```
zero_rank_valid_or_falsifiable  THM

⊢  ∀L:Sequent List.
|  ∀s∈L.(ρ(s) = 0) ⇒ ∀s∈L.|= s  ∨ (∃a:Assignment. ∃s∈L.a |≠ s)
| by UnivCD
|
1. L : Sequent List
2. ∀s∈L.(ρ(s) = 0)
⊢  ∀s∈L.|= s  ∨ (∃a:Assignment. ∃s∈L.a |≠ s)
| by MkDiscreteEqWith ‘Sequent‘ ‘eqS’ 2
|    THEN
|    MkDiscreteEqWith ‘Formula‘ ‘eqF’ 3
|
2. {Sequent=2}
3. {Formula=2}
4. ∀s∈L.(ρ(s) = 0)
⊢   by Decide ∃s∈L.disjoint(eqF;s.H;s.C)
|
|\
| 5. ∃s∈L.disjoint(eqF;s.H;s.C)
| ⊢   by Choose [2]
| |
| ⊢  ∃a:Assignment. ∃s∈L.a |≠ s
| | by Using
| |    [‘T’,Sequent;‘eq’,eqS;‘P’,(λ2s. disjoint(eqF;s.H;s.C));‘L’,L]
| |    (FwdThru
| |     Thm* ∀eq:{T=2}, P:(T→Prop), L:T List.
| |          (∃x:T. x(∈eq) L & P(x)) ⟺ ∃y∈L.P(y)
| |     {list_exists_is_member_lemma}
| |     [-1])
| |
| 6. ∃x:Sequent. x(∈eqS) L & disjoint(eqF;x.H;x.C)
| ⊢   by Repeat (Analyze -1)
| |
| 6. x : Sequent
| 7. x(∈eqS) L
| 8. disjoint(eqF;x.H;x.C)
| ⊢
| | by Assert
| |    ((λv.if ⌈v⌉(∈eqF) x.H→3_2 ;⌈v⌉(∈eqF) x.C→3_0 else 3_1 fi)
| |     ∈ Assignment)
| |
| |\
| | .....assertion .....
| | ⊢  (λv.if ⌈v⌉(∈eqF) x.H→3_2 ;⌈v⌉(∈eqF) x.C→3_0 else 3_1 fi)
| | |  ∈ Assignment  by Unfold ‘Assignment‘ 0 THEN Ext
| | |
| | 9. x1 : Var
| | ⊢  (λv.if ⌈v⌉(∈eqF) x.H→3_2 ;⌈v⌉(∈eqF) x.C→3_0 else 3_1 fi)(x1) ∈ ℕ3
| |   by Reduce 0
| |
| .\
| . 9. (λv.if ⌈v⌉(∈eqF) x.H→3_2 ;⌈v⌉(∈eqF) x.C→3_0 else 3_1 fi)
| . |  ∈ Assignment
| . ⊢
```

```
| . | by Witness λv.if ⌈v⌉(∈eqF) x.H→3₂ ;⌈v⌉(∈eqF) x.C→3₀ else 3₁ fi
| . |
| . ⊢  ∃s∈L.(λv.if ⌈v⌉(∈eqF) x.H→3₂ ;⌈v⌉(∈eqF) x.C→3₀ else 3₁ fi) |≠ s
| . | by Using [‘x’,x;‘eq’,eqS]
| . |    (BackThru
| . |     Thm* ∀eq:{T=₂}, P:(T→Prop), x:T, L:T List.
