```
spanner-root-exists  THM

⊢  ∀T:(Id→𝔹), to,from:(|T|→(IdLnk List)), f:(Edge(T)→𝔹).
|  bi-tree(T;to;from)
|  ⇒
|  spanner(f;T;to;from) ⇒ (∃i:|T|. spanner-root(f;T;to;from;i))
| by Repeat (Analyze 0 THENA Complete Auto)
|    THEN
|    AssertBY bi-graph(T;to;from) (Analyze -1)
|
1. T : Id→𝔹
2. to : |T|→(IdLnk List)
3. from : |T|→(IdLnk List)
4. f : Edge(T)→𝔹
5. bi-tree(T;to;from)
6. bi-graph(T;to;from)
7. spanner(f;T;to;from)
⊢  ∃i:|T|. spanner-root(f;T;to;from;i)
| by FwdThru
|    Thm* ∀T:(Id→𝔹), to,from:(|T|→(IdLnk List)).
|         bi-tree(T;to;from)
|         ⇒
|         (∃n:ℕ. ∀p:Edge(T) List. lpath(p) ⇒ ||p||≤n)
|    {bi_DASH_tree_DASH_diameter}
|    [-3]
|    THEN
|    Unfold `bi-tree` 5
|    THEN
|    ExRepD
|    THEN
|    RenameVar `j` 9
|    THEN
|    Decide (||to(j)|| = 0)
|
|\
| 1. T : Id→𝔹
| 2. to : |T|→(IdLnk List)
| 3. from : |T|→(IdLnk List)
| 4. f : Edge(T)→𝔹
| 5. bi-graph(T;to;from)
| 6. ∀i,j:|T|.
|    ∃p:Edge(T) List.
|    lconnects(p;i;j) & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
| 7. L : |T| List
| 8. ∀i:|T|. (i ∈ L)
| 9. j : |T|
| 10. bi-graph(T;to;from)
| 11. spanner(f;T;to;from)
| 12. n : ℕ
| 13. ∀p:Edge(T) List. lpath(p) ⇒ ||p||≤n
| 14. ||to(j)|| = 0
| ⊢  ∃i:|T|. spanner-root(f;T;to;from;i)
|   by InstConcl [j] THEN Unfold `spanner-root` 0
|      THEN
|      Unfold `l_member` -1
```

```
|      THEN
|      ExRepD
|
.\
. 1. T : Id→𝔹
. 2. to : |T|→(IdLnk List)
. 3. from : |T|→(IdLnk List)
. 4. f : Edge(T)→𝔹
. 5. bi-graph(T;to;from)
. 6. ∀i,j:|T|.
. |  ∃p:Edge(T) List.
. |  lconnects(p;i;j) & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. 7. L : |T| List
. 8. ∀i:|T|. (i ∈ L)
. 9. j : |T|
. 10. bi-graph(T;to;from)
. 11. spanner(f;T;to;from)
. 12. n : ℕ
. 13. ∀p:Edge(T) List. lpath(p) ⇒ ||p||≤n
. 14. ¬||to(j)|| = 0
. ⊢  ∃i:|T|. spanner-root(f;T;to;from;i)
. | by Assert Dec(∃i:|T|. spanner-root(f;T;to;from;i))
. |
. |\
. | .....assertion .....
. | 1. T : Id→𝔹
. | 2. to : |T|→(IdLnk List)
. | 3. from : |T|→(IdLnk List)
. | 4. f : Edge(T)→𝔹
. | 5. bi-graph(T;to;from)
. | 6. ∀i,j:|T|.
. | |  ∃p:Edge(T) List.
. | |  lconnects(p;i;j) & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. | 7. L : |T| List
. | 8. ∀i:|T|. (i ∈ L)
. | 9. j : |T|
. | 10. bi-graph(T;to;from)
. | 11. spanner(f;T;to;from)
. | 12. n : ℕ
. | 13. ∀p:Edge(T) List. lpath(p) ⇒ ||p||≤n
. | 14. ¬||to(j)|| = 0
. | ⊢  Dec(∃i:|T|. spanner-root(f;T;to;from;i))
. | | by Assert
. | |    ((∃i:|T|. spanner-root(f;T;to;from;i))
. | |     ⇐⇒
. | |     (∃i∈L.spanner-root(f;T;to;from;i)))
. | |

. | |\
. | | .....assertion .....
. | | 1. T : Id→𝔹
. | | 2. to : |T|→(IdLnk List)
. | | 3. from : |T|→(IdLnk List)
. | | 4. f : Edge(T)→𝔹
. | | 5. bi-graph(T;to;from)
. | | 6. ∀i,j:|T|.
```

```
. | |    ∃p:Edge(T) List.
. | |    lconnects(p;i;j)
. | |    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. | | 7. L : |T| List
. | | 8. ∀i:|T|. (i ∈ L)
. | | 9. j : |T|
. | | 10. bi-graph(T;to;from)
. | | 11. spanner(f;T;to;from)
. | | 12. n : ℕ
. | | 13. ∀p:Edge(T) List. lpath(p) ⇒ ||p||≤n
. | | 14. ¬||to(j)|| = 0
. | | ⊢  (∃i:|T|. spanner-root(f;T;to;from;i))
. | |    ⟺
. | |    (∃i∈L.spanner-root(f;T;to;from;i))
. | |   by Unfold `l_exists` 0 THEN ParallelLast THEN BackThruSomeHyp
. | |
. | .\
. | . 1. T : Id→ℬ
. | . 2. to : |T|→(IdLnk List)
. | . 3. from : |T|→(IdLnk List)
. | . 4. f : Edge(T)→ℬ
. | . 5. bi-graph(T;to;from)
. | . 6. ∀i,j:|T|.
. | . |  ∃p:Edge(T) List.
. | . |  lconnects(p;i;j)
. | . |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. | . 7. L : |T| List
. | . 8. ∀i:|T|. (i ∈ L)
. | . 9. j : |T|
. | . 10. bi-graph(T;to;from)
. | . 11. spanner(f;T;to;from)
. | . 12. n : ℕ
. | . 13. ∀p:Edge(T) List. lpath(p) ⇒ ||p||≤n
. | . 14. ¬||to(j)|| = 0
. | . 15. (∃i:|T|. spanner-root(f;T;to;from;i))
. | . |   ⟺
. | . |   (∃i∈L.spanner-root(f;T;to;from;i))
. | . ⊢  Dec(∃i:|T|. spanner-root(f;T;to;from;i))
. | . | by RW (AddrC [1] (HypC -1)) 0
. | . |
. | . 1. T : Id→ℬ
. | . 2. to : |T|→(IdLnk List)
. | . 3. from : |T|→(IdLnk List)
. | . 4. f : Edge(T)→ℬ
. | . 5. bi-graph(T;to;from)
. | . 6. ∀i,j:|T|.
. | .    ∃p:Edge(T) List.
