```
topsortl-properties  THM

⊢  For any graph
|  ∀the_obj:GraphObject(the_graph), L:V List.
|  (∀i:V. (i ∈ L))
|  ⇒
|  (non-trivial-loop-free(the_graph)
|   ⇒
|   topsortedl(the_graph;L;topsortl(the_obj;L)))
|  & (∀i:V. (i ∈ topsortl(the_obj;L)))
|  & no_repeats(V;topsortl(the_obj;L))
| by UnivCD
|    THEN
|    (Inst Thm: dfsl_DASH_properties [the_graph;the_obj;L])
|    THEN
|    ExRepD
|    THEN
|    (Analyze 0)
|
|\
| 1. the_graph : Graph
| 2. the_obj : GraphObject(the_graph)
| 3. L : Vertices(the_graph) List
| 4. ∀i:Vertices(the_graph). (i ∈ L)
| 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| 6. no_repeats(Vertices(the_graph)+
| |  Vertices(the_graph);dfsl(the_obj;L))
| 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| 8. ∀i:Vertices(the_graph).
| |  (i ∈ L)
| |  ⇒
| |  (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| ⊢  non-trivial-loop-free(the_graph)
| |  ⇒
| |  topsortedl(the_graph;L;topsortl(the_obj;L))
| | by (Analyze 0)
| |    THEN
| |    (Assert
| |     (∀L1,L2:Vertices(the_graph) List.
| |      L = (L1 @ L2)
| |      ⇒
| |      (∃s1,s2:Vertices(the_graph) List.
| |       topsortl(the_obj;L) = (s2 @ s1)
| |       & (∀j:Vertices(the_graph).
| |          ((j ∈ s1) ⟺ L1-the_graph->*j)
| |          & ((j ∈ s2) ⟺ L2-the_graph->*j & ¬L1-the_graph->*j)))))
| |
| |\
| | .....assertion .....
| | 1. the_graph : Graph
| | 2. the_obj : GraphObject(the_graph)
| | 3. L : Vertices(the_graph) List
| | 4. ∀i:Vertices(the_graph). (i ∈ L)
| | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| | 6. no_repeats(Vertices(the_graph)+
```

```
| |    Vertices(the_graph);dfsl(the_obj;L))
| | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| | 8. ∀i:Vertices(the_graph).
| |    (i ∈ L)
| |    ⇒
| |    (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| | 9. non-trivial-loop-free(the_graph)
| | ⊢  ∀L1,L2:Vertices(the_graph) List.
| |    L = (L1 @ L2)
| |    ⇒
| |    (∃s1,s2:Vertices(the_graph) List.
| |     topsortl(the_obj;L) = (s2 @ s1)
| |     & (∀j:Vertices(the_graph).
| |        ((j ∈ s1) ⟺ L1-the_graph->*j)
| |        & ((j ∈ s2) ⟺ L2-the_graph->*j & ¬L1-the_graph->*j)))
| |   by let h ≡ 7 in
| |      (((((Unfold ‘topsortl‘ 0) THEN (Unfold ‘dfsl-traversal‘ h)
| |          THEN
| |          (Analyze h)
| |          THEN
| |          (Analyze h+1))
| |         THENA
| |         (Auto
| |          THEN
| |          (AllHyps
| |           (λh.
| |            (Unfold ‘non-trivial-loop-free‘ h)
| |            THEN
| |            (BackThru h)))))
| |        THEN
| |        (RepeatFor 3 (ParallelOp -1))
| |        THEN
| |        ExRepD
| |        THEN
| |        (InstConcl [mapoutl(s1);mapoutl(s2)])
| |        THEN
| |        (RWO "mapoutl_append<" 0))
| |       THEN
| |       (RWO
| |        Thm* ∀s:(A+B) List, x:A. (x ∈ mapoutl(s)) ⟺ (inl(x) ∈ s)
| |        {mapoutl_member}
| |        0))
| |      THEN
| |      (Analyze 0)
| |      THEN
| |      (Try Trivial)
| |      THEN
| |      Analyze
| |
| .\
| . 1. the_graph : Graph
| . 2. the_obj : GraphObject(the_graph)
| . 3. L : Vertices(the_graph) List
| . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
```

```
| . 6. no_repeats(Vertices(the_graph)+
| . |  Vertices(the_graph);dfsl(the_obj;L))
| . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . 8. ∀i:Vertices(the_graph).
| . |  (i ∈ L)
| . |  ⇒
| . |  (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . 9. non-trivial-loop-free(the_graph)
| . 10. ∀L1,L2:Vertices(the_graph) List.
| . |   L = (L1 @ L2)
| . |   ⇒
| . |   (∃s1,s2:Vertices(the_graph) List.
| . |    topsortl(the_obj;L) = (s2 @ s1)
| . |    & (∀j:Vertices(the_graph).
| . |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . |       & ((j ∈ s2) ⟺ L2-the_graph->*j & ¬L1-the_graph->*j)))
| . ⊢  topsortedl(the_graph;L;topsortl(the_obj;L))
| . | by Assert
| . |    (∀i:Vertices(the_graph), s1,s2:Vertices(the_graph) List.
| . |     topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . |     ⇒
| . |     (∀j:Vertices(the_graph).
| . |      ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2)))
| . |
| . |\
| . | ...
| . | |
| . | |\
| . | | .....aux .....
| . | | 1. the_graph : Graph
| . | | 2. the_obj : GraphObject(the_graph)
| . | | 3. L : Vertices(the_graph) List
| . | | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . | | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . | | 6. no_repeats(Vertices(the_graph)+
| . | |    Vertices(the_graph);dfsl(the_obj;L))
| . | | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . | | 8. ∀i:Vertices(the_graph).
| . | |    (i ∈ L)
| . | |    ⇒
| . | |    (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . | | 9. non-trivial-loop-free(the_graph)
| . | | 10. ∀L1,L2:Vertices(the_graph) List.
| . | |     L = (L1 @ L2)
| . | |     ⇒
| . | |     (∃s1,s2:Vertices(the_graph) List.
| . | |      topsortl(the_obj;L) = (s2 @ s1)
| . | |      & (∀j:Vertices(the_graph).
| . | |         ((j ∈ s1) ⟺ L1-the_graph->*j)
| . | |         & ((j ∈ s2) ⟺ L2-the_graph->*j & ¬L1-the_graph->*j)))
| . | | ⊢  depthfirst-traversal(the_graph;dfsl(the_obj;L))
| . | |   by (BackThru
| . | |       Thm* ∀G:Graph.
| . | |             (∀x,y:Vertices(G). Dec(x = y))
| . | |             ⇒
```

```
| . | |            (∀s:traversal(G).
| . | |             paren(Vertices(G);s)
| . | |             ⇒
| . | |             no_repeats(Vertices(G)+Vertices(G);s)
| . | |             ⇒
| . | |             df-traversal(G;s) ⇒ depthfirst-traversal(G;s))
| . | |       {paren_DASH_df_DASH_traversal})
| . | |      THEN
| . | |      (Try
| . | |       (Using [‘the_obj’,the_obj]
| . | |        (BackThru
| . | |         Thm* For any graph
| . | |              ∀the_obj:GraphObject(the_graph), x,y:V.
| . | |              Dec(x = y)
| . | |         {decidable__equal_gr_v})))
| . | |      THEN
| . | |      (AllHyps (λh.(Analyze h) THEN (Complete Auto)))
| . | |      THEN
| . | |      (Unfold ‘depthfirst-traversal‘ -1)
| . | |
| . | .\
| . | . 1. the_graph : Graph
| . | . 2. the_obj : GraphObject(the_graph)
| . | . 3. L : Vertices(the_graph) List
| . | . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . | . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . | . 6. no_repeats(Vertices(the_graph)+
| . | . |  Vertices(the_graph);dfsl(the_obj;L))
| . | . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . | . 8. ∀i:Vertices(the_graph).
| . | . |  (i ∈ L)
| . | . |  ⇒
| . | . |  (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . | . 9. non-trivial-loop-free(the_graph)
| . | . 10. ∀L1,L2:Vertices(the_graph) List.
| . | . |   L = (L1 @ L2)
| . | . |   ⇒
| . | . |   (∃s1,s2:Vertices(the_graph) List.
| . | . |    topsortl(the_obj;L) = (s2 @ s1)
| . | . |    & (∀j:Vertices(the_graph).
| . | . |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . | . |       & ((j ∈ s2) ⟺ L2-the_graph->*j & ¬L1-the_graph->*j)))
| . | . 11. depthfirst-traversal(the_graph;dfsl(the_obj;L))
| . | . ⊢  ∀i:Vertices(the_graph), s1,s2:Vertices(the_graph) List.
| . | . |  topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . | . |  ⇒
| . | . |  (∀j:Vertices(the_graph).
| . | . |   ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . | . | by (Unfold ‘depthfirst-traversal‘ -1)
| . | . |    THEN
| . | . |    (Unfold ‘topsortl‘ 0)
| . | . |    THEN
| . | . |    (RWO
| . | . |     Thm* ∀L:(A+B) List, l1,l2:A List.
| . | . |          mapoutl(L) = (l1 @ l2)
```

```
| . | . |          ⟺
| . | . |          (∃L1,L2:(A+B) List.
| . | . |           L = (L1 @ L2)
| . | . |           & mapoutl(L1) = l1
| . | . |           & mapoutl(L2) = l2)
| . | . |     {mapoutl_is_append}
| . | . |     -4)
| . | . |    THEN
| . | . |    ExRepD
| . | . |    THEN
| . | . |    (RWO
| . | . |     Thm* ∀L:(A+B) List, l1,l2:A List.
| . | . |          mapoutl(L) = (l1 @ l2)
| . | . |          ⟺
| . | . |          (∃L1,L2:(A+B) List.
| . | . |           L = (L1 @ L2)
| . | . |           & mapoutl(L1) = l1
| . | . |           & mapoutl(L2) = l2)
| . | . |     {mapoutl_is_append}
| . | . |     -4)
| . | . |    THEN
| . | . |    ExRepD
| . | . |    THEN
| . | . |    (RWO
| . | . |     Thm* ∀L:(A+B) List, a:A.
| . | . |          mapoutl(L) = [a]
| . | . |          ⟺
| . | . |          (∃L1,L2:(A+B) List.
| . | . |           L = (L1 @ [inl(a)] @ L2)
| . | . |           & mapoutl(L1) = nil
| . | . |           & mapoutl(L2) = nil)
| . | . |     {mapoutl_is_singleton}
| . | . |     -5)
| . | . |    THEN
| . | . |    ExRepD
| . | . |
| . | . 1. the_graph : Graph
| . | . 2. the_obj : GraphObject(the_graph)
| . | . 3. L : Vertices(the_graph) List
| . | . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . | . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . | . 6. no_repeats(Vertices(the_graph)+
| . | . |  Vertices(the_graph);dfsl(the_obj;L))
| . | . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . | . 8. ∀i:Vertices(the_graph).
| . | . |  (i ∈ L)
| . | . |  ⇒
| . | . |  (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . | . 9. non-trivial-loop-free(the_graph)
| . | . 10. ∀L1,L2:Vertices(the_graph) List.
| . | . |   L = (L1 @ L2)
| . | . |   ⇒
| . | . |   (∃s1,s2:Vertices(the_graph) List.
| . | . |    topsortl(the_obj;L) = (s2 @ s1)
| . | . |    & (∀j:Vertices(the_graph).
