tail end of section 5, debugging

April 1, 2020

```
begin Lestrade execution

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


                  line90 : that Rcal1 (S) = Rcal
                   (chosenof (S))


                  {move 5}

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


                  line91 : that xx E Rcal
                   (chosenof (S))


                  {move 5}

                  >>> define line92 case2 \
                      : Fixform (chosenof S <~ \
                      xx, (Mpsubs line85 Ssubm \
                      zins Ssubm) Conj (Mpsubs \
                      thehyp Ssubm) Conj (Negeqsymm \
                      case2) Conj line91)
```

```
      line92 : [(case2_1 : that
          ~ (xx = chosenof (S))) =>
          (--- : that chosenof
          (S) <~ xx)]


      {move 5}

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


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


      {move 5}

      >>> close


   {move 5}

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


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


   {move 4}

   >>> close


{move 4}
```

```
   >>> define line95 xx : Ded line94


   line95 : [(xx_1 : obj) =>
       (--- : that (xx_1 E S) ->
       (xx_1 = chosenof (S)) V chosenof
       (S) <~ xx_1)]


   {move 3}

   >>> close


{move 3}

>>> define line96 Ssubm zins : Ug \
    line95


line96 : [(.S_1 : obj), (Ssubm_1
    : that .S_1 <<= M), (.z_1
    : obj), (zins_1 : that .z_1
    E .S_1) => (--- : that Forall
    ([(x''_2 : obj) =>
       ({def} (x''_2 E .S_1) ->
       (x''_2 = chosenof (.S_1)) V chosenof
       (.S_1) <~ x''_2 : prop)]))]


{move 2}

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


line97 : [(.S_1 : obj), (Ssubm_1
    : that .S_1 <<= M), (.z_1
    : obj), (zins_1 : that .z_1
    E .S_1) => (--- : that Exists
    ([(x'_2 : obj) =>
       ({def} (x'_2 E .S_1) & Forall
       ([(x''_4 : obj) =>
          ({def} (x''_4 E .S_1) ->
          (x''_4 = x'_2) V x'_2
```

```
        <~ x''_4 : prop)]) : prop)]))]


{move 2}

>>> open


   {move 4}

   >>> declare x66 obj


   x66 : obj


   {move 4}

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


   thehyp : that (S <<= M) & Exists
    ([(x66_3 : obj) =>
       ({def} x66_3 E S : prop)])


   {move 4}

   >>> open


      {move 5}

      >>> declare y66 obj


      y66 : obj


      {move 5}

      >>> declare yins66 that y66 \
          E S
```

```
   yins66 : that y66 E S


   {move 5}

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


   line98 : [(.y66_1 : obj), (yins66_1
       : that .y66_1 E S) =>
       (--- : that Exists ([(x'_2
          : obj) =>
          ({def} (x'_2 E S) & Forall
          ([(x''_4 : obj) =>
             ({def} (x''_4 E S) ->
             (x''_4 = x'_2) V x'_2
             <~ x''_4 : prop)]) : prop)]))]


   {move 4}

   >>> close


{move 4}

>>> define line99 thehyp : Eg \
    Simp2 thehyp line98


line99 : [(thehyp_1 : that
    (S <<= M) & Exists ([(x66_4
       : obj) =>
       ({def} x66_4 E S : prop)])) =>
    (--- : that Exists ([(x'_2
       : obj) =>
       ({def} (x'_2 E S) & Forall