. | .    lconnects(p;i;j)
. | .    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. | . 7. L : |T| List
. | . 8. ∀i:|T|. (i ∈ L)
. | . 9. j : |T|
. | . 10. bi-graph(T;to;from)
. | . 11. spanner(f;T;to;from)
. | . 12. n : ℕ
```

```
. | . 13. ∀p:Edge(T) List. lpath(p) ⇒ ||p||≤n
. | . 14. ¬||to(j)|| = 0
. | . 15. (∃i:|T|. spanner-root(f;T;to;from;i))
. | .     ⟺
. | .     (∃i∈L.spanner-root(f;T;to;from;i))
. | . ⊢  Dec((∃i∈L.spanner-root(f;T;to;from;i)))
. | .   by Unfold ‘spanner-root‘ 0 THEN Fold ‘l_all‘ 0
. |
. .\
. . 1. T : Id→𝔹
. . 2. to : |T|→(IdLnk List)
. . 3. from : |T|→(IdLnk List)
. . 4. f : Edge(T)→𝔹
. . 5. bi-graph(T;to;from)
. . 6. ∀i,j:|T|.
. . |  ∃p:Edge(T) List.
. . |  lconnects(p;i;j) & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . 7. L : |T| List
. . 8. ∀i:|T|. (i ∈ L)
. . 9. j : |T|
. . 10. bi-graph(T;to;from)
. . 11. spanner(f;T;to;from)
. . 12. n : ℕ
. . 13. ∀p:Edge(T) List. lpath(p) ⇒ ||p||≤n
. . 14. ¬||to(j)|| = 0
. . 15. Dec(∃i:|T|. spanner-root(f;T;to;from;i))
. . ⊢  ∃i:|T|. spanner-root(f;T;to;from;i)
. . | by Analyze -1 THEN Try Trivial
. . |
. . 1. T : Id→𝔹
. . 2. to : |T|→(IdLnk List)
. . 3. from : |T|→(IdLnk List)
. . 4. f : Edge(T)→𝔹
. . 5. bi-graph(T;to;from)
. . 6. ∀i,j:|T|.
. . |  ∃p:Edge(T) List.
. . |  lconnects(p;i;j) & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . 7. L : |T| List
. . 8. ∀i:|T|. (i ∈ L)
. . 9. j : |T|
. . 10. bi-graph(T;to;from)
. . 11. spanner(f;T;to;from)
. . 12. n : ℕ
. . 13. ∀p:Edge(T) List. lpath(p) ⇒ ||p||≤n
. . 14. ¬||to(j)|| = 0
. . 15. ¬(∃i:|T|. spanner-root(f;T;to;from;i))
. . ⊢  ∃i:|T|. spanner-root(f;T;to;from;i)
. . | by Assert False THEN Unfold ‘spanner-root‘ -1
. . |    THEN
. . |    Assert (∀i:|T|. (∃l∈to(i).¬f(l)))
. . |
. . |\
. . | .....assertion .....
. . | 1. T : Id→𝔹
. . | 2. to : |T|→(IdLnk List)
```

```
. . | 3. from : |T|→(IdLnk List)
. . | 4. f : Edge(T)→ℬ
. . | 5. bi-graph(T;to;from)
. . | 6. ∀i,j:|T|.
. . | |  ∃p:Edge(T) List.
. . | |  lconnects(p;i;j)
. . | |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . | 7. L : |T| List
. . | 8. ∀i:|T|. (i ∈ L)
. . | 9. j : |T|
. . | 10. bi-graph(T;to;from)
. . | 11. spanner(f;T;to;from)
. . | 12. n : ℕ
. . | 13. ∀p:Edge(T) List. lpath(p) ⇒ ||p||≤n
. . | 14. ¬||to(j)|| = 0
. . | 15. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . | ⊢  ∀i:|T|. (∃l∈to(i).¬f(l))
. . | | by Auto THEN SupposeNot THEN Analyze -3
. . | |
. . | 1. T : Id→ℬ
. . | 2. to : |T|→(IdLnk List)
. . | 3. from : |T|→(IdLnk List)
. . | 4. f : Edge(T)→ℬ
. . | 5. bi-graph(T;to;from)
. . | 6. ∀i,j:|T|.
. . | |  ∃p:Edge(T) List.
. . | |  lconnects(p;i;j)
. . | |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . | 7. L : |T| List
. . | 8. ∀i:|T|. (i ∈ L)
. . | 9. j : |T|
. . | 10. bi-graph(T;to;from)
. . | 11. spanner(f;T;to;from)
. . | 12. n : ℕ
. . | 13. ∀p:Edge(T) List. lpath(p) ⇒ ||p||≤n
. . | 14. ¬||to(j)|| = 0
. . | 15. i : |T|
. . | 16. ¬(∃l∈to(i).¬f(l))
. . | ⊢  ∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l)
. . | | by InstConcl [i] THEN SupposeNot
. . | |
. . | 1. T : Id→ℬ
. . | 2. to : |T|→(IdLnk List)
. . | 3. from : |T|→(IdLnk List)
. . | 4. f : Edge(T)→ℬ
. . | 5. bi-graph(T;to;from)
. . | 6. ∀i,j:|T|.
. . |    ∃p:Edge(T) List.
. . |    lconnects(p;i;j)
. . |    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . | 7. L : |T| List
. . | 8. ∀i:|T|. (i ∈ L)
. . | 9. j : |T|
. . | 10. bi-graph(T;to;from)
. . | 11. spanner(f;T;to;from)
```

```
. . | 12. n : ℕ
. . | 13. ∀p:Edge(T) List. lpath(p) ⇒ ||p||≤n
. . | 14. ¬||to(j)|| = 0
. . | 15. i : |T|
. . | 16. ¬(∃l∈to(i).¬f(l))
. . | 17. l : Edge(T)
. . | 18. (l ∈ to(i))
. . | 19. ¬f(l)
. . | ⊢  f(l)
. . |   by Analyze -4 THEN Unfold `l_exists` 0 THEN InstConcl [l]
. . |
. . .\
. . . 1. T : Id→𝔹
. . . 2. to : |T|→(IdLnk List)
. . . 3. from : |T|→(IdLnk List)
. . . 4. f : Edge(T)→𝔹
. . . 5. bi-graph(T;to;from)
. . . 6. ∀i,j:|T|.
. . . |  ∃p:Edge(T) List.
. . . |  lconnects(p;i;j)
. . . |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . 7. L : |T| List
. . . 8. ∀i:|T|. (i ∈ L)
. . . 9. j : |T|
. . . 10. bi-graph(T;to;from)
. . . 11. spanner(f;T;to;from)
. . . 12. n : ℕ
. . . 13. ∀p:Edge(T) List. lpath(p) ⇒ ||p||≤n
. . . 14. ¬||to(j)|| = 0
. . . 15. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . 16. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . ⊢  False
. . . | by Assert
. . . |    (∀n:ℕ.
. . . |     ∃p:Edge(T) List. lpath(p) & ||p|| = n & (∀l∈p.¬f(l)))
. . . |
. . . |\
. . . | .....assertion .....
. . . | 1. T : Id→𝔹
. . . | 2. to : |T|→(IdLnk List)
. . . | 3. from : |T|→(IdLnk List)
. . . | 4. f : Edge(T)→𝔹
. . . | 5. bi-graph(T;to;from)
. . . | 6. ∀i,j:|T|.
. . . | |  ∃p:Edge(T) List.