```

```
| . | . |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . | . |       & ((j ∈ s2) ⟺ L2-the_graph->*j & ¬L1-the_graph->*j)))
| . | . 11. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| . | . |   (∀j:Vertices(the_graph).
| . | . |    i-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j))
| . | . |   ⇒
| . | . |   dfsl(the_obj;L) = (s1 @ [inl(i)] @ s2)
| . | . |   ⇒
| . | . |   (∀j:Vertices(the_graph).
| . | . |    ¬j = i ⇒ i-the_graph->*j ⇒ (inl(j) ∈ s2))
| . | . 12. i : Vertices(the_graph)
| . | . 13. s1 : Vertices(the_graph) List
| . | . 14. s2 : Vertices(the_graph) List
| . | . 15. L1 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . 16. L2 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . 17. dfsl(the_obj;L) = (L1 @ L2)
| . | . 18. mapoutl(L1) = s1
| . | . 19. L3 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . 20. L3@0 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . 21. L2 = (L3 @ L3@0)
| . | . 22. L4 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . 23. L5 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . 24. L3 = (L4 @ [inl(i)] @ L5)
| . | . 25. mapoutl(L4) = nil
| . | . 26. mapoutl(L5) = nil
| . | . 27. mapoutl(L3@0) = s2
| . | . 28. j : Vertices(the_graph)
| . | . 29. ¬j = i
| . | . 30. i-the_graph->*j
| . | . ⊢  (j ∈ s2)  by InstHyp [i;L1 @ L4;L5 @ L3@0;j] 11
| . | . |
| . | . |\
| . | . | 1. the_graph : Graph
| . | . | 2. the_obj : GraphObject(the_graph)
| . | . | 3. L : Vertices(the_graph) List
| . | . | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . | . | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . | . | 6. no_repeats(Vertices(the_graph)+
| . | . |    Vertices(the_graph);dfsl(the_obj;L))
| . | . | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . | . | 8. ∀i:Vertices(the_graph).
| . | . |    (i ∈ L)
| . | . |    ⇒
| . | . |    (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . | . | 9. non-trivial-loop-free(the_graph)
| . | . | 10. ∀L1,L2:Vertices(the_graph) List.
| . | . |     L = (L1 @ L2)
| . | . |     ⇒
| . | . |     (∃s1,s2:Vertices(the_graph) List.
| . | . |      topsortl(the_obj;L) = (s2 @ s1)
| . | . |      & (∀j:Vertices(the_graph).
| . | . |         ((j ∈ s1) ⟺ L1-the_graph->*j)
| . | . |         & ((j ∈ s2)
| . | . |            ⟺
| . | . |            L2-the_graph->*j & ¬L1-the_graph->*j)))
```

```
| . | . | 11. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| . | . |     (∀j:Vertices(the_graph).
| . | . |      i-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j))
| . | . |     ⇒
| . | . |     dfsl(the_obj;L) = (s1 @ [inl(i)] @ s2)
| . | . |     ⇒
| . | . |     (∀j:Vertices(the_graph).
| . | . |      ¬j = i ⇒ i-the_graph->*j ⇒ (inl(j) ∈ s2))
| . | . | 12. i : Vertices(the_graph)
| . | . | 13. s1 : Vertices(the_graph) List
| . | . | 14. s2 : Vertices(the_graph) List
| . | . | 15. L1 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . | 16. L2 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . | 17. dfsl(the_obj;L) = (L1 @ L2)
| . | . | 18. mapoutl(L1) = s1
| . | . | 19. L3 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . | 20. L3@0 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . | 21. L2 = (L3 @ L3@0)
| . | . | 22. L4 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . | 23. L5 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . | 24. L3 = (L4 @ [inl(i)] @ L5)
| . | . | 25. mapoutl(L4) = nil
| . | . | 26. mapoutl(L5) = nil
| . | . | 27. mapoutl(L3@0) = s2
| . | . | 28. j : Vertices(the_graph)
| . | . | 29. ¬j = i
| . | . | 30. i-the_graph->*j
| . | . | 31. j1 : Vertices(the_graph)
| . | . | 32. i-the_graph->*j1
| . | . | ⊢  ¬non-trivial-loop(the_graph;j1)
| . | . |   by AllHyps
| . | . |      (λh.
| . | . |       (Unfold `non-trivial-loop-free` h) THEN (BackThru h))
| . | . |
| . | . |\
| . | . | .....antecedent .....
| . | . | 1. the_graph : Graph
| . | . | 2. the_obj : GraphObject(the_graph)
| . | . | 3. L : Vertices(the_graph) List
| . | . | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . | . | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . | . | 6. no_repeats(Vertices(the_graph)+
| . | . |    Vertices(the_graph);dfsl(the_obj;L))
| . | . | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . | . | 8. ∀i:Vertices(the_graph).
| . | . |    (i ∈ L)
| . | . |    ⇒
| . | . |    (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . | . | 9. non-trivial-loop-free(the_graph)
| . | . | 10. ∀L1,L2:Vertices(the_graph) List.
| . | . |     L = (L1 @ L2)
| . | . |     ⇒
| . | . |     (∃s1,s2:Vertices(the_graph) List.
| . | . |      topsortl(the_obj;L) = (s2 @ s1)
| . | . |      & (∀j:Vertices(the_graph).
```

```
| . | . |         ((j ∈ s1) ⟺ L1-the_graph->*j)
| . | . |         & ((j ∈ s2)
| . | . |            ⟺
| . | . |            L2-the_graph->*j & ¬L1-the_graph->*j)))
| . | . | 11. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| . | . |     (∀j:Vertices(the_graph).
| . | . |      i-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j))
| . | . |     ⇒
| . | . |     dfsl(the_obj;L) = (s1 @ [inl(i)] @ s2)
| . | . |     ⇒
| . | . |     (∀j:Vertices(the_graph).
| . | . |      ¬j = i ⇒ i-the_graph->*j ⇒ (inl(j) ∈ s2))
| . | . | 12. i : Vertices(the_graph)
| . | . | 13. s1 : Vertices(the_graph) List
| . | . | 14. s2 : Vertices(the_graph) List
| . | . | 15. L1 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . | 16. L2 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . | 17. dfsl(the_obj;L) = (L1 @ L2)
| . | . | 18. mapoutl(L1) = s1
| . | . | 19. L3 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . | 20. L3@0 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . | 21. L2 = (L3 @ L3@0)
| . | . | 22. L4 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . | 23. L5 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . | 24. L3 = (L4 @ [inl(i)] @ L5)
| . | . | 25. mapoutl(L4) = nil
| . | . | 26. mapoutl(L5) = nil
| . | . | 27. mapoutl(L3@0) = s2
| . | . | 28. j : Vertices(the_graph)
| . | . | 29. ¬j = i
| . | . | 30. i-the_graph->*j
| . | . | ⊢  dfsl(the_obj;L) = ((L1 @ L4) @ [inl(i)] @ L5 @ L3@0)
| . | . |   by (SubstFor dfsl(the_obj;L) 0) THEN (SubstFor L2 0)
| . | . |      THEN
| . | . |      (SubstFor L3 0)
| . | . |      THEN
| . | . |      (RWW
| . | . |       Thm* ∀a,b,c:Top List. ((a @ b) @ c) ~ (a @ b @ c)
| . | . |       {append_assoc_sq}
| . | . |       0)
| . | . |
| . | . .\
| . | . . 1. the_graph : Graph
| . | . . 2. the_obj : GraphObject(the_graph)
| . | . . 3. L : Vertices(the_graph) List
| . | . . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . | . . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . | . . 6. no_repeats(Vertices(the_graph)+
| . | . . |  Vertices(the_graph);dfsl(the_obj;L))
| . | . . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . | . . 8. ∀i:Vertices(the_graph).
| . | . . |  (i ∈ L)
| . | . . |  ⇒
| . | . . |  (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . | . . 9. non-trivial-loop-free(the_graph)
```

```
| . | . . 10. ∀L1,L2:Vertices(the_graph) List.
| . | . . |   L = (L1 @ L2)
| . | . . |   ⇒
| . | . . |   (∃s1,s2:Vertices(the_graph) List.
| . | . . |    topsortl(the_obj;L) = (s2 @ s1)
| . | . . |    & (∀j:Vertices(the_graph).
| . | . . |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . | . . |       & ((j ∈ s2)
| . | . . |          ⟺
| . | . . |          L2-the_graph->*j & ¬L1-the_graph->*j)))
| . | . . 11. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| . | . . |   (∀j:Vertices(the_graph).
| . | . . |    i-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j))
| . | . . |   ⇒
| . | . . |   dfsl(the_obj;L) = (s1 @ [inl(i)] @ s2)
| . | . . |   ⇒
| . | . . |   (∀j:Vertices(the_graph).
| . | . . |    ¬j = i ⇒ i-the_graph->*j ⇒ (inl(j) ∈ s2))
| . | . . 12. i : Vertices(the_graph)
| . | . . 13. s1 : Vertices(the_graph) List
| . | . . 14. s2 : Vertices(the_graph) List
| . | . . 15. L1 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . 16. L2 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . 17. dfsl(the_obj;L) = (L1 @ L2)
| . | . . 18. mapoutl(L1) = s1
| . | . . 19. L3 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . 20. L3@0 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . 21. L2 = (L3 @ L3@0)
| . | . . 22. L4 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . 23. L5 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . 24. L3 = (L4 @ [inl(i)] @ L5)
| . | . . 25. mapoutl(L4) = nil
| . | . . 26. mapoutl(L5) = nil
| . | . . 27. mapoutl(L3@0) = s2
| . | . . 28. j : Vertices(the_graph)
| . | . . 29. ¬j = i
| . | . . 30. i-the_graph->*j
| . | . . 31. (inl(j) ∈ L5 @ L3@0)
| . | . . ⊢  (j ∈ s2)
| . | . . | by ((RWO "mapoutl_member<" -1)
| . | . . |     THEN
| . | . . |     (RWO
| . | . . |      Thm* ∀L1,L2:(A+B) List.
| . | . . |           mapoutl(L1 @ L2) ~ (mapoutl(L1) @ mapoutl(L2))
| . | . . |      {mapoutl_append}
| . | . . |      -1)
| . | . . |     THEN
| . | . . |     (RWO
| . | . . |      Thm* ∀x:T, l1,l2:T List.
| . | . . |           (x ∈ l1 @ l2) ⟺ (x ∈ l1) ∨ (x ∈ l2)
| . | . . |      {member_append}
| . | . . |      -1))
| . | . . |    THEN
| . | . . |    (Analyze -1)
| . | . . |
```

```
| . | . . |\
| . | . . | 1. the_graph : Graph
| . | . . | 2. the_obj : GraphObject(the_graph)
| . | . . | 3. L : Vertices(the_graph) List
| . | . . | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . | . . | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . | . . | 6. no_repeats(Vertices(the_graph)+
| . | . . |    Vertices(the_graph);dfsl(the_obj;L))
| . | . . | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . | . . | 8. ∀i:Vertices(the_graph).
| . | . . |    (i ∈ L)
| . | . . |    ⇒
| . | . . |    (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . | . . | 9. non-trivial-loop-free(the_graph)
| . | . . | 10. ∀L1,L2:Vertices(the_graph) List.
| . | . . |     L = (L1 @ L2)
| . | . . |     ⇒
| . | . . |     (∃s1,s2:Vertices(the_graph) List.
| . | . . |      topsortl(the_obj;L) = (s2 @ s1)
| . | . . |      & (∀j:Vertices(the_graph).
| . | . . |         ((j ∈ s1) ⟺ L1-the_graph->*j)
| . | . . |         & ((j ∈ s2)
| . | . . |            ⟺
| . | . . |            L2-the_graph->*j & ¬L1-the_graph->*j)))
| . | . . | 11. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| . | . . |     (∀j:Vertices(the_graph).
| . | . . |      i-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j))
| . | . . |     ⇒
| . | . . |     dfsl(the_obj;L) = (s1 @ [inl(i)] @ s2)
| . | . . |     ⇒
| . | . . |     (∀j:Vertices(the_graph).
| . | . . |      ¬j = i ⇒ i-the_graph->*j ⇒ (inl(j) ∈ s2))
| . | . . | 12. i : Vertices(the_graph)
| . | . . | 13. s1 : Vertices(the_graph) List
| . | . . | 14. s2 : Vertices(the_graph) List
| . | . . | 15. L1 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . | 16. L2 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . | 17. dfsl(the_obj;L) = (L1 @ L2)
| . | . . | 18. mapoutl(L1) = s1
| . | . . | 19. L3 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . | 20. L3@0 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . | 21. L2 = (L3 @ L3@0)
| . | . . | 22. L4 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . | 23. L5 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . | 24. L3 = (L4 @ [inl(i)] @ L5)
| . | . . | 25. mapoutl(L4) = nil
| . | . . | 26. mapoutl(L5) = nil
| . | . . | 27. mapoutl(L3@0) = s2
| . | . . | 28. j : Vertices(the_graph)
| . | . . | 29. ¬j = i
| . | . . | 30. i-the_graph->*j
| . | . . | 31. (j ∈ mapoutl(L5))
| . | . . | ⊢  (j ∈ s2)
| . | . . |   by (HypSubst -6 -1) THEN (ObviousContradiction [-1])
| . | . . |
```

```
| . | . . .\
| . | . . . 1. the_graph : Graph
| . | . . . 2. the_obj : GraphObject(the_graph)
| . | . . . 3. L : Vertices(the_graph) List
| . | . . . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . | . . . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . | . . . 6. no_repeats(Vertices(the_graph)+
| . | . . .    Vertices(the_graph);dfsl(the_obj;L))
| . | . . . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . | . . . 8. ∀i:Vertices(the_graph).
| . | . . .    (i ∈ L)
| . | . . .    ⇒
| . | . . .    (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . | . . . 9. non-trivial-loop-free(the_graph)
| . | . . . 10. ∀L1,L2:Vertices(the_graph) List.
| . | . . .     L = (L1 @ L2)
| . | . . .     ⇒
| . | . . .     (∃s1,s2:Vertices(the_graph) List.
| . | . . .      topsortl(the_obj;L) = (s2 @ s1)
| . | . . .      & (∀j:Vertices(the_graph).
| . | . . .         ((j ∈ s1) ⟺ L1-the_graph->*j)
| . | . . .         & ((j ∈ s2)
| . | . . .            ⟺
| . | . . .            L2-the_graph->*j & ¬L1-the_graph->*j)))
| . | . . . 11. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| . | . . .     (∀j:Vertices(the_graph).