| . |          x(∈eq) L ⇒ P(x) ⇒ ∃y∈L.P(y)
| . |     {list_exists_is_member_lemma1})
| . |
| . |\
| . | ⊢  x(∈eqS) L  by Trivial
| . |
| . .\
| . . ⊢  (λv.if ⌈v⌉(∈eqF) x.H→3₂ ;⌈v⌉(∈eqF) x.C→3₀ else 3₁ fi) |≠ x
| . . | by New [‘hyp’;‘concl’] (Analyze -4)
| . . |    THEN
| . . |    OnHyps [-3;-2;-1;0] AbReduce
| . . |
| . . 6. hyp : Formula List
| . . 7. concl : Formula List
| . . 8. <hyp,concl>(∈eqS) L
| . . 9. disjoint(eqF;hyp;concl)
| . . 10. (λv.if ⌈v⌉(∈eqF) hyp→3₂ ;⌈v⌉(∈eqF) concl→3₀ else 3₁ fi)
| . . |   ∈ Assignment
| . . ⊢  (λv.if
| . . |       ⌈v⌉(∈eqF) hyp→3₂
| . . |      ;⌈v⌉(∈eqF) concl→3₀
| . . |      else 3₁ fi) |≠ <hyp,concl>  by Analyze 0 THEN Reduce 0
| . . |
| . . |\
| . . | ⊢  ∀F∈hyp.(λv.if
| . . | |              ⌈v⌉(∈eqF) hyp→3₂
| . . | |             ;⌈v⌉(∈eqF) concl→3₀
| . . | |            else 3₁ fi) |= F
| . . | | by Using [‘eq’,eqF]
| . . | |    (BackThru
| . . | |     Thm* ∀eq:{T=₂}, P:(T→Prop), L:T List.
| . . | |          ∀x∈L.P(x) ⟺ (∀z:T. z(∈eq) L ⇒ P(z))
| . . | |     {list_all_is_member_lemma})
| . . | |
| . . | ⊢  ∀z:Formula.
| . . | |  z(∈eqF) hyp ⇒
| . . | |  (λv.if
| . . | |       ⌈v⌉(∈eqF) hyp→3₂
| . . | |      ;⌈v⌉(∈eqF) concl→3₀
| . . | |      else 3₁ fi) |= z   by UnivCD
| . . | |

| . . | 11. z : Formula
| . . | 12. z(∈eqF) hyp
| . . | ⊢  (λv.if
| . . | |       ⌈v⌉(∈eqF) hyp→3₂
| . . | |      ;⌈v⌉(∈eqF) concl→3₀
| . . | |      else 3₁ fi) |= z
| . . | | by Using
| . . | |    [‘L’,L;‘eqS’,eqS;‘eqF’,eqF;‘hyp’,hyp;‘concl’,concl;‘z’,z]
```

```
| . . | |    (FwdThru
| . . | |     Thm* ∀L:Sequent List, eqS:{Sequent=₂}, eqF:{Formula=₂}.
| . . | |          ∀s∈L.(ρ(s) = 0) ⇒
| . . | |          (∀hyp,concl:Formula List.
| . . | |           <hyp,concl>(∈eqS) L ⇒
| . . | |           (∀z:Formula.
| . . | |            z(∈eqF) hyp ∨ z(∈eqF) concl ⇒
| . . | |            (∃v:Var. z = ⌜v⌝)))
| . . | |     {zero_rank_all_vars}
| . . | |     [8])
| . . | |
| . . | |\
| . . | | .....antecedent .....