. . . | |  lconnects(p;i;j)
. . . | |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | 7. L : |T| List
. . . | 8. ∀i:|T|. (i ∈ L)
. . . | 9. j : |T|
. . . | 10. bi-graph(T;to;from)
. . . | 11. spanner(f;T;to;from)
. . . | 12. n : ℕ
. . . | 13. ∀p:Edge(T) List. lpath(p) ⇒ ||p||≤n
. . . | 14. ¬||to(j)|| = 0
```

```
. . . | 15. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | 16. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | ⊢  ∀n:ℕ. ∃p:Edge(T) List. lpath(p) & ||p|| = n & (∀l∈p.¬f(l))
. . . | | by Thin -4 THEN Thin -4 THEN InductionOnNat
. . . | |
. . . | |\
. . . | | .....basecase .....
. . . | | 1. T : Id→ℙ
. . . | | 2. to : |T|→(IdLnk List)
. . . | | 3. from : |T|→(IdLnk List)
. . . | | 4. f : Edge(T)→ℙ
. . . | | 5. bi-graph(T;to;from)
. . . | | 6. ∀i,j:|T|.
. . . | |    ∃p:Edge(T) List.
. . . | |    lconnects(p;i;j)
. . . | |    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | | 7. L : |T| List
. . . | | 8. ∀i:|T|. (i ∈ L)
. . . | | 9. j : |T|
. . . | | 10. bi-graph(T;to;from)
. . . | | 11. spanner(f;T;to;from)
. . . | | 12. ¬||to(j)|| = 0
. . . | | 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | | 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | | ⊢  ∃p:Edge(T) List. lpath(p) & ||p|| = 0 & (∀l∈p.¬f(l))
. . . | |   by Repeat
. . . | |      (Unfolds [‘l_all‘;‘lpath‘;‘l_member‘] 0 THEN Reduce 0)
. . . | |      THEN
. . . | |      InstConcl [nil]
. . . | |      THEN
. . . | |      Reduce 0
. . . | |      THEN
. . . | |      MaAuto
. . . | |      THEN
. . . | |      ExRepD
. . . | |
. . . | .\
. . . | . .....upcase .....
. . . | . 1. T : Id→ℙ
. . . | . 2. to : |T|→(IdLnk List)
. . . | . 3. from : |T|→(IdLnk List)
. . . | . 4. f : Edge(T)→ℙ
. . . | . 5. bi-graph(T;to;from)
. . . | . 6. ∀i,j:|T|.
. . . | . |  ∃p:Edge(T) List.
. . . | . |  lconnects(p;i;j)
. . . | . |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | . 7. L : |T| List
. . . | . 8. ∀i:|T|. (i ∈ L)
. . . | . 9. j : |T|
. . . | . 10. bi-graph(T;to;from)
. . . | . 11. spanner(f;T;to;from)
. . . | . 12. ¬||to(j)|| = 0
. . . | . 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . 14. ∀i:|T|. (∃l∈to(i).¬f(l))
```

```
. . . | . 15. n : ℤ
. . . | . 16. 0<n
. . . | . 17. ∃p:Edge(T) List. lpath(p) & ||p|| = n-1 & (∀l∈p.¬f(l))
. . . | . ⊢  ∃p:Edge(T) List. lpath(p) & ||p|| = n & (∀l∈p.¬f(l))
. . . | . | by ExRepD
. . . | . |
. . . | . 1. T : Id→𝔹
. . . | . 2. to : |T|→(IdLnk List)
. . . | . 3. from : |T|→(IdLnk List)
. . . | . 4. f : Edge(T)→𝔹
. . . | . 5. bi-graph(T;to;from)
. . . | . 6. ∀i,j:|T|.
. . . | . |  ∃p:Edge(T) List.
. . . | . |  lconnects(p;i;j)
. . . | . |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | . 7. L : |T| List
. . . | . 8. ∀i:|T|. (i ∈ L)
. . . | . 9. j : |T|
. . . | . 10. bi-graph(T;to;from)
. . . | . 11. spanner(f;T;to;from)
. . . | . 12. ¬||to(j)|| = 0
. . . | . 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | . 15. n : ℤ
. . . | . 16. 0<n
. . . | . 17. p : Edge(T) List
. . . | . 18. lpath(p)
. . . | . 19. ||p|| = n-1
. . . | . 20. (∀l∈p.¬f(l))
. . . | . ⊢  ∃p:Edge(T) List. lpath(p) & ||p|| = n & (∀l∈p.¬f(l))
. . . | . | by DVar ‘p’
. . . | . |
. . . | . |\
. . . | . | 1. T : Id→𝔹
. . . | . | 2. to : |T|→(IdLnk List)
. . . | . | 3. from : |T|→(IdLnk List)
. . . | . | 4. f : Edge(T)→𝔹
. . . | . | 5. bi-graph(T;to;from)
. . . | . | 6. ∀i,j:|T|.
. . . | . | |  ∃p:Edge(T) List.
. . . | . | |  lconnects(p;i;j)
. . . | . | |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | . | 7. L : |T| List
. . . | . | 8. ∀i:|T|. (i ∈ L)
. . . | . | 9. j : |T|
. . . | . | 10. bi-graph(T;to;from)
. . . | . | 11. spanner(f;T;to;from)
. . . | . | 12. ¬||to(j)|| = 0
. . . | . | 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . | 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | . | 15. n : ℤ
. . . | . | 16. 0<n
. . . | . | 17. lpath(nil)
. . . | . | 18. ||nil|| = n-1
. . . | . | 19. (∀l∈nil.¬f(l))
```

```
. . . | . | ⊢  ∃p:Edge(T) List. lpath(p) & ||p|| = n & (∀l∈p.¬f(l))
. . . | . | | by InstHyp [j] -6 THEN Analyze -1 THEN InstConcl [[l]]
. . . | . | |    THEN
. . . | . | |    Reduce 0
. . . | . | |    THEN
. . . | . | |    MaAuto
. . . | . | |
. . . | . | |\
. . . | . | | 1. T : Id→𝔹
. . . | . | | 2. to : |T|→(IdLnk List)
. . . | . | | 3. from : |T|→(IdLnk List)
. . . | . | | 4. f : Edge(T)→𝔹
. . . | . | | 5. bi-graph(T;to;from)
. . . | . | | 6. ∀i,j:|T|.
. . . | . | |    ∃p:Edge(T) List.
. . . | . | |    lconnects(p;i;j)
. . . | . | |    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | . | | 7. L : |T| List
. . . | . | | 8. ∀i:|T|. (i ∈ L)
. . . | . | | 9. j : |T|
. . . | . | | 10. bi-graph(T;to;from)
. . . | . | | 11. spanner(f;T;to;from)
. . . | . | | 12. ¬||to(j)|| = 0
. . . | . | | 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . | | 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | . | | 15. n : ℤ
. . . | . | | 16. 0<n
. . . | . | | 17. lpath(nil)
. . . | . | | 18. ||nil|| = n-1
. . . | . | | 19. (∀l∈nil.¬f(l))
. . . | . | | 20. l : Edge(T)
. . . | . | | 21. (l ∈ to(j))
. . . | . | | 22. ¬f(l)
. . . | . | | ⊢  lpath([l])  by Unfold ‘lpath‘ 0 THEN Reduce 0
. . . | . | |
. . . | . | .\
. . . | . | . 1. T : Id→𝔹
. . . | . | . 2. to : |T|→(IdLnk List)
. . . | . | . 3. from : |T|→(IdLnk List)
. . . | . | . 4. f : Edge(T)→𝔹
. . . | . | . 5. bi-graph(T;to;from)
. . . | . | . 6. ∀i,j:|T|.