| . | . . .      i-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j))
| . | . . .     ⇒
| . | . . .     dfsl(the_obj;L) = (s1 @ [inl(i)] @ s2)
| . | . . .     ⇒
| . | . . .     (∀j:Vertices(the_graph).
| . | . . .      ¬j = i ⇒ i-the_graph->*j ⇒ (inl(j) ∈ s2))
| . | . . . 12. i : Vertices(the_graph)
| . | . . . 13. s1 : Vertices(the_graph) List
| . | . . . 14. s2 : Vertices(the_graph) List
| . | . . . 15. L1 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . . 16. L2 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . . 17. dfsl(the_obj;L) = (L1 @ L2)
| . | . . . 18. mapoutl(L1) = s1
| . | . . . 19. L3 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . . 20. L3@0 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . . 21. L2 = (L3 @ L3@0)
| . | . . . 22. L4 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . . 23. L5 : (Vertices(the_graph)+Vertices(the_graph)) List
| . | . . . 24. L3 = (L4 @ [inl(i)] @ L5)
| . | . . . 25. mapoutl(L4) = nil
| . | . . . 26. mapoutl(L5) = nil
| . | . . . 27. mapoutl(L3@0) = s2
| . | . . . 28. j : Vertices(the_graph)
| . | . . . 29. ¬j = i
| . | . . . 30. i-the_graph->*j
| . | . . . 31. (j ∈ mapoutl(L3@0))

| . | . . . ⊢  (j ∈ s2)  by HypSubst -5 -1
| . |
| . .\
```

```
| . . 1. the_graph : Graph
| . . 2. the_obj : GraphObject(the_graph)
| . . 3. L : Vertices(the_graph) List
| . . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . 6. no_repeats(Vertices(the_graph)+
| . . |  Vertices(the_graph);dfsl(the_obj;L))
| . . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . 8. ∀i:Vertices(the_graph).
| . . |  (i ∈ L)
| . . |  ⇒
| . . |  (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . . 9. non-trivial-loop-free(the_graph)
| . . 10. ∀L1,L2:Vertices(the_graph) List.
| . . |   L = (L1 @ L2)
| . . |   ⇒
| . . |   (∃s1,s2:Vertices(the_graph) List.
| . . |    topsortl(the_obj;L) = (s2 @ s1)
| . . |    & (∀j:Vertices(the_graph).
| . . |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . |       & ((j ∈ s2) ⟺ L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . 11. ∀i:Vertices(the_graph), s1,s2:Vertices(the_graph) List.
| . . |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . |   ⇒
| . . |   (∀j:Vertices(the_graph).
| . . |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . ⊢  topsortedl(the_graph;L;topsortl(the_obj;L))
| . . | by Assert (∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L)))
| . . |
| . . |\
| . . | .....assertion .....
| . . | 1. the_graph : Graph
| . . | 2. the_obj : GraphObject(the_graph)
| . . | 3. L : Vertices(the_graph) List
| . . | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . | 6. no_repeats(Vertices(the_graph)+
| . . |    Vertices(the_graph);dfsl(the_obj;L))
| . . | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . | 8. ∀i:Vertices(the_graph).
| . . |    (i ∈ L)
| . . |    ⇒
| . . |    (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . . | 9. non-trivial-loop-free(the_graph)
| . . | 10. ∀L1,L2:Vertices(the_graph) List.
| . . |     L = (L1 @ L2)
| . . |     ⇒
| . . |     (∃s1,s2:Vertices(the_graph) List.
| . . |      topsortl(the_obj;L) = (s2 @ s1)
| . . |      & (∀j:Vertices(the_graph).
| . . |         ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . |         & ((j ∈ s2) ⟺ L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . | 11. ∀i:Vertices(the_graph), s1,s2:Vertices(the_graph) List.
| . . |     topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . |     ⇒
```

```
| . . |     (∀j:Vertices(the_graph).
| . . |      ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . | ⊢  ∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L))
| . . |   by ((((ParallelOp 8) THEN (Analyze -1)) THENA EasyHyp)
| . . |       THEN
| . . |       (Analyze -1)
| . . |       THEN
| . . |       (RWO "mapoutl_member<" -1))
| . . |      THEN
| . . |      (Fold `topsortl` -1)
| . . |
| . . .\
| . . . 1. the_graph : Graph
| . . . 2. the_obj : GraphObject(the_graph)
| . . . 3. L : Vertices(the_graph) List
| . . . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . 6. no_repeats(Vertices(the_graph)+
| . . . |  Vertices(the_graph);dfsl(the_obj;L))
| . . . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . 8. ∀i:Vertices(the_graph).
| . . . |  (i ∈ L)
| . . . |  ⇒
| . . . |  (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . . . 9. non-trivial-loop-free(the_graph)
| . . . 10. ∀L1,L2:Vertices(the_graph) List.
| . . . |   L = (L1 @ L2)
| . . . |   ⇒
| . . . |   (∃s1,s2:Vertices(the_graph) List.
| . . . |    topsortl(the_obj;L) = (s2 @ s1)
| . . . |    & (∀j:Vertices(the_graph).
| . . . |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . |       & ((j ∈ s2) ⟺ L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . 11. ∀i:Vertices(the_graph), s1,s2:Vertices(the_graph) List.
| . . . |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . |   ⇒
| . . . |   (∀j:Vertices(the_graph).
| . . . |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . 12. ∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L))
| . . . ⊢  topsortedl(the_graph;L;topsortl(the_obj;L))
| . . . | by Assert
| . . . |    (∀i,j:Vertices(the_graph).
| . . . |     ¬j = i
| . . . |     ⇒
| . . . |     i-the_graph->*j ⇒ i before j ∈ topsortl(the_obj;L))
| . . . |
| . . . |\
| . . . | .....assertion .....
| . . . | 1. the_graph : Graph
| . . . | 2. the_obj : GraphObject(the_graph)
| . . . | 3. L : Vertices(the_graph) List
| . . . | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . | 6. no_repeats(Vertices(the_graph)+
| . . . | |  Vertices(the_graph);dfsl(the_obj;L))
```

```
| . . . | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . | 8. ∀i:Vertices(the_graph).
| . . . | |  (i ∈ L)
| . . . | |  ⇒
| . . . | |  (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . . . | 9. non-trivial-loop-free(the_graph)
| . . . | 10. ∀L1,L2:Vertices(the_graph) List.
| . . . | |   L = (L1 @ L2)
| . . . | |   ⇒
| . . . | |   (∃s1,s2:Vertices(the_graph) List.
| . . . | |    topsortl(the_obj;L) = (s2 @ s1)
| . . . | |    & (∀j:Vertices(the_graph).
| . . . | |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . | |       & ((j ∈ s2)
| . . . | |          ⟺
| . . . | |          L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . | 11. ∀i:Vertices(the_graph), s1,s2:Vertices(the_graph) List.
| . . . | |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . | |   ⇒
| . . . | |   (∀j:Vertices(the_graph).
| . . . | |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . | 12. ∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L))
| . . . | ⊢  ∀i,j:Vertices(the_graph).
| . . . | |  ¬j = i
| . . . | |  ⇒
| . . . | |  i-the_graph->*j ⇒ i before j ∈ topsortl(the_obj;L)
| . . . | | by (ParallelOp -1) THEN (ListMemD -4)
| . . . | |    THEN
| . . . | |    (InstHyp [i;s1;s2;j] 11)
| . . . | |    THEN
| . . . | |    (Reduce 0)
| . . . | |
| . . . | 1. the_graph : Graph
| . . . | 2. the_obj : GraphObject(the_graph)
| . . . | 3. L : Vertices(the_graph) List
| . . . | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . | 6. no_repeats(Vertices(the_graph)+
| . . . | |  Vertices(the_graph);dfsl(the_obj;L))
| . . . | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . | 8. ∀i:Vertices(the_graph).
| . . . | |  (i ∈ L)
| . . . | |  ⇒
| . . . | |  (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . . . | 9. non-trivial-loop-free(the_graph)
| . . . | 10. ∀L1,L2:Vertices(the_graph) List.
| . . . | |   L = (L1 @ L2)
| . . . | |   ⇒
| . . . | |   (∃s1,s2:Vertices(the_graph) List.
| . . . | |    topsortl(the_obj;L) = (s2 @ s1)
| . . . | |    & (∀j:Vertices(the_graph).
| . . . | |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . | |       & ((j ∈ s2)
| . . . | |          ⟺
| . . . | |          L2-the_graph->*j & ¬L1-the_graph->*j)))
```

```
| . . . | 11. ∀i:Vertices(the_graph), s1,s2:Vertices(the_graph) List.
| . . . | |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . | |   ⇒
| . . . | |   (∀j:Vertices(the_graph).
| . . . | |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . | 12. ∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L))
| . . . | 13. i : Vertices(the_graph)
| . . . | 14. (i ∈ topsortl(the_obj;L))
| . . . | 15. j : Vertices(the_graph)
| . . . | 16. ¬j = i
| . . . | 17. i-the_graph->*j
| . . . | 18. s1 : Vertices(the_graph) List
| . . . | 19. s2 : Vertices(the_graph) List
| . . . | 20. topsortl(the_obj;L) = (s1 @ [i / s2])
| . . . | 21. (j ∈ s2)
| . . . | ⊢  i before j ∈ topsortl(the_obj;L)
| . . . | | by (RWO
| . . . | |     Thm* ∀L:T List, x,y:T.
| . . . | |          x before y ∈ L
| . . . | |          ⟺
| . . . | |          (∃A,B:T List. L = (A @ B) & (x ∈ A) & (y ∈ B))
| . . . | |     {l_before_decomp}
| . . . | |     0)
| . . . | |    THEN
| . . . | |    (InstConcl [s1 @ [i];s2])
| . . . | |    THEN
| . . . | |    (Try
| . . . | |     ((RWO
| . . . | |       Thm* ∀a,b,c:Top List. ((a @ b) @ c) ∼ (a @ b @ c)
| . . . | |       {append_assoc_sq}
| . . . | |       0)
| . . . | |      THEN
| . . . | |      (Reduce 0)))
| . . . | |
| . . . | 1. the_graph : Graph
| . . . | 2. the_obj : GraphObject(the_graph)
| . . . | 3. L : Vertices(the_graph) List
| . . . | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . | 6. no_repeats(Vertices(the_graph)+
| . . . |    Vertices(the_graph);dfsl(the_obj;L))
| . . . | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . | 8. ∀i:Vertices(the_graph).
| . . . |    (i ∈ L)
| . . . |    ⇒
| . . . |    (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . . . | 9. non-trivial-loop-free(the_graph)
| . . . | 10. ∀L1,L2:Vertices(the_graph) List.
| . . . |     L = (L1 @ L2)
| . . . |     ⇒
| . . . |     (∃s1,s2:Vertices(the_graph) List.
| . . . |      topsortl(the_obj;L) = (s2 @ s1)
| . . . |      & (∀j:Vertices(the_graph).
| . . . |         ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . |         & ((j ∈ s2)
```

```
| . . . |            ⟺
| . . . |            L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . | 11. ∀i:Vertices(the_graph), s1,s2:Vertices(the_graph) List.
| . . . |     topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . |     ⇒
| . . . |     (∀j:Vertices(the_graph).
| . . . |      ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . | 12. ∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L))
| . . . | 13. i : Vertices(the_graph)
| . . . | 14. (i ∈ topsortl(the_obj;L))
| . . . | 15. j : Vertices(the_graph)
| . . . | 16. ¬j = i
| . . . | 17. i-the_graph->*j
| . . . | 18. s1 : Vertices(the_graph) List
| . . . | 19. s2 : Vertices(the_graph) List
| . . . | 20. topsortl(the_obj;L) = (s1 @ [i / s2])
| . . . | 21. (j ∈ s2)
| . . . | ⊢  (i ∈ s1 @ [i])  by ObviousConcl
| . . . |
| . . . .\
| . . . . 1. the_graph : Graph
| . . . . 2. the_obj : GraphObject(the_graph)
| . . . . 3. L : Vertices(the_graph) List
| . . . . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . 6. no_repeats(Vertices(the_graph)+
| . . . . |  Vertices(the_graph);dfsl(the_obj;L))
| . . . . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . 8. ∀i:Vertices(the_graph).
| . . . . |  (i ∈ L)
| . . . . |  ⇒
| . . . . |  (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . . . . 9. non-trivial-loop-free(the_graph)
| . . . . 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . |   L = (L1 @ L2)
| . . . . |   ⇒
| . . . . |   (∃s1,s2:Vertices(the_graph) List.
| . . . . |    topsortl(the_obj;L) = (s2 @ s1)
| . . . . |    & (∀j:Vertices(the_graph).
| . . . . |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . |       & ((j ∈ s2)
| . . . . |          ⟺
| . . . . |          L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . 11. ∀i:Vertices(the_graph), s1,s2:Vertices(the_graph) List.
