```
normalization_lemma  THM

⊢  ∀G:Sequent.
|  ∃L:Sequent List.
|  ∀s∈L.(ρ(s) = 0)
|  & (∀s∈L.|= s  ⇒ |= G )
|  & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ G)  by Analyze 0
|
1. G : Sequent
⊢  ∃L:Sequent List.
|  ∀s∈L.(ρ(s) = 0)
|  & (∀s∈L.|= s  ⇒ |= G )
|  & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ G)
| by New [‘S’] (InvImageInd ρ ℕ CompNatInd 1)
|
2. ∀S:Sequent.
|  ρ(S)<ρ(G) ⇒
|  (∃L:Sequent List.
|   ∀s∈L.(ρ(s) = 0)
|   & (∀s∈L.|= s  ⇒ |= S )
|   & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
⊢   by New [‘hyp’;‘concl’] (Analyze 1)
|
1. hyp : Formula List
2. concl : Formula List
3. ∀S:Sequent.
|  ρ(S)<ρ(<hyp,concl>) ⇒
|  (∃L:Sequent List.
|   ∀s∈L.(ρ(s) = 0)
|   & (∀s∈L.|= s  ⇒ |= S )
|   & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
⊢  ∃L:Sequent List.
|  ∀s∈L.(ρ(s) = 0)
|  & (∀s∈L.|= s  ⇒ |= <hyp,concl> )
|  & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <hyp,concl>)
| by Decide ∃f∈hyp.(ρ(f)>0)
|
|\
| 4. ∃f∈hyp.(ρ(f)>0)
| ⊢
| | by FwdThru
| |    Thm* ∀P:(T→Prop), L:T List.
| |         ∃x∈L.P(x) ⟺ (∃M,N:T List, x:T. P(x) & L = (M @ (x.N)))
| |    {list_exists_is_member_append_lemma}
| |    [-1]
| |
| 5. ∃M,N:Formula List, f:Formula. ρ(f)>0 & hyp = (M @ (f.N))
| ⊢   by Repeat (Analyze -1) THEN MoveToEnd 3
| |
| 3. ∃f∈hyp.(ρ(f)>0)
| 4. M : Formula List
| 5. N : Formula List
| 6. f : Formula
| 7. ρ(f)>0
| 8. hyp = (M @ (f.N))
```

```
| 9. ∀S:Sequent.
| |  ρ(S)<ρ(<hyp,concl>) ⇒
| |  (∃L:Sequent List.
| |   ∀s∈L.(ρ(s) = 0)
| |   & (∀s∈L.|= s  ⇒ |= S )
| |   & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
| ⊢   by OnHyps [-1;0] (λi.HypSubst -2 i)
| |
| 9. ∀S:Sequent.
| |  ρ(S)<ρ(<M @ (f.N),concl>) ⇒
| |  (∃L:Sequent List.
| |   ∀s∈L.(ρ(s) = 0)
| |   & (∀s∈L.|= s  ⇒ |= S )
| |   & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
| ⊢  ∃L:Sequent List.
| |  ∀s∈L.(ρ(s) = 0)
| |  & (∀s∈L.|= s  ⇒ |= <M @ (f.N),concl> )
| |  & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <M @ (f.N),concl>)
| | by FormulaCases -4
| |
| |\
| | 7. x : Var
| | 8. ρ(⌈x⌉)>0
| | 9. hyp = (M @ (⌈x⌉.N))
| | 10. ∀S:Sequent.
| |     ρ(S)<ρ(<M @ (⌈x⌉.N),concl>) ⇒
| |     (∃L:Sequent List.
| |      ∀s∈L.(ρ(s) = 0)
| |      & (∀s∈L.|= s  ⇒ |= S )
| |      & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
| | ⊢  ∃L:Sequent List.
| |    ∀s∈L.(ρ(s) = 0)
| |    & (∀s∈L.|= s  ⇒ |= <M @ (⌈x⌉.N),concl> )
| |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <M @ (⌈x⌉.N),concl>)
| |   by Reduce -3
| |
| |\
| | 7. x : Formula
| | 8. ρ(⌈◇⌉x)>0
| | 9. hyp = (M @ (⌈◇⌉x.N))
| | 10. ∀S:Sequent.
| | |   ρ(S)<ρ(<M @ (⌈◇⌉x.N),concl>) ⇒
| | |   (∃L:Sequent List.
| | |    ∀s∈L.(ρ(s) = 0)
| | |    & (∀s∈L.|= s  ⇒ |= S )
| | |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
| | ⊢  ∃L:Sequent List.

| | |  ∀s∈L.(ρ(s) = 0)
| | |  & (∀s∈L.|= s  ⇒ |= <M @ (⌈◇⌉x.N),concl> )
| | |  & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <M @ (⌈◇⌉x.N),concl>)
| | | by With <M @ N,x.concl> (Analyze -1)
| | |
| | 10. ρ(<M @ N,x.concl>)<ρ(<M @ (⌈◇⌉x.N),concl>) ⇒
| | |   (∃L:Sequent List.
| | |    ∀s∈L.(ρ(s) = 0)
```

```
| | |    & (∀s∈L.|= s  ⇒ |= <M @ N,x.concl> )
| | |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <M @ N,x.concl>))
| | ⊢   by Analyze -1
| | |
| | |\
| | | .....antecedent .....
| | | 9. hyp = (M @ (⌈◇⌉x.N))
| | | ⊢  ρ(<M @ N,x.concl>)<ρ(<M @ (⌈◇⌉x.N),concl>)
| | |   by SequentRankReduce 0
| | |
| | .\
| | . 10. ∃L:Sequent List.
| | . |   ∀s∈L.(ρ(s) = 0)
| | . |   & (∀s∈L.|= s  ⇒ |= <M @ N,x.concl> )
| | . |   & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <M @ N,x.concl>)
| | . ⊢   by Repeat (Analyze -1)
| | . |
| | . 10. L : Sequent List
| | . 11. ∀s∈L.(ρ(s) = 0)
| | . 12. ∀s∈L.|= s  ⇒ |= <M @ N,x.concl>
| | . 13. ∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <M @ N,x.concl>
| | . ⊢   by Witness L
| | . |
| | . |\
| | . | 14. ∀s∈L.|= s
| | . | ⊢  |= <M @ (⌈◇⌉x.N),concl>
| | . | | by BackThru
| | . | |    Thm* ∀concl,M,N:Formula List, f2:Formula.