. . . | . | .    ∃p:Edge(T) List.
. . . | . | .    lconnects(p;i;j)
. . . | . | .    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | . | . 7. L : |T| List
. . . | . | . 8. ∀i:|T|. (i ∈ L)
. . . | . | . 9. j : |T|
. . . | . | . 10. bi-graph(T;to;from)
. . . | . | . 11. spanner(f;T;to;from)
. . . | . | . 12. ¬||to(j)|| = 0
. . . | . | . 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . | . 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | . | . 15. n : ℤ
. . . | . | . 16. 0<n
```

```
. . . | . | . 17. lpath(nil)
. . . | . | . 18. ||nil|| = n-1
. . . | . | . 19. (∀l∈nil.¬f(l))
. . . | . | . 20. l : Edge(T)
. . . | . | . 21. (l ∈ to(j))
. . . | . | . 22. ¬f(l)
. . . | . | . ⊢  (∀l∈[l].¬f(l))

. . . | . | .   by RWO
. . . | . | .      Thm* ∀P:(T→Prop), x:T, L:T List.
. . . | . | .           (∀y∈[x / L].P(y)) ⟺ P(x) & (∀y∈L.P(y))
. . . | . | .      {l_all_cons}
. . . | . | .      0
. . . | . |
. . . | . .\
. . . | . . 1. T : Id→𝔹
. . . | . . 2. to : |T|→(IdLnk List)
. . . | . . 3. from : |T|→(IdLnk List)
. . . | . . 4. f : Edge(T)→𝔹
. . . | . . 5. bi-graph(T;to;from)
. . . | . . 6. ∀i,j:|T|.
. . . | . . |  ∃p:Edge(T) List.
. . . | . . |  lconnects(p;i;j)
. . . | . . |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | . . 7. L : |T| List
. . . | . . 8. ∀i:|T|. (i ∈ L)
. . . | . . 9. j : |T|
. . . | . . 10. bi-graph(T;to;from)
. . . | . . 11. spanner(f;T;to;from)
. . . | . . 12. ¬||to(j)|| = 0
. . . | . . 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . . 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | . . 15. n : ℤ
. . . | . . 16. 0<n
. . . | . . 17. u : Edge(T)
. . . | . . 18. v : Edge(T) List
. . . | . . 19. lpath([u / v])
. . . | . . 20. ||[u / v]|| = n-1
. . . | . . 21. (∀l∈[u / v].¬f(l))
. . . | . . ⊢  ∃p:Edge(T) List. lpath(p) & ||p|| = n & (∀l∈p.¬f(l))
. . . | . . | by RWO
. . . | . . |    Thm* ∀P:(T→Prop), x:T, L:T List.
. . . | . . |         (∀y∈[x / L].P(y)) ⟺ P(x) & (∀y∈L.P(y))
. . . | . . |    {l_all_cons}
. . . | . . |    -1
. . . | . . |    THEN
. . . | . . |    AssertBY (∃l∈to(source(u)).¬f(l))
. . . | . . |    (BackThruSomeHyp
. . . | . . |     THEN
. . . | . . |     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
. . . | . . |    Analyze -1
```

```
. . . | . . |    THEN
. . . | . . |    InstConcl [[l; u / v]]
. . . | . . |    THEN
. . . | . . |    All Reduce
. . . | . . |    THEN
. . . | . . |    MaAuto
. . . | . . |
. . . | . . |\
. . . | . . | 1. T : Id→𝔹
. . . | . . | 2. to : |T|→(IdLnk List)
. . . | . . | 3. from : |T|→(IdLnk List)
. . . | . . | 4. f : Edge(T)→𝔹
. . . | . . | 5. bi-graph(T;to;from)
. . . | . . | 6. ∀i,j:|T|.
. . . | . . | |  ∃p:Edge(T) List.
. . . | . . | |  lconnects(p;i;j)
. . . | . . | |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | . . | 7. L : |T| List
. . . | . . | 8. ∀i:|T|. (i ∈ L)
. . . | . . | 9. j : |T|
. . . | . . | 10. bi-graph(T;to;from)
. . . | . . | 11. spanner(f;T;to;from)
. . . | . . | 12. ¬||to(j)|| = 0
. . . | . . | 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . . | 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | . . | 15. n : ℤ
. . . | . . | 16. 0<n
. . . | . . | 17. u : Edge(T)
. . . | . . | 18. v : Edge(T) List
. . . | . . | 19. lpath([u / v])
. . . | . . | 20. ||v||+1 = n-1
. . . | . . | 21. ¬f(u)
. . . | . . | 22. (∀l∈v.¬f(l))
. . . | . . | 23. l : Edge(T)
. . . | . . | 24. (l ∈ to(source(u)))
. . . | . . | 25. ¬f(l)
. . . | . . | ⊢  lpath([l; u / v])
. . . | . . | | by RW0
. . . | . . | |    Thm* ∀l:IdLnk, p:IdLnk List.
. . . | . . | |         lpath([l / p])
. . . | . . | |         ⟺
. . . | . . | |         lpath(p)
. . . | . . | |         & (¬||p|| = 0 ∈ ℤ
. . . | . . | |            ⇒
. . . | . . | |            destination(l) = source(hd(p))
. . . | . . | |            & ¬hd(p) = lnk-inv(l))
. . . | . . | |    {lpath_cons}
. . . | . . | |    0
. . . | . . | |    THEN
. . . | . . | |    MaAuto
. . . | . . | |    THEN
. . . | . . | |    Reduce 0
. . . | . . | |
. . . | . . | |\
. . . | . . | | 1. T : Id→𝔹
```

```
. . . | . . | | 2. to : |T|→(IdLnk List)
. . . | . . | | 3. from : |T|→(IdLnk List)
. . . | . . | | 4. f : Edge(T)→𝔹
. . . | . . | | 5. bi-graph(T;to;from)
. . . | . . | | 6. ∀i,j:|T|.
. . . | . . | | |  ∃p:Edge(T) List.
. . . | . . | | |  lconnects(p;i;j)
. . . | . . | | |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | . . | | 7. L : |T| List
. . . | . . | | 8. ∀i:|T|. (i ∈ L)
. . . | . . | | 9. j : |T|
. . . | . . | | 10. bi-graph(T;to;from)
. . . | . . | | 11. spanner(f;T;to;from)
. . . | . . | | 12. ¬||to(j)|| = 0
. . . | . . | | 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . . | | 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | . . | | 15. n : ℤ
. . . | . . | | 16. 0<n
. . . | . . | | 17. u : Edge(T)
. . . | . . | | 18. v : Edge(T) List
. . . | . . | | 19. lpath([u / v])
. . . | . . | | 20. ||v||+1 = n-1
. . . | . . | | 21. ¬f(u)
. . . | . . | | 22. (∀l∈v.¬f(l))
. . . | . . | | 23. l : Edge(T)
. . . | . . | | 24. (l ∈ to(source(u)))
. . . | . . | | 25. ¬f(l)
. . . | . . | | 26. ¬||[u / v]|| = 0
. . . | . . | | ⊢  destination(l) = source(u)
. . . | . . | | | by DupHyp 5 THEN Unfold ‘bi-graph‘ -1
. . . | . . | | |    THEN
. . . | . . | | |    InstHyp [source(u)] -1
. . . | . . | | |    THEN
. . . | . . | | |    MaAuto
. . . | . . | | |
. . . | . . | | |\
. . . | . . | | | .....wf .....
