<= (((Misset_1
>>            Mbold2 thelawchooses_1) Set [(x1_116:
>>               obj) => ((Usc(.A77_1) <<= x1_116):
>>               prop)])
>>            Intersection .M_1)))]
>>         ,[(dir2_117:that ((((Misset_1 Mbold2
>>            thelawchooses_1) Set [(x1_118:obj)
>>               => ((Usc(.B77_1) <<= x1_118):
>>               prop)])
>>            Intersection .M_1) <<= (((Misset_1
>>            Mbold2 thelawchooses_1) Set [(x1_119:
>>               obj) => ((Usc(.A77_1) <<= x1_119):
>>               prop)])
>>            Intersection .M_1))) => (((Lineab13(Misset_1,
>>            thelawchooses_1,(binm77_1 Iff2 (.B77_1
>>            Uscsubs .M_1)),(.B77_1 Pairinhabited
>>            .B77_1)) Iff1 (.B77_1 Uscsubs (((Misset_1
>>            Mbold2 thelawchooses_1) Set [(x1_123:
>>               obj) => ((Usc(.B77_1) <<= x1_123):
>>               prop)])
>>            Intersection .M_1))) Mpsubs dir2_117):
```

```
>>            that (.B77_1 E (((Misset_1 Mbold2
>>            thelawchooses_1) Set [(x1_125:obj)
>>               => ((Usc(.A77_1) <<= x1_125):
>>               prop)])
>>            Intersection .M_1)))]))
>>         ) Iff2 Dediff([(dir1_130:that (.C77_1
>>            E (((Misset_1 Mbold2 thelawchooses_1)
>>            Set [(x1_131:obj) => ((Usc(.A77_1)
>>               <<= x1_131):prop)])
>>            Intersection .M_1))) => (Cases(Mboldstrongtotal2(Misset_1,
>>            thelawchooses_1,(((((Misset_1 Mbold2
>>            thelawchooses_1) Set [(x1_137:obj)
>>               => ((Usc(.C77_1) <<= x1_137):
>>               prop)])
>>            Intersection .M_1) E (Misset_1 Mbold2
>>            thelawchooses_1)) Fixform Lineb4(Misset_1,
>>            thelawchooses_1,(cinm77_1 Iff2 (.C77_1
>>            Uscsubs .M_1)),(.C77_1 Pairinhabited
>>            .C77_1))),(((((Misset_1 Mbold2 thelawchooses_1)
>>            Set [(x1_139:obj) => ((Usc(.A77_1)
>>               <<= x1_139):prop)])
>>            Intersection .M_1) E (Misset_1 Mbold2
>>            thelawchooses_1)) Fixform Lineb4(Misset_1,
>>            thelawchooses_1,(ainm77_1 Iff2 (.A77_1
>>            Uscsubs .M_1)),(.A77_1 Pairinhabited
>>            .A77_1)))),[(case2_142:that ((((Misset_1
>>               Mbold2 thelawchooses_1) Set [(x1_143:
>>                  obj) => ((Usc(.A77_1) <<=
>>                  x1_143):prop)])
>>               Intersection .M_1) <<= prime2(.thelaw_1,
>>               (((Misset_1 Mbold2 thelawchooses_1)
>>               Set [(x1_144:obj) => ((Usc(.C77_1)
>>                  <<= x1_144):prop)])
>>               Intersection .M_1)))) => ((((((Misset_1
>>               Mbold2 thelawchooses_1) Set [(x1_145:
>>                  obj) => ((Usc(.C77_1) <<=
>>                  x1_145):prop)])
>>               Intersection .M_1) <<= (((Misset_1
```

