```
bi-tree-diameter  THM

⊢  ∀T:(Id→𝔹), to,from:(|T|→(IdLnk List)).
|  bi-tree(T;to;from) ⇒ (∃n:ℕ. ∀p:Edge(T) List. lpath(p) ⇒ ||p||≤n)
| by Auto
|
1. T : Id→𝔹
2. to : |T|→(IdLnk List)
3. from : |T|→(IdLnk List)
4. bi-tree(T;to;from)
⊢  ∃n:ℕ. ∀p:Edge(T) List. lpath(p) ⇒ ||p||≤n
| by Analyze -1 THEN ExRepD THEN InstConcl [||L||]
|
|\
| 1. T : Id→𝔹
| 2. to : |T|→(IdLnk List)
| 3. from : |T|→(IdLnk List)
| 4. bi-graph(T;to;from)
| 5. ∀i,j:|T|.
| |  ∃p:Edge(T) List.
| |  lconnects(p;i;j) & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| 6. L : |T| List
| 7. ∀i:|T|. (i ∈ L)
| 8. |T|
| 9. p : Edge(T) List
| 10. lpath(p)
| ⊢  ||p||≤||L||  by SupposeNot
| |
| 1. T : Id→𝔹
| 2. to : |T|→(IdLnk List)
| 3. from : |T|→(IdLnk List)
| 4. bi-graph(T;to;from)
| 5. ∀i,j:|T|.
| |  ∃p:Edge(T) List.
| |  lconnects(p;i;j) & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| 6. L : |T| List
| 7. ∀i:|T|. (i ∈ L)
| 8. |T|
| 9. p : Edge(T) List
| 10. lpath(p)
| 11. ¬||p||≤||L||
| ⊢  ||p||≤||L||
| | by Assert (∃n:ℕ||p||, m:ℕn. source(p[m]) = source(p[n]))
| |
| |\
| | .....assertion .....
| | 1. T : Id→𝔹
| | 2. to : |T|→(IdLnk List)
| | 3. from : |T|→(IdLnk List)
| | 4. bi-graph(T;to;from)
| | 5. ∀i,j:|T|.
| | |  ∃p:Edge(T) List.
| | |  lconnects(p;i;j) & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| | 6. L : |T| List
| | 7. ∀i:|T|. (i ∈ L)
```

```
| | 8. |T|
| | 9. p : Edge(T) List
| | 10. lpath(p)
| | 11. ¬||p||≤||L||
| | ⊢  ∃n:ℕ||p||, m:ℕn. source(p[m]) = source(p[n])
| | | by DupHyp 7 THEN RenameVar ‘g’ -1
| | |    THEN
| | |    Repeat (Unfolds [‘all‘;‘l_member‘;‘exists‘] -1)
| | |
| | 1. T : Id→𝔹
| | 2. to : |T|→(IdLnk List)
| | 3. from : |T|→(IdLnk List)
| | 4. bi-graph(T;to;from)
| | 5. ∀i,j:|T|.
| | |  ∃p:Edge(T) List.
| | |  lconnects(p;i;j) & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| | 6. L : |T| List
| | 7. ∀i:|T|. (i ∈ L)
| | 8. |T|
| | 9. p : Edge(T) List
| | 10. lpath(p)
| | 11. ¬||p||≤||L||
| | 12. i:|T|→i@0:ℕ×i@0<||L|| & i = L[i@0]
| | ⊢  ∃n:ℕ||p||, m:ℕn. source(p[m]) = source(p[n])
| | | by SupposeNot THEN MaAuto
| | |    THEN
| | |    Try
| | |    (BackThru
| | |     Thm* ∀T:(Id→𝔹), to,from:(|T|→(IdLnk List)), u:Edge(T).
| | |          bi-graph(T;to;from) ⇒ source(u) ∈ |T|
| | |     {src_DASH_edge})
| | |    THEN
| | |    Assert False
| | |    THEN
| | |    Assert (∃h:(ℕ||p||→|T|). Inj(ℕ||p||; |T|; h))
| | |
| | |\
| | | .....assertion .....
| | | 1. T : Id→𝔹
| | | 2. to : |T|→(IdLnk List)
| | | 3. from : |T|→(IdLnk List)
| | | 4. bi-graph(T;to;from)
| | | 5. ∀i,j:|T|.
| | | |  ∃p:Edge(T) List.