       ([(x''_4 : obj) =>
          ({def} (x''_4 E S) ->
          (x''_4 = x'_2) V x'_2
          <~ x''_4 : prop)]) : prop)]))]


{move 3}
```

```
      >>> close


   {move 3}

   >>> define line10 S : Ded line99


   line10 : [(S_1 : obj) => (---
       : that ((S_1 <<= M) & Exists
       ([(x66_4 : obj) =>
          ({def} x66_4 E S_1 : prop)])) ->
       Exists ([(x'_3 : obj) =>
          ({def} (x'_3 E S_1) & Forall
          ([(x''_5 : obj) =>
             ({def} (x''_5 E S_1) ->
             (x''_5 = x'_3) V x'_3
             <~ x''_5 : prop)]) : prop)]))]


   {move 2}

   >>> close


{move 2}

>>> define line11 : Ug line10


line11 : that Forall ([(x''_2 : obj) =>
    ({def} ((x''_2 <<= M) & Exists
    ([(x66_5 : obj) =>
       ({def} x66_5 E x''_2 : prop)])) ->
    Exists ([(x'_4 : obj) =>
       ({def} (x'_4 E x''_2) & Forall
       ([(x''_6 : obj) =>
          ({def} (x''_6 E x''_2) ->
          (x''_6 = x'_4) V x'_4 <~
          x''_6 : prop)]) : prop)]) : prop)])


{move 1}

>>> close
```

```
{move 1}

>>> comment the following line will not \
    run until we work on definition expansion \
    control in the text above


{move 1}

>>> define line12 Misset thelawchooses \
    : line11


line12 : [(.M_1 : obj), (Misset_1
    : that Isset (.M_1)), (.thelaw_1
    : [(S_2 : obj) => (--- : obj)]), (thelawchooses_1
    : [(.S_2 : obj), (subsetev_2 : that
       .S_2 <<= .M_1), (inev_2 : that
       Exists ([(x_4 : obj) =>
          ({def} x_4 E .S_2 : prop)])) =>
       (--- : that .thelaw_1 (.S_2) E .S_2)]) =>
    ({def} Ug ([(S_2 : obj) =>
       ({def} Ded ([(thehyp_3 : that
          (S_2 <<= .M_1) & Exists ([(x66_6
             : obj) =>
             ({def} x66_6 E S_2 : prop)])) =>
          ({def} Simp2 (thehyp_3) Eg
          [(.y66_4 : obj), (yins66_4
             : that .y66_4 E S_2) =>
             ({def} .thelaw_1 ((Misset_1
             Mbold2 thelawchooses_1 Set
             [(x1_8 : obj) =>
                ({def} S_2 <<= x1_8 : prop)]) Intersection
             .M_1) Ei1 ((.thelaw_1 ((Misset_1
             Mbold2 thelawchooses_1 Set
             [(x1_11 : obj) =>
                ({def} S_2 <<= x1_11 : prop)]) Intersection
             .M_1) E S_2) Fixform Lineb27
             (Misset_1, thelawchooses_1, Simp1
             (thehyp_3), .y66_4 Ei1
             yins66_4)) Conj Ug ([(xx_7
                : obj) =>
                ({def} Ded ([(thehyp_8
                   : that xx_7 E S_2) =>
                   ({def} Cases (Excmid
```

```
(xx_7 = .thelaw_1 ((Misset_1
Mbold2 thelawchooses_1
Set [(x1_14 : obj) =>
   ({def} S_2 <<= x1_14
   : prop)]) Intersection
.M_1)), [(case1_9
   : that xx_7 = .thelaw_1
   ((Misset_1 Mbold2
   thelawchooses_1 Set
   [(x1_14 : obj) =>
      ({def} S_2 <<=
      x1_14 : prop)]) Intersection
   .M_1)) =>
   ({def} <<<~ (Misset_1, thelawchooses_1, .thelaw_1
   ((Misset_1 Mbold2
   thelawchooses_1 Set
   [(x1_14 : obj) =>
      ({def} S_2 <<=
      x1_14 : prop)]) Intersection
   .M_1), xx_7) Add1
   case1_9 : that (xx_7
   = .thelaw_1 ((Misset_1
   Mbold2 thelawchooses_1
   Set [(x1_14 : obj) =>
      ({def} S_2 <<=