```
>>                   Mbold2 thelawchooses_1) Set [(x1_146:
>>                      obj) => ((Usc(.A77_1) <<=
>>                      x1_146):prop)])
>>                   Intersection .M_1)) Giveup (Subs(Eqsymm(((.thelaw_1((((Misset_1
>>                   Mbold2 thelawchooses_1) Set [(x1_151:
>>                      obj) => ((Usc(.C77_1) <<=
>>                      x1_151):prop)])
>>                   Intersection .M_1)) = .C77_1)
>>                   Fixform Inusc1(Lineb27(Misset_1,
>>                   thelawchooses_1,(cinm77_1 Iff2
>>                   (.C77_1 Uscsubs .M_1)),(.C77_1
>>                   Pairinhabited .C77_1))))),[(z1_153:
>>                      obj) => ((z1_153 E prime2(.thelaw_1,
>>                      (((Misset_1 Mbold2 thelawchooses_1)
>>                      Set [(x1_154:obj) => ((Usc(.C77_1)
>>                         <<= x1_154):prop)])
>>                      Intersection .M_1))):prop)]
>>                   ,(dir1_130 Mpsubs case2_142))
>>                   Mp primefact3(Misset_1,thelawchooses_1,
>>                   (((Misset_1 Mbold2 thelawchooses_1)
>>                   Set [(x1_157:obj) => ((Usc(.C77_1)
>>                      <<= x1_157):prop)])
>>                   Intersection .M_1)))):that ((((Misset_1
>>                   Mbold2 thelawchooses_1) Set [(x1_158:
>>                      obj) => ((Usc(.C77_1) <<=
>>                      x1_158):prop)])
>>                   Intersection .M_1) <<= (((Misset_1
>>                   Mbold2 thelawchooses_1) Set [(x1_159:
>>                      obj) => ((Usc(.A77_1) <<=
>>                      x1_159):prop)])
>>                   Intersection .M_1)))]
>>                ,[(case1_160:that ((((Misset_1 Mbold2
>>                   thelawchooses_1) Set [(x1_161:
>>                      obj) => ((Usc(.C77_1) <<=
>>                      x1_161):prop)])
>>                   Intersection .M_1) <<= (((Misset_1
>>                   Mbold2 thelawchooses_1) Set [(x1_162:
>>                      obj) => ((Usc(.A77_1) <<=
```

```
>>                   x1_162):prop)])
>>                Intersection .M_1))) => (case1_160:
>>                that ((((Misset_1 Mbold2 thelawchooses_1)
>>                Set [(x1_163:obj) => ((Usc(.C77_1)
>>                   <<= x1_163):prop)])
>>                Intersection .M_1) <<= (((Misset_1
>>                Mbold2 thelawchooses_1) Set [(x1_164:
>>                   obj) => ((Usc(.A77_1) <<=
>>                   x1_164):prop)])
>>                Intersection .M_1))))])
>>             :that ((((Misset_1 Mbold2 thelawchooses_1)
>>             Set [(x1_165:obj) => ((Usc(.C77_1)
>>                <<= x1_165):prop)])
>>             Intersection .M_1) <<= (((Misset_1
>>             Mbold2 thelawchooses_1) Set [(x1_166:
>>                obj) => ((Usc(.A77_1) <<= x1_166):
>>                prop)])
>>             Intersection .M_1)))]
>>          ,[(dir2_167:that ((((Misset_1 Mbold2
>>             thelawchooses_1) Set [(x1_168:obj)
>>                => ((Usc(.C77_1) <<= x1_168):
>>                prop)])
>>             Intersection .M_1) <<= (((Misset_1
>>             Mbold2 thelawchooses_1) Set [(x1_169:
>>                obj) => ((Usc(.A77_1) <<= x1_169):
>>                prop)])
>>             Intersection .M_1))) => (((Lineab13(Misset_1,
>>             thelawchooses_1,(cinm77_1 Iff2 (.C77_1
>>             Uscsubs .M_1)),(.C77_1 Pairinhabited
>>             .C77_1)) Iff1 (.C77_1 Uscsubs (((Misset_1
>>             Mbold2 thelawchooses_1) Set [(x1_173:
>>                obj) => ((Usc(.C77_1) <<= x1_173):
>>                prop)])
>>             Intersection .M_1))) Mpsubs dir2_167):
>>             that (.C77_1 E (((Misset_1 Mbold2
>>             thelawchooses_1) Set [(x1_175:obj)
>>                => ((Usc(.A77_1) <<= x1_175):
>>                prop)])
```