| | | |  lconnects(p;i;j)
| | | |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| | | 6. L : |T| List
| | | 7. ∀i:|T|. (i ∈ L)
| | | 8. |T|
| | | 9. p : Edge(T) List
| | | 10. lpath(p)
| | | 11. ¬||p||≤||L||
| | | 12. i:|T|→i@0:ℕ×i@0<||L|| & i = L[i@0]
| | | 13. ¬(∃n:ℕ||p||, m:ℕn. source(p[m]) = source(p[n]))
| | | ⊢  ∃h:(ℕ||p||→|T|). Inj(ℕ||p||; |T|; h)
```

```
| | | | by InstConcl [λx.source(p[x])]
| | | |    THEN
| | | |    Try
| | | |    (BackThru
| | | |     Thm* ∀T:(Id→𝔹), to,from:(|T|→(IdLnk List)), u:Edge(T).
| | | |          bi-graph(T;to;from) ⇒ source(u) ∈ |T|
| | | |     {src_DASH_edge})
| | | |
| | | 1. T : Id→𝔹
| | | 2. to : |T|→(IdLnk List)
| | | 3. from : |T|→(IdLnk List)
| | | 4. bi-graph(T;to;from)
| | | 5. ∀i,j:|T|.
| | | |  ∃p:Edge(T) List.
| | | |  lconnects(p;i;j)
| | | |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| | | 6. L : |T| List
| | | 7. ∀i:|T|. (i ∈ L)
| | | 8. |T|
| | | 9. p : Edge(T) List
| | | 10. lpath(p)
| | | 11. ¬||p||≤||L||
| | | 12. i:|T|→i@0:ℕ×i@0<||L|| & i = L[i@0]
| | | 13. ¬(∃n:ℕ||p||, m:ℕn. source(p[m]) = source(p[n]))
| | | ⊢  Inj(ℕ||p||; |T|; λx.source(p[x]))
| | | | by Analyze 0 THEN All Reduce
| | | |    THEN
| | | |    Try
| | | |    (BackThru
| | | |     Thm* ∀T:(Id→𝔹), to,from:(|T|→(IdLnk List)), u:Edge(T).
| | | |          bi-graph(T;to;from) ⇒ source(u) ∈ |T|
| | | |     {src_DASH_edge})
| | | |
| | | 1. T : Id→𝔹
| | | 2. to : |T|→(IdLnk List)
| | | 3. from : |T|→(IdLnk List)
| | | 4. bi-graph(T;to;from)
| | | 5. ∀i,j:|T|.
| | | |  ∃p:Edge(T) List.
| | | |  lconnects(p;i;j)
| | | |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| | | 6. L : |T| List
| | | 7. ∀i:|T|. (i ∈ L)
| | | 8. |T|
| | | 9. p : Edge(T) List
| | | 10. lpath(p)
| | | 11. ¬||p||≤||L||
| | | 12. i:|T|→i@0:ℕ×i@0<||L|| & i = L[i@0]
| | | 13. ¬(∃n:ℕ||p||, m:ℕn. source(p[m]) = source(p[n]))
| | | 14. a1 : ℕ||p||
| | | 15. a2 : ℕ||p||
| | | 16. source(p[a1]) = source(p[a2])
| | | ⊢  a1 = a2  by Decide (a1≤a2)
| | | |
| | | |\
```

```
| | | | 1. T : Id→𝔹
| | | | 2. to : |T|→(IdLnk List)
| | | | 3. from : |T|→(IdLnk List)
| | | | 4. bi-graph(T;to;from)
| | | | 5. ∀i,j:|T|.
| | | | |  ∃p:Edge(T) List.
| | | | |  lconnects(p;i;j)
| | | | |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| | | | 6. L : |T| List
| | | | 7. ∀i:|T|. (i ∈ L)
| | | | 8. |T|
| | | | 9. p : Edge(T) List
| | | | 10. lpath(p)
| | | | 11. ¬||p||≤||L||
| | | | 12. i:|T|→i@0:ℕ×i@0<||L|| & i = L[i@0]
| | | | 13. ¬(∃n:ℕ||p||, m:ℕn. source(p[m]) = source(p[n]))
| | | | 14. a1 : ℕ||p||
| | | | 15. a2 : ℕ||p||
| | | | 16. source(p[a1]) = source(p[a2])
| | | | 17. a1≤a2
| | | | ⊢  a1 = a2  by Decide (a2≤a1)
| | | | |
| | | | 1. T : Id→𝔹
| | | | 2. to : |T|→(IdLnk List)
| | | | 3. from : |T|→(IdLnk List)
| | | | 4. bi-graph(T;to;from)
| | | | 5. ∀i,j:|T|.