| . . . . |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . |   ⇒
| . . . . |   (∀j:Vertices(the_graph).
| . . . . |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . 12. ∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L))
| . . . . 13. ∀i,j:Vertices(the_graph).
| . . . . |   ¬j = i
| . . . . |   ⇒
| . . . . |   i-the_graph->*j ⇒ i before j ∈ topsortl(the_obj;L)
| . . . . ⊢  topsortedl(the_graph;L;topsortl(the_obj;L))
| . . . . | by Assert
```

```
| . . . . |    (∀i,j:Vertices(the_graph).
| . . . . |     j before i ∈ L
| . . . . |     ⇒
| . . . . |     ¬j-the_graph->*i
| . . . . |     ⇒
| . . . . |     (∀k:Vertices(the_graph).
| . . . . |      k before j ∈ L ⇒ ¬k-the_graph->*i)
| . . . . |     ⇒
| . . . . |     i before j ∈ topsortl(the_obj;L))
| . . . . |
| . . . . |\
| . . . . | .....assertion .....
| . . . . | 1. the_graph : Graph
| . . . . | 2. the_obj : GraphObject(the_graph)
| . . . . | 3. L : Vertices(the_graph) List
| . . . . | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . | 6. no_repeats(Vertices(the_graph)+
| . . . . | |  Vertices(the_graph);dfsl(the_obj;L))
| . . . . | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . | 8. ∀i:Vertices(the_graph).
| . . . . | |  (i ∈ L)
| . . . . | |  ⇒
| . . . . | |  (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . . . . | 9. non-trivial-loop-free(the_graph)
| . . . . | 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . | |   L = (L1 @ L2)
| . . . . | |   ⇒
| . . . . | |   (∃s1,s2:Vertices(the_graph) List.
| . . . . | |    topsortl(the_obj;L) = (s2 @ s1)
| . . . . | |    & (∀j:Vertices(the_graph).
| . . . . | |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . | |       & ((j ∈ s2)
| . . . . | |          ⟺
| . . . . | |          L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . | 11. ∀i:Vertices(the_graph)
| . . . . | |   , s1,s2:Vertices(the_graph) List.
| . . . . | |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . | |   ⇒
| . . . . | |   (∀j:Vertices(the_graph).
| . . . . | |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . | 12. ∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L))
| . . . . | 13. ∀i,j:Vertices(the_graph).
| . . . . | |   ¬j = i
| . . . . | |   ⇒
| . . . . | |   i-the_graph->*j ⇒ i before j ∈ topsortl(the_obj;L)
| . . . . | ⊢  ∀i,j:Vertices(the_graph).
| . . . . | |  j before i ∈ L
| . . . . | |  ⇒
| . . . . | |  ¬j-the_graph->*i
| . . . . | |  ⇒
| . . . . | |  (∀k:Vertices(the_graph).
| . . . . | |   k before j ∈ L ⇒ ¬k-the_graph->*i)
| . . . . | |  ⇒
| . . . . | |  i before j ∈ topsortl(the_obj;L)
```

```
| . . . . | | by Auto
| . . . . | |    THEN
| . . . . | |    (RWO
| . . . . | |     Thm* ∀L:T List, x,y:T.
| . . . . | |          x before y ∈ L
| . . . . | |          ⟺
| . . . . | |          (∃A,B:T List. L = (A @ B) & (x ∈ A) & (y ∈ B))
| . . . . | |     {l_before_decomp}
| . . . . | |     -3)
| . . . . | |    THEN
| . . . . | |    ExRepD
| . . . . | |    THEN
| . . . . | |    (ListMemD2 -4)
| . . . . | |    THEN
| . . . . | |    (Try
| . . . . | |     (Using [‘the_obj’,the_obj]
| . . . . | |      (BackThru
| . . . . | |       Thm* For any graph
| . . . . | |            ∀the_obj:GraphObject(the_graph), x,y:V.
| . . . . | |            Dec(x = y)
| . . . . | |       {decidable__equal_gr_v})))
| . . . . | |    THEN
| . . . . | |    (InstHyp [s1 @ [j];s2 @ B] 10)
| . . . . | |
| . . . . | |\
| . . . . | | .....antecedent .....
| . . . . | | 1. the_graph : Graph
| . . . . | | 2. the_obj : GraphObject(the_graph)
| . . . . | | 3. L : Vertices(the_graph) List
| . . . . | | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . | | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . | | 6. no_repeats(Vertices(the_graph)+
| . . . . | |    Vertices(the_graph);dfsl(the_obj;L))
| . . . . | | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . | | 8. ∀i:Vertices(the_graph).
| . . . . | |    (i ∈ L)
| . . . . | |    ⇒
| . . . . | |    (inr(i) ∈ dfsl(the_obj;L))
| . . . . | |    & (inl(i) ∈ dfsl(the_obj;L))
| . . . . | | 9. non-trivial-loop-free(the_graph)
| . . . . | | 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . | |     L = (L1 @ L2)
| . . . . | |     ⇒
| . . . . | |     (∃s1,s2:Vertices(the_graph) List.
| . . . . | |      topsortl(the_obj;L) = (s2 @ s1)
| . . . . | |      & (∀j:Vertices(the_graph).
| . . . . | |         ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . | |         & ((j ∈ s2)
| . . . . | |            ⟺
| . . . . | |            L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . | | 11. ∀i:Vertices(the_graph)
| . . . . | |     , s1,s2:Vertices(the_graph) List.
| . . . . | |     topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . | |     ⇒
| . . . . | |     (∀j:Vertices(the_graph).
```

```
| . . . . | |      ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . | | 12. ∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L))
| . . . . | | 13. ∀i,j:Vertices(the_graph).
| . . . . | |     ¬j = i
| . . . . | |     ⇒
| . . . . | |     i-the_graph->*j ⇒ i before j ∈ topsortl(the_obj;L)
| . . . . | | 14. i : Vertices(the_graph)
| . . . . | | 15. j : Vertices(the_graph)
| . . . . | | 16. A : Vertices(the_graph) List
| . . . . | | 17. B : Vertices(the_graph) List
| . . . . | | 18. L = (A @ B)
| . . . . | | 19. (j ∈ A)
| . . . . | | 20. (i ∈ B)
| . . . . | | 21. ¬j-the_graph->*i
| . . . . | | 22. ∀k:Vertices(the_graph).
| . . . . | |     k before j ∈ L ⇒ ¬k-the_graph->*i
| . . . . | | 23. s1 : Vertices(the_graph) List
| . . . . | | 24. s2 : Vertices(the_graph) List
| . . . . | | 25. A = (s1 @ [j / s2])
| . . . . | | 26. ¬(j ∈ s1)
| . . . . | | ⊢  L = ((s1 @ [j]) @ s2 @ B)
| . . . . | |   by ((SubstFor L 0) THEN (SubstFor A 0)
| . . . . | |       THEN
| . . . . | |       (RWW
| . . . . | |        Thm* ∀a,b,c:Top List. ((a @ b) @ c) ~ (a @ b @ c)
| . . . . | |        {append_assoc_sq}
| . . . . | |        0))
| . . . . | |      THEN
| . . . . | |      (Reduce 0)
| . . . . | |
| . . . . | .\
| . . . . | . 1. the_graph : Graph
| . . . . | . 2. the_obj : GraphObject(the_graph)
| . . . . | . 3. L : Vertices(the_graph) List
| . . . . | . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . | . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . 6. no_repeats(Vertices(the_graph)+
| . . . . | . |  Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . | . 8. ∀i:Vertices(the_graph).
| . . . . | . |  (i ∈ L)
| . . . . | . |  ⇒
| . . . . | . |  (inr(i) ∈ dfsl(the_obj;L))
| . . . . | . |  & (inl(i) ∈ dfsl(the_obj;L))
| . . . . | . 9. non-trivial-loop-free(the_graph)
| . . . . | . 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . | . |   L = (L1 @ L2)
| . . . . | . |   ⇒
| . . . . | . |   (∃s1,s2:Vertices(the_graph) List.
| . . . . | . |    topsortl(the_obj;L) = (s2 @ s1)
| . . . . | . |    & (∀j:Vertices(the_graph).
| . . . . | . |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . | . |       & ((j ∈ s2)
| . . . . | . |          ⟺
| . . . . | . |          L2-the_graph->*j & ¬L1-the_graph->*j)))
```

```
| . . . . | . 11. ∀i:Vertices(the_graph)
| . . . . | . |   , s1,s2:Vertices(the_graph) List.
| . . . . | . |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . | . |   ⇒
| . . . . | . |   (∀j:Vertices(the_graph).
| . . . . | . |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . | . 12. ∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L))
| . . . . | . 13. ∀i,j:Vertices(the_graph).
| . . . . | . |   ¬j = i
| . . . . | . |   ⇒
| . . . . | . |   i-the_graph->*j ⇒ i before j ∈ topsortl(the_obj;L)
| . . . . | . 14. i : Vertices(the_graph)
| . . . . | . 15. j : Vertices(the_graph)
| . . . . | . 16. A : Vertices(the_graph) List
| . . . . | . 17. B : Vertices(the_graph) List
| . . . . | . 18. L = (A @ B)
| . . . . | . 19. (j ∈ A)
| . . . . | . 20. (i ∈ B)
| . . . . | . 21. ¬j-the_graph->*i
| . . . . | . 22. ∀k:Vertices(the_graph).
| . . . . | . |   k before j ∈ L ⇒ ¬k-the_graph->*i
| . . . . | . 23. s1 : Vertices(the_graph) List
| . . . . | . 24. s2 : Vertices(the_graph) List
| . . . . | . 25. A = (s1 @ [j / s2])
| . . . . | . 26. ¬(j ∈ s1)
| . . . . | . 27. ∃s1@0,s2@0:Vertices(the_graph) List.
| . . . . | . |   topsortl(the_obj;L) = (s2@0 @ s1@0)
| . . . . | . |   & (∀j@0:Vertices(the_graph).
| . . . . | . |      ((j@0 ∈ s1@0) ⟺ s1 @ [j]-the_graph->*j@0)
| . . . . | . |      & ((j@0 ∈ s2@0)
| . . . . | . |         ⟺
| . . . . | . |         s2 @ B-the_graph->*j@0
| . . . . | . |         & ¬s1 @ [j]-the_graph->*j@0))
| . . . . | . ⊢  i before j ∈ topsortl(the_obj;L)
| . . . . | . | by (RWO
| . . . . | . |     Thm* ∀L:T List, x,y:T.
| . . . . | . |          x before y ∈ L
| . . . . | . |          ⟺
| . . . . | . |          (∃A,B:T List.
| . . . . | . |           L = (A @ B) & (x ∈ A) & (y ∈ B))
| . . . . | . |     {l_before_decomp}
| . . . . | . |     0)
| . . . . | . |    THEN
| . . . . | . |    ExRepD
| . . . . | . |    THEN
| . . . . | . |    (InstConcl [s2@0;s1@0])
| . . . . | . |
| . . . . | . |\
| . . . . | . | 1. the_graph : Graph
| . . . . | . | 2. the_obj : GraphObject(the_graph)
| . . . . | . | 3. L : Vertices(the_graph) List
| . . . . | . | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . | . | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . | 6. no_repeats(Vertices(the_graph)+
| . . . . | . | |  Vertices(the_graph);dfsl(the_obj;L))
```

```
| . . . . . | . . | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . . | . . | 8. ∀i:Vertices(the_graph).
| . . . . . | . . | |  (i ∈ L)
| . . . . . | . . | |  ⇒
| . . . . . | . . | |  (inr(i) ∈ dfsl(the_obj;L))
| . . . . . | . . | |  & (inl(i) ∈ dfsl(the_obj;L))
| . . . . . | . . | 9. non-trivial-loop-free(the_graph)
| . . . . . | . . | 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . . | . . | |   L = (L1 @ L2)
| . . . . . | . . | |   ⇒
| . . . . . | . . | |   (∃s1,s2:Vertices(the_graph) List.
| . . . . . | . . | |    topsortl(the_obj;L) = (s2 @ s1)
| . . . . . | . . | |    & (∀j:Vertices(the_graph).
| . . . . . | . . | |        ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . . | . . | |        & ((j ∈ s2)
| . . . . . | . . | |           ⟺
| . . . . . | . . | |           L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . . | . . | 11. ∀i:Vertices(the_graph)
| . . . . . | . . | |   , s1,s2:Vertices(the_graph) List.
| . . . . . | . . | |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . . | . . | |   ⇒
| . . . . . | . . | |   (∀j:Vertices(the_graph).
| . . . . . | . . | |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . . | . . | 12. ∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L))
| . . . . . | . . | 13. ∀i,j:Vertices(the_graph).