| . . | | ⊢  z(∈eqF) hyp ∨ z(∈eqF) concl  by Choose [1] THEN Trivial
| . . | |
| . . | .\
| . . | . 13. ∃v:Var. z = ⌜v⌝
| . . | . ⊢   by Analyze -1
| . . | . |
| . . | . 13. v : Var
| . . | . 14. z = ⌜v⌝
| . . | . ⊢   by HypSubst -1 0
| . . | . |
| . . | . ⊢  (λv.if
| . . | . |       ⌜v⌝(∈eqF) hyp→3₂
| . . | . |      ;⌜v⌝(∈eqF) concl→3₀
| . . | . |      else 3₁ fi) |= ⌜v⌝   by UnfoldTopAb 0 THEN Reduce 0
| . . | . |
| . . | . ⊢  if ⌜v⌝(∈eqF) hyp→3₂ ;⌜v⌝(∈eqF) concl→3₀ else 3₁ fi = 3₂
| . . | . | by FwdThru Thm* ∀P:𝔹. P ⟺ P = true₂    {assert_iff_btrue}
| . . | . |    [-3]
| . . | . |    THEN
| . . | . |    HypSubst -2 -1
| . . | . |    THEN
| . . | . |    HypSubst -1 0
| . . | . |
| . . | . 15. ⌜v⌝(∈eqF) hyp = true₂
| . . | . ⊢  if true₂→3₂ ;⌜v⌝(∈eqF) concl→3₀ else 3₁ fi = 3₂
| . . | .   by Reduce 0
| . . |
| . . .\
| . . . ⊢  ∀F∈concl.(λv.if
| . . . |               ⌜v⌝(∈eqF) hyp→3₂
| . . . |              ;⌜v⌝(∈eqF) concl→3₀
| . . . |              else 3₁ fi) |≠ F
| . . . | by Using [‘eq’,eqF]
| . . . |    (BackThru
| . . . |     Thm* ∀eq:{T=₂}, P:(T→Prop), L:T List.
| . . . |          ∀x∈L.P(x) ⟺ (∀z:T. z(∈eq) L ⇒ P(z))
| . . . |     {list_all_is_member_lemma})
| . . . |
| . . . ⊢  ∀z:Formula.
| . . . |  z(∈eqF) concl ⇒
| . . . |  (λv.if
| . . . |       ⌜v⌝(∈eqF) hyp→3₂
```

```
| . . . |      ;⌜v⌝(∈eqF) concl→3₀
| . . . |      else 3₁ fi) |≠ z  by UnivCD
| . . . |
| . . . 11. z : Formula
| . . . 12. z(∈eqF) concl
| . . . ⊢  (λv.if
| . . . |       ⌜v⌝(∈eqF) hyp→3₂
| . . . |      ;⌜v⌝(∈eqF) concl→3₀
| . . . |      else 3₁ fi) |≠ z
| . . . | by Using
| . . . |    ['L',L;'eqS',eqS;'eqF',eqF;'hyp',hyp;'concl',concl;'z',z]
| . . . |    (FwdThru
| . . . |     Thm* ∀L:Sequent List, eqS:{Sequent=₂}, eqF:{Formula=₂}.
| . . . |          ∀s∈L.(ρ(s) = 0) ⇒
| . . . |          (∀hyp,concl:Formula List.
| . . . |           <hyp,concl>(∈eqS) L ⇒
| . . . |           (∀z:Formula.
| . . . |            z(∈eqF) hyp ∨ z(∈eqF) concl ⇒
| . . . |            (∃v:Var. z = ⌜v⌝)))
| . . . |     {zero_rank_all_vars}
| . . . |     [-5])
| . . . |
| . . . |\
| . . . | .....antecedent .....
| . . . | ⊢  z(∈eqF) hyp ∨ z(∈eqF) concl  by Choose [2] THEN Trivial
| . . . |
| . . . .\
| . . . . 13. ∃v:Var. z = ⌜v⌝
| . . . . ⊢   by Analyze -1
| . . . . |
| . . . . 13. v : Var
| . . . . 14. z = ⌜v⌝
| . . . . ⊢   by HypSubst -1 0
| . . . . |
| . . . . ⊢  (λv.if
| . . . . |       ⌜v⌝(∈eqF) hyp→3₂
| . . . . |      ;⌜v⌝(∈eqF) concl→3₀

| . . . . |      else 3₁ fi) |≠ ⌜v⌝  by UnfoldTopAb 0 THEN AbReduce 0
| . . . . |
| . . . . ⊢  if ⌜v⌝(∈eqF) hyp→3₂ ;⌜v⌝(∈eqF) concl→3₀ else 3₁ fi = 3₀
| . . . . | by FwdThru Thm* ∀P:𝔹. P ⟺ P = true₂    {assert_iff_btrue}
| . . . . |    [-3]
| . . . . |    THEN
| . . . . |    HypSubst -2 -1
| . . . . |    THEN
| . . . . |    HypSubst -1 0
| . . . . |
| . . . . 15. ⌜v⌝(∈eqF) concl = true₂
| . . . . ⊢  if ⌜v⌝(∈eqF) hyp→3₂ ;true₂→3₀ else 3₁ fi = 3₀
| . . . . | by IfThenElseCases 0
| . . . . |
| . . . . |\
| . . . . | .....truecase .....