| | | |    ∃p:Edge(T) List.
| | | |    lconnects(p;i;j)
| | | |    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| | | | 6. L : |T| List
| | | | 7. ∀i:|T|. (i ∈ L)
| | | | 8. |T|
| | | | 9. p : Edge(T) List
| | | | 10. lpath(p)
| | | | 11. ¬||p||≤||L||
| | | | 12. i:|T|→i@0:ℕ×i@0<||L|| & i = L[i@0]
| | | | 13. ¬(∃n:ℕ||p||, m:ℕn. source(p[m]) = source(p[n]))
| | | | 14. a1 : ℕ||p||
| | | | 15. a2 : ℕ||p||
| | | | 16. source(p[a1]) = source(p[a2])
| | | | 17. a1≤a2
| | | | 18. ¬a2≤a1
| | | | ⊢  a1 = a2
| | | |   by Analyze -6 THEN InstConcl [a2;a1]
| | | |      THEN
| | | |      Try
| | | |      (BackThru
| | | |       Thm* ∀T:(Id→𝔹), to,from:(|T|→(IdLnk List)), u:Edge(T).
| | | |            bi-graph(T;to;from) ⇒ source(u) ∈ |T|
| | | |       {src_DASH_edge})
| | | |
| | | .\
| | | . 1. T : Id→𝔹
| | | . 2. to : |T|→(IdLnk List)
```

```
| | | . 3. from : |T|→(IdLnk List)
| | | . 4. bi-graph(T;to;from)
| | | . 5. ∀i,j:|T|.
| | | .    ∃p:Edge(T) List.
| | | .    lconnects(p;i;j)
| | | .    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| | | . 6. L : |T| List
| | | . 7. ∀i:|T|. (i ∈ L)
| | | . 8. |T|
| | | . 9. p : Edge(T) List
| | | . 10. lpath(p)
| | | . 11. ¬||p||≤||L||
| | | . 12. i:|T|→i@0:ℕ×i@0<||L|| & i = L[i@0]
| | | . 13. ¬(∃n:ℕ||p||, m:ℕn. source(p[m]) = source(p[n]))
| | | . 14. a1 : ℕ||p||
| | | . 15. a2 : ℕ||p||
| | | . 16. source(p[a1]) = source(p[a2])
| | | . 17. ¬a1≤a2
| | | . ⊢  a1 = a2
| | | .   by Analyze -5 THEN InstConcl [a1;a2]
| | | .      THEN
| | | .      Try
| | | .      (BackThru
| | | .       Thm* ∀T:(Id→𝔹), to,from:(|T|→(IdLnk List)), u:Edge(T).
| | | .            bi-graph(T;to;from) ⇒ source(u) ∈ |T|
| | | .       {src_DASH_edge})
| | |
| | .\
| | . 1. T : Id→𝔹
| | . 2. to : |T|→(IdLnk List)
| | . 3. from : |T|→(IdLnk List)
| | . 4. bi-graph(T;to;from)
| | . 5. ∀i,j:|T|.
| | . |  ∃p:Edge(T) List.
| | . |  lconnects(p;i;j)
| | . |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| | . 6. L : |T| List
| | . 7. ∀i:|T|. (i ∈ L)
| | . 8. |T|
| | . 9. p : Edge(T) List
| | . 10. lpath(p)
| | . 11. ¬||p||≤||L||
| | . 12. i:|T|→i@0:ℕ×i@0<||L|| & i = L[i@0]
| | . 13. ¬(∃n:ℕ||p||, m:ℕn. source(p[m]) = source(p[n]))
| | . 14. ∃h:(ℕ||p||→|T|). Inj(ℕ||p||; |T|; h)
| | . ⊢  False  by Assert (∃k:(|T|→ℕ||L||). Inj(|T|; ℕ||L||; k))
| | . |
| | . |\
| | . | .....assertion .....
| | . | 1. T : Id→𝔹
| | . | 2. to : |T|→(IdLnk List)
| | . | 3. from : |T|→(IdLnk List)
| | . | 4. bi-graph(T;to;from)
| | . | 5. ∀i,j:|T|.
| | . | |  ∃p:Edge(T) List.