| | . | |          |= <M @ N,f2.concl>  ⇒ |= <M @ (⌈◇⌉f2.N),concl>
| | . | |    {formula_not_left_sound}
| | . | |
| | . | ⊢  |= <M @ N,x.concl>   by HypBackchain
| | . |
| | . .\
| | . . 14. a : Assignment
| | . . 15. ∃s∈L.a |≠ s
| | . . ⊢  a |≠ <M @ (⌈◇⌉x.N),concl>
| | . . | by BackThru
| | . . |    Thm* ∀concl,M,N:Formula List, f2:Formula, a:Assignment.
| | . . |          a |≠ <M @ N,f2.concl> ⟺ a |≠ <M @ (⌈◇⌉f2.N),concl>
| | . . |    {formula_not_left_falsifiable}
| | . . |
| | . . ⊢  a |≠ <M @ N,x.concl>  by HypBackchain
| |
| |\
| | 7. x1 : Formula
| | 8. x2 : Formula
| | 9. ρ(x1⌈∧⌉x2)>0
| | 10. hyp = (M @ (x1⌈∧⌉x2.N))
| | 11. ∀S:Sequent.
| | |   ρ(S)<ρ(<M @ (x1⌈∧⌉x2.N),concl>) ⇒
| | |   (∃L:Sequent List.
| | |    ∀s∈L.(ρ(s) = 0)
| | |    & (∀s∈L.|= s  ⇒ |= S )
| | |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
```

```
| | ⊢  ∃L:Sequent List.
| | |  ∀s∈L.(ρ(s) = 0)
| | |  & (∀s∈L.|= s  ⇒ |= <M @ (x1⌈∧⌉x2.N),concl> )
| | |  & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <M @ (x1⌈∧⌉x2.N),concl>)
| | | by With <[x1; x2/ M @ N],concl> (Analyze -1)
| | |
| | 11. ρ(<[x1; x2/ M @ N],concl>)<ρ(<M @ (x1⌈∧⌉x2.N),concl>) ⇒
| | |    (∃L:Sequent List.
| | |     ∀s∈L.(ρ(s) = 0)
| | |     & (∀s∈L.|= s  ⇒ |= <[x1; x2/ M @ N],concl> )
| | |     & (∀a:Assignment.
| | |        ∃s∈L.a |≠ s ⇒ a |≠ <[x1; x2/ M @ N],concl>))
| | ⊢   by Analyze -1
| | |
| | |\
| | | .....antecedent .....
| | | 10. hyp = (M @ (x1⌈∧⌉x2.N))
| | | ⊢  ρ(<[x1; x2/ M @ N],concl>)<ρ(<M @ (x1⌈∧⌉x2.N),concl>)
| | |   by SequentRankReduce 0
| | |
| | .\
| | . 11. ∃L:Sequent List.
| | . |    ∀s∈L.(ρ(s) = 0)
| | . |    & (∀s∈L.|= s  ⇒ |= <[x1; x2/ M @ N],concl> )
| | . |    & (∀a:Assignment.
| | . |       ∃s∈L.a |≠ s ⇒ a |≠ <[x1; x2/ M @ N],concl>)
| | . ⊢   by Repeat (Analyze -1)
| | . |
| | . 11. L : Sequent List
| | . 12. ∀s∈L.(ρ(s) = 0)
| | . 13. ∀s∈L.|= s  ⇒ |= <[x1; x2/ M @ N],concl>
| | . 14. ∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <[x1; x2/ M @ N],concl>
| | . ⊢   by Witness L
| | . |
| | . |\
| | . | 15. ∀s∈L.|= s
| | . | ⊢  |= <M @ (x1⌈∧⌉x2.N),concl>

| | . | | by BackThru
| | . | |    Thm* ∀concl,M,N:Formula List, q,r:Formula.
| | . | |         |= <[q; r/ M @ N],concl>  ⇒
| | . | |         |= <M @ (q⌈∧⌉r.N),concl>
| | . | |    {formula_and_left_sound}
| | . | |
| | . | ⊢  |= <[x1; x2/ M @ N],concl>   by HypBackchain
| | . |
| | . .\
| | . . 15. a : Assignment
| | . . 16. ∃s∈L.a |≠ s
| | . . ⊢  a |≠ <M @ (x1⌈∧⌉x2.N),concl>
| | . . | by BackThru
| | . . |    Thm* ∀concl,M,N:Formula List, q,r:Formula, a:Assignment.
| | . . |         a |≠ <[q; r/ M @ N],concl>
| | . . |         ⟺
| | . . |         a |≠ <M @ (q⌈∧⌉r.N),concl>
| | . . |    {formula_and_left_falsifiable}
```

```
| | . . |
| | . . ⊢  a |≠ <[x1; x2/ M @ N],concl>  by HypBackchain
| |
| |\
| | 7. x1 : Formula
| | 8. x2 : Formula
| | 9. ρ(x1⌈∨⌉x2)>0
| | 10. hyp = (M @ (x1⌈∨⌉x2.N))
| | 11. ∀S:Sequent.
| | |   ρ(S)<ρ(<M @ (x1⌈∨⌉x2.N),concl>) ⇒
| | |   (∃L:Sequent List.
| | |    ∀s∈L.(ρ(s) = 0)
| | |    & (∀s∈L.|= s  ⇒ |= S )
| | |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
| | ⊢  ∃L:Sequent List.
| | |  ∀s∈L.(ρ(s) = 0)
| | |  & (∀s∈L.|= s  ⇒ |= <M @ (x1⌈∨⌉x2.N),concl> )
| | |  & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <M @ (x1⌈∨⌉x2.N),concl>)
| | | by CopyToHyp -1 -1
| | |
| | 11. ∀S:Sequent.
| | |   ρ(S)<ρ(<M @ (x1⌈∨⌉x2.N),concl>) ⇒
| | |   (∃L:Sequent List.
| | |    ∀s∈L.(ρ(s) = 0)
| | |    & (∀s∈L.|= s  ⇒ |= S )
| | |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
| | 12. ∀S:Sequent.
| | |   ρ(S)<ρ(<M @ (x1⌈∨⌉x2.N),concl>) ⇒
| | |   (∃L:Sequent List.