| . . . . | 16. ⌜v⌝(∈eqF) hyp = true₂
| . . . . | ⊢  3₂ = 3₀
```

```
| . . . . | | by OnHyps [-2;-2]
| . . . . | |    (λi.
| . . . . | |     FwdThru
| . . . . | |     Thm* ∀P:𝔹. P ⟺ P = true₂    {assert_iff_btrue}
| . . . . | |     [i])
| . . . . | |
| . . . . | 17. ⌈v⌉(∈eqF) concl
| . . . . | 18. ⌈v⌉(∈eqF) hyp
| . . . . | ⊢   by Unfold ‘disjoint‘ -10
| . . . . | |
| . . . . | 9. ∀x∈hyp.¬₂x(∈eqF) concl
| . . . . | 10. (λv.if
| . . . . | |       ⌈v⌉(∈eqF) hyp→3₂
| . . . . | |      ;⌈v⌉(∈eqF) concl→3₀
| . . . . | |      else 3₁ fi)
| . . . . | |   ∈ Assignment
| . . . . | 11. z : Formula
| . . . . | 12. z(∈eqF) concl
| . . . . | 13. v : Var
| . . . . | 14. z = ⌈v⌉
| . . . . | 15. ⌈v⌉(∈eqF) concl = true₂
| . . . . | 16. ⌈v⌉(∈eqF) hyp = true₂
| . . . . | 17. ⌈v⌉(∈eqF) concl
| . . . . | 18. ⌈v⌉(∈eqF) hyp
| . . . . | ⊢
| . . . . | | by Using
| . . . . | |    [‘T’,Formula
| . . . . | |    ;‘eq’,eqF
| . . . . | |    ;‘P’,(λ₂x. ¬₂x(∈eqF) concl)
| . . . . | |    ;‘L’,hyp
| . . . . | |    ;‘z’,⌈v⌉]
| . . . . | |    (FwdThru
| . . . . | |     Thm* ∀eq:{T=₂}, P:(T→Prop), L:T List.
| . . . . | |          ∀x∈L.P(x) ⟺ (∀z:T. z(∈eq) L ⇒ P(z))
| . . . . | |     {list_all_is_member_lemma}
| . . . . | |     [-10])
| . . . . | |
| . . . . | 19. ¬₂⌈v⌉(∈eqF) concl
| . . . . | ⊢
| . . . . | | by FwdThru Thm* ∀p:𝔹. ¬₂p ⟺ ¬p    {assert_of_bnot}
| . . . . | |    [-1]
| . . . . | |
| . . . . | 20. ¬⌈v⌉(∈eqF) concl
| . . . . | ⊢   by Analyze -1
| . . . . |
| . . . . .\
| . . . . . .....falsecase .....
| . . . . . 16. ⌈v⌉(∈eqF) hyp = false₂
| . . . . . ⊢  if true₂→ 3₀ else 3₁ fi = 3₀  by Reduce 0
|
.\
. 5. ¬∃s∈L.disjoint(eqF;s.H;s.C)
. ⊢
. | by FwdThru
. |    Thm* ∀P:(T→Prop), L:T List. ¬∃x∈L.P(x) ⟺ ∀x∈L.(¬P(x))
```

```
. |    {not_list_exists_iff_list_all_not}
. |    [-1]
. |
. 6. ∀s∈L.(¬disjoint(eqF;s.H;s.C))
. ⊢   by Thin -2 THEN Choose [1]
. |
. 5. ∀s∈L.(¬disjoint(eqF;s.H;s.C))
. ⊢  ∀s∈L.|= s
. | by Using [‘eq’,eqS]
. |    (BackThru
. |     Thm* ∀eq:{T=₂}, P:(T→Prop), L:T List.