```

```
| | . | |  lconnects(p;i;j)
| | . | |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| | . | 6. L : |T| List
| | . | 7. ∀i:|T|. (i ∈ L)
| | . | 8. |T|
| | . | 9. p : Edge(T) List
| | . | 10. lpath(p)
| | . | 11. ¬||p||≤||L||
| | . | 12. i:|T|→i@0:ℕ×i@0<||L|| & i = L[i@0]
| | . | 13. ¬(∃n:ℕ||p||, m:ℕn. source(p[m]) = source(p[n]))
| | . | 14. ∃h:(ℕ||p||→|T|). Inj(ℕ||p||; |T|; h)
| | . | ⊢  ∃k:(|T|→ℕ||L||). Inj(|T|; ℕ||L||; k)
| | . | | by InstConcl [λi.1of(g(i))]
| | . | |
| | . | |\
| | . | | .....subterm .....
| | . | | 1. T : Id→𝔹
| | . | | 2. to : |T|→(IdLnk List)
| | . | | 3. from : |T|→(IdLnk List)
| | . | | 4. bi-graph(T;to;from)
| | . | | 5. ∀i,j:|T|.
| | . | |    ∃p:Edge(T) List.
| | . | |    lconnects(p;i;j)
| | . | |    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| | . | | 6. L : |T| List
| | . | | 7. ∀i:|T|. (i ∈ L)
| | . | | 8. |T|
| | . | | 9. p : Edge(T) List
| | . | | 10. lpath(p)
| | . | | 11. ¬||p||≤||L||
| | . | | 12. g : i:|T|→i@0:ℕ×i@0<||L|| & i = L[i@0]
| | . | | 13. ¬(∃n:ℕ||p||, m:ℕn. source(p[m]) = source(p[n]))
| | . | | 14. ∃h:(ℕ||p||→|T|). Inj(ℕ||p||; |T|; h)
| | . | | 15. i : |T|
| | . | | ⊢  g(i) ∈ a:ℕ||L||×a<||L|| & i = L[a]
| | . | |   by DoSubsume THEN Analyze 0 THEN Analyze -1
| | . | |      THEN
| | . | |      Analyze -1
| | . | |
| | . | .\
| | . | . 1. T : Id→𝔹
| | . | . 2. to : |T|→(IdLnk List)
| | . | . 3. from : |T|→(IdLnk List)
| | . | . 4. bi-graph(T;to;from)
| | . | . 5. ∀i,j:|T|.
| | . | . |  ∃p:Edge(T) List.
| | . | . |  lconnects(p;i;j)
| | . | . |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| | . | . 6. L : |T| List
| | . | . 7. ∀i:|T|. (i ∈ L)
| | . | . 8. |T|
| | . | . 9. p : Edge(T) List
| | . | . 10. lpath(p)
| | . | . 11. ¬||p||≤||L||
| | . | . 12. g : i:|T|→i@0:ℕ×i@0<||L|| & i = L[i@0]
```

```
| | . | . 13. ¬(∃n:ℕ||p||, m:ℕn. source(p[m]) = source(p[n]))
| | . | . 14. ∃h:(ℕ||p||→|T|). Inj(ℕ||p||; |T|; h)
| | . | . ⊢  Inj(|T|; ℕ||L||; λi.1of(g(i)))
| | . | . | by Unfold ‘inject‘ 0 THEN Reduce 0 THEN Analyze 0
| | . | . |    THEN
| | . | . |    Analyze 0
| | . | . |    THEN
| | . | . |    GenConclAtAddr [1;2;1]
| | . | . |    THEN
| | . | . |    Analyze -2
| | . | . |    THEN
| | . | . |    Analyze -2
| | . | . |    THEN
| | . | . |    GenConclAtAddr [1;3;1]
| | . | . |    THEN
| | . | . |    Analyze -2
| | . | . |    THEN
| | . | . |    Analyze -2
| | . | . |    THEN
| | . | . |    Reduce 0
| | . | . |
| | . | . 1. T : Id→ℙ
| | . | . 2. to : |T|→(IdLnk List)
| | . | . 3. from : |T|→(IdLnk List)
| | . | . 4. bi-graph(T;to;from)
| | . | . 5. ∀i,j:|T|.