| | |    ∀s∈L.(ρ(s) = 0)
| | |    & (∀s∈L.|= s  ⇒ |= S )
| | |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
| | ⊢
| | | by With <x1.(M @ N),concl> (Analyze -2)
| | |    THEN
| | |    With <x2.(M @ N),concl> (Analyze -2)
| | |
| | 11. ρ(<x1.(M @ N),concl>)<ρ(<M @ (x1⌈∨⌉x2.N),concl>) ⇒
| | |   (∃L:Sequent List.
| | |    ∀s∈L.(ρ(s) = 0)
| | |    & (∀s∈L.|= s  ⇒ |= <x1.(M @ N),concl> )
| | |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <x1.(M @ N),concl>))
| | 12. ρ(<x2.(M @ N),concl>)<ρ(<M @ (x1⌈∨⌉x2.N),concl>) ⇒
| | |   (∃L:Sequent List.
| | |    ∀s∈L.(ρ(s) = 0)
| | |    & (∀s∈L.|= s  ⇒ |= <x2.(M @ N),concl> )
| | |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <x2.(M @ N),concl>))
| | ⊢   by Analyze -2
| | |
| | |\
| | | .....antecedent .....
| | | 11. ρ(<x2.(M @ N),concl>)<ρ(<M @ (x1⌈∨⌉x2.N),concl>) ⇒
| | |     (∃L:Sequent List.
| | |      ∀s∈L.(ρ(s) = 0)
| | |      & (∀s∈L.|= s  ⇒ |= <x2.(M @ N),concl> )
```

```
| | |      & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <x2.(M @ N),concl>))
| | | ⊢  ρ(<x1.(M @ N),concl>)<ρ(<M @ (x1⌈∨⌉x2.N),concl>)
| | |   by SequentRankReduce 0 THEN GenConclOnAps
| | |
| | .\
| | . 11. ρ(<x2.(M @ N),concl>)<ρ(<M @ (x1⌈∨⌉x2.N),concl>) ⇒
| | . |   (∃L:Sequent List.
| | . |    ∀s∈L.(ρ(s) = 0)
| | . |     & (∀s∈L.|= s  ⇒ |= <x2.(M @ N),concl> )
| | . |     & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <x2.(M @ N),concl>))
| | . 12. ∃L:Sequent List.
| | . |   ∀s∈L.(ρ(s) = 0)
| | . |   & (∀s∈L.|= s  ⇒ |= <x1.(M @ N),concl> )
| | . |   & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <x1.(M @ N),concl>)
| | . ⊢   by Analyze -2
| | . |
| | . |\
| | . | .....antecedent .....
| | . | 11. ∃L:Sequent List.
| | . |     ∀s∈L.(ρ(s) = 0)
| | . |     & (∀s∈L.|= s  ⇒ |= <x1.(M @ N),concl> )
| | . |     & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <x1.(M @ N),concl>)
| | . | ⊢  ρ(<x2.(M @ N),concl>)<ρ(<M @ (x1⌈∨⌉x2.N),concl>)
| | . |   by SequentRankReduce 0 THEN GenConclOnAps
| | . |
| | . .\
| | . . 11. ∃L:Sequent List.
| | . . |   ∀s∈L.(ρ(s) = 0)
| | . . |   & (∀s∈L.|= s  ⇒ |= <x1.(M @ N),concl> )
| | . . |   & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <x1.(M @ N),concl>)
| | . . 12. ∃L:Sequent List.
| | . . |   ∀s∈L.(ρ(s) = 0)
| | . . |   & (∀s∈L.|= s  ⇒ |= <x2.(M @ N),concl> )
| | . . |   & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <x2.(M @ N),concl>)
| | . . ⊢   by Repeat (Analyze -2) THEN Repeat (Analyze -1)
| | . . |
| | . . 11. L : Sequent List
| | . . 12. ∀s∈L.(ρ(s) = 0)
| | . . 13. ∀s∈L.|= s  ⇒ |= <x1.(M @ N),concl>
| | . . 14. ∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <x1.(M @ N),concl>
| | . . 15. L1 : Sequent List
| | . . 16. ∀s∈L1.(ρ(s) = 0)
| | . . 17. ∀s∈L1.|= s  ⇒ |= <x2.(M @ N),concl>
| | . . 18. ∀a:Assignment. ∃s∈L1.a |≠ s ⇒ a |≠ <x2.(M @ N),concl>
| | . . ⊢   by Witness L @ L1
| | . . |
| | . . |\
| | . . | ⊢  ∀s∈(L @ L1).(ρ(s) = 0)
| | . . |   by BackThru
| | . . |      Thm* ∀P:(T→Prop), L,M:T List.
| | . . |           ∀x∈(L @ M).P(x) ⟺ ∀x∈L.P(x) & ∀x∈M.P(x)
| | . . |      {list_all_append_lemma}
| | . . |
| | . . |\
| | . . | 19. ∀s∈(L @ L1).|= s
```

```
| | . . | ⊢  |= <M @ (x1⌈∨⌉x2.N),concl>
| | . . | | by FwdThru
| | . . | |    Thm* ∀P:(T→Prop), L,M:T List.
| | . . | |         ∀x∈(L @ M).P(x) ⟺ ∀x∈L.P(x) & ∀x∈M.P(x)
| | . . | |    {list_all_append_lemma}
| | . . | |    [-1]
| | . . | |    THEN
| | . . | |    Sel 2
| | . . | |    (FwdThru
| | . . | |     Thm* ∀P:(T→Prop), L,M:T List.
| | . . | |          ∀x∈(L @ M).P(x) ⟺ ∀x∈L.P(x) & ∀x∈M.P(x)
| | . . | |     {list_all_append_lemma}
| | . . | |     [-2])
| | . . | |
| | . . | 20. ∀s∈L.|= s
| | . . | 21. ∀s∈L1.|= s
| | . . | ⊢
| | . . |   by BackThru
| | . . |      Thm* ∀concl,M,N:Formula List, q,r:Formula.
| | . . |           |= <q.(M @ N),concl>  ⇒
| | . . |           |= <r.(M @ N),concl>  ⇒ |= <M @ (q⌈∨⌉r.N),concl>
| | . . |      {formula_or_left_sound}
| | . . |      THEN
| | . . |      HypBackchain
| | . . |
| | . . .\
| | . . . 19. a : Assignment
| | . . . 20. ∃s∈(L @ L1).a |≠ s
| | . . . ⊢  a |≠ <M @ (x1⌈∨⌉x2.N),concl>
| | . . . | by BackThru
| | . . . |    Thm* ∀concl,M,N:Formula List, q,r:Formula
| | . . . |         , a:Assignment.
