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

January 5, 2022

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.

```
begin Lestrade execution

   >>> comment load whatismath4

   {move 1}

   >>> open

      {move 2}

      >>> clearcurrent
{move 2}

      >>> 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), (.B1_1
    : obj), (aev_1 : that .A1_1 E Mbold), (bev_1
    : that .B1_1 E Mbold) => (---
    : that (.B1_1 <<= prime2 ([(S'_4
       : obj) =>
       ({def} thelaw (S'_4) : obj)], .A1_1)) V .A1_1
    <<= .B1_1)]

{move 1}

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

Mboldtotal : [(.A1_1 : obj), (.B1_1
    : obj), (aev_1 : that .A1_1 E Mbold), (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, [(S'_2
    : obj) =>
    ({def} thelaw (S'_2) : obj)], Misset
 Mbold2 thelawchooses)
```

```
    {move 1}
end Lestrade execution
```

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 **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.

```
begin 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_2 : obj) =>
    ({def} x2_2 E P : prop)])

{move 2}

>>> declare x obj

x : obj

{move 2}

>>> open

   {move 3}

   >>> 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
```

```
        that P0 E Mbold

        {move 3}

        >>> 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}
end Lestrade execution
```

P_0 is in **M**.

```
begin Lestrade execution

         >>> define p0 : thelaw P0

         p0 : obj

         {move 2}

         >>> goal that p0 E P

         that p0 E P

         {move 3}

         >>> open

            {move 4}

            >>> 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

            that z E P0

            {move 4}
```

```
>>> define line6 z : Ui z, Separation4 \
    Refleq P0

line6 : [(z_1 : obj) => (---
    : that (z_1 E M Set [(x_4
       : obj) =>
       ({def} Forall ([(B_5
          : obj) =>
          ({def} (B_5 E Pset) ->
          x_4 E B_5 : prop)]) : prop)]) ==
    (z_1 E M) & Forall ([(B_4
       : obj) =>
       ({def} (B_4 E Pset) ->
       z_1 E B_4 : 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

   {move 5}

   >>> declare B obj

   B : obj

   {move 5}

   >>> open
```

```
   {move 6}

   >>> declare Bev that B E Pset

   Bev : that B E Pset

   {move 6}

   >>> goal that z E B

   that z E B

   {move 6}

   >>> 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

{move 5}

>>> define line9 B : Ded line8

line9 : [(B_1 : obj) =>
    (--- : that (B_1 E Pset) ->
    z E B_1)]

{move 4}
```

```
   >>> close

{move 4}

>>> define line10 zev : Ug line9

line10 : [(.z_1 : obj), (zev_1
    : that .z_1 E P) => (---
    : that Forall ([(x''_2
       : obj) =>
       ({def} (x''_2 E Pset) ->
       .z_1 E x''_2 : 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_2 : obj) =>
          ({def} w_2 E P0 : prop)]))]

   {move 3}

   >>> close

{move 3}

>>> define line13 : Eg Pev2 line12

line13 : that Exists ([(w_2 : obj) =>
    ({def} w_2 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

   {move 4}

   >>> declare absurdhyp that ~ (p0 \
       E P)

   absurdhyp : that ~ (p0 E P)

   {move 4}

   >>> open

      {move 5}

      >>> declare Q obj

      Q : obj

      {move 5}

      >>> open

         {move 6}

         >>> 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

   {move 7}

   >>> 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

   {move 6}

   >>> 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

{move 5}

>>> define line20 Q : Ded \
    line19

line20 : [(Q_1 : obj) =>
    (--- : that (Q_1 E P) ->
    Q_1 E prime (P0))]
```

```
   {move 4}

   >>> save

   {move 5}

   >>> close

{move 4}

>>> 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 ([(S'_3 : obj) =>
       ({def} thelaw (S'_3) : obj)], 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

      {move 4}

      >>> close

   {move 3}

   >>> define line27 : Dneg Negintro \
     line26

   line27 : that p0 E P

   {move 2}
end Lestrade execution
```

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

```
begin Lestrade execution

   >>> declare P1 obj

   P1 : obj

   {move 3}

   >>> goal that ~ ((thelaw P1) E prime \
```

```
    P1)

that ~ (thelaw (P1) E prime (P1))

{move 3}

>>> open

   {move 4}

   >>> 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
```

```
   {move 3}

   >>> define primefact1 P1 : Negintro \
       line29

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

   {move 2}

   >>> save

   {move 3}

   >>> close

{move 2}

>>> 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

{move 2}
```

```
   >>> close

{move 1}

>>> 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_2 : obj), (subsetev_2 : that
       .S_2 <<= .M_1), (inev_2 : that
       Exists ([(x_4 : obj) =>
          ({def} x_4 E .S_2 : prop)])) =>
       (--- : that .thelaw_1 (.S_2) E .S_2)]), (P3_1
    : obj) =>
    ({def} Negintro ([(neghyp_2 : that
       .thelaw_1 (P3_1) E prime2 (.thelaw_1, P3_1)) =>
       ({def} Inusc2 (.thelaw_1 (P3_1)) Mp
       Simp2 (Separation5 (neghyp_2)) : that
       ??)]) : that ~ (.thelaw_1 (P3_1) E prime2
    (.thelaw_1, P3_1)))]

primefact3 : [(.M_1 : obj), (Misset_1
    : that Isset (.M_1)), (.thelaw_1
    : [(S_2 : obj) => (--- : obj)]), (thelawchooses_1
    : [(.S_2 : obj), (subsetev_2 : that
       .S_2 <<= .M_1), (inev_2 : that
       Exists ([(x_4 : obj) =>
          ({def} x_4 E .S_2 : prop)])) =>
       (--- : that .thelaw_1 (.S_2) E .S_2)]), (P3_1
```

```
    : obj) => (--- : that ~ (.thelaw_1
    (P3_1) E prime2 (.thelaw_1, P3_1)))]

{move 0}

>>> open

   {move 2}

   >>> define primefact4 P2 : primefact3 \
       Misset, thelawchooses, P2

   primefact4 : [(P2_1 : obj) => (---
       : that ~ (thelaw (P2_1) E prime2
       ([(S'_4 : obj) =>
          ({def} thelaw (S'_4) : obj)], P2_1)))]

   {move 1}

   >>> open

      {move 3}

      >>> define primefact P1 : primefact4 \
          P1

      primefact : [(P1_1 : obj) =>
          (--- : that ~ (thelaw (P1_1) E prime2
          ([(S'_4 : obj) =>
             ({def} thelaw (S'_4) : obj)], P1_1)))]

      {move 2}
end Lestrade execution
```

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 **M** which includes P and whose distinguished element is in P.

```
begin Lestrade execution

         >>> open

            {move 4}

            >>> 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

            that P0 <<= P1

            {move 4}

            >>> open
```

```
{move 5}

>>> declare z obj

z : obj

{move 5}

>>> open

   {move 6}

   >>> declare zev that z E P0

   zev : that z E P0

   {move 6}

   >>> goal that z E P1

   that z E P1

   {move 6}

   >>> 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

   {move 5}

   >>> 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}
end Lestrade execution
```

P_0 is a subset of P_1.

```
begin Lestrade execution

                >>> goal that P0 <<= prime \
                    P1

                that P0 <<= prime (P1)

                {move 5}

                >>> goal that ~ (P1 <<= P0)

                that ~ (P1 <<= P0)

                {move 5}

                >>> open

                   {move 6}

                   >>> 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

{move 5}

>>> 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}
```
end Lestrade execution

and in fact a subset of P_1'

begin Lestrade execution

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

that ~ (thelaw (P1) E P)

{move 5}

>>> open

   {move 6}
```

```
                >>> 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

             {move 5}

             >>> define line39 : Negintro \
                 line38

             line39 : that ~ (thelaw (P1) E P)

             {move 4}
end Lestrade execution
```

so the distinguished element of P_1 is not in P.

```
begin Lestrade execution

             >>> close
```

```
{move 4}

>>> 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
```

```
{move 3}

>>> 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}
end Lestrade execution
```

Some results are recapitulated at lower moves.

```
begin 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

    that P1 = P0

    {move 3}

    >>> open

       {move 4}

       >>> 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

     {move 3}

     >>> 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

  {move 2}
end Lestrade execution
```

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.

```
begin 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}
end Lestrade execution
```

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 $\mathbf{M}$ containing it) and the function $\mathcal{R}$ defined by Zermelo, $\mathcal{R}(x)$ being $\mathcal{R}_1(\{x\})$, the intersection of all elements of $\mathbf{M}$ containing x.