| | . | .    ∃p:Edge(T) List.
| | . | .    lconnects(p;i;j)
| | . | .    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| | . | . 6. L : |T| List
| | . | . 7. ∀i:|T|. (i ∈ L)
| | . | . 8. |T|
| | . | . 9. p : Edge(T) List
| | . | . 10. lpath(p)
| | . | . 11. ¬||p||≤||L||
| | . | . 12. g : i:|T|→i@0:ℕ×i@0<||L|| & i = L[i@0]
| | . | . 13. ¬(∃n:ℕ||p||, m:ℕn. source(p[m]) = source(p[n]))
| | . | . 14. ∃h:(ℕ||p||→|T|). Inj(ℕ||p||; |T|; h)
| | . | . 15. a1 : |T|
| | . | . 16. a2 : |T|
| | . | . 17. i@0 : ℕ
| | . | . 18. v2 : i@0<||L||
| | . | . 19. v3 : a1 = L[i@0]
| | . | . 20. g(a1) = <i@0,v2,v3>
| | . | . 21. i1 : ℕ
| | . | . 22. v5 : i1<||L||
| | . | . 23. v6 : a2 = L[i1]
| | . | . 24. g(a2) = <i1,v5,v6>
| | . | . 25. i@0 = i1
| | . | . ⊢  a1 = a2  by RevHypSubst’ -1 -3
| | . |
| | . .\
| | . . 1. T : Id→ℙ
| | . . 2. to : |T|→(IdLnk List)
| | . . 3. from : |T|→(IdLnk List)
```

```
| | . . 4. bi-graph(T;to;from)
| | . . 5. ∀i,j:|T|.
| | . . |  ∃p:Edge(T) List.
| | . . |  lconnects(p;i;j)
| | . . |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| | . . 6. L : |T| List
| | . . 7. ∀i:|T|. (i ∈ L)
| | . . 8. |T|
| | . . 9. p : Edge(T) List
| | . . 10. lpath(p)
| | . . 11. ¬||p||≤||L||
| | . . 12. i:|T|→i@0:ℕ×i@0<||L|| & i = L[i@0]
| | . . 13. ¬(∃n:ℕ||p||, m:ℕn. source(p[m]) = source(p[n]))
| | . . 14. ∃h:(ℕ||p||→|T|). Inj(ℕ||p||; |T|; h)
| | . . 15. ∃k:(|T|→ℕ||L||). Inj(|T|; ℕ||L||; k)
| | . . ⊢  False
| | . . | by ExRepD

| | . . |    THEN
| | . . |    Inst
| | . . |    Thm* ∀n,m:ℕ, f:(ℕn→ℕm). Inj(ℕn; ℕm; f) ⇒ n≤m
| | . . |    {pigeon_DASH_hole}
| | . . |    [||p||;||L||;k o h]
| | . . |
| | . . .....antecedent .....
| | . . 1. T : Id→𝔹
| | . . 2. to : |T|→(IdLnk List)
| | . . 3. from : |T|→(IdLnk List)
| | . . 4. bi-graph(T;to;from)
| | . . 5. ∀i,j:|T|.
| | . .    ∃p:Edge(T) List.
| | . .    lconnects(p;i;j)
| | . .    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| | . . 6. L : |T| List
| | . . 7. ∀i:|T|. (i ∈ L)
| | . . 8. |T|
| | . . 9. p : Edge(T) List
| | . . 10. lpath(p)
| | . . 11. ¬||p||≤||L||
| | . . 12. i:|T|→i@0:ℕ×i@0<||L|| & i = L[i@0]
| | . . 13. ¬(∃n:ℕ||p||, m:ℕn. source(p[m]) = source(p[n]))
| | . . 14. h : ℕ||p||→|T|
| | . . 15. Inj(ℕ||p||; |T|; h)
| | . . 16. k : |T|→ℕ||L||
| | . . 17. Inj(|T|; ℕ||L||; k)
| | . . ⊢  Inj(ℕ||p||; ℕ||L||; k o h)
| | . .   by All (λh.Unfold ‘inject‘ h) THEN Reduce 0 THEN BackThru -6
| | . .      THEN
| | . .      BackThru -4
| |
| .\
| . 1. T : Id→𝔹
| . 2. to : |T|→(IdLnk List)
| . 3. from : |T|→(IdLnk List)
| . 4. bi-graph(T;to;from)
| . 5. ∀i,j:|T|.