```
>>            Intersection .M_1)))]))
>>         )))):that <<<~(Misset_1,thelawchooses_1,
>>         .A77_1,.C77_1))])
>>      :that ((<<<~(Misset_1,thelawchooses_1,
>>      .A77_1,.B77_1) & <<<~(Misset_1,thelawchooses_1,
>>      .B77_1,.C77_1)) -> <<<~(Misset_1,thelawchooses_1,
>>      .A77_1,.C77_1)))]
>>   {move 0}



open

   define line84 ainm binm cinm: lineb84 \
      Misset, thelawchooses,ainm binm cinm


>>    line84: [(.a_1:obj),(ainm_1:that (.a_1
>>         E M)),(.b_1:obj),(binm_1:that (.b_1
>>         E M)),(.c_1:obj),(cinm_1:that (.c_1
>>         E M)) => (---:that ((<<<~(Misset,thelawchooses,
>>         .a_1,.b_1) & <<<~(Misset,thelawchooses,
>>         .b_1,.c_1)) -> <<<~(Misset,thelawchooses,
>>         .a_1,.c_1)))]
>>      {move 1}
```

The purported order is transitive. It really is a strict linear order, it's all true!

Our aim now is to show that the order is well-founded, so a well-ordering.

```
Lestrade execution:

%%  we have shown that <~ is a linear order.
% line67 = trichotomy, line69 irreflexive, line76 asymmetric, line84 = transitive

% it remains to show that it is well-founded.
```

```
   open

      declare S obj

>>       S: obj {move 3}



      declare Ssubm that S <<= M

>>       Ssubm: that (S <<= M) {move 3}



      declare z obj

>>       z: obj {move 3}



      declare zins that z E S

>>       zins: that (z E S) {move 3}



      define chosenof S: thelaw(Rcal1 S)


>>       chosenof: [(S_1:obj) => (---:obj)]
>>         {move 2}



      goal that chosenof S E S
```

```
>>      Goal: that (chosenof(S) E S)

     define line85 Ssubm zins: Fixform(chosenof \
        S E S,Line27 Ssubm, Ei1 z zins)

>>      line85: [(.S_1:obj),(Ssubm_1:that (.S_1
>>           <<= M)),(.z_1:obj),(zins_1:that
>>           (.z_1 E .S_1)) => (---:that (chosenof(.S_1)
>>           E .S_1))]
>>        {move 2}



     open

        declare xx obj

>>         xx: obj {move 4}



        goal that Forall[xx => (xx E S) \
              -> (xx = chosenof S) V (chosenof \
              S <~ xx)] \




>>         Goal: that Forall([(xx_277:obj)
>>              => (((xx_277 E S) -> ((xx_277
>>              = chosenof(S)) V (chosenof(S)
>>              <~ xx_277))):prop)])
>>

        open

           declare thehyp that xx E S
```

```
>>              thehyp: that (xx E S) {move 5}



             define line86 thehyp: Excmid(xx \
                = chosenof S)

>>              line86: [(thehyp_1:that (xx E
>>                   S)) => (---:that ((xx = chosenof(S))
>>                   V ~((xx = chosenof(S)))))]
>>                {move 4}



             open

                declare case1 that xx = chosenof \
                   S

>>                 case1: that (xx = chosenof(S))
>>                   {move 6}



                declare case2 that ~(xx = \
                   chosenof S)

>>                 case2: that ~((xx = chosenof(S)))
>>                   {move 6}



                define line87 case1: Add1(chosenof \
                   S <~ xx,case1)

>>                 line87: [(case1_1:that (xx
>>                      = chosenof(S))) => (---:
>>                      that ((xx = chosenof(S))
```