```
begin Lestrade execution

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

      that thelaw (Rcal (x)) = x

      {move 2}

      >>> 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_3 : obj) =>
             ({def} x2_3 E .P_1 : prop)])) =>
          (--- : that thelaw (Rcal1 (.P_1)) E .P_1)]

      {move 1}

      >>> save

      {move 2}
```

```
   >>> close

{move 1}

>>> 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 E P77]

Pev277 : that Exists ([(x77_2 : obj) =>
    ({def} x77_2 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_2 : obj), (subsetev_2 : that
       .S_2 <<= .M_1), (inev_2 : that
```

```
   Exists ([(x_4 : obj) =>
      ({def} x_4 E .S_2 : prop)])) =>
   (--- : that .thelaw_1 (.S_2) E .S_2)]), (.P77_1
: obj), (Pev77_1 : that .P77_1 <<=
.M_1), (Pev277_1 : that Exists ([(x77_3
   : obj) =>
   ({def} x77_3 E .P77_1 : prop)])) =>
({def} (.thelaw_1 ((Misset_1 Mbold2
thelawchooses_1 Set [(x1_6 : obj) =>
   ({def} .P77_1 <<= x1_6 : prop)]) Intersection
.M_1) E .P77_1) Fixform Dneg (Negintro
([(absurdhyp_4 : that ~ (.thelaw_1
   ((Misset_1 Mbold2 thelawchooses_1
   Set [(x1_10 : obj) =>
      ({def} .P77_1 <<= x1_10 : prop)]) Intersection
   .M_1) E .P77_1)) =>
   ({def} Inusc2 (.thelaw_1 ((Misset_1
   Mbold2 thelawchooses_1 Set [(x1_9
      : obj) =>
      ({def} .P77_1 <<= x1_9 : prop)]) Intersection
   .M_1)) Mp Simp2 (((prime2 (.thelaw_1, (Misset_1
   Mbold2 thelawchooses_1 Set [(x1_13
      : obj) =>
      ({def} .P77_1 <<= x1_13 : prop)]) Intersection
   .M_1) E Misset_1 Mbold2 thelawchooses_1
   Set [(x1_11 : obj) =>
      ({def} .P77_1 <<= x1_11 : prop)]) Fixform
   ((((Misset_1 Mbold2 thelawchooses_1
   Set [(x1_16 : obj) =>
      ({def} .P77_1 <<= x1_16 : prop)]) Intersection
   .M_1) E Misset_1 Mbold2 thelawchooses_1) Fixform
   (((Misset_1 Mbold2 thelawchooses_1
   Set [(x1_18 : obj) =>
      ({def} .P77_1 <<= x1_18 : 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) =>
   ({def} .P77_1 <<= x1_19 : prop)])) Conj
(.M_1 E Misset_1 Mbold2 thelawchooses_1
Set [(x1_18 : obj) =>
   ({def} .P77_1 <<= x1_18 : prop)]) Fixform
Simp1 (Misset_1 Mboldtheta2 thelawchooses_1) Conj
Pev77_1 Iff2 .M_1 Ui Separation4
(Refleq (Misset_1 Mbold2 thelawchooses_1
Set [(x1_21 : obj) =>
   ({def} .P77_1 <<= x1_21 : prop)])) Mp
.M_1 Ui (Misset_1 Mbold2 thelawchooses_1
Set [(x1_17 : obj) =>
   ({def} .P77_1 <<= x1_17 : prop)]) Ui
Simp2 (Simp2 (Simp2 (Misset_1
Mboldtheta2 thelawchooses_1)))) Mp
((Misset_1 Mbold2 thelawchooses_1
Set [(x1_15 : obj) =>
   ({def} .P77_1 <<= x1_15 : 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_16
   : obj) =>
   ({def} .P77_1 <<= x1_16 : prop)]) Intersection
.M_1)) Fixform Ug ([(Q_14 : obj) =>
   ({def} Ded ([(Qev_15 : that
      Q_14 E .P77_1) =>
      ({def} (Q_14 E prime2 (.thelaw_1, (Misset_1
      Mbold2 thelawchooses_1 Set
      [(x1_20 : obj) =>
         ({def} .P77_1 <<= x1_20
         : prop)]) Intersection
      .M_1)) Fixform ((Q_14
      E (Misset_1 Mbold2 thelawchooses_1
      Set [(x1_22 : obj) =>
         ({def} .P77_1 <<= x1_22
         : prop)]) Intersection
      .M_1) Fixform Qev_15 Mpsubs
```

```
Pev77_1 Conj Ug ([(B_22
   : obj) =>
   ({def} Ded ([(Bev_23
      : that B_22 E Misset_1
      Mbold2 thelawchooses_1
      Set [(x1_26 : obj) =>
         ({def} .P77_1 <<=
         x1_26 : prop)]) =>
      ({def} Qev_15 Mpsubs
      Simp2 (Bev_23 Iff1
      B_22 Ui Separation4
      (Refleq (Misset_1
      Mbold2 thelawchooses_1
      Set [(x1_30 : obj) =>
         ({def} .P77_1 <<=
         x1_30 : prop)]))) : that
      Q_14 E B_22)]) : that
   (B_22 E Misset_1 Mbold2
   thelawchooses_1 Set [(x1_25
      : obj) =>
      ({def} .P77_1 <<= x1_25
      : prop)]) -> Q_14
   E B_22)]) Iff2 Q_14
Ui Separation4 (Refleq ((Misset_1
Mbold2 thelawchooses_1 Set
[(x1_25 : obj) =>
   ({def} .P77_1 <<= x1_25
   : prop)]) Intersection
.M_1))) Conj Negintro ([(eqtest_19
   : that Q_14 E Usc (.thelaw_1
   ((Misset_1 Mbold2 thelawchooses_1
   Set [(x1_25 : obj) =>
      ({def} .P77_1 <<= x1_25
      : prop)]) Intersection
   .M_1))) =>
   ({def} Qev_15 Mp Eqsymm
   (Inusc1 (eqtest_19)) Subs1
   absurdhyp_4 : that ??)]) Iff2
```

```
      Q_14 Ui Separation4 (Refleq
      (prime2 (.thelaw_1, (Misset_1
      Mbold2 thelawchooses_1 Set
      [(x1_23 : obj) =>
         ({def} .P77_1 <<= x1_23
         : prop)]) Intersection
      .M_1))) : that Q_14 E prime2
      (.thelaw_1, (Misset_1 Mbold2
      thelawchooses_1 Set [(x1_19
         : obj) =>
         ({def} .P77_1 <<= x1_19
         : prop)]) Intersection
      .M_1))]) : that (Q_14
   E .P77_1) -> Q_14 E prime2 (.thelaw_1, (Misset_1
   Mbold2 thelawchooses_1 Set [(x1_19
      : obj) =>
      ({def} .P77_1 <<= x1_19 : prop)]) Intersection
   .M_1))]) Conj (.P77_1 = 0) Add2
Pev277_1 Conj Separation3 (Refleq
(prime2 (.thelaw_1, (Misset_1
Mbold2 thelawchooses_1 Set [(x1_19
   : obj) =>
   ({def} .P77_1 <<= x1_19 : prop)]) Intersection
.M_1))) Iff2 prime2 (.thelaw_1, (Misset_1
Mbold2 thelawchooses_1 Set [(x1_14
   : obj) =>
   ({def} .P77_1 <<= x1_14 : prop)]) Intersection
.M_1) Ui Separation4 (Refleq (Misset_1
Mbold2 thelawchooses_1 Set [(x1_14
   : obj) =>
   ({def} .P77_1 <<= x1_14 : prop)]))) Mp
prime2 (.thelaw_1, (Misset_1
Mbold2 thelawchooses_1 Set [(x1_12
   : obj) =>
   ({def} .P77_1 <<= x1_12 : prop)]) Intersection
.M_1) Ui Simp2 (((.thelaw_1
((Misset_1 Mbold2 thelawchooses_1
Set [(x1_16 : obj) =>
```

```
   ({def} .P77_1 <<= x1_16 : prop)]) Intersection
.M_1) E (Misset_1 Mbold2 thelawchooses_1
Set [(x1_15 : obj) =>
   ({def} .P77_1 <<= x1_15 : prop)]) Intersection
.M_1) Fixform thelawchooses_1 (.M_1
Set [(x_14 : obj) =>
   ({def} Forall ([(B_15 : obj) =>
      ({def} (B_15 E Misset_1
      Mbold2 thelawchooses_1 Set
      [(x1_18 : obj) =>
         ({def} .P77_1 <<= x1_18
         : prop)]) -> x_14 E B_15
      : prop)]) : prop)], Misset_1
Sepsub2 Refleq ((Misset_1 Mbold2
thelawchooses_1 Set [(x1_17 : obj) =>
   ({def} .P77_1 <<= x1_17 : prop)]) Intersection
.M_1), Pev277_1 Eg [(.z_14 : obj), (zev_14
   : that .z_14 E .P77_1) =>
   ({def} Exists ([(w_16 : obj) =>
      ({def} w_16 E (Misset_1
      Mbold2 thelawchooses_1 Set
      [(x1_19 : obj) =>
         ({def} .P77_1 <<= x1_19
         : prop)]) Intersection
      .M_1 : prop)]) Fixform
   .z_14 Ei1 (.z_14 E (Misset_1
   Mbold2 thelawchooses_1 Set [(x1_20
      : obj) =>
      ({def} .P77_1 <<= x1_20 : prop)]) Intersection
   .M_1) Fixform zev_14 Mpsubs
   Pev77_1 Conj Ug ([(B_20 : obj) =>
      ({def} Ded ([(Bev_21 : that
         B_20 E Misset_1 Mbold2
         thelawchooses_1 Set [(x1_24
            : obj) =>
            ({def} .P77_1 <<= x1_24
            : prop)]) =>
         ({def} zev_14 Mpsubs Simp2
```

```
          (Bev_21 Iff1 B_20 Ui Separation4
          (Refleq (Misset_1 Mbold2
          thelawchooses_1 Set [(x1_28
             : obj) =>
             ({def} .P77_1 <<= x1_28
             : prop)]))) : that
          .z_14 E B_20)]) : that
       (B_20 E Misset_1 Mbold2 thelawchooses_1
       Set [(x1_23 : obj) =>
          ({def} .P77_1 <<= x1_23
          : prop)]) -> .z_14 E B_20)]) Iff2
    .z_14 Ui Separation4 (Refleq
    ((Misset_1 Mbold2 thelawchooses_1
    Set [(x1_23 : obj) =>
       ({def} .P77_1 <<= x1_23 : prop)]) Intersection
    .M_1)) : that Exists ([(w_15
       : obj) =>
       ({def} w_15 E (Misset_1
       Mbold2 thelawchooses_1 Set
       [(x1_18 : obj) =>
          ({def} .P77_1 <<= x1_18
          : prop)]) Intersection
       .M_1 : prop)]))])) Iff1
.thelaw_1 ((Misset_1 Mbold2 thelawchooses_1
Set [(x1_15 : obj) =>
   ({def} .P77_1 <<= x1_15 : prop)]) Intersection
.M_1) Ui Separation4 (Refleq ((Misset_1
Mbold2 thelawchooses_1 Set [(x1_16
   : obj) =>
   ({def} .P77_1 <<= x1_16 : prop)]) Intersection
.M_1))) Iff1 .thelaw_1 ((Misset_1
Mbold2 thelawchooses_1 Set [(x1_11
   : obj) =>
   ({def} .P77_1 <<= x1_11 : prop)]) Intersection
.M_1) Ui Separation4 (Refleq (prime2
(.thelaw_1, (Misset_1 Mbold2
thelawchooses_1 Set [(x1_13 : obj) =>
   ({def} .P77_1 <<= x1_13 : prop)]) Intersection
```