```

```
| . |  ∃p:Edge(T) List.
| . |  lconnects(p;i;j) & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| . 6. L : |T| List
| . 7. ∀i:|T|. (i ∈ L)
| . 8. |T|
| . 9. p : Edge(T) List
| . 10. lpath(p)
| . 11. ¬||p||≤||L||
| . 12. ∃n:ℕ||p||, m:ℕn. source(p[m]) = source(p[n])
| . ⊢  ||p||≤||L||  by ExRepD
| . |
| . 1. T : Id→𝔹
| . 2. to : |T|→(IdLnk List)
| . 3. from : |T|→(IdLnk List)
| . 4. bi-graph(T;to;from)
| . 5. ∀i,j:|T|.
| . |  ∃p:Edge(T) List.
| . |  lconnects(p;i;j) & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| . 6. L : |T| List
| . 7. ∀i:|T|. (i ∈ L)
| . 8. |T|
| . 9. p : Edge(T) List
| . 10. lpath(p)
| . 11. ¬||p||≤||L||
| . 12. n : ℕ||p||
| . 13. m : ℕn
| . 14. source(p[m]) = source(p[n])
| . ⊢  ||p||≤||L||  by Auto
| . |
| . 1. T : Id→𝔹
| . 2. to : |T|→(IdLnk List)
| . 3. from : |T|→(IdLnk List)
| . 4. bi-graph(T;to;from)
| . 5. ∀i,j:|T|.
| . |  ∃p:Edge(T) List.
| . |  lconnects(p;i;j) & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| . 6. L : |T| List
| . 7. ∀i:|T|. (i ∈ L)
| . 8. |T|
| . 9. p : Edge(T) List
| . 10. lpath(p)
| . 11. ¬||p||≤||L||
| . 12. n : ℕ||p||
| . 13. m : ℕn
| . 14. source(p[m]) = source(p[n])
| . ⊢  ||p||≤||L||
| . | by InstHyp [source(p[m]);source(p[n])] 5 THEN ExRepD
| . |    THEN
| . |    InstHyp [nil] -1
| . |
| . |\
| . | .....antecedent .....
| . | 1. T : Id→𝔹
| . | 2. to : |T|→(IdLnk List)
| . | 3. from : |T|→(IdLnk List)
```

```
| . | 4. bi-graph(T;to;from)
| . | 5. ∀i,j:|T|.
| . | |  ∃p:Edge(T) List.
| . | |  lconnects(p;i;j)
| . | |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| . | 6. L : |T| List
| . | 7. ∀i:|T|. (i ∈ L)
| . | 8. |T|
| . | 9. p : Edge(T) List
| . | 10. lpath(p)
| . | 11. ¬||p||≤||L||
| . | 12. n : ℕ||p||
| . | 13. m : ℕn
| . | 14. source(p[m]) = source(p[n])
| . | 15. p@0 : Edge(T) List
| . | 16. lconnects(p@0;source(p[m]);source(p[n]))
| . | 17. ∀q:Edge(T) List.
| . | |   lconnects(q;source(p[m]);source(p[n])) ⇒ q = p@0
| . | ⊢  lconnects(nil;source(p[m]);source(p[n]))
| . | | by Unfold ‘lconnects‘ 0 THEN Reduce 0
| . | |
| . | |\
| . | | 1. T : Id→𝔹
| . | | 2. to : |T|→(IdLnk List)
| . | | 3. from : |T|→(IdLnk List)
| . | | 4. bi-graph(T;to;from)
| . | | 5. ∀i,j:|T|.
| . | |    ∃p:Edge(T) List.
| . | |    lconnects(p;i;j)
| . | |    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| . | | 6. L : |T| List
| . | | 7. ∀i:|T|. (i ∈ L)
| . | | 8. |T|
| . | | 9. p : Edge(T) List
| . | | 10. lpath(p)
| . | | 11. ¬||p||≤||L||
| . | | 12. n : ℕ||p||
| . | | 13. m : ℕn
| . | | 14. source(p[m]) = source(p[n])
| . | | 15. p@0 : Edge(T) List
| . | | 16. lconnects(p@0;source(p[m]);source(p[n]))
| . | | 17. ∀q:Edge(T) List.