```
>>                      V (chosenof(S) <~ xx)))]
>>                   {move 5}



               goal that Rcal1 S = Rcal chosenof \
                  S

>>                Goal: that (Rcal1(S) = Rcal(chosenof(S)))


               define line88: Fixform(Rcal1 \
                  S E Mbold,Line4 Ssubm, Ei1 \
                  z zins)

>>                line88: [(---:that (Rcal1(S)
>>                     E Mbold))]
>>                  {move 5}


% will be using Line41 to show Rcal1 S = Rcal(chosenof S)


               define line89: Iff2(Mpsubs \
                  line85 Ssubm zins, Linea13 \
                  Ssubm , Ei1 z zins,Uscsubs \
                  chosenof S Rcal1 S)

>>                line89: [(---:that (Usc(chosenof(S))
>>                     <<= Rcal1(S)))]
>>                  {move 5}



               define linea90: (Line4 Ssubm, \
                  Ei1 z zins) Conj line89 Conj \
                  (Inusc2 chosenof S)
```

```
>>                  linea90: [(---:that (((((Misset
>>                       Mbold2 thelawchooses) Set
>>                       [(x1_3:obj) => ((S <<=
>>                          x1_3):prop)])
>>                       Intersection M) E (Misset
>>                       Mbold2 thelawchooses))
>>                       & ((Usc(chosenof(S)) <<=
>>                       Rcal1(S)) & (chosenof(S)
>>                       E (chosenof(S) ; chosenof(S))))))]
>>                    {move 5}



               define line90: Fixform(Rcal1 \
                  S = Rcal chosenof S,Line41 \
                  (Iff2 Mpsubs line85 Ssubm \
                  zins Ssubm,Uscsubs chosenof \
                  S M,Pairinhabited chosenof \
                  S chosenof S,linea90))

>>                  line90: [(---:that (Rcal1(S)
>>                       = Rcal(chosenof(S))))]
>>                    {move 5}



               define line91: Subs1 line90, \
                  Mpsubs thehyp, Linea13 Ssubm \
                  , Ei1 z zins

>>                  line91: [(---:that (xx E Rcal(chosenof(S))))]
>>                    {move 5}



               define line92 case2: Fixform(chosenof \
                  S <~ xx,(Mpsubs line85 Ssubm \
                  zins Ssubm) Conj (Mpsubs \
```

```
                  thehyp Ssubm) Conj (Negeqsymm \
                  case2) Conj line91)

>>                line92: [(case2_1:that ~((xx
>>                     = chosenof(S)))) => (---:
>>                     that (chosenof(S) <~ xx))]
>>                  {move 5}



               define line93 case2: Add2(xx=chosenof \
                  S,line92 case2)

>>                line93: [(case2_1:that ~((xx
>>                     = chosenof(S)))) => (---:
>>                     that ((xx = chosenof(S))
>>                     V (chosenof(S) <~ xx)))]
>>                  {move 5}



               close

            define line94 thehyp: Cases line86 \
               thehyp, line87, line93

>>             line94: [(thehyp_1:that (xx E
>>                  S)) => (---:that ((xx = chosenof(S))
>>                  V (chosenof(S) <~ xx)))]
>>               {move 4}



            close

         define line95 xx: Ded line94

>>          line95: [(xx_1:obj) => (---:that
```

```
>>              ((xx_1 E S) -> ((xx_1 = chosenof(S))
>>              V (chosenof(S) <~ xx_1))))]
>>           {move 3}
```

```
   close

define line96 Ssubm zins: Ug line95
```

```
>>     line96: [(.S_1:obj),(Ssubm_1:that (.S_1
>>          <<= M)),(.z_1:obj),(zins_1:that
>>          (.z_1 E .S_1)) => (---:that Forall([(xx_16:
>>             obj) => (((xx_16 E .S_1) -> ((xx_16
>>             = chosenof(.S_1)) V (chosenof(.S_1)
>>             <~ xx_16))):prop)]))
>>          ]
>>       {move 2}
```

```
define line97 Ssubm zins: Ei1 chosenof \
   S,Conj (line85 Ssubm zins,line96 Ssubm \
   zins)
```