```
      .M_1)))) : that ??)])) : that
    .thelaw_1 ((Misset_1 Mbold2 thelawchooses_1
    Set [(x1_5 : obj) =>
      ({def} .P77_1 <<= x1_5 : prop)]) Intersection
    .M_1) E .P77_1)]

Lineb27 : [(.M_1 : obj), (Misset_1
    : that Isset (.M_1)), (.thelaw_1
    : [(S_2 : obj) => (--- : obj)]), (thelawchooses_1
    : [(.S_2 : obj), (subsetev_2 : that
      .S_2 <<= .M_1), (inev_2 : that
      Exists ([(x_4 : obj) =>
        ({def} x_4 E .S_2 : prop)])) =>
      (--- : that .thelaw_1 (.S_2) E .S_2)]), (.P77_1
    : obj), (Pev77_1 : that .P77_1 <<=
    .M_1), (Pev277_1 : that Exists ([(x77_3
      : obj) =>
      ({def} x77_3 E .P77_1 : prop)])) =>
    (--- : that .thelaw_1 ((Misset_1
    Mbold2 thelawchooses_1 Set [(x1_5
      : obj) =>
      ({def} .P77_1 <<= x1_5 : prop)]) Intersection
    .M_1) E .P77_1)]

{move 0}

>>> open

   {move 2}

   >>> define Line27 Pev Pev2 : Lineb27 \
       Misset, thelawchooses, Pev, Pev2

   Line27 : [(.P_1 : obj), (Pev_1
       : that .P_1 <<= M), (Pev2_1 : that
       Exists ([(x2_3 : obj) =>
         ({def} x2_3 E .P_1 : prop)])) =>
       (--- : that thelaw ((Misset Mbold2
```

```
    thelawchooses Set [(x1_5 : obj) =>
       ({def} .P_1 <<= x1_5 : prop)]) Intersection
    M) E .P_1)]

{move 1}

>>> declare xinm that x E M

xinm : that x E M

{move 2}

>>> open

   {move 3}

   >>> define line42 : Iff2 xinm, Uscsubs \
       x M

   line42 : that Usc (x) <<= M

   {move 2}

   >>> define line43 : Pairinhabited \
       x x

   line43 : that Exists ([(u_2 : obj) =>
       ({def} u_2 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

  {move 2}

  >>> define line45 xinm : line44

  line45 : [(.x_1 : obj), (xinm_1
      : that .x_1 E M) => (--- : that
      thelaw (Rcal (.x_1)) = .x_1)]

  {move 1}
end Lestrade execution
```

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.

```
begin 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_3 : obj) =>
         ({def} x2_3 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_4 : obj) =>
         ({def} .P_1 <<= x1_4 : prop)]) Intersection
      M)]

   {move 1}

   >>> save

   {move 2}

   >>> close

{move 1}

>>> 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_2 : obj), (subsetev_2 : that
   .S_2 <<= .M_1), (inev_2 : that
   Exists ([(x_4 : obj) =>
      ({def} x_4 E .S_2 : prop)])) =>
   (--- : that .thelaw_1 (.S_2) E .S_2)]), (.P77_1
: obj), (Pev77_1 : that .P77_1 <<=
.M_1), (Pev277_1 : that Exists ([(x77_3
   : obj) =>
   ({def} x77_3 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) =>
({def} Dneg (Negintro ([(sillyhyp_3
   : that ~ (.Q77_1 = (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_8 : obj) =>
      ({def} .P77_1 <<= x1_8 : prop)]) Intersection
   .M_1)) =>
   ({def} Simp2 (Simp2 (phypsq77_1)) Mp
   Negintro ([(sillyhyp_5 : that
      .thelaw_1 (.Q77_1) E .P77_1) =>
      ({def} sillyhyp_5 Mpsubs (.P77_1
      <<= (Misset_1 Mbold2 thelawchooses_1
      Set [(x1_12 : obj) =>
         ({def} .P77_1 <<= x1_12 : prop)]) Intersection
      .M_1) Fixform Ug ([(z_11
         : obj) =>
         ({def} Ded ([(zev2_12
            : that z_11 E .P77_1) =>
            ({def} (z_11 E (Misset_1
            Mbold2 thelawchooses_1
            Set [(x1_16 : obj) =>
               ({def} .P77_1 <<= x1_16
               : prop)]) Intersection
            .M_1) Fixform zev2_12
            Mpsubs Pev77_1 Conj Ug
            ([(B_16 : obj) =>
```

```
      ({def} Ded ([(Bev_17
         : that B_16 E Misset_1
         Mbold2 thelawchooses_1
         Set [(x1_20 : obj) =>
            ({def} .P77_1
            <<= x1_20 : prop)]) =>
         ({def} zev2_12 Mpsubs
         Simp2 (Bev_17 Iff1
         B_16 Ui Separation4
         (Refleq (Misset_1
         Mbold2 thelawchooses_1
         Set [(x1_24 : obj) =>
            ({def} .P77_1
            <<= x1_24 : prop)]))) : that
         z_11 E B_16)]) : that
      (B_16 E Misset_1 Mbold2
      thelawchooses_1 Set
      [(x1_19 : obj) =>
         ({def} .P77_1 <<=
         x1_19 : prop)]) ->
      z_11 E B_16)]) Iff2
   z_11 Ui Separation4 (Refleq
   ((Misset_1 Mbold2 thelawchooses_1
   Set [(x1_19 : obj) =>
      ({def} .P77_1 <<= x1_19
      : prop)]) Intersection
   .M_1)) : that z_11 E (Misset_1
   Mbold2 thelawchooses_1
   Set [(x1_15 : obj) =>
      ({def} .P77_1 <<= x1_15
      : prop)]) Intersection
   .M_1)]) : that (z_11
E .P77_1) -> z_11 E (Misset_1
Mbold2 thelawchooses_1 Set
[(x1_15 : obj) =>
   ({def} .P77_1 <<= x1_15
   : prop)]) Intersection
.M_1)]) Conj Simp1 (Simp2
```

```
(Pev77_1)) Conj Separation3
(Refleq ((Misset_1 Mbold2
thelawchooses_1 Set [(x1_15
   : obj) =>
   ({def} .P77_1 <<= x1_15 : prop)]) Intersection
.M_1)) Mpsubs (((Misset_1
Mbold2 thelawchooses_1 Set [(x1_11
   : obj) =>
   ({def} .P77_1 <<= x1_11 : 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_3), (((Misset_1
Mbold2 thelawchooses_1 Set [(x1_14
   : obj) =>
   ({def} .P77_1 <<= x1_14 : prop)]) Intersection
.M_1) E Misset_1 Mbold2 thelawchooses_1) Fixform
(((Misset_1 Mbold2 thelawchooses_1
Set [(x1_16 : obj) =>
   ({def} .P77_1 <<= x1_16 : prop)]) <<=
Misset_1 Mbold2 thelawchooses_1) Fixform
Separation3 (Refleq (Misset_1
Mbold2 thelawchooses_1)) Sepsub2
Refleq (Misset_1 Mbold2 thelawchooses_1
Set [(x1_17 : obj) =>
   ({def} .P77_1 <<= x1_17 : prop)])) Conj
(.M_1 E Misset_1 Mbold2 thelawchooses_1
Set [(x1_16 : obj) =>
   ({def} .P77_1 <<= x1_16 : prop)]) Fixform
Simp1 (Misset_1 Mboldtheta2
thelawchooses_1) Conj Pev77_1
Iff2 .M_1 Ui Separation4 (Refleq
(Misset_1 Mbold2 thelawchooses_1
Set [(x1_19 : obj) =>
   ({def} .P77_1 <<= x1_19 : prop)])) Mp
.M_1 Ui (Misset_1 Mbold2 thelawchooses_1
Set [(x1_15 : obj) =>
```

```
   ({def} .P77_1 <<= x1_15 : prop)]) Ui
Simp2 (Simp2 (Simp2 (Misset_1
Mboldtheta2 thelawchooses_1)))) Ds1
Negintro ([(sillyhyp_10 : that
   .Q77_1 <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_14
      : obj) =>
      ({def} .P77_1 <<= x1_14
      : prop)]) Intersection
   .M_1) =>
   ({def} sillyhyp_10 Antisymsub
   (((Misset_1 Mbold2 thelawchooses_1
   Set [(x1_16 : obj) =>
      ({def} .P77_1 <<= x1_16
      : prop)]) Intersection
   .M_1) <<= .Q77_1) Fixform
   Ug ([(z_15 : obj) =>
      ({def} Ded ([(zev_16
         : that z_15 E (Misset_1
         Mbold2 thelawchooses_1
         Set [(x1_20 : obj) =>
            ({def} .P77_1 <<=
            x1_20 : prop)]) Intersection
         .M_1) =>
         ({def} ((.Q77_1 E Misset_1
         Mbold2 thelawchooses_1
         Set [(x1_20 : obj) =>
            ({def} .P77_1 <<=
            x1_20 : prop)]) Fixform
         Simp1 (Simp1 (phypsq77_1) Conj
         Simp1 (Simp2 (phypsq77_1)) Conj
         sillyhyp_3) Conj Simp1
         (Simp2 (Simp1 (phypsq77_1) Conj
         Simp1 (Simp2 (phypsq77_1)) Conj
         sillyhyp_3)) Iff2
         .Q77_1 Ui Separation4
         (Refleq (Misset_1
         Mbold2 thelawchooses_1
```