| . . . . . | . . | |   ¬j = i
| . . . . . | . . | |   ⇒
| . . . . . | . . | |   i-the_graph->*j
| . . . . . | . . | |   ⇒
| . . . . . | . . | |   i before j ∈ topsortl(the_obj;L)
| . . . . . | . . | 14. i : Vertices(the_graph)
| . . . . . | . . | 15. j : Vertices(the_graph)
| . . . . . | . . | 16. A : Vertices(the_graph) List
| . . . . . | . . | 17. B : Vertices(the_graph) List
| . . . . . | . . | 18. L = (A @ B)
| . . . . . | . . | 19. (j ∈ A)
| . . . . . | . . | 20. (i ∈ B)
| . . . . . | . . | 21. ¬j-the_graph->*i
| . . . . . | . . | 22. ∀k:Vertices(the_graph).
| . . . . . | . . | |   k before j ∈ L ⇒ ¬k-the_graph->*i
| . . . . . | . . | 23. s1 : Vertices(the_graph) List
| . . . . . | . . | 24. s2 : Vertices(the_graph) List
| . . . . . | . . | 25. A = (s1 @ [j / s2])
| . . . . . | . . | 26. ¬(j ∈ s1)
| . . . . . | . . | 27. s1@0 : Vertices(the_graph) List
| . . . . . | . . | 28. s2@0 : Vertices(the_graph) List
| . . . . . | . . | 29. topsortl(the_obj;L) = (s2@0 @ s1@0)
| . . . . . | . . | 30. ∀j@0:Vertices(the_graph).
| . . . . . | . . | |   ((j@0 ∈ s1@0) ⟺ s1 @ [j]-the_graph->*j@0)
| . . . . . | . . | |   & ((j@0 ∈ s2@0)
| . . . . . | . . | |      ⟺
| . . . . . | . . | |      s2 @ B-the_graph->*j@0
| . . . . . | . . | |      & ¬s1 @ [j]-the_graph->*j@0)
| . . . . . | . . | ⊢  (i ∈ s2@0)
| . . . . . | . . | | by ((InstHyp [i] -1) THEN (Analyze -1)
```

```
| . . . . | . | |      THEN
| . . . . | . | |      (Analyze -1)
| . . . . | . | |      THEN
| . . . . | . | |      (Analyze -1)
| . . . . | . | |      THEN
| . . . . | . | |      (RWO
| . . . . | . | |       Thm* For any graph
| . . . . | . | |            ∀A,B:V List, i:V.
| . . . . | . | |            A @ B-->*i ⟺ A-->*i ∨ B-->*i
| . . . . | . | |       {list_DASH_connect_DASH_append}
| . . . . | . | |       0))
| . . . . | . | |     THEN
| . . . . | . | |     (RWW
| . . . . | . | |      Thm* For any graph
| . . . . | . | |           ∀i,j:V. [i]-->*j ⟺ i-the_graph->*j
| . . . . | . | |      {list_DASH_connect_DASH_singleton}
| . . . . | . | |      0)
| . . . . | . | |
| . . . . | . | |\
| . . . . | . | | 1. the_graph : Graph
| . . . . | . | | 2. the_obj : GraphObject(the_graph)
| . . . . | . | | 3. L : Vertices(the_graph) List
| . . . . | . | | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . | . | | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . | | 6. no_repeats(Vertices(the_graph)+
| . . . . | . | |    Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . | | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . | . | | 8. ∀i:Vertices(the_graph).
| . . . . | . | |    (i ∈ L)
| . . . . | . | |    ⇒
| . . . . | . | |    (inr(i) ∈ dfsl(the_obj;L))
| . . . . | . | |    & (inl(i) ∈ dfsl(the_obj;L))
| . . . . | . | | 9. non-trivial-loop-free(the_graph)
| . . . . | . | | 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . | . | |      L = (L1 @ L2)
| . . . . | . | |      ⇒
| . . . . | . | |      (∃s1,s2:Vertices(the_graph) List.
| . . . . | . | |       topsortl(the_obj;L) = (s2 @ s1)
| . . . . | . | |       & (∀j:Vertices(the_graph).
| . . . . | . | |          ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . | . | |          & ((j ∈ s2)
| . . . . | . | |             ⟺
| . . . . | . | |             L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . | . | | 11. ∀i:Vertices(the_graph)
| . . . . | . | |      , s1,s2:Vertices(the_graph) List.
| . . . . | . | |      topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . | . | |      ⇒
| . . . . | . | |      (∀j:Vertices(the_graph).
| . . . . | . | |       ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . | . | | 12. ∀i:Vertices(the_graph).
| . . . . | . | |      (i ∈ topsortl(the_obj;L))
| . . . . | . | | 13. ∀i,j:Vertices(the_graph).
| . . . . | . | |      ¬j = i
| . . . . | . | |      ⇒
| . . . . | . | |      i-the_graph->*j
```

```
| . . . . | . | |     ⇒
| . . . . | . | |     i before j ∈ topsortl(the_obj;L)
| . . . . | . | | 14. i : Vertices(the_graph)
| . . . . | . | | 15. j : Vertices(the_graph)
| . . . . | . | | 16. A : Vertices(the_graph) List
| . . . . | . | | 17. B : Vertices(the_graph) List
| . . . . | . | | 18. L = (A @ B)
| . . . . | . | | 19. (j ∈ A)
| . . . . | . | | 20. (i ∈ B)
| . . . . | . | | 21. ¬j-the_graph->*i
| . . . . | . | | 22. ∀k:Vertices(the_graph).
| . . . . | . | |     k before j ∈ L ⇒ ¬k-the_graph->*i
| . . . . | . | | 23. s1 : Vertices(the_graph) List
| . . . . | . | | 24. s2 : Vertices(the_graph) List
| . . . . | . | | 25. A = (s1 @ [j / s2])
| . . . . | . | | 26. ¬(j ∈ s1)
| . . . . | . | | 27. s1@0 : Vertices(the_graph) List
| . . . . | . | | 28. s2@0 : Vertices(the_graph) List
| . . . . | . | | 29. topsortl(the_obj;L) = (s2@0 @ s1@0)
| . . . . | . | | 30. ∀j@0:Vertices(the_graph).
| . . . . | . | |     ((j@0 ∈ s1@0) ⟺ s1 @ [j]-the_graph->*j@0)
| . . . . | . | |     & ((j@0 ∈ s2@0)
| . . . . | . | |        ⟺
| . . . . | . | |        s2 @ B-the_graph->*j@0
| . . . . | . | |        & ¬s1 @ [j]-the_graph->*j@0)
| . . . . | . | | 31. (i ∈ s1@0) ⇒ s1 @ [j]-the_graph->*i
| . . . . | . | | 32. (i ∈ s1@0) ⇐ s1 @ [j]-the_graph->*i
| . . . . | . | | 33. (i ∈ s2@0)
| . . . . | . | |     ⇒
| . . . . | . | |     s2 @ B-the_graph->*i & ¬s1 @ [j]-the_graph->*i
| . . . . | . | | ⊢  s2-the_graph->*i ∨ B-the_graph->*i
| . . . . | . | |   by OrRight THEN (Unfold ‘list-connect‘ 0)
| . . . . | . | |      THEN
| . . . . | . | |      (Unfold ‘l_exists‘ 0)
| . . . . | . | |      THEN
| . . . . | . | |      (InstConcl [i])
| . . . . | . | |      THEN
| . . . . | . | |      Connect
| . . . . | . | |
| . . . . | . | .\
| . . . . | . | . 1. the_graph : Graph
| . . . . | . | . 2. the_obj : GraphObject(the_graph)
| . . . . | . | . 3. L : Vertices(the_graph) List
| . . . . | . | . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . | . | . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . | . 6. no_repeats(Vertices(the_graph)+
| . . . . | . | . |  Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . | . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . | . | . 8. ∀i:Vertices(the_graph).
| . . . . | . | . |  (i ∈ L)
| . . . . | . | . |  ⇒
| . . . . | . | . |  (inr(i) ∈ dfsl(the_obj;L))
| . . . . | . | . |  & (inl(i) ∈ dfsl(the_obj;L))
| . . . . | . | . 9. non-trivial-loop-free(the_graph)
| . . . . | . | . 10. ∀L1,L2:Vertices(the_graph) List.
```

```
| . . . . | . | . |   L = (L1 @ L2)
| . . . . | . | . |   ⇒
| . . . . | . | . |   (∃s1,s2:Vertices(the_graph) List.
| . . . . | . | . |    topsortl(the_obj;L) = (s2 @ s1)
| . . . . | . | . |    & (∀j:Vertices(the_graph).
| . . . . | . | . |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . | . | . |       & ((j ∈ s2)
| . . . . | . | . |          ⟺
| . . . . | . | . |          L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . | . | . 11. ∀i:Vertices(the_graph)
| . . . . | . | . |   , s1,s2:Vertices(the_graph) List.
| . . . . | . | . |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . | . | . |   ⇒
| . . . . | . | . |   (∀j:Vertices(the_graph).
| . . . . | . | . |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . | . | . 12. ∀i:Vertices(the_graph).
| . . . . | . | . |   (i ∈ topsortl(the_obj;L))
| . . . . | . | . 13. ∀i,j:Vertices(the_graph).
| . . . . | . | . |   ¬j = i
| . . . . | . | . |   ⇒
| . . . . | . | . |   i-the_graph->*j
| . . . . | . | . |   ⇒
| . . . . | . | . |   i before j ∈ topsortl(the_obj;L)
| . . . . | . | . 14. i : Vertices(the_graph)
| . . . . | . | . 15. j : Vertices(the_graph)
| . . . . | . | . 16. A : Vertices(the_graph) List
| . . . . | . | . 17. B : Vertices(the_graph) List
| . . . . | . | . 18. L = (A @ B)
| . . . . | . | . 19. (j ∈ A)
| . . . . | . | . 20. (i ∈ B)
| . . . . | . | . 21. ¬j-the_graph->*i
| . . . . | . | . 22. ∀k:Vertices(the_graph).
| . . . . | . | . |   k before j ∈ L ⇒ ¬k-the_graph->*i
| . . . . | . | . 23. s1 : Vertices(the_graph) List
| . . . . | . | . 24. s2 : Vertices(the_graph) List
| . . . . | . | . 25. A = (s1 @ [j / s2])
| . . . . | . | . 26. ¬(j ∈ s1)
| . . . . | . | . 27. s1@0 : Vertices(the_graph) List
| . . . . | . | . 28. s2@0 : Vertices(the_graph) List
| . . . . | . | . 29. topsortl(the_obj;L) = (s2@0 @ s1@0)
| . . . . | . | . 30. ∀j@0:Vertices(the_graph).
| . . . . | . | . |   ((j@0 ∈ s1@0) ⟺ s1 @ [j]-the_graph->*j@0)
| . . . . | . | . |   & ((j@0 ∈ s2@0)
| . . . . | . | . |      ⟺
| . . . . | . | . |      s2 @ B-the_graph->*j@0
| . . . . | . | . |      & ¬s1 @ [j]-the_graph->*j@0)
| . . . . | . | . 31. (i ∈ s1@0) ⇒ s1 @ [j]-the_graph->*i
| . . . . | . | . 32. (i ∈ s1@0) ⇐ s1 @ [j]-the_graph->*i
| . . . . | . | . 33. (i ∈ s2@0)
| . . . . | . | . |   ⇒
| . . . . | . | . |   s2 @ B-the_graph->*i & ¬s1 @ [j]-the_graph->*i
| . . . . | . | . ⊢  ¬(s1-the_graph->*i ∨ j-the_graph->*i)
| . . . . | . | . | by (Analyze 0) THEN (Analyze -1)
| . . . . | . | . |
| . . . . | . | . 1. the_graph : Graph
```

```
| . . . . | . | . 2. the_obj : GraphObject(the_graph)
| . . . . | . | . 3. L : Vertices(the_graph) List
| . . . . | . | . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . | . | . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . | . 6. no_repeats(Vertices(the_graph)+
| . . . . | . | . |  Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . | . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . | . | . 8. ∀i:Vertices(the_graph).
| . . . . | . | . |  (i ∈ L)
| . . . . | . | . |  ⇒
| . . . . | . | . |  (inr(i) ∈ dfsl(the_obj;L))
| . . . . | . | . |  & (inl(i) ∈ dfsl(the_obj;L))
| . . . . | . | . 9. non-trivial-loop-free(the_graph)
| . . . . | . | . 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . | . | . |   L = (L1 @ L2)
| . . . . | . | . |   ⇒
| . . . . | . | . |   (∃s1,s2:Vertices(the_graph) List.
| . . . . | . | . |    topsortl(the_obj;L) = (s2 @ s1)
| . . . . | . | . |    & (∀j:Vertices(the_graph).
| . . . . | . | . |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . | . | . |       & ((j ∈ s2)
| . . . . | . | . |          ⟺
| . . . . | . | . |          L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . | . | . 11. ∀i:Vertices(the_graph)
| . . . . | . | . |   , s1,s2:Vertices(the_graph) List.
| . . . . | . | . |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . | . | . |   ⇒
| . . . . | . | . |   (∀j:Vertices(the_graph).
| . . . . | . | . |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . | . | . 12. ∀i:Vertices(the_graph).
| . . . . | . | . |   (i ∈ topsortl(the_obj;L))
| . . . . | . | . 13. ∀i,j:Vertices(the_graph).
| . . . . | . | . |   ¬j = i
| . . . . | . | . |   ⇒
| . . . . | . | . |   i-the_graph->*j
| . . . . | . | . |   ⇒
| . . . . | . | . |   i before j ∈ topsortl(the_obj;L)
| . . . . | . | . 14. i : Vertices(the_graph)
| . . . . | . | . 15. j : Vertices(the_graph)
| . . . . | . | . 16. A : Vertices(the_graph) List
| . . . . | . | . 17. B : Vertices(the_graph) List
| . . . . | . | . 18. L = (A @ B)
| . . . . | . | . 19. (j ∈ A)
| . . . . | . | . 20. (i ∈ B)
| . . . . | . | . 21. ¬j-the_graph->*i
| . . . . | . | . 22. ∀k:Vertices(the_graph).
| . . . . | . | . |   k before j ∈ L ⇒ ¬k-the_graph->*i
| . . . . | . | . 23. s1 : Vertices(the_graph) List
| . . . . | . | . 24. s2 : Vertices(the_graph) List
| . . . . | . | . 25. A = (s1 @ [j / s2])
| . . . . | . | . 26. ¬(j ∈ s1)
| . . . . | . | . 27. s1@0 : Vertices(the_graph) List
| . . . . | . | . 28. s2@0 : Vertices(the_graph) List
| . . . . | . | . 29. topsortl(the_obj;L) = (s2@0 @ s1@0)
| . . . . | . | . 30. ∀j@0:Vertices(the_graph).