```
>>     line97: [(.S_1:obj),(Ssubm_1:that (.S_1
>>          <<= M)),(.z_1:obj),(zins_1:that
>>          (.z_1 E .S_1)) => (---:that Exists([(x_5:
>>             obj) => (((x_5 E .S_1) & Forall([(xx_6:
>>                obj) => (((xx_6 E .S_1) ->
>>                ((xx_6 = x_5) V (x_5 <~ xx_6))):
>>                prop)]))
>>             :prop)]))
>>          ]
>>       {move 2}
```

```
      open

         declare x66 obj

>>          x66: obj {move 4}



         declare thehyp that (S <<= M ) & \
            Exists[x66 => x66 E S] \




>>          thehyp: that ((S <<= M) & Exists([(x66_1:
>>               obj) => ((x66_1 E S):prop)]))
>>            {move 4}



         open

            declare y66 obj

>>             y66: obj {move 5}



            declare yins66 that y66 E S

>>             yins66: that (y66 E S) {move
>>               5}



            define line98 yins66 : line97 \
               Simp1 thehyp yins66
```

```
>>             line98: [(.y66_1:obj),(yins66_1:
>>                  that (.y66_1 E S)) => (---:
>>                  that Exists([(x_3:obj) =>
>>                     (((x_3 E S) & Forall([(xx_4:
>>                        obj) => (((xx_4 E S)
>>                        -> ((xx_4 = x_3) V (x_3
>>                        <~ xx_4))):prop)]))
>>                     :prop)]))
>>                  ]
>>               {move 4}



          close

       define line99 thehyp: Eg Simp2 thehyp \
          line98

>>          line99: [(thehyp_1:that ((S <<=
>>               M) & Exists([(x66_2:obj) => ((x66_2
>>                  E S):prop)]))
>>               ) => (---:that Exists([(x_11:
>>                  obj) => (((x_11 E S) & Forall([(xx_12:
>>                     obj) => (((xx_12 E S) ->
>>                     ((xx_12 = x_11) V (x_11
>>                     <~ xx_12))):prop)]))
>>                  :prop)]))
>>               ]
>>            {move 3}



       close

    define line100 S: Ded line99

>>       line100: [(S_1:obj) => (---:that (((S_1
```

```
>>             <<= M) & Exists([(x66_17:obj) =>
>>                ((x66_17 E S_1):prop)]))
>>             -> Exists([(x_18:obj) => (((x_18
>>                E S_1) & Forall([(xx_19:obj)
>>                   => (((xx_19 E S_1) -> ((xx_19
>>                   = x_18) V (x_18 <~ xx_19))):
>>                   prop)]))
>>                :prop)]))
>>             )]
>>          {move 2}
```

```
      close

   define line101: Ug line100
```

```
>>    line101: [(---:that Forall([(S_50:obj)
>>          => ((((S_50 <<= M) & Exists([(x66_51:
>>             obj) => ((x66_51 E S_50):prop)]))
>>          -> Exists([(x_52:obj) => (((x_52
>>             E S_50) & Forall([(xx_53:obj)
>>                => (((xx_53 E S_50) -> ((xx_53
>>                = x_52) V (x_52 <~ xx_53))):
>>                prop)]))
>>             :prop)]))
>>          :prop)]))
>>       ]
>>    {move 1}
```

```
   close
comment the following line will not run until we work on definition expansion cont