| | . . . |         a |≠ <q.(M @ N),concl> ∨ a |≠ <r.(M @ N),concl>
| | . . . |         ⟺
| | . . . |         a |≠ <M @ (q⌈∨⌉r.N),concl>
| | . . . |    {formula_or_left_falsifiable}
| | . . . |
| | . . . ⊢  a |≠ <x1.(M @ N),concl> ∨ a |≠ <x2.(M @ N),concl>
| | . . . | by FwdThru
| | . . . |    Thm* ∀P:(T→Prop), L,M:T List.
| | . . . |         ∃x∈(L @ M).P(x) ⟺ ∃x∈L.P(x) ∨ ∃x∈M.P(x)
| | . . . |    {list_exists_append_lemma}
| | . . . |    [-1]
| | . . . |    THEN
| | . . . |    Analyze -1
| | . . . |
| | . . . |\
| | . . . | 21. ∃s∈L.a |≠ s
| | . . . | ⊢   by Choose [1] THEN HypBackchain
| | . . . |
| | . . . .\
| | . . . . 21. ∃s∈L1.a |≠ s
| | . . . . ⊢   by Choose [2] THEN HypBackchain
| |
| .\
```

```
| . 7. y1 : Formula
| . 8. y2 : Formula
| . 9. ρ(y1⌈⇒⌉y2)>0
| . 10. hyp = (M @ (y1⌈⇒⌉y2.N))
| . 11. ∀S:Sequent.
| . |    ρ(S)<ρ(<M @ (y1⌈⇒⌉y2.N),concl>) ⇒
| . |    (∃L:Sequent List.
| . |     ∀s∈L.(ρ(s) = 0)
| . |     & (∀s∈L.|= s  ⇒  |= S )
| . |     & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
| . ⊢  ∃L:Sequent List.
| . |  ∀s∈L.(ρ(s) = 0)
| . |  & (∀s∈L.|= s  ⇒  |= <M @ (y1⌈⇒⌉y2.N),concl> )
| . |  & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <M @ (y1⌈⇒⌉y2.N),concl>)
| . | by CopyToHyp -1 -1
| . |
| . 11. ∀S:Sequent.

| . |    ρ(S)<ρ(<M @ (y1⌈⇒⌉y2.N),concl>) ⇒
| . |    (∃L:Sequent List.
| . |     ∀s∈L.(ρ(s) = 0)
| . |     & (∀s∈L.|= s  ⇒  |= S )
| . |     & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
| . 12. ∀S:Sequent.
| . |    ρ(S)<ρ(<M @ (y1⌈⇒⌉y2.N),concl>) ⇒
| . |    (∃L:Sequent List.
| . |     ∀s∈L.(ρ(s) = 0)
| . |     & (∀s∈L.|= s  ⇒  |= S )
| . |     & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
| . ⊢
| . | by With <y2.(M @ N),concl> (Analyze -2)
| . |     THEN
| . |     With <M @ N,y1.concl> (Analyze -2)
| . |
| . 11. ρ(<y2.(M @ N),concl>)<ρ(<M @ (y1⌈⇒⌉y2.N),concl>) ⇒
| . |    (∃L:Sequent List.
| . |     ∀s∈L.(ρ(s) = 0)
| . |     & (∀s∈L.|= s  ⇒  |= <y2.(M @ N),concl> )
| . |     & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <y2.(M @ N),concl>))
| . 12. ρ(<M @ N,y1.concl>)<ρ(<M @ (y1⌈⇒⌉y2.N),concl>) ⇒
| . |    (∃L:Sequent List.
| . |     ∀s∈L.(ρ(s) = 0)
| . |     & (∀s∈L.|= s  ⇒  |= <M @ N,y1.concl> )
| . |     & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <M @ N,y1.concl>))
| . ⊢   by Analyze -2
| . |
| . |\
| . | .....antecedent .....
| . | 11. ρ(<M @ N,y1.concl>)<ρ(<M @ (y1⌈⇒⌉y2.N),concl>) ⇒
| . |      (∃L:Sequent List.
| . |       ∀s∈L.(ρ(s) = 0)
| . |       & (∀s∈L.|= s  ⇒  |= <M @ N,y1.concl> )
| . |       & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <M @ N,y1.concl>))
| . | ⊢  ρ(<y2.(M @ N),concl>)<ρ(<M @ (y1⌈⇒⌉y2.N),concl>)
| . |   by SequentRankReduce 0 THEN GenConclOnAps
| . |
```

```
| . .\
| . . 11. ρ(<M @ N,y1.concl>)<ρ(<M @ (y1⌜⇒⌝y2.N),concl>) ⇒
| . . |   (∃L:Sequent List.
| . . |    ∀s∈L.(ρ(s) = 0)
| . . |    & (∀s∈L.|= s  ⇒ |= <M @ N,y1.concl> )
| . . |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <M @ N,y1.concl>))
| . . 12. ∃L:Sequent List.
| . . |   ∀s∈L.(ρ(s) = 0)
| . . |   & (∀s∈L.|= s  ⇒ |= <y2.(M @ N),concl> )
| . . |   & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <y2.(M @ N),concl>)
| . . ⊢   by Analyze -2
| . . |
| . . |\
| . . | .....antecedent .....
| . . | 11. ∃L:Sequent List.
| . . |     ∀s∈L.(ρ(s) = 0)
| . . |     & (∀s∈L.|= s  ⇒ |= <y2.(M @ N),concl> )
| . . |     & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <y2.(M @ N),concl>)
| . . | ⊢  ρ(<M @ N,y1.concl>)<ρ(<M @ (y1⌜⇒⌝y2.N),concl>)
| . . |   by SequentRankReduce 0 THEN GenConclOnAps
| . . |
| . . .\
| . . . 11. ∃L:Sequent List.
| . . . |   ∀s∈L.(ρ(s) = 0)
| . . . |   & (∀s∈L.|= s  ⇒ |= <y2.(M @ N),concl> )
| . . . |   & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <y2.(M @ N),concl>)
| . . . 12. ∃L:Sequent List.