. . . | . . | | | 1. T : Id→𝔹
. . . | . . | | | 2. to : |T|→(IdLnk List)
. . . | . . | | | 3. from : |T|→(IdLnk List)
. . . | . . | | | 4. f : Edge(T)→𝔹
. . . | . . | | | 5. bi-graph(T;to;from)
. . . | . . | | | 6. ∀i,j:|T|.
. . . | . . | | |    ∃p:Edge(T) List.
. . . | . . | | |    lconnects(p;i;j)
. . . | . . | | |    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | . . | | | 7. L : |T| List
. . . | . . | | | 8. ∀i:|T|. (i ∈ L)
. . . | . . | | | 9. j : |T|
. . . | . . | | | 10. bi-graph(T;to;from)
. . . | . . | | | 11. spanner(f;T;to;from)
. . . | . . | | | 12. ¬||to(j)|| = 0
. . . | . . | | | 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . . | | | 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | . . | | | 15. n : ℤ
```

```
. . . | . . | | 16. 0<n
. . . | . . | | 17. u : Edge(T)
. . . | . . | | 18. v : Edge(T) List
. . . | . . | | 19. lpath([u / v])
. . . | . . | | 20. ||v||+1 = n-1
. . . | . . | | 21. ¬f(u)
. . . | . . | | 22. (∀l∈v.¬f(l))
. . . | . . | | 23. l : Edge(T)
. . . | . . | | 24. (l ∈ to(source(u)))
. . . | . . | | 25. ¬f(l)
. . . | . . | | 26. ¬||[u / v]|| = 0
. . . | . . | | 27. ∀i:|T|.
. . . | . . | |     (∀l∈to(i).destination(l) = i
. . . | . . | |     & T(source(l))
. . . | . . | |     & (l ∈ from(source(l)))
. . . | . . | |     & (lnk-inv(l) ∈ from(i)))
. . . | . . | |     & (∀l∈from(i).source(l) = i
. . . | . . | |       & T(destination(l))
. . . | . . | |       & (l ∈ to(destination(l)))
. . . | . . | |       & (lnk-inv(l) ∈ to(i)))
. . . | . . | | ⊢  source(u) ∈ |T|
. . . | . . | |   by 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. f : Edge(T)→𝔹
. . . | . . | . 5. bi-graph(T;to;from)
. . . | . . | . 6. ∀i,j:|T|.
. . . | . . | . |  ∃p:Edge(T) List.
. . . | . . | . |  lconnects(p;i;j)
. . . | . . | . |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | . . | . 7. L : |T| List
. . . | . . | . 8. ∀i:|T|. (i ∈ L)
. . . | . . | . 9. j : |T|
. . . | . . | . 10. bi-graph(T;to;from)
. . . | . . | . 11. spanner(f;T;to;from)
. . . | . . | . 12. ¬||to(j)|| = 0
. . . | . . | . 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . . | . 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | . . | . 15. n : ℤ
. . . | . . | . 16. 0<n
. . . | . . | . 17. u : Edge(T)
. . . | . . | . 18. v : Edge(T) List
. . . | . . | . 19. lpath([u / v])
. . . | . . | . 20. ||v||+1 = n-1
. . . | . . | . 21. ¬f(u)
. . . | . . | . 22. (∀l∈v.¬f(l))
. . . | . . | . 23. l : Edge(T)
. . . | . . | . 24. (l ∈ to(source(u)))
```

```
. . . | . . | | . 25. ¬f(l)
. . . | . . | | . 26. ¬||[u / v]|| = 0
. . . | . . | | . 27. ∀i:|T|.
. . . | . . | | . |   (∀l∈to(i).destination(l) = i
. . . | . . | | . |   & T(source(l))
. . . | . . | | . |   & (l ∈ from(source(l)))
. . . | . . | | . |   & (lnk-inv(l) ∈ from(i)))
. . . | . . | | . |   & (∀l∈from(i).source(l) = i
. . . | . . | | . |     & T(destination(l))
. . . | . . | | . |     & (l ∈ to(destination(l)))
. . . | . . | | . |     & (lnk-inv(l) ∈ to(i)))
. . . | . . | | . 28. (∀l∈to(source(u)).destination(l) = source(u)
. . . | . . | | . |   & T(source(l))
. . . | . . | | . |   & (l ∈ from(source(l)))
. . . | . . | | . |   & (lnk-inv(l) ∈ from(source(u))))
. . . | . . | | . 29. (∀l∈from(source(u)).source(l) = source(u)
. . . | . . | | . |   & T(destination(l))
. . . | . . | | . |   & (l ∈ to(destination(l)))
. . . | . . | | . |   & (lnk-inv(l) ∈ to(source(u))))
. . . | . . | | . ⊢  destination(l) = source(u)
. . . | . . | | . | by Unfold ‘l_all‘ -2 THEN InstHyp [l] -2
. . . | . . | | . |    THEN
. . . | . . | | . |    MaAuto
. . . | . . | | . |    THEN
. . . | . . | | . |    All (Unfold ‘bi-graph-to‘)
. . . | . . | | . |
. . . | . . | | . 1. T : Id→𝔹
. . . | . . | | . 2. to : |T|→(IdLnk List)
. . . | . . | | . 3. from : |T|→(IdLnk List)
. . . | . . | | . 4. f : Edge(T)→𝔹
. . . | . . | | . 5. bi-graph(T;to;from)
. . . | . . | | . 6. ∀i,j:|T|.
. . . | . . | | . |  ∃p:Edge(T) List.
. . . | . . | | . |  lconnects(p;i;j)
. . . | . . | | . |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | . . | | . 7. L : |T| List
. . . | . . | | . 8. ∀i:|T|. (i ∈ L)
. . . | . . | | . 9. j : |T|
. . . | . . | | . 10. bi-graph(T;to;from)
. . . | . . | | . 11. spanner(f;T;to;from)
. . . | . . | | . 12. ¬||to(j)|| = 0
. . . | . . | | . 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . . | | . 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | . . | | . 15. n : ℤ
. . . | . . | | . 16. 0<n
. . . | . . | | . 17. u : Edge(T)
. . . | . . | | . 18. v : Edge(T) List
. . . | . . | | . 19. lpath([u / v])
. . . | . . | | . 20. ||v||+1 = n-1
. . . | . . | | . 21. ¬f(u)
. . . | . . | | . 22. (∀l∈v.¬f(l))
. . . | . . | | . 23. l : Edge(T)
. . . | . . | | . 24. (l ∈ to(source(u)))
. . . | . . | | . 25. ¬f(l)
. . . | . . | | . 26. ¬||[u / v]|| = 0
```

```
. . . | . . | | . 27. ∀i:|T|.