```
                    Set [(x1_23 : obj) =>
                       ({def} .P77_1 <<=
                       x1_23 : prop)]))) Mp
                    .Q77_1 Ui Simp2 (Separation5
                    (zev_16)) : that
                    z_15 E .Q77_1)]) : that
                 (z_15 E (Misset_1 Mbold2
                 thelawchooses_1 Set [(x1_19
                    : obj) =>
                    ({def} .P77_1 <<= x1_19
                    : prop)]) Intersection
                 .M_1) -> z_15 E .Q77_1)]) Conj
              Separation3 (Refleq ((Misset_1
              Mbold2 thelawchooses_1 Set
              [(x1_19 : obj) =>
                 ({def} .P77_1 <<= x1_19
                 : prop)]) Intersection
              .M_1)) Conj Simp2 (Simp2
              (Simp1 (Simp2 (Simp1 (phypsq77_1) Conj
              Simp1 (Simp2 (phypsq77_1)) Conj
              sillyhyp_3)))) Mp Simp2
              (Simp2 (Simp1 (phypsq77_1) Conj
              Simp1 (Simp2 (phypsq77_1)) Conj
              sillyhyp_3)) : that ??)]) Mp
           primefact3 (Misset_1, thelawchooses_1, .Q77_1) : that
           ??)]) : that ??)])) : that
     .Q77_1 = (Misset_1 Mbold2 thelawchooses_1
     Set [(x1_4 : obj) =>
        ({def} .P77_1 <<= x1_4 : prop)]) Intersection
     .M_1)]

Lineb41 : [(.M_1 : obj), (Misset_1
     : that Isset (.M_1)), (.thelaw_1
     : [(S_2 : obj) => (--- : obj)]), (thelawchooses_1
     : [(.S_2 : obj), (subsetev_2 : that
        .S_2 <<= .M_1), (inev_2 : that
        Exists ([(x_4 : obj) =>
           ({def} x_4 E .S_2 : prop)])) =>
```

```
      (--- : that .thelaw_1 (.S_2) E .S_2)]), (.P77_1
   : obj), (Pev77_1 : that .P77_1 <<=
   .M_1), (Pev277_1 : that Exists ([(x77_3
      : obj) =>
      ({def} x77_3 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) =>
   (--- : that .Q77_1 = (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_4 : obj) =>
      ({def} .P77_1 <<= x1_4 : prop)]) Intersection
   .M_1)]

{move 0}

>>> open

   {move 2}

   >>> 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_3 : obj) =>
          ({def} x2_3 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_4 : obj) =>
          ({def} .P_1 <<= x1_4 : 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

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

{move 2}

>>> open

   {move 3}

   >>> 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

{move 2}

>>> 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}
end Lestrade execution
```

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 $Q = \mathcal{R}(x)$.

```
begin 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

   {move 2}

   >>> close

{move 1}

>>> 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_2 : obj), (subsetev_2 : that
       .S_2 <<= .M_1), (inev_2 : that
       Exists ([(x_4 : obj) =>
          ({def} x_4 E .S_2 : prop)])) =>
       (--- : that .thelaw_1 (.S_2) E .S_2)]), (A37_1
    : obj), (B37_1 : obj) =>
    ({def} (A37_1 E .M_1) & (B37_1
    E .M_1) & ~ (A37_1 = B37_1) & B37_1
    E (Misset_1 Mbold2 thelawchooses_1
    Set [(x1_7 : obj) =>
       ({def} Usc (A37_1) <<= x1_7 : prop)]) Intersection
    .M_1 : prop)]

<<<~ : [(.M_1 : obj), (Misset_1 : that
    Isset (.M_1)), (.thelaw_1 : [(S_2
       : obj) => (--- : obj)]), (thelawchooses_1
    : [(.S_2 : obj), (subsetev_2 : that
       .S_2 <<= .M_1), (inev_2 : that
       Exists ([(x_4 : obj) =>
          ({def} x_4 E .S_2 : prop)])) =>
       (--- : that .thelaw_1 (.S_2) E .S_2)]), (A37_1
    : obj), (B37_1 : obj) => (---
    : prop)]
```

```
   {move 0}

   >>> open

      {move 2}

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

      <~ : [(a_1 : obj), (b_1 : obj) =>
          (--- : prop)]

      {move 1}
end Lestrade execution
```

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.

```
begin Lestrade execution

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

      that (b E Rcal (a)) == Rcal (b) <<=
       Rcal (a)

      {move 2}

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

      Linea4 : [(.P_1 : obj), (Pev_1
          : that .P_1 <<= M), (Pev2_1 : that
          Exists ([(x2_3 : obj) =>
```

```
            ({def} x2_3 E .P_1 : prop)])) =>
         (--- : that ((Mbold Set [(x1_4
            : obj) =>
            ({def} .P_1 <<= x1_4 : prop)]) Intersection
         M) E Mbold)]

   {move 1}

   >>> save

   {move 2}

   >>> close

{move 1}

>>> 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_2 : obj), (subsetev_2 : that
       .S_2 <<= .M_1), (inev_2 : that
       Exists ([(x_4 : obj) =>
          ({def} x_4 E .S_2 : prop)])) =>
       (--- : that .thelaw_1 (.S_2) E .S_2)]), (.P77_1
    : obj), (Pev77_1 : that .P77_1 <<=
    .M_1), (Pev277_1 : that Exists ([(x77_3
       : obj) =>
       ({def} x77_3 E .P77_1 : prop)])) =>
    ({def} (((Misset_1 Mbold2 thelawchooses_1
    Set [(x1_5 : obj) =>
       ({def} .P77_1 <<= x1_5 : prop)]) Intersection
    .M_1) E Misset_1 Mbold2 thelawchooses_1) Fixform
    (((Misset_1 Mbold2 thelawchooses_1
    Set [(x1_6 : obj) =>
       ({def} .P77_1 <<= x1_6 : prop)]) Intersection
```

```
    .M_1) E Misset_1 Mbold2 thelawchooses_1) Fixform
    (((Misset_1 Mbold2 thelawchooses_1
    Set [(x1_8 : obj) =>
       ({def} .P77_1 <<= x1_8 : prop)]) <<=
    Misset_1 Mbold2 thelawchooses_1) Fixform
    Separation3 (Refleq (Misset_1 Mbold2
    thelawchooses_1)) Sepsub2 Refleq
    (Misset_1 Mbold2 thelawchooses_1 Set
    [(x1_9 : obj) =>
       ({def} .P77_1 <<= x1_9 : prop)])) Conj
    (.M_1 E Misset_1 Mbold2 thelawchooses_1
    Set [(x1_8 : obj) =>
       ({def} .P77_1 <<= x1_8 : prop)]) Fixform
    Simp1 (Misset_1 Mboldtheta2 thelawchooses_1) Conj
    Pev77_1 Iff2 .M_1 Ui Separation4 (Refleq
    (Misset_1 Mbold2 thelawchooses_1 Set
    [(x1_11 : obj) =>
       ({def} .P77_1 <<= x1_11 : prop)])) Mp
    .M_1 Ui (Misset_1 Mbold2 thelawchooses_1
    Set [(x1_7 : obj) =>
       ({def} .P77_1 <<= x1_7 : prop)]) Ui
    Simp2 (Simp2 (Simp2 (Misset_1 Mboldtheta2
    thelawchooses_1))) : that ((Misset_1
    Mbold2 thelawchooses_1 Set [(x1_4
       : obj) =>
       ({def} .P77_1 <<= x1_4 : prop)]) Intersection
    .M_1) E Misset_1 Mbold2 thelawchooses_1)]

Lineb4 : [(.M_1 : obj), (Misset_1
    : that Isset (.M_1)), (.thelaw_1
    : [(S_2 : obj) => (--- : obj)]), (thelawchooses_1
    : [(.S_2 : obj), (subsetev_2 : that
       .S_2 <<= .M_1), (inev_2 : that
       Exists ([(x_4 : obj) =>
          ({def} x_4 E .S_2 : prop)])) =>
       (--- : that .thelaw_1 (.S_2) E .S_2)]), (.P77_1
    : obj), (Pev77_1 : that .P77_1 <<=
    .M_1), (Pev277_1 : that Exists ([(x77_3
```

```
      : obj) =>
      ({def} x77_3 E .P77_1 : prop)])) =>
   (--- : that ((Misset_1 Mbold2 thelawchooses_1
   Set [(x1_4 : obj) =>
      ({def} .P77_1 <<= x1_4 : prop)]) Intersection
   .M_1) E Misset_1 Mbold2 thelawchooses_1)]

{move 0}

>>> open

   {move 2}

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

   Line4 : [(.P_1 : obj), (Pev_1
       : that .P_1 <<= M), (Pev2_1 : that
       Exists ([(x2_3 : obj) =>
          ({def} x2_3 E .P_1 : prop)])) =>
       (--- : that ((Misset Mbold2 thelawchooses
       Set [(x1_4 : obj) =>
          ({def} .P_1 <<= x1_4 : 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_3 : obj) =>
          ({def} x2_3 E .P_1 : prop)])) =>
       (--- : that .P_1 <<= Rcal1 (.P_1))]

   {move 1}

   >>> save

   {move 2}

   >>> close

{move 1}

>>> 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_2 : obj), (subsetev_2 : that
       .S_2 <<= .M_1), (inev_2 : that
       Exists ([(x_4 : obj) =>
          ({def} x_4 E .S_2 : prop)])) =>
       (--- : that .thelaw_1 (.S_2) E .S_2)]), (.P77_1
    : obj), (Pev77_1 : that .P77_1 <<=
    .M_1), (Pev277_1 : that Exists ([(x77_3
```