| . . . |   ∀s∈L.(ρ(s) = 0)
| . . . |   & (∀s∈L.|= s  ⇒ |= <M @ N,y1.concl> )
| . . . |   & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <M @ N,y1.concl>)
| . . . ⊢   by Repeat (Analyze -2) THEN Repeat (Analyze -1)
| . . . |
| . . . 11. L : Sequent List
| . . . 12. ∀s∈L.(ρ(s) = 0)
| . . . 13. ∀s∈L.|= s  ⇒ |= <y2.(M @ N),concl>
| . . . 14. ∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <y2.(M @ N),concl>
| . . . 15. L1 : Sequent List
| . . . 16. ∀s∈L1.(ρ(s) = 0)
| . . . 17. ∀s∈L1.|= s  ⇒ |= <M @ N,y1.concl>
| . . . 18. ∀a:Assignment. ∃s∈L1.a |≠ s ⇒ a |≠ <M @ N,y1.concl>
| . . . ⊢   by Witness L @ L1
| . . . |
| . . . |\
| . . . | ⊢  ∀s∈(L @ L1).(ρ(s) = 0)
| . . . |   by BackThru
| . . . |      Thm* ∀P:(T→Prop), L,M:T List.
| . . . |           ∀x∈(L @ M).P(x) ⟺ ∀x∈L.P(x) & ∀x∈M.P(x)
| . . . |      {list_all_append_lemma}
| . . . |
| . . . |\
| . . . | 19. ∀s∈(L @ L1).|= s
| . . . | ⊢  |= <M @ (y1⌜⇒⌝y2.N),concl>
| . . . | | by FwdThru
| . . . | |    Thm* ∀P:(T→Prop), L,M:T List.
| . . . | |         ∀x∈(L @ M).P(x) ⟺ ∀x∈L.P(x) & ∀x∈M.P(x)
```

```
| . . . | |    {list_all_append_lemma}
| . . . | |    [-1]
| . . . | |    THEN
| . . . | |    Sel 2
| . . . | |    (FwdThru
| . . . | |     Thm* ∀P:(T→Prop), L,M:T List.
| . . . | |           ∀x∈(L @ M).P(x) ⟺ ∀x∈L.P(x) & ∀x∈M.P(x)
| . . . | |     {list_all_append_lemma}
| . . . | |     [-2])
| . . . | |
| . . . | 20. ∀s∈L.|= s
| . . . | 21. ∀s∈L1.|= s
| . . . | ⊢
| . . . |   by BackThru
| . . . |      Thm* ∀concl,M,N:Formula List, q,r:Formula.
| . . . |           |= <M @ N,q.concl>  ⇒
| . . . |           |= <r.(M @ N),concl>  ⇒
| . . . |           |= <M @ (q⌈⇒⌉r.N),concl>
| . . . |      {formula_imp_left_sound}
| . . . |      THEN
| . . . |      HypBackchain
| . . . |
| . . . .\
| . . . . 19. a : Assignment
| . . . . 20. ∃s∈(L @ L1).a |≠ s
| . . . . ⊢  a |≠ <M @ (y1⌈⇒⌉y2.N),concl>
| . . . . | by BackThru
| . . . . |    Thm* ∀concl,M,N:Formula List, q,r:Formula
| . . . . |         , a:Assignment.
| . . . . |         a |≠ <r.(M @ N),concl> ∨ a |≠ <M @ N,q.concl>
| . . . . |         ⟺
| . . . . |         a |≠ <M @ (q⌈⇒⌉r.N),concl>
| . . . . |    {formula_imp_left_falsifiable}
| . . . . |
| . . . . ⊢  a |≠ <y2.(M @ N),concl> ∨ a |≠ <M @ N,y1.concl>
| . . . . | by FwdThru
| . . . . |    Thm* ∀P:(T→Prop), L,M:T List.
| . . . . |         ∃x∈(L @ M).P(x) ⟺ ∃x∈L.P(x) ∨ ∃x∈M.P(x)
| . . . . |    {list_exists_append_lemma}
| . . . . |    [-1]
| . . . . |    THEN
| . . . . |    Analyze -1
| . . . . |
| . . . . |\
| . . . . | 21. ∃s∈L.a |≠ s
| . . . . | ⊢   by Choose [1] THEN HypBackchain
| . . . . |
| . . . . .\
| . . . . . 21. ∃s∈L1.a |≠ s
| . . . . . ⊢   by Choose [2] THEN HypBackchain
|
.\
. 4. ¬∃f∈hyp.(ρ(f)>0)
. ⊢   by Decide ∃f∈concl.(ρ(f)>0)
. |
```

```
. |\
. | 5. ∃f∈concl.(ρ(f)>0)
. | ⊢
. | | by Thin -2
. | |    THEN
. | |    FwdThru
. | |    Thm* ∀P:(T→Prop), L:T List.
. | |         ∃x∈L.P(x) ⟺ (∃M,N:T List, x:T. P(x) & L = (M @ (x.N)))
. | |    {list_exists_is_member_append_lemma}
. | |    [-1]
. | |    THEN
. | |    Repeat (Analyze -1)
. | |
. | 4. ∃f∈concl.(ρ(f)>0)
. | 5. M : Formula List
. | 6. N : Formula List
. | 7. f : Formula
. | 8. ρ(f)>0
. | 9. concl = (M @ (f.N))
. | ⊢   by MoveToEnd 3 THEN OnHyps [-1;0] (λi.HypSubst -2 i)
. | |
. | 3. ∃f∈concl.(ρ(f)>0)
. | 4. M : Formula List
. | 5. N : Formula List
. | 6. f : Formula
. | 7. ρ(f)>0
. | 8. concl = (M @ (f.N))
. | 9. ∀S:Sequent.
. | |  ρ(S)<ρ(<hyp,M @ (f.N)>) ⇒
. | |  (∃L:Sequent List.
. | |   ∀s∈L.(ρ(s) = 0)
. | |   & (∀s∈L.|= s  ⇒ |= S )
. | |   & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
. | ⊢  ∃L:Sequent List.
. | |  ∀s∈L.(ρ(s) = 0)
. | |  & (∀s∈L.|= s  ⇒ |= <hyp,M @ (f.N)> )
. | |  & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <hyp,M @ (f.N)>)
. | | by FormulaCases -4
. | |
. | |\
. | | 7. x : Var
. | | 8. ρ(⌈x⌉)>0
. | | 9. concl = (M @ (⌈x⌉.N))
. | | 10. ∀S:Sequent.
. | |     ρ(S)<ρ(<hyp,M @ (⌈x⌉.N)>) ⇒
. | |     (∃L:Sequent List.
. | |      ∀s∈L.(ρ(s) = 0)
. | |      & (∀s∈L.|= s  ⇒ |= S )
. | |      & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
. | | ⊢  ∃L:Sequent List.