      x1_14 : prop)]) Intersection
   .M_1)) V <<<~ (Misset_1, thelawchooses_1, .thelaw_1
   ((Misset_1 Mbold2
   thelawchooses_1 Set
   [(x1_14 : obj) =>
      ({def} S_2 <<=
      x1_14 : prop)]) Intersection
   .M_1), xx_7))], [(case2_9
   : that ~ (xx_7 = .thelaw_1
   ((Misset_1 Mbold2
   thelawchooses_1 Set
   [(x1_15 : obj) =>
      ({def} S_2 <<=
      x1_15 : prop)]) Intersection
   .M_1))) =>
   ({def} (xx_7 = .thelaw_1
   ((Misset_1 Mbold2
   thelawchooses_1 Set
   [(x1_14 : obj) =>
      ({def} S_2 <<=
      x1_14 : prop)]) Intersection
```

```
.M_1)) Add2 <<<~
(Misset_1, thelawchooses_1, .thelaw_1
((Misset_1 Mbold2
thelawchooses_1 Set
[(x1_15 : obj) =>
   ({def} S_2 <<=
   x1_15 : prop)]) Intersection
.M_1), xx_7) Fixform
((.thelaw_1 ((Misset_1
Mbold2 thelawchooses_1
Set [(x1_18 : obj) =>
   ({def} S_2 <<=
   x1_18 : prop)]) Intersection
.M_1) E S_2) Fixform
Lineb27 (Misset_1, thelawchooses_1, Simp1
(thehyp_3), .y66_4
Ei1 yins66_4)) Mpsubs
Simp1 (thehyp_3) Conj
thehyp_8 Mpsubs Simp1
(thehyp_3) Conj
Negeqsymm (case2_9) Conj
((((Misset_1
Mbold2 thelawchooses_1
Set [(x1_19 : obj) =>
   ({def} S_2 <<=
   x1_19 : prop)]) Intersection
.M_1) = (Misset_1
Mbold2 thelawchooses_1
Set [(x1_19 : obj) =>
   ({def} Usc (.thelaw_1
   ((Misset_1 Mbold2
   thelawchooses_1
   Set [(x1_24
      : obj) =>
      ({def} S_2
      <<= x1_24 : prop)]) Intersection
   .M_1)) <<= x1_19
   : prop)]) Intersection
.M_1) Fixform Lineb41
(Misset_1, thelawchooses_1, ((.thelaw_1
((Misset_1 Mbold2
thelawchooses_1 Set
[(x1_24 : obj) =>
   ({def} S_2 <<=
   x1_24 : prop)]) Intersection
.M_1) E S_2) Fixform
```

```
Lineb27 (Misset_1, thelawchooses_1, Simp1
(thehyp_3), .y66_4
Ei1 yins66_4)) Mpsubs
Simp1 (thehyp_3) Iff2
.thelaw_1 ((Misset_1
Mbold2 thelawchooses_1
Set [(x1_22 : obj) =>
   ({def} S_2 <<=
   x1_22 : prop)]) Intersection
.M_1) Uscsubs .M_1, .thelaw_1
((Misset_1 Mbold2
thelawchooses_1 Set
[(x1_21 : obj) =>
   ({def} S_2 <<=
   x1_21 : prop)]) Intersection
.M_1) Pairinhabited
.thelaw_1 ((Misset_1
Mbold2 thelawchooses_1
Set [(x1_21 : obj) =>
   ({def} S_2 <<=
   x1_21 : prop)]) Intersection
.M_1), Lineb4 (Misset_1, thelawchooses_1, Simp1
(thehyp_3), .y66_4
Ei1 yins66_4) Conj
((.thelaw_1 ((Misset_1
Mbold2 thelawchooses_1
Set [(x1_26 : obj) =>
   ({def} S_2 <<=
   x1_26 : prop)]) Intersection
.M_1) E S_2) Fixform
Lineb27 (Misset_1, thelawchooses_1, Simp1
(thehyp_3), .y66_4
Ei1 yins66_4)) Mpsubs
Lineab13 (Misset_1, thelawchooses_1, Simp1
(thehyp_3), .y66_4
Ei1 yins66_4) Iff2
.thelaw_1 ((Misset_1
Mbold2 thelawchooses_1
Set [(x1_24 : obj) =>
   ({def} S_2 <<=
   x1_24 : prop)]) Intersection
.M_1) Uscsubs (Misset_1
Mbold2 thelawchooses_1
Set [(x1_23 : obj) =>
   ({def} S_2 <<=
   x1_23 : prop)]) Intersection
```

```
      .M_1 Conj Inusc2
      (.thelaw_1 ((Misset_1
      Mbold2 thelawchooses_1
      Set [(x1_23 : obj) =>
         ({def} S_2 <<=
         x1_23 : prop)]) Intersection
      .M_1)))) Subs1
      thehyp_8 Mpsubs Lineab13
      (Misset_1, thelawchooses_1, Simp1
      (thehyp_3), .y66_4
      Ei1 yins66_4) : that
      (xx_7 = .thelaw_1
      ((Misset_1 Mbold2
      thelawchooses_1 Set
      [(x1_14 : obj) =>
         ({def} S_2 <<=