. . . | . . | | . |   (∀l∈to(i).destination(l) = i
. . . | . . | | . |   & T(source(l))
. . . | . . | | . |   & (l ∈ from(source(l)))
. . . | . . | | . |   & (lnk-inv(l) ∈ from(i)))
. . . | . . | | . |   & (∀l∈from(i).source(l) = i
. . . | . . | | . |     & T(destination(l))
. . . | . . | | . |     & (l ∈ to(destination(l)))
. . . | . . | | . |     & (lnk-inv(l) ∈ to(i)))
. . . | . . | | . 28. ∀l:IdLnk.
. . . | . . | | . |   (l ∈ to(source(u)))
. . . | . . | | . |   ⇒
. . . | . . | | . |   destination(l) = source(u)
. . . | . . | | . |   & T(source(l))
. . . | . . | | . |   & (l ∈ from(source(l)))
. . . | . . | | . |   & (lnk-inv(l) ∈ from(source(u)))
. . . | . . | | . 29. (∀l∈from(source(u)).source(l) = source(u)
. . . | . . | | . |   & T(destination(l))
. . . | . . | | . |   & (l ∈ to(destination(l)))
. . . | . . | | . |   & (lnk-inv(l) ∈ to(source(u))))
. . . | . . | | . ⊢  (l ∈ to(source(u)))
. . . | . . | | . | by Using [‘A’,Edge(T)]
. . . | . . | | . |    (BackThru
. . . | . . | | . |     Thm* ∀L:A List, x:A.
. . . | . . | | . |          (A ⊆r B) ⇒ (x ∈ L) ⇒ (x ∈ L)
. . . | . . | | . |     {l_member_subtype})
. . . | . . | | . |    THEN
. . . | . . | | . |    Try (Complete Auto)
. . . | . . | | . |
. . . | . . | | . |\
. . . | . . | | . | .....wf .....
. . . | . . | | . | 1. T : Id→𝔹
. . . | . . | | . | 2. to : |T|→(IdLnk List)
. . . | . . | | . | 3. from : |T|→(IdLnk List)
. . . | . . | | . | 4. f : Edge(T)→𝔹
. . . | . . | | . | 5. bi-graph(T;to;from)
. . . | . . | | . | 6. ∀i,j:|T|.
. . . | . . | | . | |  ∃p:Edge(T) List.
. . . | . . | | . | |  lconnects(p;i;j)
. . . | . . | | . | |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | . . | | . | 7. L : |T| List
. . . | . . | | . | 8. ∀i:|T|. (i ∈ L)
. . . | . . | | . | 9. j : |T|
. . . | . . | | . | 10. bi-graph(T;to;from)
. . . | . . | | . | 11. spanner(f;T;to;from)
. . . | . . | | . | 12. ¬||to(j)|| = 0
. . . | . . | | . | 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . . | | . | 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | . . | | . | 15. n : ℤ
. . . | . . | | . | 16. 0<n
. . . | . . | | . | 17. u : Edge(T)
. . . | . . | | . | 18. v : Edge(T) List
. . . | . . | | . | 19. lpath([u / v])
. . . | . . | | . | 20. ||v||+1 = n-1
. . . | . . | | . | 21. ¬f(u)
```

```
. . . | . . | | . | 22. (∀l∈v.¬f(l))
. . . | . . | | . | 23. l : Edge(T)
. . . | . . | | . | 24. (l ∈ to(source(u)))
. . . | . . | | . | 25. ¬f(l)
. . . | . . | | . | 26. ¬||[u / v]|| = 0
. . . | . . | | . | 27. ∀i:|T|.
. . . | . . | | . | |   (∀l∈to(i).destination(l) = i
. . . | . . | | . | |   & T(source(l))
. . . | . . | | . | |   & (l ∈ from(source(l)))
. . . | . . | | . | |   & (lnk-inv(l) ∈ from(i)))
. . . | . . | | . | |   & (∀l∈from(i).source(l) = i
. . . | . . | | . | |     & T(destination(l))
. . . | . . | | . | |     & (l ∈ to(destination(l)))
. . . | . . | | . | |     & (lnk-inv(l) ∈ to(i)))
. . . | . . | | . | 28. ∀l:IdLnk.
. . . | . . | | . | |   (l ∈ to(source(u)))
. . . | . . | | . | |   ⇒
. . . | . . | | . | |   destination(l) = source(u)
. . . | . . | | . | |   & T(source(l))
. . . | . . | | . | |   & (l ∈ from(source(l)))
. . . | . . | | . | |   & (lnk-inv(l) ∈ from(source(u)))
. . . | . . | | . | 29. (∀l∈from(source(u)).source(l) = source(u)
. . . | . . | | . | |   & T(destination(l))
. . . | . . | | . | |   & (l ∈ to(destination(l)))
. . . | . . | | . | |   & (lnk-inv(l) ∈ to(source(u))))
. . . | . . | | . | ⊢  to(source(u)) ∈ Edge(T) List
. . . | . . | | . | | by Subst ((to(source(u))) ∼ to(source(u))) 0
. . . | . . | | . | |
. . . | . . | | . | |\
. . . | . . | | . | | .....equality .....
. . . | . . | | . | | 1. T : Id→𝔹
. . . | . . | | . | | 2. to : |T|→(IdLnk List)
. . . | . . | | . | | 3. from : |T|→(IdLnk List)
. . . | . . | | . | | 4. f : Edge(T)→𝔹
. . . | . . | | . | | 5. bi-graph(T;to;from)
. . . | . . | | . | | 6. ∀i,j:|T|.
. . . | . . | | . | |    ∃p:Edge(T) List.
. . . | . . | | . | |    lconnects(p;i;j)
. . . | . . | | . | |    & (∀q:Edge(T) List.
. . . | . . | | . | |       lconnects(q;i;j) ⇒ q = p)
. . . | . . | | . | | 7. L : |T| List
. . . | . . | | . | | 8. ∀i:|T|. (i ∈ L)
. . . | . . | | . | | 9. j : |T|
. . . | . . | | . | | 10. bi-graph(T;to;from)
. . . | . . | | . | | 11. spanner(f;T;to;from)
. . . | . . | | . | | 12. ¬||to(j)|| = 0
. . . | . . | | . | | 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . . | | . | | 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | . . | | . | | 15. n : ℤ
. . . | . . | | . | | 16. 0<n
. . . | . . | | . | | 17. u : Edge(T)
. . . | . . | | . | | 18. v : Edge(T) List
. . . | . . | | . | | 19. lpath([u / v])
. . . | . . | | . | | 20. ||v||+1 = n-1
. . . | . . | | . | | 21. ¬f(u)
```

```
. . . | . . | . | . | 22. (∀l∈v.¬f(l))
. . . | . . | . | . | 23. l : Edge(T)
. . . | . . | . | . | 24. (l ∈ to(source(u)))
. . . | . . | . | . | 25. ¬f(l)
. . . | . . | . | . | 26. ¬||[u / v]|| = 0
. . . | . . | . | . | 27. ∀i:|T|.
. . . | . . | . | .     (∀l∈to(i).destination(l) = i
. . . | . . | . | .     & T(source(l))
. . . | . . | . | .     & (l ∈ from(source(l)))
. . . | . . | . | .     & (lnk-inv(l) ∈ from(i)))
. . . | . . | . | .     & (∀l∈from(i).source(l) = i
. . . | . . | . | .       & T(destination(l))
. . . | . . | . | .       & (l ∈ to(destination(l)))
. . . | . . | . | .       & (lnk-inv(l) ∈ to(i)))
. . . | . . | . | . | 28. ∀l:IdLnk.