. | |    ∀s∈L.(ρ(s) = 0)
. | |    & (∀s∈L.|= s  ⇒ |= <hyp,M @ (⌈x⌉.N)> )
. | |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <hyp,M @ (⌈x⌉.N)>)
. | |   by Reduce -3
. | |
```

```
. | |\
. | | 7. x : Formula
. | | 8. ρ(⌈◇⌉x)>0
. | | 9. concl = (M @ (⌈◇⌉x.N))
. | | 10. ∀S:Sequent.
. | | |    ρ(S)<ρ(<hyp,M @ (⌈◇⌉x.N)>) ⇒
. | | |    (∃L:Sequent List.
. | | |     ∀s∈L.(ρ(s) = 0)
. | | |     & (∀s∈L.|= s  ⇒ |= S )
. | | |     & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
. | | ⊢  ∃L:Sequent List.
. | | |  ∀s∈L.(ρ(s) = 0)
. | | |  & (∀s∈L.|= s  ⇒ |= <hyp,M @ (⌈◇⌉x.N)> )
. | | |  & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <hyp,M @ (⌈◇⌉x.N)>)
. | | | by With <x.hyp,M @ N> (Analyze -1)
. | | |
. | | 10. ρ(<x.hyp,M @ N>)<ρ(<hyp,M @ (⌈◇⌉x.N)>) ⇒
. | | |    (∃L:Sequent List.
. | | |     ∀s∈L.(ρ(s) = 0)
. | | |     & (∀s∈L.|= s  ⇒ |= <x.hyp,M @ N> )
. | | |     & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <x.hyp,M @ N>))
. | | ⊢   by Analyze -1
. | | |
. | | |\
. | | | .....antecedent .....
. | | | 9. concl = (M @ (⌈◇⌉x.N))
. | | | ⊢  ρ(<x.hyp,M @ N>)<ρ(<hyp,M @ (⌈◇⌉x.N)>)
. | | |   by SequentRankReduce 0
. | | |
. | | .\
. | | . 10. ∃L:Sequent List.
. | | . |   ∀s∈L.(ρ(s) = 0)
. | | . |   & (∀s∈L.|= s  ⇒ |= <x.hyp,M @ N> )
. | | . |   & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <x.hyp,M @ N>)
. | | . ⊢   by Repeat (Analyze -1)
. | | . |
. | | . 10. L : Sequent List
. | | . 11. ∀s∈L.(ρ(s) = 0)
. | | . 12. ∀s∈L.|= s  ⇒ |= <x.hyp,M @ N>
. | | . 13. ∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <x.hyp,M @ N>
. | | . ⊢   by Witness L
. | | . |
. | | . |\
. | | . | 14. ∀s∈L.|= s
. | | . | ⊢  |= <hyp,M @ (⌈◇⌉x.N)>
. | | . | | by BackThru
. | | . | |    Thm* ∀hyp,M,N:Formula List, f2:Formula.
. | | . | |          |= <f2.hyp,M @ N>  ⇒ |= <hyp,M @ (⌈◇⌉f2.N)>
. | | . | |    {formula_not_right_sound}
. | | . | |
. | | . | ⊢  |= <x.hyp,M @ N>   by HypBackchain
. | | . |
. | | . .\
. | | . . 14. a : Assignment
. | | . . 15. ∃s∈L.a |≠ s
```

```
. | | . . ⊢  a |≠ <hyp,M @ (⌈◇⌉x.N)>
. | | . . | by BackThru
. | | . . |    Thm* ∀hyp,M,N:Formula List, f2:Formula, a:Assignment.
. | | . . |         a |≠ <f2.hyp,M @ N> ⟺ a |≠ <hyp,M @ (⌈◇⌉f2.N)>
. | | . . |    {formula_not_right_falsifiable}
. | | . . |
. | | . . ⊢  a |≠ <x.hyp,M @ N>  by HypBackchain
. | |
. | |\
. | | 7. x1 : Formula
. | | 8. x2 : Formula
. | | 9. ρ(x1⌈∧⌉x2)>0
. | | 10. concl = (M @ (x1⌈∧⌉x2.N))
. | | 11. ∀S:Sequent.
. | | |   ρ(S)<ρ(<hyp,M @ (x1⌈∧⌉x2.N)>) ⇒
. | | |   (∃L:Sequent List.
. | | |    ∀s∈L.(ρ(s) = 0)
. | | |    & (∀s∈L.|= s  ⇒ |= S )
. | | |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
. | | ⊢  ∃L:Sequent List.
. | | |  ∀s∈L.(ρ(s) = 0)
. | | |  & (∀s∈L.|= s  ⇒ |= <hyp,M @ (x1⌈∧⌉x2.N)> )
. | | |  & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <hyp,M @ (x1⌈∧⌉x2.N)>)
. | | | by CopyToHyp -1 -1
. | | |
. | | 11. ∀S:Sequent.
. | | |   ρ(S)<ρ(<hyp,M @ (x1⌈∧⌉x2.N)>) ⇒
. | | |   (∃L:Sequent List.
. | | |    ∀s∈L.(ρ(s) = 0)
. | | |    & (∀s∈L.|= s  ⇒ |= S )
. | | |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
. | | 12. ∀S:Sequent.
. | | |   ρ(S)<ρ(<hyp,M @ (x1⌈∧⌉x2.N)>) ⇒
. | | |   (∃L:Sequent List.
. | | |    ∀s∈L.(ρ(s) = 0)
. | | |    & (∀s∈L.|= s  ⇒ |= S )
. | | |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
. | | ⊢
. | | | by With <hyp,x1.(M @ N)> (Analyze -2)
. | | |    THEN
. | | |    With <hyp,x2.(M @ N)> (Analyze -2)
. | | |

. | | 11. ρ(<hyp,x1.(M @ N)>)<ρ(<hyp,M @ (x1⌈∧⌉x2.N)>) ⇒
. | | |   (∃L:Sequent List.
. | | |    ∀s∈L.(ρ(s) = 0)
. | | |    & (∀s∈L.|= s  ⇒ |= <hyp,x1.(M @ N)> )
. | | |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <hyp,x1.(M @ N)>))
. | | 12. ρ(<hyp,x2.(M @ N)>)<ρ(<hyp,M @ (x1⌈∧⌉x2.N)>) ⇒
. | | |   (∃L:Sequent List.