```
   : obj) =>
   ({def} x77_3 E .P77_1 : prop)])) =>
({def} (.P77_1 <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_5 : obj) =>
   ({def} .P77_1 <<= x1_5 : prop)]) Intersection
.M_1) Fixform (.P77_1 <<= (Misset_1
Mbold2 thelawchooses_1 Set [(x1_6
   : obj) =>
   ({def} .P77_1 <<= x1_6 : prop)]) Intersection
.M_1) Fixform Ug ([(z_5 : obj) =>
   ({def} Ded ([(zev2_6 : that
      z_5 E .P77_1) =>
      ({def} (z_5 E (Misset_1 Mbold2
      thelawchooses_1 Set [(x1_10
         : obj) =>
         ({def} .P77_1 <<= x1_10 : prop)]) Intersection
      .M_1) Fixform zev2_6 Mpsubs
      Pev77_1 Conj Ug ([(B_10 : obj) =>
         ({def} Ded ([(Bev_11 : that
            B_10 E Misset_1 Mbold2
            thelawchooses_1 Set [(x1_14
               : obj) =>
               ({def} .P77_1 <<= x1_14
               : prop)]) =>
            ({def} zev2_6 Mpsubs Simp2
            (Bev_11 Iff1 B_10 Ui Separation4
            (Refleq (Misset_1 Mbold2
            thelawchooses_1 Set [(x1_18
               : obj) =>
               ({def} .P77_1 <<= x1_18
               : prop)]))) : that
            z_5 E B_10)]) : that
         (B_10 E Misset_1 Mbold2 thelawchooses_1
         Set [(x1_13 : obj) =>
            ({def} .P77_1 <<= x1_13
            : prop)]) -> z_5 E B_10)]) Iff2
      z_5 Ui Separation4 (Refleq ((Misset_1
      Mbold2 thelawchooses_1 Set [(x1_13
```

```
              : obj) =>
              ({def} .P77_1 <<= x1_13 : prop)]) Intersection
           .M_1)) : that z_5 E (Misset_1
           Mbold2 thelawchooses_1 Set [(x1_9
              : obj) =>
              ({def} .P77_1 <<= x1_9 : prop)]) Intersection
           .M_1)]) : that (z_5 E .P77_1) ->
        z_5 E (Misset_1 Mbold2 thelawchooses_1
        Set [(x1_9 : obj) =>
           ({def} .P77_1 <<= x1_9 : prop)]) Intersection
        .M_1)]) Conj Simp1 (Simp2 (Pev77_1)) Conj
     Separation3 (Refleq ((Misset_1 Mbold2
     thelawchooses_1 Set [(x1_9 : obj) =>
        ({def} .P77_1 <<= x1_9 : prop)]) Intersection
     .M_1)) : that .P77_1 <<= (Misset_1
     Mbold2 thelawchooses_1 Set [(x1_4
        : obj) =>
        ({def} .P77_1 <<= x1_4 : prop)]) Intersection
     .M_1)]

Lineab13 : [(.M_1 : obj), (Misset_1
     : that Isset (.M_1)), (.thelaw_1
     : [(S_2 : obj) => (--- : obj)]), (thelawchooses_1
     : [(.S_2 : obj), (subsetev_2 : that
        .S_2 <<= .M_1), (inev_2 : that
        Exists ([(x_4 : obj) =>
           ({def} x_4 E .S_2 : prop)])) =>
        (--- : that .thelaw_1 (.S_2) E .S_2)]), (.P77_1
     : obj), (Pev77_1 : that .P77_1 <<=
     .M_1), (Pev277_1 : that Exists ([(x77_3
        : obj) =>
        ({def} x77_3 E .P77_1 : prop)])) =>
     (--- : that .P77_1 <<= (Misset_1
     Mbold2 thelawchooses_1 Set [(x1_4
        : obj) =>
        ({def} .P77_1 <<= x1_4 : prop)]) Intersection
     .M_1)]
```

```
   {move 0}

   >>> open

      {move 2}

      >>> define Linea13 Pev Pev2 : Lineab13 \
          Misset, thelawchooses, Pev, Pev2

      Linea13 : [(.P_1 : obj), (Pev_1
          : that .P_1 <<= M), (Pev2_1 : that
          Exists ([(x2_3 : obj) =>
             ({def} x2_3 E .P_1 : prop)])) =>
          (--- : that .P_1 <<= (Misset Mbold2
          thelawchooses Set [(x1_4 : obj) =>
             ({def} .P_1 <<= x1_4 : 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}
end Lestrade execution
```

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

```
begin Lestrade execution

      >>> open

         {move 3}
```

```
>>> 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
 ([(S'_4 : obj) =>
    ({def} thelaw (S'_4) : obj)], Rcal
 (b))) V Rcal (b) <<= Rcal
 (a)

{move 2}

>>> open

   {move 4}

   >>> 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

{move 3}
```

```
      >>> 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

   {move 2}

   >>> define line58 ainm binm : Dediff \
       line56, line57

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

   {move 1}
end Lestrade execution
```

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.

```
begin Lestrade execution
```

```
>>> goal that (a = b) V (a <~ b) V (b <~ \
    a)

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

{move 2}

>>> 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

   {move 3}

   >>> 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

   {move 4}

   >>> 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

  {move 3}

  >>> 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

{move 2}

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

linea67 : [(.a_1 : obj), (.b_1
    : obj), (ainm_1 : that .a_1 E M), (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

    {move 2}

    >>> close

{move 1}

>>> 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_2 : obj), (subsetev_2 : that
       .S_2 <<= .M_1), (inev_2 : that
       Exists ([(x_4 : obj) =>
          ({def} x_4 E .S_2 : prop)])) =>
       (--- : that .thelaw_1 (.S_2) E .S_2)]), (.A77_1
    : obj), (.B77_1 : obj), (ainm77_1
    : that .A77_1 E .M_1), (binm77_1
    : that .B77_1 E .M_1) =>
    ({def} Cases (Excmid (.A77_1 = .B77_1), [(case1_2
       : that .A77_1 = .B77_1) =>
       ({def} (<<<~ (Misset_1, thelawchooses_1, .A77_1, .B77_1) V <<<~
       (Misset_1, thelawchooses_1, .B77_1, .A77_1)) Add1
       case1_2 : that (.A77_1 = .B77_1) V <<<~
       (Misset_1, thelawchooses_1, .A77_1, .B77_1) V <<<~
       (Misset_1, thelawchooses_1, .B77_1, .A77_1))], [(case2_2
       : that ~ (.A77_1 = .B77_1)) =>
       ({def} Cases (Mboldtotal2 (Misset_1, thelawchooses_1, (((Misset_1
       Mbold2 thelawchooses_1 Set [(x1_8
          : obj) =>
          ({def} Usc (.A77_1) <<= x1_8
          : 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_8
          : obj) =>
          ({def} Usc (.B77_1) <<= x1_8
          : 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_3
   : that ((Misset_1 Mbold2 thelawchooses_1
   Set [(x1_7 : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_7 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_7
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_7 : prop)]) Intersection
   .M_1) =>
   ({def} (.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_2
   Conj casea1_3 Iff2 Dediff ([(dir1_11
      : that .B77_1 E (Misset_1
      Mbold2 thelawchooses_1 Set
      [(x1_15 : obj) =>
         ({def} Usc (.A77_1) <<=
         x1_15 : prop)]) Intersection
      .M_1) =>
      ({def} Cases (Mboldstrongtotal2
      (Misset_1, thelawchooses_1, (((Misset_1
      Mbold2 thelawchooses_1 Set
      [(x1_17 : obj) =>
         ({def} Usc (.B77_1) <<=
         x1_17 : 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_17 : obj) =>
         ({def} Usc (.A77_1) <<=
         x1_17 : 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_12
   : that ((Misset_1 Mbold2
   thelawchooses_1 Set [(x1_16
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_16 : prop)]) Intersection
   .M_1) <<= prime2 (.thelaw_1, (Misset_1
   Mbold2 thelawchooses_1
   Set [(x1_17 : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_17 : prop)]) Intersection
   .M_1)) =>
   ({def} (((Misset_1
   Mbold2 thelawchooses_1
   Set [(x1_16 : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_16 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_16
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_16 : prop)]) Intersection
   .M_1) Giveup Subs (Eqsymm
   ((.thelaw_1 ((Misset_1
   Mbold2 thelawchooses_1
   Set [(x1_21 : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_21 : 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_15
      : obj) =>
      ({def} z1_15 E prime2
      (.thelaw_1, (Misset_1
      Mbold2 thelawchooses_1
      Set [(x1_19 : obj) =>
```

```
      ({def} Usc (.B77_1) <<=
      x1_19 : prop)]) Intersection
   .M_1) : prop)], dir1_11
Mpsubs case2_12) Mp primefact3
(Misset_1, thelawchooses_1, (Misset_1
Mbold2 thelawchooses_1
Set [(x1_17 : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_17 : prop)]) Intersection
.M_1) : that ((Misset_1
Mbold2 thelawchooses_1
Set [(x1_15 : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_15 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_15
   : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_15 : prop)]) Intersection
.M_1)], [(case1_12
: that ((Misset_1 Mbold2
thelawchooses_1 Set [(x1_16
   : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_16 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_16
   : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_16 : prop)]) Intersection
.M_1) =>
({def} case1_12 : that
((Misset_1 Mbold2 thelawchooses_1
Set [(x1_15 : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_15 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_15
```