. . . | . . | . | .     (l ∈ to(source(u)))
. . . | . . | . | .     ⇒
. . . | . . | . | .     destination(l) = source(u)
. . . | . . | . | .     & T(source(l))
. . . | . . | . | .     & (l ∈ from(source(l)))
. . . | . . | . | .     & (lnk-inv(l) ∈ from(source(u)))
. . . | . . | . | . | 29. (∀l∈from(source(u)).source(l) = source(u)

. . . | . . | . | .     & T(destination(l))
. . . | . . | . | .     & (l ∈ to(destination(l)))
. . . | . . | . | .     & (lnk-inv(l) ∈ to(source(u))))
. . . | . . | . | . ⊢  (to(source(u))) ∼ to(source(u))
. . . | . . | . | .   by Unfold ‘bi-graph-to‘ 0
. . . | . . | . |
. . . | . . | . | .\
. . . | . . | . | . . 1. T : Id→𝔹
. . . | . . | . | . . 2. to : |T|→(IdLnk List)
. . . | . . | . | . . 3. from : |T|→(IdLnk List)
. . . | . . | . | . . 4. f : Edge(T)→𝔹
. . . | . . | . | . . 5. bi-graph(T;to;from)
. . . | . . | . | . . 6. ∀i,j:|T|.
. . . | . . | . | . .    ∃p:Edge(T) List.
. . . | . . | . | . .    lconnects(p;i;j)
. . . | . . | . | . .    & (∀q:Edge(T) List.
. . . | . . | . | . .       lconnects(q;i;j) ⇒ q = p)
. . . | . . | . | . . 7. L : |T| List
. . . | . . | . | . . 8. ∀i:|T|. (i ∈ L)
. . . | . . | . | . . 9. j : |T|
. . . | . . | . | . . 10. bi-graph(T;to;from)
. . . | . . | . | . . 11. spanner(f;T;to;from)
. . . | . . | . | . . 12. ¬||to(j)|| = 0
. . . | . . | . | . . 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . . | . | . . 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | . . | . | . . 15. n : ℤ
. . . | . . | . | . . 16. 0<n
. . . | . . | . | . . 17. u : Edge(T)
. . . | . . | . | . . 18. v : Edge(T) List
. . . | . . | . | . . 19. lpath([u / v])
. . . | . . | . | . . 20. ||v||+1 = n-1
. . . | . . | . | . . 21. ¬f(u)
. . . | . . | . | . . 22. (∀l∈v.¬f(l))
```

```
. . . | . . | | . | . 23. l : Edge(T)
. . . | . . | | . | . 24. (l ∈ to(source(u)))
. . . | . . | | . | . 25. ¬f(l)
. . . | . . | | . | . 26. ¬||[u / v]|| = 0
. . . | . . | | . | . 27. ∀i:|T|.
. . . | . . | | . | .     (∀l∈to(i).destination(l) = i
. . . | . . | | . | .     & T(source(l))
. . . | . . | | . | .     & (l ∈ from(source(l)))
. . . | . . | | . | .     & (lnk-inv(l) ∈ from(i)))
. . . | . . | | . | .     & (∀l∈from(i).source(l) = i
. . . | . . | | . | .       & T(destination(l))
. . . | . . | | . | .       & (l ∈ to(destination(l)))
. . . | . . | | . | .       & (lnk-inv(l) ∈ to(i)))
. . . | . . | | . | . 28. ∀l:IdLnk.
. . . | . . | | . | .     (l ∈ to(source(u)))
. . . | . . | | . | .     ⇒
. . . | . . | | . | .     destination(l) = source(u)
. . . | . . | | . | .     & T(source(l))
. . . | . . | | . | .     & (l ∈ from(source(l)))
. . . | . . | | . | .     & (lnk-inv(l) ∈ from(source(u)))
. . . | . . | | . | . 29. (∀l∈from(source(u)).source(l) = source(u)
. . . | . . | | . | .     & T(destination(l))
. . . | . . | | . | .     & (l ∈ to(destination(l)))
. . . | . . | | . | .     & (lnk-inv(l) ∈ to(source(u))))
. . . | . . | | . | . ⊢  to(source(u)) ∈ Edge(T) List
. . . | . . | | . | .   by Auto
. . . | . . | | . | .      THEN
. . . | . . | | . | .      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. f : Edge(T)→𝔹
. . . | . . | | . . 5. bi-graph(T;to;from)
. . . | . . | | . . 6. ∀i,j:|T|.
. . . | . . | | . .    ∃p:Edge(T) List.
. . . | . . | | . .    lconnects(p;i;j)
. . . | . . | | . .    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | . . | | . . 7. L : |T| List
. . . | . . | | . . 8. ∀i:|T|. (i ∈ L)
. . . | . . | | . . 9. j : |T|
. . . | . . | | . . 10. bi-graph(T;to;from)
. . . | . . | | . . 11. spanner(f;T;to;from)
. . . | . . | | . . 12. ¬||to(j)|| = 0
. . . | . . | | . . 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . . | | . . 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | . . | | . . 15. n : ℤ
. . . | . . | | . . 16. 0<n
. . . | . . | | . . 17. u : Edge(T)
```

```
. . . | . . | | . . 18. v : Edge(T) List
. . . | . . | | . . 19. lpath([u / v])
. . . | . . | | . . 20. ||v||+1 = n-1
. . . | . . | | . . 21. ¬f(u)
. . . | . . | | . . 22. (∀l∈v.¬f(l))
. . . | . . | | . . 23. l : Edge(T)
. . . | . . | | . . 24. (l ∈ to(source(u)))
. . . | . . | | . . 25. ¬f(l)
. . . | . . | | . . 26. ¬||[u / v]|| = 0
. . . | . . | | . . 27. ∀i:|T|.
. . . | . . | | . .     (∀l∈to(i).destination(l) = i
. . . | . . | | . .     & T(source(l))
. . . | . . | | . .     & (l ∈ from(source(l)))
. . . | . . | | . .     & (lnk-inv(l) ∈ from(i)))
. . . | . . | | . .     & (∀l∈from(i).source(l) = i
. . . | . . | | . .       & T(destination(l))
. . . | . . | | . .       & (l ∈ to(destination(l)))
. . . | . . | | . .       & (lnk-inv(l) ∈ to(i)))
. . . | . . | | . . 28. ∀l:IdLnk.
. . . | . . | | . .     (l ∈ to(source(u)))
. . . | . . | | . .     ⇒
. . . | . . | | . .     destination(l) = source(u)
. . . | . . | | . .     & T(source(l))
. . . | . . | | . .     & (l ∈ from(source(l)))
. . . | . . | | . .     & (lnk-inv(l) ∈ from(source(u)))
. . . | . . | | . . 29. (∀l∈from(source(u)).source(l) = source(u)
. . . | . . | | . .     & T(destination(l))
. . . | . . | | . .     & (l ∈ to(destination(l)))
. . . | . . | | . .     & (lnk-inv(l) ∈ to(source(u))))
. . . | . . | | . . ⊢  Edge(T) ⊆r IdLnk  by Analyze 0 THEN MaAuto
. . . | . . | |
. . . | . . | .\
. . . | . . | . 1. T : Id→𝔹
. . . | . . | . 2. to : |T|→(IdLnk List)
. . . | . . | . 3. from : |T|→(IdLnk List)
. . . | . . | . 4. f : Edge(T)→𝔹
. . . | . . | . 5. bi-graph(T;to;from)
. . . | . . | . 6. ∀i,j:|T|.