. |          ∀x∈L.P(x) ⟺ (∀z:T. z(∈eq) L ⇒ P(z))
. |     {list_all_is_member_lemma})
. |
. ⊢  ∀z:Sequent. z(∈eqS) L ⇒ |= z   by UnivCD
. |
. 6. z : Sequent
. 7. z(∈eqS) L
. ⊢  |= z
. | by Using
. |    [‘T’,Sequent
. |    ;‘eq’,eqS
. |    ;‘P’,(λ₂s. ¬disjoint(eqF;s.H;s.C))
. |    ;‘L’,L
. |    ;‘z’,z]
. |    (FwdThru
. |     Thm* ∀eq:{T=₂}, P:(T→Prop), L:T List.
. |          ∀x∈L.P(x) ⟺ (∀z:T. z(∈eq) L ⇒ P(z))
. |     {list_all_is_member_lemma}
. |     [-3])
. |
. 8. ¬disjoint(eqF;z.H;z.C)
. ⊢
. | by FwdThru
. |    Thm* ∀eq:{T=₂}, L,M:T List.
. |         ¬disjoint(eq;L;M) ⇒ (∃x:T. x(∈eq) L & x(∈eq) M)
. |    {not_disjoint_is_member_lemma}
. |    [-1]
. |    THEN
. |    Repeat (Analyze -1)
. |
. 9. x : Formula
. 10. x(∈eqF) z.H
. 11. x(∈eqF) z.C
. ⊢   by OnHyps [-4;-4] Thin
. |
. 7. x : Formula
. 8. x(∈eqF) z.H
. 9. x(∈eqF) z.C
. ⊢   by Analyze 0
. |
. 10. a : Full(z)
. ⊢  a |= z  by AbSetHD -1
. |
. 10. a : Assignment
```

```
. 11. a |= z ∨ a |≠ z
. ⊢   by Analyze -1 THENL [Trivial;Id]
. |
. 11. a |≠ z
. ⊢   by Analyze -1
. |
. 11. ∀F∈z.H.a |= F
. 12. ∀F∈z.C.a |≠ F
. ⊢
. | by Using [‘T’,Formula;‘eq’,eqF;‘P’,(λ2F. a |= F );‘L’,z.H;‘z’,x]
. |    (FwdThru
. |     Thm* ∀eq:{T=2}, P:(T→Prop), L:T List.
. |          ∀x∈L.P(x) ⟺ (∀z:T. z(∈eq) L ⇒ P(z))
. |     {list_all_is_member_lemma}
. |     [11])
. |
. 13. a |= x
. ⊢
. | by Using [‘T’,Formula;‘eq’,eqF;‘P’,(λ2F. a |≠ F);‘L’,z.C;‘z’,x]
. |    (FwdThru
. |     Thm* ∀eq:{T=2}, P:(T→Prop), L:T List.
. |          ∀x∈L.P(x) ⟺ (∀z:T. z(∈eq) L ⇒ P(z))
. |     {list_all_is_member_lemma}
. |     [12])
. |
. 14. a |≠ x
. ⊢
. | by FwdThru
. |    Thm* ∀a:Assignment, F:Formula. a |≠ F ⇒ ¬a |= F
. |    {formula_falsifiable_lemma1}
. |    [-1]
. |
. 15. ¬a |= x
. ⊢   by Analyze -1 THEN Trivial
END
```