define line102 Misset thelawchooses: line101
```

```
>> line102: [(.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([(S_9:obj) => (Ded([(thehyp_13:
>>            that ((S_9 <<= .M_1) & Exists([(x66_14:
>>               obj) => ((x66_14 E S_9):prop)]))
>>            ) => ((Simp2(thehyp_13) Eg [(.y66_19:
>>               obj),(yins66_19:that (.y66_19
>>               E S_9)) => ((.thelaw_1((((Misset_1
>>               Mbold2 thelawchooses_1) Set [(x1_20:
>>                  obj) => ((S_9 <<= x1_20):prop)])
>>               Intersection .M_1)) Ei1 (((.thelaw_1((((Misset_1
>>               Mbold2 thelawchooses_1) Set [(x1_24:
>>                  obj) => ((S_9 <<= x1_24):prop)])
>>               Intersection .M_1)) E S_9) Fixform
>>               Lineb27(Misset_1,thelawchooses_1,
>>               Simp1(thehyp_13),(.y66_19 Ei1
>>               yins66_19))) Conj Ug([(xx_33:
>>                  obj) => (Ded([(thehyp_36:that
>>                     (xx_33 E S_9)) => (Cases(Excmid((xx_33
>>                     = .thelaw_1((((Misset_1
>>                     Mbold2 thelawchooses_1)
>>                     Set [(x1_39:obj) => ((S_9
>>                        <<= x1_39):prop)])
>>                     Intersection .M_1)))),[(case1_42:
>>                        that (xx_33 = .thelaw_1((((Misset_1
>>                        Mbold2 thelawchooses_1)
>>                        Set [(x1_43:obj) =>
>>                           ((S_9 <<= x1_43):
>>                           prop)])
>>                        Intersection .M_1))))
>>                        => ((<<<~(Misset_1,thelawchooses_1,
>>                        .thelaw_1((((Misset_1
```

```
>>                            Mbold2 thelawchooses_1)
>>                            Set [(x1_44:obj) =>
>>                               ((S_9 <<= x1_44):
>>                               prop)])
>>                            Intersection .M_1)),
>>                            xx_33) Add1 case1_42):
>>                            that ((xx_33 = .thelaw_1((((Misset_1
>>                            Mbold2 thelawchooses_1)
>>                            Set [(x1_46:obj) =>
>>                               ((S_9 <<= x1_46):
>>                               prop)])
>>                            Intersection .M_1)))
>>                            V <<<~(Misset_1,thelawchooses_1,
>>                            .thelaw_1((((Misset_1
>>                            Mbold2 thelawchooses_1)
>>                            Set [(x1_47:obj) =>
>>                               ((S_9 <<= x1_47):
>>                               prop)])
>>                            Intersection .M_1)),
>>                            xx_33)))]
>>                         ,[(case2_48:that ~((xx_33
>>                            = .thelaw_1((((Misset_1
>>                            Mbold2 thelawchooses_1)
>>                            Set [(x1_49:obj) =>
>>                               ((S_9 <<= x1_49):
>>                               prop)])
>>                            Intersection .M_1)))))
>>                            => (((xx_33 = .thelaw_1((((Misset_1
>>                            Mbold2 thelawchooses_1)
>>                            Set [(x1_50:obj) =>
>>                               ((S_9 <<= x1_50):
>>                               prop)])
>>                            Intersection .M_1)))
>>                            Add2 (<<<~(Misset_1,
>>                            thelawchooses_1,.thelaw_1((((Misset_1
>>                            Mbold2 thelawchooses_1)
>>                            Set [(x1_52:obj) =>
>>                               ((S_9 <<= x1_52):
```

```
>>                            prop)])
>>                         Intersection .M_1)),
>>                         xx_33) Fixform ((((.thelaw_1((((Misset_1
>>                         Mbold2 thelawchooses_1)
>>                         Set [(x1_55:obj) =>
>>                            ((S_9 <<= x1_55):
>>                            prop)])
>>                         Intersection .M_1))
>>                         E S_9) Fixform Lineb27(Misset_1,
>>                         thelawchooses_1,Simp1(thehyp_13),
>>                         (.y66_19 Ei1 yins66_19)))
>>                         Mpsubs Simp1(thehyp_13))
>>                         Conj ((thehyp_36 Mpsubs
>>                         Simp1(thehyp_13)) Conj
>>                         (Negeqsymm(case2_48)
>>                         Conj ((((((Misset_1
>>                         Mbold2 thelawchooses_1)
>>                         Set [(x1_73:obj) =>
>>                            ((S_9 <<= x1_73):
>>                            prop)])
>>                         Intersection .M_1) =
>>                         (((Misset_1 Mbold2 thelawchooses_1)
>>                         Set [(x1_74:obj) =>
>>                            ((Usc(.thelaw_1((((Misset_1
>>                            Mbold2 thelawchooses_1)
>>                            Set [(x1_75:obj)
>>                               => ((S_9 <<= x1_75):
>>                               prop)])
>>                            Intersection .M_1)))
>>                            <<= x1_74):prop)])
>>                         Intersection .M_1))
>>                         Fixform Lineb41(Misset_1,
>>                         thelawchooses_1,((((.thelaw_1((((Misset_1
>>                         Mbold2 thelawchooses_1)
>>                         Set [(x1_79:obj) =>
>>                            ((S_9 <<= x1_79):
>>                            prop)])
>>                         Intersection .M_1))
```