```

```
| . . . . | . | . |   ((j@0 ∈ s1@0) ⟺ s1 @ [j]-the_graph->*j@0)
| . . . . | . | . |   & ((j@0 ∈ s2@0)
| . . . . | . | . |      ⟺
| . . . . | . | . |      s2 @ B-the_graph->*j@0
| . . . . | . | . |      & ¬s1 @ [j]-the_graph->*j@0)
| . . . . | . | . 31. (i ∈ s1@0) ⇒ s1 @ [j]-the_graph->*i
| . . . . | . | . 32. (i ∈ s1@0) ⇐ s1 @ [j]-the_graph->*i
| . . . . | . | . 33. (i ∈ s2@0)
| . . . . | . | . |   ⇒
| . . . . | . | . |   s2 @ B-the_graph->*i & ¬s1 @ [j]-the_graph->*i
| . . . . | . | . 34. s1-the_graph->*i
| . . . . | . | . ⊢  False
| . . . . | . | . | by (Analyze -1) THEN (Analyze -1)
| . . . . | . | . |    THEN
| . . . . | . | . |    (Assert y before j ∈ L)
| . . . . | . | . |    THEN
| . . . . | . | . |    (AllHyps
| . . . . | . | . |     (λh.(InstHyp [y] h) THEN (Complete Auto)))
| . . . . | . | . |
| . . . . | . | . 1. the_graph : Graph
| . . . . | . | . 2. the_obj : GraphObject(the_graph)
| . . . . | . | . 3. L : Vertices(the_graph) List
| . . . . | . | . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . | . | . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . | . 6. no_repeats(Vertices(the_graph)+
| . . . . | . | . |  Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . | . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . | . | . 8. ∀i:Vertices(the_graph).
| . . . . | . | . |  (i ∈ L)
| . . . . | . | . |  ⇒
| . . . . | . | . |  (inr(i) ∈ dfsl(the_obj;L))
| . . . . | . | . |  & (inl(i) ∈ dfsl(the_obj;L))
| . . . . | . | . 9. non-trivial-loop-free(the_graph)
| . . . . | . | . 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . | . | . |   L = (L1 @ L2)
| . . . . | . | . |   ⇒
| . . . . | . | . |   (∃s1,s2:Vertices(the_graph) List.
| . . . . | . | . |    topsortl(the_obj;L) = (s2 @ s1)
| . . . . | . | . |    & (∀j:Vertices(the_graph).
| . . . . | . | . |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . | . | . |       & ((j ∈ s2)
| . . . . | . | . |          ⟺
| . . . . | . | . |          L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . | . | . 11. ∀i:Vertices(the_graph)
| . . . . | . | . |   , s1,s2:Vertices(the_graph) List.
| . . . . | . | . |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . | . | . |   ⇒
| . . . . | . | . |   (∀j:Vertices(the_graph).
| . . . . | . | . |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . | . | . 12. ∀i:Vertices(the_graph).
| . . . . | . | . |   (i ∈ topsortl(the_obj;L))
| . . . . | . | . 13. ∀i,j:Vertices(the_graph).
| . . . . | . | . |   ¬j = i
| . . . . | . | . |   ⇒
| . . . . | . | . |   i-the_graph->*j
```

```
| . . . . | . | . |   ⇒
| . . . . | . | . |   i before j ∈ topsortl(the_obj;L)
| . . . . | . | . 14. i : Vertices(the_graph)
| . . . . | . | . 15. j : Vertices(the_graph)
| . . . . | . | . 16. A : Vertices(the_graph) List
| . . . . | . | . 17. B : Vertices(the_graph) List
| . . . . | . | . 18. L = (A @ B)
| . . . . | . | . 19. (j ∈ A)
| . . . . | . | . 20. (i ∈ B)
| . . . . | . | . 21. ¬j-the_graph->*i
| . . . . | . | . 22. ∀k:Vertices(the_graph).
| . . . . | . | . |   k before j ∈ L ⇒ ¬k-the_graph->*i
| . . . . | . | . 23. s1 : Vertices(the_graph) List
| . . . . | . | . 24. s2 : Vertices(the_graph) List
| . . . . | . | . 25. A = (s1 @ [j / s2])
| . . . . | . | . 26. ¬(j ∈ s1)
| . . . . | . | . 27. s1@0 : Vertices(the_graph) List
| . . . . | . | . 28. s2@0 : Vertices(the_graph) List
| . . . . | . | . 29. topsortl(the_obj;L) = (s2@0 @ s1@0)
| . . . . | . | . 30. ∀j@0:Vertices(the_graph).
| . . . . | . | . |   ((j@0 ∈ s1@0) ⟺ s1 @ [j]-the_graph->*j@0)
| . . . . | . | . |   & ((j@0 ∈ s2@0)
| . . . . | . | . |      ⟺
| . . . . | . | . |      s2 @ B-the_graph->*j@0
| . . . . | . | . |      & ¬s1 @ [j]-the_graph->*j@0)
| . . . . | . | . 31. (i ∈ s1@0) ⇒ s1 @ [j]-the_graph->*i
| . . . . | . | . 32. (i ∈ s1@0) ⇐ s1 @ [j]-the_graph->*i
| . . . . | . | . 33. (i ∈ s2@0)
| . . . . | . | . |   ⇒
| . . . . | . | . |   s2 @ B-the_graph->*i & ¬s1 @ [j]-the_graph->*i
| . . . . | . | . 34. y : Vertices(the_graph)
| . . . . | . | . 35. (y ∈ s1)
| . . . . | . | . 36. y-the_graph->*i
| . . . . | . | . ⊢  y before j ∈ L
| . . . . | . | . | by (RWO
| . . . . | . | . |     Thm* ∀L:T List, x,y:T.
| . . . . | . | . |          x before y ∈ L
| . . . . | . | . |          ⟺
| . . . . | . | . |          (∃A,B:T List.
| . . . . | . | . |           L = (A @ B) & (x ∈ A) & (y ∈ B))
| . . . . | . | . |     {l_before_decomp}
| . . . . | . | . |     0)
| . . . . | . | . |    THEN
| . . . . | . | . |    (InstConcl [s1;[j / (s2 @ B)]])
| . . . . | . | . |
| . . . . | . | . |\
| . . . . | . | . | 1. the_graph : Graph
| . . . . | . | . | 2. the_obj : GraphObject(the_graph)
| . . . . | . | . | 3. L : Vertices(the_graph) List
| . . . . | . | . | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . | . | . | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . | . | 6. no_repeats(Vertices(the_graph)+
| . . . . | . | . | |  Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . | . | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . | . | . | 8. ∀i:Vertices(the_graph).
```

```
| . . . . | . | . | |  (i ∈ L)
| . . . . | . | . | |  ⇒
| . . . . | . | . | |  (inr(i) ∈ dfsl(the_obj;L))
| . . . . | . | . | |  & (inl(i) ∈ dfsl(the_obj;L))
| . . . . | . | . | 9. non-trivial-loop-free(the_graph)
| . . . . | . | . | 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . | . | . | |   L = (L1 @ L2)
| . . . . | . | . | |   ⇒
| . . . . | . | . | |   (∃s1,s2:Vertices(the_graph) List.
| . . . . | . | . | |    topsortl(the_obj;L) = (s2 @ s1)
| . . . . | . | . | |    & (∀j:Vertices(the_graph).
| . . . . | . | . | |        ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . | . | . | |        & ((j ∈ s2)
| . . . . | . | . | |            ⟺
| . . . . | . | . | |            L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . | . | . | 11. ∀i:Vertices(the_graph)
| . . . . | . | . | |   , s1,s2:Vertices(the_graph) List.
| . . . . | . | . | |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . | . | . | |   ⇒
| . . . . | . | . | |   (∀j:Vertices(the_graph).
| . . . . | . | . | |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . | . | . | 12. ∀i:Vertices(the_graph).
| . . . . | . | . | |   (i ∈ topsortl(the_obj;L))
| . . . . | . | . | 13. ∀i,j:Vertices(the_graph).
| . . . . | . | . | |   ¬j = i
| . . . . | . | . | |   ⇒
| . . . . | . | . | |   i-the_graph->*j
| . . . . | . | . | |   ⇒
| . . . . | . | . | |   i before j ∈ topsortl(the_obj;L)
| . . . . | . | . | 14. i : Vertices(the_graph)
| . . . . | . | . | 15. j : Vertices(the_graph)
| . . . . | . | . | 16. A : Vertices(the_graph) List
| . . . . | . | . | 17. B : Vertices(the_graph) List
| . . . . | . | . | 18. L = (A @ B)
| . . . . | . | . | 19. (j ∈ A)
| . . . . | . | . | 20. (i ∈ B)
| . . . . | . | . | 21. ¬j-the_graph->*i
| . . . . | . | . | 22. ∀k:Vertices(the_graph).
| . . . . | . | . | |   k before j ∈ L ⇒ ¬k-the_graph->*i
| . . . . | . | . | 23. s1 : Vertices(the_graph) List
| . . . . | . | . | 24. s2 : Vertices(the_graph) List
| . . . . | . | . | 25. A = (s1 @ [j / s2])
| . . . . | . | . | 26. ¬(j ∈ s1)
| . . . . | . | . | 27. s1@0 : Vertices(the_graph) List
| . . . . | . | . | 28. s2@0 : Vertices(the_graph) List
| . . . . | . | . | 29. topsortl(the_obj;L) = (s2@0 @ s1@0)
| . . . . | . | . | 30. ∀j@0:Vertices(the_graph).
| . . . . | . | . | |   ((j@0 ∈ s1@0) ⟺ s1 @ [j]-the_graph->*j@0)
| . . . . | . | . | |   & ((j@0 ∈ s2@0)
| . . . . | . | . | |      ⟺
| . . . . | . | . | |      s2 @ B-the_graph->*j@0
| . . . . | . | . | |      & ¬s1 @ [j]-the_graph->*j@0)
| . . . . | . | . | 31. (i ∈ s1@0) ⇒ s1 @ [j]-the_graph->*i
| . . . . | . | . | 32. (i ∈ s1@0) ⇐ s1 @ [j]-the_graph->*i
| . . . . | . | . | 33. (i ∈ s2@0)
```

```
| . . . . | . | . | |   ⇒
| . . . . | . | . | |   s2 @ B-the_graph->*i & ¬s1 @ [j]-the_graph->*i
| . . . . | . | . | 34. y : Vertices(the_graph)
| . . . . | . | . | 35. (y ∈ s1)
| . . . . | . | . | 36. y-the_graph->*i
| . . . . | . | . | ⊢  L = (s1 @ [j / (s2 @ B)])
| . . . . | . | . | | by ((SubstFor L 0) THEN (SubstFor A1 0)
| . . . . | . | . | |     THEN
| . . . . | . | . | |     (RWW
| . . . . | . | . | |      Thm* ∀a,b,c:Top List.