```
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_15 : prop)]) Intersection
   .M_1)]) : that ((Misset_1
Mbold2 thelawchooses_1 Set
[(x1_14 : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_14 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_14
   : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_14 : prop)]) Intersection
.M_1)], [(dir2_11 : that
((Misset_1 Mbold2 thelawchooses_1
Set [(x1_15 : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_15 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_15
   : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_15 : prop)]) Intersection
.M_1) =>
({def} 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_16 : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_16 : prop)]) Intersection
.M_1 Mpsubs dir2_11 : that
.B77_1 E (Misset_1 Mbold2
thelawchooses_1 Set [(x1_14
   : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_14 : 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_3
: that ((Misset_1 Mbold2 thelawchooses_1
Set [(x1_7 : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_7 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_7
   : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_7 : prop)]) Intersection
.M_1) =>
({def} (.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_2) Conj casea2_3 Iff2
Dediff ([(dir1_11 : that .A77_1
   E (Misset_1 Mbold2 thelawchooses_1
   Set [(x1_15 : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_15 : prop)]) Intersection
   .M_1) =>
   ({def} Cases (Mboldstrongtotal2
   (Misset_1, thelawchooses_1, (((Misset_1
   Mbold2 thelawchooses_1 Set
   [(x1_17 : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_17 : 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_17 : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_17 : 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_12
   : that ((Misset_1 Mbold2
   thelawchooses_1 Set [(x1_16
      : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_16 : prop)]) Intersection
   .M_1) <<= prime2 (.thelaw_1, (Misset_1
   Mbold2 thelawchooses_1
   Set [(x1_17 : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_17 : prop)]) Intersection
   .M_1)) =>
   ({def} (((Misset_1
   Mbold2 thelawchooses_1
   Set [(x1_16 : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_16 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_16
      : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_16 : prop)]) Intersection
   .M_1) Giveup Subs (Eqsymm
   ((.thelaw_1 ((Misset_1
   Mbold2 thelawchooses_1
   Set [(x1_21 : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_21 : 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_15
      : obj) =>
      ({def} z1_15 E prime2
      (.thelaw_1, (Misset_1
```

```
   Mbold2 thelawchooses_1
   Set [(x1_19 : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_19 : prop)]) Intersection
   .M_1) : prop)], dir1_11
Mpsubs case2_12) Mp primefact3
(Misset_1, thelawchooses_1, (Misset_1
Mbold2 thelawchooses_1
Set [(x1_17 : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_17 : prop)]) Intersection
.M_1) : that ((Misset_1
Mbold2 thelawchooses_1
Set [(x1_15 : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_15 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_15
   : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_15 : prop)]) Intersection
.M_1)], [(case1_12
: that ((Misset_1 Mbold2
thelawchooses_1 Set [(x1_16
   : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_16 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_16
   : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_16 : prop)]) Intersection
.M_1) =>
({def} case1_12 : that
((Misset_1 Mbold2 thelawchooses_1
Set [(x1_15 : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_15 : prop)]) Intersection
```

```
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_15
      : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_15 : prop)]) Intersection
   .M_1)]) : that ((Misset_1
Mbold2 thelawchooses_1 Set
[(x1_14 : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_14 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_14
   : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_14 : prop)]) Intersection
.M_1)], [(dir2_11 : that
((Misset_1 Mbold2 thelawchooses_1
Set [(x1_15 : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_15 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_15
   : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_15 : prop)]) Intersection
.M_1) =>
({def} 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_16 : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_16 : prop)]) Intersection
.M_1 Mpsubs dir2_11 : that
.A77_1 E (Misset_1 Mbold2
thelawchooses_1 Set [(x1_14
   : obj) =>
```

```
                 ({def} Usc (.B77_1) <<=
                 x1_14 : 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)
           (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))]

   lineb67 : [(.M_1 : obj), (Misset_1
         : that Isset (.M_1)), (.thelaw_1
         : [(S_2 : obj) => (--- : obj)]), (thelawchooses_1
         : [(.S_2 : obj), (subsetev_2 : that
           .S_2 <<= .M_1), (inev_2 : that
           Exists ([(x_4 : obj) =>
             ({def} x_4 E .S_2 : prop)])) =>
           (--- : that .thelaw_1 (.S_2) E .S_2)]), (.A77_1
         : obj), (.B77_1 : obj), (ainm77_1
         : that .A77_1 E .M_1), (binm77_1
         : that .B77_1 E .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))]

   {move 0}
end Lestrade execution
```

The purported order is trichotomous (so total).

```
begin Lestrade execution

   >>> open

      {move 2}

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

```
line67 : [(.a_1 : obj), (.b_1
    : obj), (ainm_1 : that .a_1 E M), (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)

that ~ (a <~ a)

{move 2}

>>> open

   {move 3}

   >>> 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

{move 2}

>>> define line69 ainm : Negintro line68
```

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

      {move 1}
end Lestrade execution
```

The purported order is irreflexive.

```
begin Lestrade execution

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

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

      {move 2}

      >>> open

         {move 3}

         >>> 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

   {move 4}

   >>> 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

   {move 3}

   >>> define line75 thehyp : Negintro \
       line74

   line75 : [(thehyp_1 : that a <~
       b) => (--- : that ~ (b <~
       a))]

   {move 2}

   >>> close

{move 2}

>>> define linea76 ainm binm : Ded \
    line75

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

```
   {move 1}

   >>> save

   {move 2}

   >>> close

{move 1}

>>> 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_2 : obj), (subsetev_2 : that
       .S_2 <<= .M_1), (inev_2 : that
       Exists ([(x_4 : obj) =>
          ({def} x_4 E .S_2 : prop)])) =>
       (--- : that .thelaw_1 (.S_2) E .S_2)]), (.A77_1
    : obj), (.B77_1 : obj), (ainm77_1
    : that .A77_1 E .M_1), (binm77_1
    : that .B77_1 E .M_1) =>
    ({def} Ded ([(thehyp_2 : that <<<~
       (Misset_1, thelawchooses_1, .A77_1, .B77_1)) =>
       ({def} Negintro ([(sillyhyp_3
          : that <<<~ (Misset_1, thelawchooses_1, .B77_1, .A77_1)) =>
          ({def} ((.thelaw_1 ((Misset_1
          Mbold2 thelawchooses_1 Set [(x1_10
             : obj) =>
             ({def} Usc (.A77_1) <<=
             x1_10 : 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_11
   : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_11 : 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_2))) Iff1
Dediff ([(dir1_10 : that .B77_1
   E (Misset_1 Mbold2 thelawchooses_1
   Set [(x1_14 : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_14 : prop)]) Intersection
   .M_1) =>
   ({def} Cases (Mboldstrongtotal2
   (Misset_1, thelawchooses_1, (((Misset_1
   Mbold2 thelawchooses_1 Set
   [(x1_16 : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_16 : 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_16 : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_16 : 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_11
      : that ((Misset_1 Mbold2
      thelawchooses_1 Set [(x1_15
         : obj) =>
         ({def} Usc (.A77_1) <<=
```

```
   x1_15 : prop)]) Intersection
.M_1) <<= prime2 (.thelaw_1, (Misset_1
Mbold2 thelawchooses_1
Set [(x1_16 : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_16 : prop)]) Intersection
.M_1)) =>
({def} (((Misset_1
Mbold2 thelawchooses_1
Set [(x1_15 : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_15 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_15
   : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_15 : prop)]) Intersection
.M_1) Giveup Subs (Eqsymm
((.thelaw_1 ((Misset_1
Mbold2 thelawchooses_1
Set [(x1_20 : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_20 : 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_14
   : obj) =>
   ({def} z1_14 E prime2
   (.thelaw_1, (Misset_1
   Mbold2 thelawchooses_1
   Set [(x1_18 : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_18 : prop)]) Intersection
   .M_1) : prop)], dir1_10
Mpsubs case2_11) Mp primefact3
(Misset_1, thelawchooses_1, (Misset_1
Mbold2 thelawchooses_1
```

```
   Set [(x1_16 : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_16 : prop)]) Intersection
   .M_1) : that ((Misset_1
   Mbold2 thelawchooses_1
   Set [(x1_14 : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_14 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_14
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_14 : prop)]) Intersection
   .M_1)], [(case1_11
   : that ((Misset_1 Mbold2
   thelawchooses_1 Set [(x1_15
      : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_15 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_15
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_15 : prop)]) Intersection
   .M_1) =>
   ({def} case1_11 : that
   ((Misset_1 Mbold2 thelawchooses_1
   Set [(x1_14 : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_14 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_14
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_14 : prop)]) Intersection
   .M_1)]) : that ((Misset_1
Mbold2 thelawchooses_1 Set
[(x1_13 : obj) =>
```

```
      ({def} Usc (.B77_1) <<=
      x1_13 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_13
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_13 : prop)]) Intersection
   .M_1)], [(dir2_10 : that
   ((Misset_1 Mbold2 thelawchooses_1
   Set [(x1_14 : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_14 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_14
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_14 : prop)]) Intersection
   .M_1) =>
   ({def} 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_15 : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_15 : prop)]) Intersection
   .M_1 Mpsubs dir2_10 : that
   .B77_1 E (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_13
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_13 : prop)]) Intersection
   .M_1)]) Antisymsub Simp2
(Simp2 (Simp2 (sillyhyp_3))) Iff1
Dediff ([(dir1_10 : that .A77_1
   E (Misset_1 Mbold2 thelawchooses_1
   Set [(x1_14 : obj) =>
      ({def} Usc (.B77_1) <<=
```

```
   x1_14 : prop)]) Intersection
.M_1) =>
({def} Cases (Mboldstrongtotal2
(Misset_1, thelawchooses_1, (((Misset_1
Mbold2 thelawchooses_1 Set
[(x1_16 : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_16 : 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_16 : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_16 : 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_11
   : that ((Misset_1 Mbold2
   thelawchooses_1 Set [(x1_15
      : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_15 : prop)]) Intersection
   .M_1) <<= prime2 (.thelaw_1, (Misset_1
   Mbold2 thelawchooses_1
   Set [(x1_16 : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_16 : prop)]) Intersection
   .M_1)) =>
   ({def} (((Misset_1
   Mbold2 thelawchooses_1
   Set [(x1_15 : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_15 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_15
```

```
   : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_15 : prop)]) Intersection
.M_1) Giveup Subs (Eqsymm
((.thelaw_1 ((Misset_1
Mbold2 thelawchooses_1
Set [(x1_20 : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_20 : 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_14
   : obj) =>
   ({def} z1_14 E prime2
   (.thelaw_1, (Misset_1
   Mbold2 thelawchooses_1
   Set [(x1_18 : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_18 : prop)]) Intersection
   .M_1) : prop)], dir1_10
Mpsubs case2_11) Mp primefact3
(Misset_1, thelawchooses_1, (Misset_1
Mbold2 thelawchooses_1
Set [(x1_16 : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_16 : prop)]) Intersection
.M_1) : that ((Misset_1
Mbold2 thelawchooses_1
Set [(x1_14 : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_14 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_14
   : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_14 : prop)]) Intersection
.M_1)], [(case1_11
```

```
   : that ((Misset_1 Mbold2
   thelawchooses_1 Set [(x1_15
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_15 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_15
      : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_15 : prop)]) Intersection
   .M_1) =>
   ({def} case1_11 : that
   ((Misset_1 Mbold2 thelawchooses_1
   Set [(x1_14 : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_14 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_14
      : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_14 : prop)]) Intersection
   .M_1)]) : that ((Misset_1
Mbold2 thelawchooses_1 Set
[(x1_13 : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_13 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_13
   : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_13 : prop)]) Intersection
.M_1)], [(dir2_10 : that
((Misset_1 Mbold2 thelawchooses_1
Set [(x1_14 : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_14 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_14
```

```
                  : obj) =>
                  ({def} Usc (.B77_1) <<=
                  x1_14 : prop)]) Intersection
               .M_1) =>
               ({def} 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_15 : obj) =>
                  ({def} Usc (.A77_1) <<=
                  x1_15 : prop)]) Intersection
               .M_1 Mpsubs dir2_10 : that
               .A77_1 E (Misset_1 Mbold2
               thelawchooses_1 Set [(x1_13
                  : obj) =>
                  ({def} Usc (.B77_1) <<=
                  x1_13 : prop)]) Intersection
               .M_1)])) Mp Simp1 (Simp2
            (Simp2 (sillyhyp_3))) : that
            ??)]) : that ~ (<<<~ (Misset_1, thelawchooses_1, .B77_1, .A77_1)))
      <<<~ (Misset_1, thelawchooses_1, .A77_1, .B77_1) ->
      ~ (<<<~ (Misset_1, thelawchooses_1, .B77_1, .A77_1)))]

lineb76 : [(.M_1 : obj), (Misset_1
      : that Isset (.M_1)), (.thelaw_1
      : [(S_2 : obj) => (--- : obj)]), (thelawchooses_1
      : [(.S_2 : obj), (subsetev_2 : that
         .S_2 <<= .M_1), (inev_2 : that
         Exists ([(x_4 : obj) =>
            ({def} x_4 E .S_2 : prop)])) =>
         (--- : that .thelaw_1 (.S_2) E .S_2)]), (.A77_1
      : obj), (.B77_1 : obj), (ainm77_1
      : that .A77_1 E .M_1), (binm77_1
      : that .B77_1 E .M_1) => (--- : that
      <<<~ (Misset_1, thelawchooses_1, .A77_1, .B77_1) ->
      ~ (<<<~ (Misset_1, thelawchooses_1, .B77_1, .A77_1)))]
```

```
   {move 0}

   >>> open

      {move 2}

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

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

      {move 1}
end Lestrade execution
```