```
>>                          E S_9) Fixform Lineb27(Misset_1,
>>                          thelawchooses_1,Simp1(thehyp_13),
>>                          (.y66_19 Ei1 yins66_19)))
>>                          Mpsubs Simp1(thehyp_13))
>>                          Iff2 (.thelaw_1((((Misset_1
>>                          Mbold2 thelawchooses_1)
>>                          Set [(x1_84:obj) =>
>>                             ((S_9 <<= x1_84):
>>                             prop)])
>>                          Intersection .M_1))
>>                          Uscsubs .M_1)),(.thelaw_1((((Misset_1
>>                          Mbold2 thelawchooses_1)
>>                          Set [(x1_85:obj) =>
>>                             ((S_9 <<= x1_85):
>>                             prop)])
>>                          Intersection .M_1))
>>                          Pairinhabited .thelaw_1((((Misset_1
>>                          Mbold2 thelawchooses_1)
>>                          Set [(x1_86:obj) =>
>>                             ((S_9 <<= x1_86):
>>                             prop)])
>>                          Intersection .M_1))),
>>                          (Lineb4(Misset_1,thelawchooses_1,
>>                          Simp1(thehyp_13),(.y66_19
>>                          Ei1 yins66_19)) Conj
>>                          (((((.thelaw_1((((Misset_1
>>                          Mbold2 thelawchooses_1)
>>                          Set [(x1_101:obj) =>
>>                             ((S_9 <<= x1_101):
>>                             prop)])
>>                          Intersection .M_1))
>>                          E S_9) Fixform Lineb27(Misset_1,
>>                          thelawchooses_1,Simp1(thehyp_13),
>>                          (.y66_19 Ei1 yins66_19)))
>>                          Mpsubs Lineab13(Misset_1,
>>                          thelawchooses_1,Simp1(thehyp_13),
>>                          (.y66_19 Ei1 yins66_19)))
>>                          Iff2 (.thelaw_1((((Misset_1
```

```
>>                          Mbold2 thelawchooses_1)
>>                          Set [(x1_109:obj) =>
>>                             ((S_9 <<= x1_109):
>>                             prop)])
>>                          Intersection .M_1))
>>                          Uscsubs (((Misset_1
>>                          Mbold2 thelawchooses_1)
>>                          Set [(x1_110:obj) =>
>>                             ((S_9 <<= x1_110):
>>                             prop)])
>>                          Intersection .M_1)))
>>                          Conj Inusc2(.thelaw_1((((Misset_1
>>                          Mbold2 thelawchooses_1)
>>                          Set [(x1_114:obj) =>
>>                             ((S_9 <<= x1_114):
>>                             prop)])
>>                          Intersection .M_1)))))))
>>                          Subs1 (thehyp_36 Mpsubs
>>                          Lineab13(Misset_1,thelawchooses_1,
>>                          Simp1(thehyp_13),(.y66_19
>>                          Ei1 yins66_19))))))))):
>>                          that ((xx_33 = .thelaw_1((((Misset_1
>>                          Mbold2 thelawchooses_1)
>>                          Set [(x1_119:obj) =>
>>                             ((S_9 <<= x1_119):
>>                             prop)])
>>                          Intersection .M_1)))
>>                          V <<<~(Misset_1,thelawchooses_1,
>>                          .thelaw_1((((Misset_1
>>                          Mbold2 thelawchooses_1)
>>                          Set [(x1_120:obj) =>
>>                             ((S_9 <<= x1_120):
>>                             prop)])
>>                          Intersection .M_1)),
>>                          xx_33)))])
>>                       :that ((xx_33 = .thelaw_1((((Misset_1
>>                       Mbold2 thelawchooses_1)
>>                       Set [(x1_121:obj) => ((S_9
```