| . . . . | . | . | |            ((a @ b) @ c) ∼ (a @ b @ c)
| . . . . | . | . | |      {append_assoc_sq}
| . . . . | . | . | |      0))
| . . . . | . | . | |    THEN
| . . . . | . | . | |    (Reduce 0)
| . . . . | . | . | |
| . . . . | . | . | 1. the_graph : Graph
| . . . . | . | . | 2. the_obj : GraphObject(the_graph)
| . . . . | . | . | 3. L : Vertices(the_graph) List
| . . . . | . | . | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . | . | . | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . | . | 6. no_repeats(Vertices(the_graph)+
| . . . . | . | . |    Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . | . | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . | . | . | 8. ∀i:Vertices(the_graph).
| . . . . | . | . |    (i ∈ L)
| . . . . | . | . |    ⇒
| . . . . | . | . |    (inr(i) ∈ dfsl(the_obj;L))
| . . . . | . | . |    & (inl(i) ∈ dfsl(the_obj;L))
| . . . . | . | . | 9. non-trivial-loop-free(the_graph)
| . . . . | . | . | 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . | . | . |     L = (L1 @ L2)
| . . . . | . | . |     ⇒
| . . . . | . | . |     (∃s1,s2:Vertices(the_graph) List.
| . . . . | . | . |      topsortl(the_obj;L) = (s2 @ s1)
| . . . . | . | . |      & (∀j:Vertices(the_graph).
| . . . . | . | . |         ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . | . | . |         & ((j ∈ s2)
| . . . . | . | . |            ⟺
| . . . . | . | . |            L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . | . | . | 11. ∀i:Vertices(the_graph)
| . . . . | . | . |     , s1,s2:Vertices(the_graph) List.
| . . . . | . | . |     topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . | . | . |     ⇒
| . . . . | . | . |     (∀j:Vertices(the_graph).
| . . . . | . | . |      ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . | . | . | 12. ∀i:Vertices(the_graph).
| . . . . | . | . |     (i ∈ topsortl(the_obj;L))
| . . . . | . | . | 13. ∀i,j:Vertices(the_graph).
| . . . . | . | . |     ¬j = i
| . . . . | . | . |     ⇒
| . . . . | . | . |     i-the_graph->*j
| . . . . | . | . |     ⇒
| . . . . | . | . |     i before j ∈ topsortl(the_obj;L)
| . . . . | . | . | 14. i : Vertices(the_graph)
```

```
| . . . . | . | . | 15. j : Vertices(the_graph)
| . . . . | . | . | 16. A : Vertices(the_graph) List
| . . . . | . | . | 17. B : Vertices(the_graph) List
| . . . . | . | . | 18. L = (A @ B)
| . . . . | . | . | 19. (j ∈ A)
| . . . . | . | . | 20. (i ∈ B)
| . . . . | . | . | 21. ¬j-the_graph->*i
| . . . . | . | . | 22. ∀k:Vertices(the_graph).
| . . . . | . | . |     k before j ∈ L ⇒ ¬k-the_graph->*i
| . . . . | . | . | 23. s1 : Vertices(the_graph) List
| . . . . | . | . | 24. s2 : Vertices(the_graph) List
| . . . . | . | . | 25. A = (s1 @ [j / s2])
| . . . . | . | . | 26. ¬(j ∈ s1)
| . . . . | . | . | 27. s1@0 : Vertices(the_graph) List
| . . . . | . | . | 28. s2@0 : Vertices(the_graph) List
| . . . . | . | . | 29. topsortl(the_obj;L) = (s2@0 @ s1@0)
| . . . . | . | . | 30. ∀j@0:Vertices(the_graph).
| . . . . | . | . |     ((j@0 ∈ s1@0) ⟺ s1 @ [j]-the_graph->*j@0)
| . . . . | . | . |     & ((j@0 ∈ s2@0)
| . . . . | . | . |        ⟺
| . . . . | . | . |        s2 @ B-the_graph->*j@0
| . . . . | . | . |        & ¬s1 @ [j]-the_graph->*j@0)
| . . . . | . | . | 31. (i ∈ s1@0) ⇒ s1 @ [j]-the_graph->*i
| . . . . | . | . | 32. (i ∈ s1@0) ⇐ s1 @ [j]-the_graph->*i
| . . . . | . | . | 33. (i ∈ s2@0)
| . . . . | . | . |     ⇒
| . . . . | . | . |     s2 @ B-the_graph->*i & ¬s1 @ [j]-the_graph->*i
| . . . . | . | . | 34. y : Vertices(the_graph)
| . . . . | . | . | 35. (y ∈ s1)
| . . . . | . | . | 36. y-the_graph->*i
| . . . . | . | . | ⊢  (A @ B) = (s1 @ [j / (s2 @ B)])
| . . . . | . | . |   by ((SubstFor L 0) THEN (SubstFor A 0)
| . . . . | . | . |       THEN
| . . . . | . | . |       (RWW
| . . . . | . | . |        Thm* ∀a,b,c:Top List.
| . . . . | . | . |             ((a @ b) @ c) ~ (a @ b @ c)
| . . . . | . | . |       {append_assoc_sq}
| . . . . | . | . |        0))
| . . . . | . | . |      THEN
| . . . . | . | . |      (Reduce 0)
| . . . . | . | . |
| . . . . | . | . .\
| . . . . | . | . . 1. the_graph : Graph
| . . . . | . | . . 2. the_obj : GraphObject(the_graph)
| . . . . | . | . . 3. L : Vertices(the_graph) List
| . . . . | . | . . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . | . | . . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . | . . 6. no_repeats(Vertices(the_graph)+
| . . . . | . | . .    Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . | . . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . | . | . . 8. ∀i:Vertices(the_graph).
| . . . . | . | . .    (i ∈ L)
| . . . . | . | . .    ⇒
| . . . . | . | . .    (inr(i) ∈ dfsl(the_obj;L))
| . . . . | . | . .    & (inl(i) ∈ dfsl(the_obj;L))
```

```
| . . . . | . | . . 9. non-trivial-loop-free(the_graph)
| . . . . | . | . . 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . | . | . .     L = (L1 @ L2)
| . . . . | . | . .     ⇒
| . . . . | . | . .     (∃s1,s2:Vertices(the_graph) List.
| . . . . | . | . .      topsortl(the_obj;L) = (s2 @ s1)
| . . . . | . | . .      & (∀j:Vertices(the_graph).
| . . . . | . | . .         ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . | . | . .         & ((j ∈ s2)
| . . . . | . | . .            ⟺
| . . . . | . | . .            L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . | . | . . 11. ∀i:Vertices(the_graph)
| . . . . | . | . .     , s1,s2:Vertices(the_graph) List.
| . . . . | . | . .     topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . | . | . .     ⇒
| . . . . | . | . .     (∀j:Vertices(the_graph).
| . . . . | . | . .      ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . | . | . . 12. ∀i:Vertices(the_graph).
| . . . . | . | . .     (i ∈ topsortl(the_obj;L))
| . . . . | . | . . 13. ∀i,j:Vertices(the_graph).
| . . . . | . | . .     ¬j = i
| . . . . | . | . .     ⇒
| . . . . | . | . .     i-the_graph->*j
| . . . . | . | . .     ⇒
| . . . . | . | . .     i before j ∈ topsortl(the_obj;L)
| . . . . | . | . . 14. i : Vertices(the_graph)
| . . . . | . | . . 15. j : Vertices(the_graph)
| . . . . | . | . . 16. A : Vertices(the_graph) List
| . . . . | . | . . 17. B : Vertices(the_graph) List
| . . . . | . | . . 18. L = (A @ B)
| . . . . | . | . . 19. (j ∈ A)
| . . . . | . | . . 20. (i ∈ B)
| . . . . | . | . . 21. ¬j-the_graph->*i
| . . . . | . | . . 22. ∀k:Vertices(the_graph).
| . . . . | . | . .     k before j ∈ L ⇒ ¬k-the_graph->*i
| . . . . | . | . . 23. s1 : Vertices(the_graph) List
| . . . . | . | . . 24. s2 : Vertices(the_graph) List
| . . . . | . | . . 25. A = (s1 @ [j / s2])
| . . . . | . | . . 26. ¬(j ∈ s1)
| . . . . | . | . . 27. s1@0 : Vertices(the_graph) List
| . . . . | . | . . 28. s2@0 : Vertices(the_graph) List
| . . . . | . | . . 29. topsortl(the_obj;L) = (s2@0 @ s1@0)
| . . . . | . | . . 30. ∀j@0:Vertices(the_graph).
| . . . . | . | . .     ((j@0 ∈ s1@0) ⟺ s1 @ [j]-the_graph->*j@0)
| . . . . | . | . .     & ((j@0 ∈ s2@0)
| . . . . | . | . .        ⟺
| . . . . | . | . .        s2 @ B-the_graph->*j@0
| . . . . | . | . .        & ¬s1 @ [j]-the_graph->*j@0)
| . . . . | . | . . 31. (i ∈ s1@0) ⇒ s1 @ [j]-the_graph->*i
| . . . . | . | . . 32. (i ∈ s1@0) ⇐ s1 @ [j]-the_graph->*i
| . . . . | . | . . 33. (i ∈ s2@0)
| . . . . | . | . .     ⇒

| . . . . | . | . .     s2 @ B-the_graph->*i & ¬s1 @ [j]-the_graph->*i
| . . . . | . | . . 34. y : Vertices(the_graph)
| . . . . | . | . . 35. (y ∈ s1)
```

```
| . . . . | . | . . 36. y-the_graph->*i
| . . . . | . | . . ⊢  (j ∈ [j / (s2 @ B)])  by ObviousConcl
| . . . . | . |
| . . . . | . .\
| . . . . | . . 1. the_graph : Graph
| . . . . | . . 2. the_obj : GraphObject(the_graph)
| . . . . | . . 3. L : Vertices(the_graph) List
| . . . . | . . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . | . . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . . 6. no_repeats(Vertices(the_graph)+
| . . . . | . .    Vertices(the_graph);dfsl(the_obj;L))
| . . . . | . . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . | . . 8. ∀i:Vertices(the_graph).
| . . . . | . .    (i ∈ L)
| . . . . | . .    ⇒
| . . . . | . .    (inr(i) ∈ dfsl(the_obj;L))
| . . . . | . .    & (inl(i) ∈ dfsl(the_obj;L))
| . . . . | . . 9. non-trivial-loop-free(the_graph)
| . . . . | . . 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . | . .     L = (L1 @ L2)
| . . . . | . .     ⇒
| . . . . | . .     (∃s1,s2:Vertices(the_graph) List.
| . . . . | . .      topsortl(the_obj;L) = (s2 @ s1)
| . . . . | . .      & (∀j:Vertices(the_graph).
| . . . . | . .         ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . | . .         & ((j ∈ s2)
| . . . . | . .            ⟺
| . . . . | . .            L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . | . . 11. ∀i:Vertices(the_graph)
| . . . . | . .     , s1,s2:Vertices(the_graph) List.
| . . . . | . .     topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . | . .     ⇒
| . . . . | . .     (∀j:Vertices(the_graph).
| . . . . | . .      ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . | . . 12. ∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L))
| . . . . | . . 13. ∀i,j:Vertices(the_graph).
| . . . . | . .     ¬j = i
| . . . . | . .     ⇒
| . . . . | . .     i-the_graph->*j
| . . . . | . .     ⇒
| . . . . | . .     i before j ∈ topsortl(the_obj;L)
| . . . . | . . 14. i : Vertices(the_graph)
| . . . . | . . 15. j : Vertices(the_graph)
| . . . . | . . 16. A : Vertices(the_graph) List
| . . . . | . . 17. B : Vertices(the_graph) List
| . . . . | . . 18. L = (A @ B)
| . . . . | . . 19. (j ∈ A)
| . . . . | . . 20. (i ∈ B)
| . . . . | . . 21. ¬j-the_graph->*i
| . . . . | . . 22. ∀k:Vertices(the_graph).
| . . . . | . .     k before j ∈ L ⇒ ¬k-the_graph->*i
| . . . . | . . 23. s1 : Vertices(the_graph) List
| . . . . | . . 24. s2 : Vertices(the_graph) List
| . . . . | . . 25. A = (s1 @ [j / s2])
| . . . . | . . 26. ¬(j ∈ s1)
```

```
| . . . . | . . 27. s1@0 : Vertices(the_graph) List
| . . . . | . . 28. s2@0 : Vertices(the_graph) List
| . . . . | . . 29. topsortl(the_obj;L) = (s2@0 @ s1@0)
| . . . . | . . 30. ∀j@0:Vertices(the_graph).
| . . . . | . .     ((j@0 ∈ s1@0) ⟺ s1 @ [j]-the_graph->*j@0)
| . . . . | . .     & ((j@0 ∈ s2@0)
| . . . . | . .        ⟺
| . . . . | . .        s2 @ B-the_graph->*j@0
| . . . . | . .        & ¬s1 @ [j]-the_graph->*j@0)
| . . . . | . . ⊢  (j ∈ s1@0)
| . . . . | . .   by ((InstHyp [j] -1) THEN (Analyze -1)
| . . . . | . .       THEN
| . . . . | . .       (Analyze -2)
| . . . . | . .       THEN
| . . . . | . .       (Analyze -2)
| . . . . | . .       THEN
| . . . . | . .       (RWO
| . . . . | . .        Thm* For any graph
| . . . . | . .             ∀A,B:V List, i:V.