The purported order is asymmetric.

```
begin 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
```

```
that ((a <~ b) & b <~ c) -> a <~
 c

{move 2}

>>> open

   {move 3}

   >>> 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

   {move 4}

   >>> 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
```

```
   {move 3}

   >>> 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

{move 2}

>>> define linea84 ainm binm cinm : Ded \
    line83

linea84 : [(.a_1 : obj), (.b_1
    : obj), (ainm_1 : that .a_1 E M), (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

   {move 2}

   >>> close

{move 1}

>>> 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_2 : obj), (subsetev_2 : that
       .S_2 <<= .M_1), (inev_2 : that
       Exists ([(x_4 : obj) =>
          ({def} x_4 E .S_2 : prop)])) =>
       (--- : that .thelaw_1 (.S_2) E .S_2)]), (.A77_1
    : obj), (.B77_1 : obj), (ainm77_1
    : that .A77_1 E .M_1), (binm77_1
    : that .B77_1 E .M_1), (.C77_1 : obj), (cinm77_1
    : that .C77_1 E .M_1) =>
    ({def} Ded ([(thehyp_2 : that <<<~
       (Misset_1, thelawchooses_1, .A77_1, .B77_1) & <<<~
```

```
(Misset_1, thelawchooses_1, .B77_1, .C77_1)) =>
({def} <<<~ (Misset_1, thelawchooses_1, .A77_1, .C77_1) Fixform
ainm77_1 Conj cinm77_1 Conj Negintro
([(sillyhyp_7 : that .A77_1 = .C77_1) =>
   ({def} Eqsymm (sillyhyp_7) Subs1
   Simp2 (thehyp_2) Mp Simp1 (thehyp_2) Mp
   lineb76 (Misset_1, thelawchooses_1, ainm77_1, binm77_1) : that
   ??)]) Conj Simp2 (Simp2 (Simp2
(Simp2 (thehyp_2)))) Iff1
Dediff ([(dir1_10 : that .C77_1
   E (Misset_1 Mbold2 thelawchooses_1
   Set [(x1_14 : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_14 : prop)]) Intersection
   .M_1) =>
   ({def} Cases (Mboldstrongtotal2
   (Misset_1, thelawchooses_1, (((Misset_1
   Mbold2 thelawchooses_1 Set [(x1_16
      : obj) =>
      ({def} Usc (.C77_1) <<=
      x1_16 : 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_16
      : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_16 : 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_11
      : that ((Misset_1 Mbold2
      thelawchooses_1 Set [(x1_15
         : obj) =>
         ({def} Usc (.B77_1) <<=
         x1_15 : prop)]) Intersection
```

```
.M_1) <<= prime2 (.thelaw_1, (Misset_1
Mbold2 thelawchooses_1 Set
[(x1_16 : obj) =>
   ({def} Usc (.C77_1) <<=
   x1_16 : prop)]) Intersection
.M_1)) =>
({def} (((Misset_1 Mbold2
thelawchooses_1 Set [(x1_15
   : obj) =>
   ({def} Usc (.C77_1) <<=
   x1_15 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_15
   : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_15 : prop)]) Intersection
.M_1) Giveup Subs (Eqsymm
((.thelaw_1 ((Misset_1
Mbold2 thelawchooses_1 Set
[(x1_20 : obj) =>
   ({def} Usc (.C77_1) <<=
   x1_20 : 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_14
   : obj) =>
   ({def} z1_14 E prime2
   (.thelaw_1, (Misset_1
   Mbold2 thelawchooses_1
   Set [(x1_18 : obj) =>
      ({def} Usc (.C77_1) <<=
      x1_18 : prop)]) Intersection
   .M_1) : prop)], dir1_10
Mpsubs case2_11) Mp primefact3
(Misset_1, thelawchooses_1, (Misset_1
Mbold2 thelawchooses_1 Set
[(x1_16 : obj) =>
```

```
      ({def} Usc (.C77_1) <<=
      x1_16 : prop)]) Intersection
   .M_1) : that ((Misset_1
   Mbold2 thelawchooses_1 Set
   [(x1_14 : obj) =>
      ({def} Usc (.C77_1) <<=
      x1_14 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_14
      : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_14 : prop)]) Intersection
   .M_1)], [(case1_11 : that
   ((Misset_1 Mbold2 thelawchooses_1
   Set [(x1_15 : obj) =>
      ({def} Usc (.C77_1) <<=
      x1_15 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_15
      : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_15 : prop)]) Intersection
   .M_1) =>
   ({def} case1_11 : that ((Misset_1
   Mbold2 thelawchooses_1 Set
   [(x1_14 : obj) =>
      ({def} Usc (.C77_1) <<=
      x1_14 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_14
      : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_14 : prop)]) Intersection
   .M_1)]) : that ((Misset_1
Mbold2 thelawchooses_1 Set [(x1_13
   : obj) =>
   ({def} Usc (.C77_1) <<=
   x1_13 : prop)]) Intersection
```

```
    .M_1) <<= (Misset_1 Mbold2
    thelawchooses_1 Set [(x1_13
       : obj) =>
       ({def} Usc (.B77_1) <<=
       x1_13 : prop)]) Intersection
    .M_1)], [(dir2_10 : that
    ((Misset_1 Mbold2 thelawchooses_1
    Set [(x1_14 : obj) =>
       ({def} Usc (.C77_1) <<=
       x1_14 : prop)]) Intersection
    .M_1) <<= (Misset_1 Mbold2
    thelawchooses_1 Set [(x1_14
       : obj) =>
       ({def} Usc (.B77_1) <<=
       x1_14 : prop)]) Intersection
    .M_1) =>
    ({def} 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_15 : obj) =>
       ({def} Usc (.C77_1) <<=
       x1_15 : prop)]) Intersection
    .M_1 Mpsubs dir2_10 : that .C77_1
    E (Misset_1 Mbold2 thelawchooses_1
    Set [(x1_13 : obj) =>
       ({def} Usc (.B77_1) <<=
       x1_13 : prop)]) Intersection
    .M_1)]) Transsub Simp2 (Simp2
 (Simp2 (Simp1 (thehyp_2)))) Iff1
 Dediff ([(dir1_10 : that .B77_1
    E (Misset_1 Mbold2 thelawchooses_1
    Set [(x1_14 : obj) =>
       ({def} Usc (.A77_1) <<=
       x1_14 : prop)]) Intersection
    .M_1) =>
    ({def} Cases (Mboldstrongtotal2
    (Misset_1, thelawchooses_1, (((Misset_1
```

```
Mbold2 thelawchooses_1 Set [(x1_16
   : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_16 : 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_16
   : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_16 : 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_11
   : that ((Misset_1 Mbold2
   thelawchooses_1 Set [(x1_15
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_15 : prop)]) Intersection
   .M_1) <<= prime2 (.thelaw_1, (Misset_1
   Mbold2 thelawchooses_1 Set
   [(x1_16 : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_16 : prop)]) Intersection
   .M_1)) =>
   ({def} (((Misset_1 Mbold2
   thelawchooses_1 Set [(x1_15
      : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_15 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_15
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_15 : prop)]) Intersection
   .M_1) Giveup Subs (Eqsymm
```

```
((.thelaw_1 ((Misset_1
Mbold2 thelawchooses_1 Set
[(x1_20 : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_20 : 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_14
   : obj) =>
   ({def} z1_14 E prime2
   (.thelaw_1, (Misset_1
   Mbold2 thelawchooses_1
   Set [(x1_18 : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_18 : prop)]) Intersection
   .M_1) : prop)], dir1_10
Mpsubs case2_11) Mp primefact3
(Misset_1, thelawchooses_1, (Misset_1
Mbold2 thelawchooses_1 Set
[(x1_16 : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_16 : prop)]) Intersection
.M_1) : that ((Misset_1
Mbold2 thelawchooses_1 Set
[(x1_14 : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_14 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_14
   : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_14 : prop)]) Intersection
.M_1)], [(case1_11 : that
((Misset_1 Mbold2 thelawchooses_1