. | | |    ∀s∈L.(ρ(s) = 0)
. | | |    & (∀s∈L.|= s  ⇒ |= <hyp,x2.(M @ N)> )
. | | |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <hyp,x2.(M @ N)>))
. | | ⊢
. | | | by RepeatFor 2
. | | |    (Analyze -2
```

```
. | | |     THENL
. | | |     [SequentRankReduce 0 THEN GenConclOnAps;Id])
. | | |
. | | 11. ∃L:Sequent List.
. | | |   ∀s∈L.(ρ(s) = 0)
. | | |   & (∀s∈L.|= s  ⇒ |= <hyp,x1.(M @ N)> )
. | | |   & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <hyp,x1.(M @ N)>)
. | | 12. ∃L:Sequent List.
. | | |   ∀s∈L.(ρ(s) = 0)
. | | |   & (∀s∈L.|= s  ⇒ |= <hyp,x2.(M @ N)> )
. | | |   & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <hyp,x2.(M @ N)>)
. | | ⊢   by Repeat (Analyze -2) THEN Repeat (Analyze -1)
. | | |
. | | 11. L : Sequent List
. | | 12. ∀s∈L.(ρ(s) = 0)
. | | 13. ∀s∈L.|= s  ⇒ |= <hyp,x1.(M @ N)>
. | | 14. ∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <hyp,x1.(M @ N)>
. | | 15. L1 : Sequent List
. | | 16. ∀s∈L1.(ρ(s) = 0)
. | | 17. ∀s∈L1.|= s  ⇒ |= <hyp,x2.(M @ N)>
. | | 18. ∀a:Assignment. ∃s∈L1.a |≠ s ⇒ a |≠ <hyp,x2.(M @ N)>
. | | ⊢   by Witness L @ L1
. | | |
. | | |\
. | | | ⊢  ∀s∈(L @ L1).(ρ(s) = 0)
. | | |   by BackThru
. | | |      Thm* ∀P:(T→Prop), L,M:T List.
. | | |           ∀x∈(L @ M).P(x) ⟺ ∀x∈L.P(x) & ∀x∈M.P(x)
. | | |      {list_all_append_lemma}
. | | |
. | | |\
. | | | 19. ∀s∈(L @ L1).|= s
. | | | ⊢  |= <hyp,M @ (x1⌈∧⌉x2.N)>
. | | | | by FwdThru
. | | | |    Thm* ∀P:(T→Prop), L,M:T List.
. | | | |         ∀x∈(L @ M).P(x) ⟺ ∀x∈L.P(x) & ∀x∈M.P(x)
. | | | |    {list_all_append_lemma}
. | | | |    [-1]
. | | | |    THEN
. | | | |    Sel 2
. | | | |    (FwdThru
. | | | |     Thm* ∀P:(T→Prop), L,M:T List.
. | | | |          ∀x∈(L @ M).P(x) ⟺ ∀x∈L.P(x) & ∀x∈M.P(x)
. | | | |     {list_all_append_lemma}
. | | | |     [-2])
. | | | |
. | | | 20. ∀s∈L.|= s
. | | | 21. ∀s∈L1.|= s
. | | | ⊢
. | | |   by BackThru
. | | |      Thm* ∀hyp,M,N:Formula List, q,r:Formula.
. | | |           |= <hyp,q.(M @ N)>  ⇒
. | | |           |= <hyp,r.(M @ N)>  ⇒ |= <hyp,M @ (q⌈∧⌉r.N)>
. | | |      {formula_and_right_sound}
. | | |      THEN
```

```
. | | |      HypBackchain
. | | |
. | | .\
. | | . 19. a : Assignment
. | | . 20. ∃s∈(L @ L1).a |≠ s
. | | . ⊢  a |≠ <hyp,M @ (x1⌜∧⌝x2.N)>
. | | . | by BackThru
. | | . |    Thm* ∀hyp,M,N:Formula List, q,r:Formula, a:Assignment.
. | | . |         a |≠ <hyp,q.(M @ N)> ∨ a |≠ <hyp,r.(M @ N)>
. | | . |         ⟺
. | | . |         a |≠ <hyp,M @ (q⌜∧⌝r.N)>
. | | . |    {formula_and_right_falsifiable}
. | | . |
. | | . ⊢  a |≠ <hyp,x1.(M @ N)> ∨ a |≠ <hyp,x2.(M @ N)>
. | | . | by FwdThru
. | | . |    Thm* ∀P:(T→Prop), L,M:T List.
. | | . |         ∃x∈(L @ M).P(x) ⟺ ∃x∈L.P(x) ∨ ∃x∈M.P(x)
. | | . |    {list_exists_append_lemma}
. | | . |    [-1]
. | | . |    THEN
. | | . |    Analyze -1
. | | . |
. | | . |\
. | | . | 21. ∃s∈L.a |≠ s
. | | . | ⊢   by Choose [1] THEN HypBackchain
. | | . |
. | | . .\
. | | . . 21. ∃s∈L1.a |≠ s
. | | . . ⊢   by Choose [2] THEN HypBackchain
. | |
. | |\
. | | 7. x1 : Formula

. | | 8. x2 : Formula
. | | 9. ρ(x1⌜∨⌝x2)>0
. | | 10. concl = (M @ (x1⌜∨⌝x2.N))
. | | 11. ∀S:Sequent.
. | | |   ρ(S)<ρ(<hyp,M @ (x1⌜∨⌝x2.N)>) ⇒
. | | |   (∃L:Sequent List.
. | | |    ∀s∈L.(ρ(s) = 0)
. | | |    & (∀s∈L.|= s  ⇒ |= S )
. | | |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
. | | ⊢  ∃L:Sequent List.
. | | |  ∀s∈L.(ρ(s) = 0)
. | | |  & (∀s∈L.|= s  ⇒ |= <hyp,M @ (x1⌜∨⌝x2.N)> )
. | | |  & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <hyp,M @ (x1⌜∨⌝x2.N)>)
. | | | by With <hyp,[x1; x2/ M @ N]> (Analyze -1)
. | | |
. | | 11. ρ(<hyp,[x1; x2/ M @ N]>)<ρ(<hyp,M @ (x1⌜∨⌝x2.N)>) ⇒
. | | |   (∃L:Sequent List.
. | | |    ∀s∈L.(ρ(s) = 0)
. | | |    & (∀s∈L.|= s  ⇒ |= <hyp,[x1; x2/ M @ N]> )
. | | |    & (∀a:Assignment.