```
>>                          <<= x1_121):prop)])
>>                       Intersection .M_1))) V
>>                       <<<~(Misset_1,thelawchooses_1,
>>                       .thelaw_1((((Misset_1 Mbold2
>>                       thelawchooses_1) Set [(x1_122:
>>                          obj) => ((S_9 <<= x1_122):
>>                          prop)])
>>                       Intersection .M_1)),xx_33)))])
>>                    :that ((xx_33 E S_9) -> ((xx_33
>>                    = .thelaw_1((((Misset_1 Mbold2
>>                    thelawchooses_1) Set [(x1_123:
>>                       obj) => ((S_9 <<= x1_123):
>>                       prop)])
>>                    Intersection .M_1))) V <<<~(Misset_1,
>>                    thelawchooses_1,.thelaw_1((((Misset_1
>>                    Mbold2 thelawchooses_1) Set
>>                    [(x1_124:obj) => ((S_9 <<=
>>                       x1_124):prop)])
>>                    Intersection .M_1)),xx_33))))]))
>>                 ):that Exists([(x_125:obj) =>
>>                    (((x_125 E S_9) & Forall([(xx_126:
>>                       obj) => (((xx_126 E S_9)
>>                       -> ((xx_126 = x_125) V
>>                       <<<~(Misset_1,thelawchooses_1,
>>                       x_125,xx_126))):prop)]))
>>                    :prop)]))
>>                 ])
>>              :that Exists([(x_127:obj) => (((x_127
>>                 E S_9) & Forall([(xx_128:obj)
>>                    => (((xx_128 E S_9) -> ((xx_128
>>                    = x_127) V <<<~(Misset_1,thelawchooses_1,
>>                    x_127,xx_128))):prop)]))
>>                 :prop)]))
>>              ])
>>           :that (((S_9 <<= .M_1) & Exists([(x66_129:
>>              obj) => ((x66_129 E S_9):prop)]))
>>           -> Exists([(x_130:obj) => (((x_130
>>              E S_9) & Forall([(xx_131:obj) =>
```

```
>>              (((xx_131 E S_9) -> ((xx_131
>>              = x_130) V <<<~(Misset_1,thelawchooses_1,
>>              x_130,xx_131))):prop)]))
>>           :prop)]))
>>        )])
>>     :that Forall([(S_132:obj) => ((((S_132
>>        <<= .M_1) & Exists([(x66_133:obj) =>
>>           ((x66_133 E S_132):prop)]))
>>        -> Exists([(x_134:obj) => (((x_134
>>           E S_132) & Forall([(xx_135:obj)
>>              => (((xx_135 E S_132) -> ((xx_135
>>              = x_134) V <<<~(Misset_1,thelawchooses_1,
>>              x_134,xx_135))):prop)]))
>>           :prop)]))
>>        :prop)]))
>>     ]
>>   {move 0}
```

We prove that a nonempty subset S of M has a minimal element in the order. The minimal element is the distinguished element s of $\mathcal{R}_1(S)$. One shows that $\mathcal{R}_1(S) = \mathcal{R}(s)$, from which it follows readily that s is an element of S and minimal in the order we defined.

This completes the proof that if we have a method of choosing a distinguished element from each subset of M, we can well-order M.

It remains to show that the Axiom of Choice in its usual form allows us to choose distinguished elements as required.