Set [(x1_15 : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_15 : prop)]) Intersection
```

```
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_15
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_15 : prop)]) Intersection
   .M_1) =>
   ({def} case1_11 : that ((Misset_1
   Mbold2 thelawchooses_1 Set
   [(x1_14 : obj) =>
      ({def} Usc (.B77_1) <<=
      x1_14 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_14
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_14 : prop)]) Intersection
   .M_1)]) : that ((Misset_1
Mbold2 thelawchooses_1 Set [(x1_13
   : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_13 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_13
   : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_13 : prop)]) Intersection
.M_1)], [(dir2_10 : that
((Misset_1 Mbold2 thelawchooses_1
Set [(x1_14 : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_14 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_14
   : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_14 : prop)]) Intersection
.M_1) =>
({def} 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_15 : obj) =>
   ({def} Usc (.B77_1) <<=
   x1_15 : prop)]) Intersection
.M_1 Mpsubs dir2_10 : that .B77_1
E (Misset_1 Mbold2 thelawchooses_1
Set [(x1_13 : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_13 : prop)]) Intersection
.M_1)]) Iff2 Dediff ([(dir1_8
: that .C77_1 E (Misset_1 Mbold2
thelawchooses_1 Set [(x1_12
   : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_12 : prop)]) Intersection
.M_1) =>
({def} Cases (Mboldstrongtotal2
(Misset_1, thelawchooses_1, (((Misset_1
Mbold2 thelawchooses_1 Set [(x1_14
   : obj) =>
   ({def} Usc (.C77_1) <<=
   x1_14 : 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_14
   : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_14 : 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_9
   : that ((Misset_1 Mbold2
   thelawchooses_1 Set [(x1_13
```

```
   : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_13 : prop)]) Intersection
.M_1) <<= prime2 (.thelaw_1, (Misset_1
Mbold2 thelawchooses_1 Set
[(x1_14 : obj) =>
   ({def} Usc (.C77_1) <<=
   x1_14 : prop)]) Intersection
.M_1)) =>
({def} (((Misset_1 Mbold2
thelawchooses_1 Set [(x1_13
   : obj) =>
   ({def} Usc (.C77_1) <<=
   x1_13 : prop)]) Intersection
.M_1) <<= (Misset_1 Mbold2
thelawchooses_1 Set [(x1_13
   : obj) =>
   ({def} Usc (.A77_1) <<=
   x1_13 : prop)]) Intersection
.M_1) Giveup Subs (Eqsymm
((.thelaw_1 ((Misset_1
Mbold2 thelawchooses_1 Set
[(x1_18 : obj) =>
   ({def} Usc (.C77_1) <<=
   x1_18 : 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_12
   : obj) =>
   ({def} z1_12 E prime2
   (.thelaw_1, (Misset_1
   Mbold2 thelawchooses_1
   Set [(x1_16 : obj) =>
      ({def} Usc (.C77_1) <<=
      x1_16 : prop)]) Intersection
   .M_1) : prop)], dir1_8
Mpsubs case2_9) Mp primefact3
```

```
   (Misset_1, thelawchooses_1, (Misset_1
   Mbold2 thelawchooses_1 Set
   [(x1_14 : obj) =>
      ({def} Usc (.C77_1) <<=
      x1_14 : prop)]) Intersection
   .M_1) : that ((Misset_1
   Mbold2 thelawchooses_1 Set
   [(x1_12 : obj) =>
      ({def} Usc (.C77_1) <<=
      x1_12 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_12
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_12 : prop)]) Intersection
   .M_1)], [(case1_9 : that
   ((Misset_1 Mbold2 thelawchooses_1
   Set [(x1_13 : obj) =>
      ({def} Usc (.C77_1) <<=
      x1_13 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_13
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_13 : prop)]) Intersection
   .M_1) =>
   ({def} case1_9 : that ((Misset_1
   Mbold2 thelawchooses_1 Set
   [(x1_12 : obj) =>
      ({def} Usc (.C77_1) <<=
      x1_12 : prop)]) Intersection
   .M_1) <<= (Misset_1 Mbold2
   thelawchooses_1 Set [(x1_12
      : obj) =>
      ({def} Usc (.A77_1) <<=
      x1_12 : prop)]) Intersection
   .M_1)]) : that ((Misset_1
Mbold2 thelawchooses_1 Set [(x1_11
```

```
            : obj) =>
            ({def} Usc (.C77_1) <<=
            x1_11 : prop)]) Intersection
         .M_1) <<= (Misset_1 Mbold2
         thelawchooses_1 Set [(x1_11
            : obj) =>
            ({def} Usc (.A77_1) <<=
            x1_11 : prop)]) Intersection
         .M_1)], [(dir2_8 : that
         ((Misset_1 Mbold2 thelawchooses_1
         Set [(x1_12 : obj) =>
            ({def} Usc (.C77_1) <<=
            x1_12 : prop)]) Intersection
         .M_1) <<= (Misset_1 Mbold2
         thelawchooses_1 Set [(x1_12
            : obj) =>
            ({def} Usc (.A77_1) <<=
            x1_12 : prop)]) Intersection
         .M_1) =>
         ({def} 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_13 : obj) =>
            ({def} Usc (.C77_1) <<=
            x1_13 : prop)]) Intersection
         .M_1 Mpsubs dir2_8 : that .C77_1
         E (Misset_1 Mbold2 thelawchooses_1
         Set [(x1_11 : obj) =>
            ({def} Usc (.A77_1) <<=
            x1_11 : prop)]) Intersection
         .M_1)]) : that <<<~ (Misset_1, thelawchooses_1, .A77_1, .C77_1))])
   (<<<~ (Misset_1, thelawchooses_1, .A77_1, .B77_1) & <<<~
   (Misset_1, thelawchooses_1, .B77_1, .C77_1)) ->
   <<<~ (Misset_1, thelawchooses_1, .A77_1, .C77_1))]

lineb84 : [(.M_1 : obj), (Misset_1
   : that Isset (.M_1)), (.thelaw_1
```

```
      : [(S_2 : obj) => (--- : obj)]), (thelawchooses_1
      : [(.S_2 : obj), (subsetev_2 : that
         .S_2 <<= .M_1), (inev_2 : that
         Exists ([(x_4 : obj) =>
            ({def} x_4 E .S_2 : prop)])) =>
         (--- : that .thelaw_1 (.S_2) E .S_2)]), (.A77_1
      : obj), (.B77_1 : obj), (ainm77_1
      : that .A77_1 E .M_1), (binm77_1
      : that .B77_1 E .M_1), (.C77_1 : obj), (cinm77_1
      : that .C77_1 E .M_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

      {move 2}

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

      line84 : [(.a_1 : obj), (.b_1
          : obj), (ainm_1 : that .a_1 E M), (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}
end Lestrade execution
```

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.

```
begin Lestrade execution

      >>> open

         {move 3}

         >>> 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

that chosenof (S) E S

{move 3}

>>> 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

   {move 4}

   >>> declare xx obj

   xx : obj

   {move 4}

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

   that Forall ([(xx : obj) =>
       ({def} (xx E S) -> (xx
       = chosenof (S)) V chosenof
       (S) <~ xx : prop)])
```

```
{move 4}

>>> open

   {move 5}

   >>> 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

      {move 6}

      >>> 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

that Rcal1 (S) = Rcal
 (chosenof (S))

{move 6}

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

line88 : that Rcal1 (S) E Mbold

{move 5}

>>> 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_5 : obj) =>
    ({def} S <<= x1_5 : prop)]) Intersection
 M) E Misset Mbold2 thelawchooses) & (Usc
 (chosenof (S)) <<=
 Rcal1 (S)) & chosenof
 (S) E chosenof (S) ; chosenof
 (S)

{move 5}
end Lestrade execution
```