         x1_14 : prop)]) Intersection
      .M_1)) V <<<~ (Misset_1, thelawchooses_1, .thelaw_1
      ((Misset_1 Mbold2
      thelawchooses_1 Set
      [(x1_14 : obj) =>
         ({def} S_2 <<=
         x1_14 : prop)]) Intersection
      .M_1), xx_7))]) : that
   (xx_7 = .thelaw_1 ((Misset_1
   Mbold2 thelawchooses_1
   Set [(x1_13 : obj) =>
      ({def} S_2 <<= x1_13
      : prop)]) Intersection
   .M_1)) V <<<~ (Misset_1, thelawchooses_1, .thelaw_1
   ((Misset_1 Mbold2
   thelawchooses_1 Set
   [(x1_13 : obj) =>
      ({def} S_2 <<= x1_13
      : prop)]) Intersection
   .M_1), xx_7))]) : that
(xx_7 E S_2) -> (xx_7
= .thelaw_1 ((Misset_1
Mbold2 thelawchooses_1
Set [(x1_13 : obj) =>
   ({def} S_2 <<= x1_13
   : prop)]) Intersection
.M_1)) V <<<~ (Misset_1, thelawchooses_1, .thelaw_1
((Misset_1 Mbold2 thelawchooses_1
Set [(x1_13 : obj) =>
   ({def} S_2 <<= x1_13
```

```
                  : prop)]) Intersection
               .M_1), xx_7))]) : that
            Exists ([(x'_5 : obj) =>
               ({def} (x'_5 E S_2) & Forall
               ([(x''_7 : obj) =>
                  ({def} (x''_7 E S_2) ->
                  (x''_7 = x'_5) V <<<~
                  (Misset_1, thelawchooses_1, x'_5, x''_7) : prop)]) : prop)]))] : tl
         Exists ([(x'_4 : obj) =>
            ({def} (x'_4 E S_2) & Forall
            ([(x''_6 : obj) =>
               ({def} (x''_6 E S_2) ->
               (x''_6 = x'_4) V <<<~
               (Misset_1, thelawchooses_1, x'_4, x''_6) : prop)]) : prop)]))]) : tha
      ((S_2 <<= .M_1) & Exists ([(x66_5
         : obj) =>
         ({def} x66_5 E S_2 : prop)])) ->
      Exists ([(x'_4 : obj) =>
         ({def} (x'_4 E S_2) & Forall
         ([(x''_6 : obj) =>
            ({def} (x''_6 E S_2) ->
            (x''_6 = x'_4) V <<<~ (Misset_1, thelawchooses_1, x'_4, x''_6) : prop)])
   Forall ([(x''_2 : obj) =>
      ({def} ((x''_2 <<= .M_1) & Exists
      ([(x66_5 : obj) =>
         ({def} x66_5 E x''_2 : prop)])) ->
      Exists ([(x'_4 : obj) =>
         ({def} (x'_4 E x''_2) & Forall
         ([(x''_6 : obj) =>
            ({def} (x''_6 E x''_2) ->
            (x''_6 = x'_4) V <<<~ (Misset_1, thelawchooses_1, x'_4, x''_6) : prop)])


line12 : [(.M_1 : obj), (Misset_1
    : that Isset (.M_1)), (.thelaw_1
    : [(S_2 : obj) => (--- : obj)]), (thelawchooses_1
    : [(.S_2 : obj), (subsetev_2 : that
       .S_2 <<= .M_1), (inev_2 : that
       Exists ([(x_4 : obj) =>
          ({def} x_4 E .S_2 : prop)])) =>
       (--- : that .thelaw_1 (.S_2) E .S_2)]) =>
    (--- : that Forall ([(x''_2 : obj) =>
       ({def} ((x''_2 <<= .M_1) & Exists
       ([(x66_5 : obj) =>
          ({def} x66_5 E x''_2 : prop)])) ->
       Exists ([(x'_4 : obj) =>
```

```
                   ({def} (x'_4 E x''_2) & Forall
                   ([(x''_6 : obj) =>
                      ({def} (x''_6 E x''_2) ->
                      (x''_6 = x'_4) V <<<~ (Misset_1, thelawchooses_1, x'_4, x''_6) : prop)])

   {move 0}
end Lestrade execution
```

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

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

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