| . | |     lconnects(q;source(p[m]);source(p[n])) ⇒ q = p@0
| . | | ⊢  lpath(nil)  by Unfold ‘lpath‘ 0 THEN Reduce 0
| . | |
| . | .\
| . | . 1. T : Id→𝔹
| . | . 2. to : |T|→(IdLnk List)
| . | . 3. from : |T|→(IdLnk List)
| . | . 4. bi-graph(T;to;from)
| . | . 5. ∀i,j:|T|.
| . | .    ∃p:Edge(T) List.
| . | .    lconnects(p;i;j)
| . | .    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| . | . 6. L : |T| List
```

```
| . | . 7. ∀i:|T|. (i ∈ L)
| . | . 8. |T|
| . | . 9. p : Edge(T) List
| . | . 10. lpath(p)
| . | . 11. ¬||p||≤||L||
| . | . 12. n : ℕ||p||
| . | . 13. m : ℕn
| . | . 14. source(p[m]) = source(p[n])
| . | . 15. p@0 : Edge(T) List
| . | . 16. lconnects(p@0;source(p[m]);source(p[n]))
| . | . 17. ∀q:Edge(T) List.
| . | .     lconnects(q;source(p[m]);source(p[n])) ⇒ q = p@0
| . | . 18. 0 = 0
| . | . ⊢  source(p[m]) = source(p[n])  by HypSubst -5 0 THEN MaAuto
| . |
| . .\
| . . 1. T : Id→𝔹
| . . 2. to : |T|→(IdLnk List)
| . . 3. from : |T|→(IdLnk List)
| . . 4. bi-graph(T;to;from)
| . . 5. ∀i,j:|T|.
| . . |  ∃p:Edge(T) List.
| . . |  lconnects(p;i;j)
| . . |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| . . 6. L : |T| List
| . . 7. ∀i:|T|. (i ∈ L)
| . . 8. |T|
| . . 9. p : Edge(T) List
| . . 10. lpath(p)
| . . 11. ¬||p||≤||L||
| . . 12. n : ℕ||p||
| . . 13. m : ℕn
| . . 14. source(p[m]) = source(p[n])
| . . 15. p@0 : Edge(T) List
| . . 16. lconnects(p@0;source(p[m]);source(p[n]))
| . . 17. ∀q:Edge(T) List.
| . . |   lconnects(q;source(p[m]);source(p[n])) ⇒ q = p@0
| . . 18. nil = p@0
| . . ⊢  ||p||≤||L||
| . . | by Inst
| . . |    Thm* ∀p:IdLnk List, i:ℕ||p||, j:ℕ(i+1).
| . . |         lpath(p)
| . . |         ⇒
| . . |         lconnects(l_interval(p;j;i);source(p[j]);source(p[i]))
| . . |    {lpath_DASH_members_DASH_connected}
| . . |    [p;n;m]
| . . |    THENA
| . . |    MaAuto
| . . |    THEN
| . . |    InstHyp [l_interval(p;m;n)] -3
| . . |
| . . 1. T : Id→𝔹
| . . 2. to : |T|→(IdLnk List)
| . . 3. from : |T|→(IdLnk List)
| . . 4. bi-graph(T;to;from)
```

```
| . . 5. ∀i,j:|T|.
| . . |  ∃p:Edge(T) List.
| . . |  lconnects(p;i;j)
| . . |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| . . 6. L : |T| List
| . . 7. ∀i:|T|. (i ∈ L)
| . . 8. |T|
| . . 9. p : Edge(T) List
| . . 10. lpath(p)
| . . 11. ¬||p||≤||L||
| . . 12. n : ℕ||p||
| . . 13. m : ℕn
| . . 14. source(p[m]) = source(p[n])
| . . 15. p@0 : Edge(T) List
| . . 16. lconnects(p@0;source(p[m]);source(p[n]))
| . . 17. ∀q:Edge(T) List.
| . . |   lconnects(q;source(p[m]);source(p[n])) ⇒ q = p@0
| . . 18. nil = p@0
| . . 19. lconnects(l_interval(p;m;n);source(p[m]);source(p[n]))
| . . 20. l_interval(p;m;n) = p@0
| . . ⊢  ||p||≤||L||  by Assert (||l_interval(p;m;n)|| = 0)
| . . |
| . . |\
| . . | .....assertion .....