. | | |       ∃s∈L.a |≠ s ⇒ a |≠ <hyp,[x1; x2/ M @ N]>))
. | | ⊢
. | | | by Analyze -1
```

```
. | | |    THENL
. | | |    [SequentRankReduce 0 THEN GenConclOnAps
. | | |    ;Repeat (Analyze -1)]
. | | |
. | | 11. L : Sequent List
. | | 12. ∀s∈L.(ρ(s) = 0)
. | | 13. ∀s∈L.|= s  ⇒ |= <hyp,[x1; x2/ M @ N]>
. | | 14. ∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <hyp,[x1; x2/ M @ N]>
. | | ⊢   by Witness L
. | | |
. | | |\
. | | | 15. ∀s∈L.|= s
. | | | ⊢  |= <hyp,M @ (x1⌈∨⌉x2.N)>
. | | |   by BackThru
. | | |      Thm* ∀hyp,M,N:Formula List, q,r:Formula.
. | | |           |= <hyp,[q; r/ M @ N]>  ⇒ |= <hyp,M @ (q⌈∨⌉r.N)>
. | | |      {formula_or_right_sound}
. | | |      THEN
. | | |      HypBackchain
. | | |
. | | .\
. | | . 15. a : Assignment
. | | . 16. ∃s∈L.a |≠ s
. | | . ⊢  a |≠ <hyp,M @ (x1⌈∨⌉x2.N)>
. | | .   by BackThru
. | | .      Thm* ∀hyp,M,N:Formula List, q,r:Formula, a:Assignment.
. | | .           a |≠ <hyp,[q; r/ M @ N]> ⟺ a |≠ <hyp,M @ (q⌈∨⌉r.N)>
. | | .      {formula_or_right_falsifiable}
. | | .      THEN
. | | .      HypBackchain
. | |
. | .\
. | . 7. y1 : Formula
. | . 8. y2 : Formula
. | . 9. ρ(y1⌈⇒⌉y2)>0
. | . 10. concl = (M @ (y1⌈⇒⌉y2.N))
. | . 11. ∀S:Sequent.
. | . |   ρ(S)<ρ(<hyp,M @ (y1⌈⇒⌉y2.N)>) ⇒
. | . |   (∃L:Sequent List.
. | . |    ∀s∈L.(ρ(s) = 0)
. | . |    & (∀s∈L.|= s  ⇒ |= S )
. | . |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ S))
. | . ⊢  ∃L:Sequent List.
. | . |  ∀s∈L.(ρ(s) = 0)
. | . |  & (∀s∈L.|= s  ⇒ |= <hyp,M @ (y1⌈⇒⌉y2.N)> )
. | . |  & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <hyp,M @ (y1⌈⇒⌉y2.N)>)
. | . | by With <y1.hyp,y2.(M @ N)> (Analyze -1)
. | . |
. | . 11. ρ(<y1.hyp,y2.(M @ N)>)<ρ(<hyp,M @ (y1⌈⇒⌉y2.N)>) ⇒
. | . |   (∃L:Sequent List.
. | . |    ∀s∈L.(ρ(s) = 0)
. | . |    & (∀s∈L.|= s  ⇒ |= <y1.hyp,y2.(M @ N)> )
. | . |    & (∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <y1.hyp,y2.(M @ N)>))
. | . ⊢
. | . | by Analyze -1
```

```
. | . |    THENL
. | . |    [SequentRankReduce 0 THEN GenConclOnAps
. | . |    ;Repeat (Analyze -1)]
. | . |
. | . 11. L : Sequent List
. | . 12. ∀s∈L.(ρ(s) = 0)
. | . 13. ∀s∈L.|= s  ⇒ |= <y1.hyp,y2.(M @ N)>
. | . 14. ∀a:Assignment. ∃s∈L.a |≠ s ⇒ a |≠ <y1.hyp,y2.(M @ N)>
. | . ⊢   by Witness L
. | . |
. | . |\
. | . | 15. ∀s∈L.|= s
. | . | ⊢  |= <hyp,M @ (y1⌈⇒⌉y2.N)>
. | . |   by BackThru
. | . |      Thm* ∀hyp,M,N:Formula List, q,r:Formula.
. | . |           |= <q.hyp,r.(M @ N)>  ⇒ |= <hyp,M @ (q⌈⇒⌉r.N)>
. | . |      {formula_imp_right_sound}
. | . |      THEN
. | . |      HypBackchain
. | . |
. | . .\
. | . . 15. a : Assignment
. | . . 16. ∃s∈L.a |≠ s
. | . . ⊢  a |≠ <hyp,M @ (y1⌈⇒⌉y2.N)>
. | . .   by BackThru
. | . .      Thm* ∀hyp,M,N:Formula List, q,r:Formula, a:Assignment.
. | . .           a |≠ <q.hyp,r.(M @ N)> ⟺ a |≠ <hyp,M @ (q⌈⇒⌉r.N)>
. | . .      {formula_imp_right_falsifiable}
. | . .      THEN
. | . .      HypBackchain
. |
. .\
. . 5. ¬∃f∈concl.(ρ(f)>0)
. . ⊢   by Assert (ρ(<hyp,concl>) = 0)
. . |
. . |\
. . | .....assertion .....
. . | ⊢  ρ(<hyp,concl>) = 0
. . |   by BackThru
. . |      Thm* ∀hyp,concl:Formula List.
. . |           ¬∃f∈hyp.(ρ(f)>0) ⇒
. . |           ¬∃f∈concl.(ρ(f)>0) ⇒ ρ(<hyp,concl>) = 0
. . |      {not_list_exists_rank_gt_0_rank_0}
. . |
. . .\
. . . 6. ρ(<hyp,concl>) = 0
. . . ⊢   by Witness [<hyp,concl>]
. . . |
. . . ⊢  ∀s∈[<hyp,concl>].(ρ(s) = 0)
. . . |  & (∀s∈[<hyp,concl>].|= s  ⇒ |= <hyp,concl> )
. . . |  & (∀a:Assignment.
. . . |     ∃s∈[<hyp,concl>].a |≠ s ⇒ a |≠ <hyp,concl>)
. . . | by AbReduce -1 THEN AbReduce 0
. . . |
. . . 6. ρ(hyp)+ρ(concl) = 0
```

```
. . . 7. a : Assignment
. . . 8. a |≠ <hyp,concl> ∨ False
. . . ⊢  a |≠ <hyp,concl>  by Analyze -1
END
```