. . . | . . | . |  ∃p:Edge(T) List.
. . . | . . | . |  lconnects(p;i;j)
. . . | . . | . |  & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | . . | . 7. L : |T| List
. . . | . . | . 8. ∀i:|T|. (i ∈ L)
. . . | . . | . 9. j : |T|
. . . | . . | . 10. bi-graph(T;to;from)
. . . | . . | . 11. spanner(f;T;to;from)
. . . | . . | . 12. ¬||to(j)|| = 0
. . . | . . | . 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . . | . 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | . . | . 15. n : ℤ
. . . | . . | . 16. 0<n
. . . | . . | . 17. u : Edge(T)
. . . | . . | . 18. v : Edge(T) List
. . . | . . | . 19. lpath([u / v])
. . . | . . | . 20. ||v||+1 = n-1
```

```
. . . | . . | . 21. ¬f(u)
. . . | . . | . 22. (∀l∈v.¬f(l))
. . . | . . | . 23. l : Edge(T)
. . . | . . | . 24. (l ∈ to(source(u)))
. . . | . . | . 25. ¬f(l)
. . . | . . | . 26. ¬||[u / v]|| = 0
. . . | . . | . ⊢  ¬u = lnk-inv(l)  by Analyze 0 THENA MaAuto
. . . | . . | . |
. . . | . . | . 1. T : Id→𝔹
. . . | . . | . 2. to : |T|→(IdLnk List)
. . . | . . | . 3. from : |T|→(IdLnk List)
. . . | . . | . 4. f : Edge(T)→𝔹
. . . | . . | . 5. bi-graph(T;to;from)
. . . | . . | . 6. ∀i,j:|T|.
. . . | . . | .    ∃p:Edge(T) List.
. . . | . . | .    lconnects(p;i;j)
. . . | . . | .    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | . . | . 7. L : |T| List
. . . | . . | . 8. ∀i:|T|. (i ∈ L)
. . . | . . | . 9. j : |T|
. . . | . . | . 10. bi-graph(T;to;from)
. . . | . . | . 11. spanner(f;T;to;from)
. . . | . . | . 12. ¬||to(j)|| = 0
. . . | . . | . 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . . | . 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | . . | . 15. n : ℤ
. . . | . . | . 16. 0<n
. . . | . . | . 17. u : Edge(T)
. . . | . . | . 18. v : Edge(T) List
. . . | . . | . 19. lpath([u / v])
. . . | . . | . 20. ||v||+1 = n-1
. . . | . . | . 21. ¬f(u)
. . . | . . | . 22. (∀l∈v.¬f(l))
. . . | . . | . 23. l : Edge(T)
. . . | . . | . 24. (l ∈ to(source(u)))
. . . | . . | . 25. ¬f(l)
. . . | . . | . 26. ¬||[u / v]|| = 0
. . . | . . | . 27. u = lnk-inv(l)
. . . | . . | . ⊢  False
. . . | . . | .   by AllHyps
. . . | . . | .      (λh.
. . . | . . | .       Unfold `spanner` h THEN Analyze h
. . . | . . | .       THEN
. . . | . . | .       InstHyp [l] h
. . . | . . | .       THEN
. . . | . . | .       Unfold `bi-graph-inv` -1
. . . | . . | .       THEN
. . . | . . | .       RevHypSubst' -2 -1
. . . | . . | .       THEN
. . . | . . | .       HypSubst' -1 -4
. . . | . . | .       THEN
. . . | . . | .       RW assert_pushdownC -4)
. . . | . . |
. . . | . . .\
. . . | . . . 1. T : Id→𝔹
```

```
. . . | . . . 2. to : |T|→(IdLnk List)
. . . | . . . 3. from : |T|→(IdLnk List)

. . . | . . . 4. f : Edge(T)→𝔹
. . . | . . . 5. bi-graph(T;to;from)
. . . | . . . 6. ∀i,j:|T|.
. . . | . . .    ∃p:Edge(T) List.
. . . | . . .    lconnects(p;i;j)
. . . | . . .    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . | . . . 7. L : |T| List
. . . | . . . 8. ∀i:|T|. (i ∈ L)
. . . | . . . 9. j : |T|
. . . | . . . 10. bi-graph(T;to;from)
. . . | . . . 11. spanner(f;T;to;from)
. . . | . . . 12. ¬||to(j)|| = 0
. . . | . . . 13. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . | . . . 14. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . | . . . 15. n : ℤ
. . . | . . . 16. 0<n
. . . | . . . 17. u : Edge(T)
. . . | . . . 18. v : Edge(T) List
. . . | . . . 19. lpath([u / v])
. . . | . . . 20. ||v||+1 = n-1
. . . | . . . 21. ¬f(u)
. . . | . . . 22. (∀l∈v.¬f(l))
. . . | . . . 23. l : Edge(T)
. . . | . . . 24. (l ∈ to(source(u)))
. . . | . . . 25. ¬f(l)
. . . | . . . ⊢  (∀l∈[l; u / v].¬f(l))
. . . | . . .   by RWW
. . . | . . .      Thm* ∀P:(T→Prop), x:T, L:T List.
. . . | . . .           (∀y∈[x / L].P(y)) ⟺ P(x) & (∀y∈L.P(y))
. . . | . . .      {l_all_cons}
. . . | . . .      0
. . . |
. . . .\
. . . . 1. T : Id→𝔹
. . . . 2. to : |T|→(IdLnk List)
. . . . 3. from : |T|→(IdLnk List)
. . . . 4. f : Edge(T)→𝔹
. . . . 5. bi-graph(T;to;from)
. . . . 6. ∀i,j:|T|.
. . . .    ∃p:Edge(T) List.
. . . .    lconnects(p;i;j)
. . . .    & (∀q:Edge(T) List. lconnects(q;i;j) ⇒ q = p)
. . . . 7. L : |T| List
. . . . 8. ∀i:|T|. (i ∈ L)
. . . . 9. j : |T|
. . . . 10. bi-graph(T;to;from)
. . . . 11. spanner(f;T;to;from)
. . . . 12. n : ℕ
. . . . 13. ∀p:Edge(T) List. lpath(p) ⇒ ||p||≤n
. . . . 14. ¬||to(j)|| = 0
. . . . 15. ¬(∃i:|T|. ∀l:Edge(T). (l ∈ to(i)) ⇒ f(l))
. . . . 16. ∀i:|T|. (∃l∈to(i).¬f(l))
. . . . 17. ∀n:ℕ. ∃p:Edge(T) List. lpath(p) & ||p|| = n & (∀l∈p.¬f(l))
```

```
. . . . ⊢  False
. . . .   by InstHyp [n+1] -1 THEN ExRepD
. . . .      THEN
. . . .      AllHyps (λh.InstHyp [p] h THEN Complete Auto)
END
```