| . . | 1. T : Id→𝔹
| . . | 2. to : |T|→(IdLnk List)
| . . | 3. from : |T|→(IdLnk List)
| . . | 4. bi-graph(T;to;from)
| . . | 5. ∀i,j:|T|.
| . . |    ∃p:Edge(T) List.
| . . |    lconnects(p;i;j)
| . . |    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| . . | 6. L : |T| List
| . . | 7. ∀i:|T|. (i ∈ L)
| . . | 8. |T|
| . . | 9. p : Edge(T) List
| . . | 10. lpath(p)
| . . | 11. ¬||p||≤||L||
| . . | 12. n : ℕ||p||
| . . | 13. m : ℕn
| . . | 14. source(p[m]) = source(p[n])
| . . | 15. p@0 : Edge(T) List
| . . | 16. lconnects(p@0;source(p[m]);source(p[n]))
| . . | 17. ∀q:Edge(T) List.
| . . |     lconnects(q;source(p[m]);source(p[n])) ⇒ q = p@0
| . . | 18. nil = p@0
| . . | 19. lconnects(l_interval(p;m;n);source(p[m]);source(p[n]))
| . . | 20. l_interval(p;m;n) = p@0
| . . | ⊢  ||l_interval(p;m;n)|| = 0
| . . |   by (HypSubst -1 0) THEN (RevHypSubst -3 0) THEN (Reduce 0)
| . . |
| . . .\
| . . . 1. T : Id→𝔹
| . . . 2. to : |T|→(IdLnk List)
| . . . 3. from : |T|→(IdLnk List)
```

```
| . . . 4. bi-graph(T;to;from)
| . . . 5. ∀i,j:|T|.
| . . .    ∃p:Edge(T) List.
| . . .    lconnects(p;i;j)
| . . .    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| . . . 6. L : |T| List
| . . . 7. ∀i:|T|. (i ∈ L)
| . . . 8. |T|
| . . . 9. p : Edge(T) List
| . . . 10. lpath(p)
| . . . 11. ¬||p||≤||L||
| . . . 12. n : ℕ||p||
| . . . 13. m : ℕn
| . . . 14. source(p[m]) = source(p[n])
| . . . 15. p@0 : Edge(T) List
| . . . 16. lconnects(p@0;source(p[m]);source(p[n]))
| . . . 17. ∀q:Edge(T) List.
| . . .     lconnects(q;source(p[m]);source(p[n])) ⇒ q = p@0
| . . . 18. nil = p@0
| . . . 19. lconnects(l_interval(p;m;n);source(p[m]);source(p[n]))
| . . . 20. l_interval(p;m;n) = p@0
| . . . 21. ||l_interval(p;m;n)|| = 0
| . . . ⊢  ||p||≤||L||
| . . .   by RW0
| . . .      Thm* ∀l:T List, i:ℕ||l||, j:ℕ(i+1).
| . . .           ||l_interval(l;j;i)|| = i-j ∈ ℤ
| . . .      {length_l_interval}
| . . .      -1
|
|\
| .....unproved Inclusion subgoal .....
| 1. T : Id→𝔹
| 2. to : |T|→(IdLnk List)
| 3. from : |T|→(IdLnk List)
| 4. bi-graph(T;to;from)
| 5. ∀i,j:|T|.
|    ∃p:Edge(T) List.
|    lconnects(p;i;j) & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| 6. L : |T| List
| 7. ∀i:|T|. (i ∈ L)
| 8. |T|
| 9. Edge(T) List
| 10. Edge(T) List
| 11. u : Edge(T)
| ⊢  u ∈ IdLnk  by MaAuto
|
.\
. .....unproved Inclusion subgoal .....
. 1. T : Id→𝔹
. 2. to : |T|→(IdLnk List)
. 3. from : |T|→(IdLnk List)
. 4. bi-graph(T;to;from)
. 5. ∀i,j:|T|.
.    ∃p:Edge(T) List.
.    lconnects(p;i;j) & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
```

```
. 6. L : |T| List
. 7. ∀i:|T|. (i ∈ L)
. 8. |T|
. 9. ℕ
. 10. Edge(T) List
. 11. Edge(T) List
. 12. u : Edge(T)
. ⊢  u ∈ IdLnk  by MaAuto
END
```