| . . . . | . .             A @ B-->*i ⟺ A-->*i ∨ B-->*i
| . . . . | . .        {list_DASH_connect_DASH_append}
| . . . . | . .        0))
| . . . . | . .      THEN
| . . . . | . .      (RWO
| . . . . | . .       Thm* For any graph
| . . . . | . .            ∀i,j:V. [i]-->*j ⟺ i-the_graph->*j
| . . . . | . .       {list_DASH_connect_DASH_singleton}
| . . . . | . .       0)
| . . . . | . .      THEN
| . . . . | . .      OrRight
| . . . . | . .      THEN
| . . . . | . .      Connect
| . . . . |
| . . . . .\
| . . . . . 1. the_graph : Graph
| . . . . . 2. the_obj : GraphObject(the_graph)
| . . . . . 3. L : Vertices(the_graph) List
| . . . . . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . . 6. no_repeats(Vertices(the_graph)+
| . . . . . |  Vertices(the_graph);dfsl(the_obj;L))
| . . . . . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . . 8. ∀i:Vertices(the_graph).
| . . . . . |  (i ∈ L)
| . . . . . |  ⇒
| . . . . . |  (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
| . . . . . 9. non-trivial-loop-free(the_graph)
| . . . . . 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . . |   L = (L1 @ L2)
| . . . . . |   ⇒
| . . . . . |   (∃s1,s2:Vertices(the_graph) List.
| . . . . . |    topsortl(the_obj;L) = (s2 @ s1)
| . . . . . |    & (∀j:Vertices(the_graph).
| . . . . . |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . . |       & ((j ∈ s2)
```

```
| . . . . . |          ⟺
| . . . . . |          L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . . 11. ∀i:Vertices(the_graph)
| . . . . . |   , s1,s2:Vertices(the_graph) List.
| . . . . . |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . . |   ⇒
| . . . . . |   (∀j:Vertices(the_graph).
| . . . . . |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . . 12. ∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L))
| . . . . . 13. ∀i,j:Vertices(the_graph).
| . . . . . |   ¬j = i
| . . . . . |   ⇒
| . . . . . |   i-the_graph->*j ⇒ i before j ∈ topsortl(the_obj;L)
| . . . . . 14. ∀i,j:Vertices(the_graph).
| . . . . . |   j before i ∈ L
| . . . . . |   ⇒
| . . . . . |   ¬j-the_graph->*i
| . . . . . |   ⇒
| . . . . . |   (∀k:Vertices(the_graph).
| . . . . . |    k before j ∈ L ⇒ ¬k-the_graph->*i)
| . . . . . |   ⇒
| . . . . . |   i before j ∈ topsortl(the_obj;L)
| . . . . . ⊢  topsortedl(the_graph;L;topsortl(the_obj;L))
| . . . . . | by Assert
| . . . . . |    (∀i,j,k:Vertices(the_graph).
| . . . . . |     k-the_graph->*j
| . . . . . |     ⇒
| . . . . . |     ¬k-the_graph->*i
| . . . . . |     ⇒
| . . . . . |     (∀k':Vertices(the_graph).
| . . . . . |      k' before k ∈ L ⇒ ¬k'-the_graph->*i)
| . . . . . |     ⇒
| . . . . . |     i before j ∈ topsortl(the_obj;L))
| . . . . . |
| . . . . . |\
| . . . . . | .....assertion .....
| . . . . . | 1. the_graph : Graph
| . . . . . | 2. the_obj : GraphObject(the_graph)
| . . . . . | 3. L : Vertices(the_graph) List
| . . . . . | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . . | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . . | 6. no_repeats(Vertices(the_graph)+
| . . . . . | |  Vertices(the_graph);dfsl(the_obj;L))
| . . . . . | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . . | 8. ∀i:Vertices(the_graph).
| . . . . . | |  (i ∈ L)
| . . . . . | |  ⇒
| . . . . . | |  (inr(i) ∈ dfsl(the_obj;L))
| . . . . . | |  & (inl(i) ∈ dfsl(the_obj;L))
| . . . . . | 9. non-trivial-loop-free(the_graph)
| . . . . . | 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . . | |   L = (L1 @ L2)
| . . . . . | |   ⇒
| . . . . . | |   (∃s1,s2:Vertices(the_graph) List.
| . . . . . | |    topsortl(the_obj;L) = (s2 @ s1)
```

```
| . . . . . . | |    & (∀j:Vertices(the_graph).
| . . . . . . | |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . . . | |       & ((j ∈ s2)
| . . . . . . | |          ⟺
| . . . . . . | |          L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . . . | 11. ∀i:Vertices(the_graph)
| . . . . . . | |   , s1,s2:Vertices(the_graph) List.
| . . . . . . | |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . . . | |   ⇒
| . . . . . . | |   (∀j:Vertices(the_graph).
| . . . . . . | |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . . . | 12. ∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L))
| . . . . . . | 13. ∀i,j:Vertices(the_graph).
| . . . . . . | |   ¬j = i
| . . . . . . | |   ⇒
| . . . . . . | |   i-the_graph->*j ⇒ i before j ∈ topsortl(the_obj;L)
| . . . . . . | 14. ∀i,j:Vertices(the_graph).
| . . . . . . | |   j before i ∈ L
| . . . . . . | |   ⇒
| . . . . . . | |   ¬j-the_graph->*i
| . . . . . . | |   ⇒
| . . . . . . | |   (∀k:Vertices(the_graph).
| . . . . . . | |    k before j ∈ L ⇒ ¬k-the_graph->*i)
| . . . . . . | |   ⇒
| . . . . . . | |   i before j ∈ topsortl(the_obj;L)
| . . . . . . | ⊢  ∀i,j,k:Vertices(the_graph).
| . . . . . . | |  k-the_graph->*j
| . . . . . . | |  ⇒
| . . . . . . | |  ¬k-the_graph->*i
| . . . . . . | |  ⇒
| . . . . . . | |  (∀k':Vertices(the_graph).
| . . . . . . | |   k' before k ∈ L ⇒ ¬k'-the_graph->*i)
| . . . . . . | |  ⇒
| . . . . . . | |  i before j ∈ topsortl(the_obj;L)
| . . . . . . | | by Auto
| . . . . . . | |    THEN
| . . . . . . | |    (Assert i before k ∈ topsortl(the_obj;L))
| . . . . . . | |
| . . . . . . | |\
| . . . . . . | | .....assertion .....
| . . . . . . | | 1. the_graph : Graph
| . . . . . . | | 2. the_obj : GraphObject(the_graph)
| . . . . . . | | 3. L : Vertices(the_graph) List
| . . . . . . | | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . . . | | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . . . | | 6. no_repeats(Vertices(the_graph)+
| . . . . . . | | |  Vertices(the_graph);dfsl(the_obj;L))
| . . . . . . | | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . . . | | 8. ∀i:Vertices(the_graph).
| . . . . . . | | |  (i ∈ L)
| . . . . . . | | |  ⇒
| . . . . . . | | |  (inr(i) ∈ dfsl(the_obj;L))
| . . . . . . | | |  & (inl(i) ∈ dfsl(the_obj;L))
| . . . . . . | | 9. non-trivial-loop-free(the_graph)
| . . . . . . | | 10. ∀L1,L2:Vertices(the_graph) List.
```

```
| . . . . . | | |   L = (L1 @ L2)
| . . . . . | | |   ⇒
| . . . . . | | |   (∃s1,s2:Vertices(the_graph) List.
| . . . . . | | |    topsortl(the_obj;L) = (s2 @ s1)
| . . . . . | | |    & (∀j:Vertices(the_graph).
| . . . . . | | |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . . | | |       & ((j ∈ s2)
| . . . . . | | |          ⟺
| . . . . . | | |          L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . . | | 11. ∀i:Vertices(the_graph)
| . . . . . | | |   , s1,s2:Vertices(the_graph) List.
| . . . . . | | |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . . | | |   ⇒
| . . . . . | | |   (∀j:Vertices(the_graph).
| . . . . . | | |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . . | | 12. ∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L))
| . . . . . | | 13. ∀i,j:Vertices(the_graph).
| . . . . . | | |   ¬j = i
| . . . . . | | |   ⇒
| . . . . . | | |   i-the_graph->*j
| . . . . . | | |   ⇒
| . . . . . | | |   i before j ∈ topsortl(the_obj;L)
| . . . . . | | 14. ∀i,j:Vertices(the_graph).
| . . . . . | | |   j before i ∈ L
| . . . . . | | |   ⇒
| . . . . . | | |   ¬j-the_graph->*i
| . . . . . | | |   ⇒
| . . . . . | | |   (∀k:Vertices(the_graph).
| . . . . . | | |    k before j ∈ L ⇒ ¬k-the_graph->*i)
| . . . . . | | |   ⇒
| . . . . . | | |   i before j ∈ topsortl(the_obj;L)
| . . . . . | | 15. i : Vertices(the_graph)
| . . . . . | | 16. j : Vertices(the_graph)
| . . . . . | | 17. k : Vertices(the_graph)
| . . . . . | | 18. k-the_graph->*j
| . . . . . | | 19. ¬k-the_graph->*i
| . . . . . | | 20. ∀k':Vertices(the_graph).
| . . . . . | | |   k' before k ∈ L ⇒ ¬k'-the_graph->*i
| . . . . . | | ⊢  i before k ∈ topsortl(the_obj;L)  by BackThru 14
| . . . . . | | |
| . . . . . | | 1. the_graph : Graph
| . . . . . | | 2. the_obj : GraphObject(the_graph)
| . . . . . | | 3. L : Vertices(the_graph) List
| . . . . . | | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . . | | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . . | | 6. no_repeats(Vertices(the_graph)+
| . . . . . | | |  Vertices(the_graph);dfsl(the_obj;L))
| . . . . . | | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . . | | 8. ∀i:Vertices(the_graph).
| . . . . . | | |  (i ∈ L)
| . . . . . | | |  ⇒
| . . . . . | | |  (inr(i) ∈ dfsl(the_obj;L))
| . . . . . | | |  & (inl(i) ∈ dfsl(the_obj;L))
| . . . . . | | 9. non-trivial-loop-free(the_graph)
| . . . . . | | 10. ∀L1,L2:Vertices(the_graph) List.
```

```
| . . . . . . | | |   L = (L1 @ L2)
| . . . . . . | | |   ⇒
| . . . . . . | | |   (∃s1,s2:Vertices(the_graph) List.
| . . . . . . | | |    topsortl(the_obj;L) = (s2 @ s1)
| . . . . . . | | |    & (∀j:Vertices(the_graph).
| . . . . . . | | |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . . . | | |       & ((j ∈ s2)
| . . . . . . | | |          ⟺
| . . . . . . | | |          L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . . . | | 11. ∀i:Vertices(the_graph)
| . . . . . . | | |   , s1,s2:Vertices(the_graph) List.
| . . . . . . | | |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . . . | | |   ⇒
| . . . . . . | | |   (∀j:Vertices(the_graph).
| . . . . . . | | |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . . . | | 12. ∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L))
| . . . . . . | | 13. ∀i,j:Vertices(the_graph).
| . . . . . . | | |   ¬j = i
| . . . . . . | | |   ⇒
| . . . . . . | | |   i-the_graph->*j
| . . . . . . | | |   ⇒
| . . . . . . | | |   i before j ∈ topsortl(the_obj;L)
| . . . . . . | | 14. ∀i,j:Vertices(the_graph).
| . . . . . . | | |   j before i ∈ L
| . . . . . . | | |   ⇒
| . . . . . . | | |   ¬j-the_graph->*i
| . . . . . . | | |   ⇒
| . . . . . . | | |   (∀k:Vertices(the_graph).
| . . . . . . | | |    k before j ∈ L ⇒ ¬k-the_graph->*i)
| . . . . . . | | |   ⇒
| . . . . . . | | |   i before j ∈ topsortl(the_obj;L)
| . . . . . . | | 15. i : Vertices(the_graph)
| . . . . . . | | 16. j : Vertices(the_graph)
| . . . . . . | | 17. k : Vertices(the_graph)
| . . . . . . | | 18. k-the_graph->*j
| . . . . . . | | 19. ¬k-the_graph->*i
| . . . . . . | | 20. ∀k':Vertices(the_graph).
| . . . . . . | | |   k' before k ∈ L ⇒ ¬k'-the_graph->*i
| . . . . . . | | ⊢  k before i ∈ L
| . . . . . . | | | by (Inst
| . . . . . . | | |     Thm* ∀x,y:T, L:T List.
| . . . . . . | | |          (x ∈ L)
| . . . . . . | | |          ⇒
| . . . . . . | | |          (y ∈ L)
| . . . . . . | | |          ⇒
| . . . . . . | | |          x = y ∨ x before y ∈ L ∨ y before x ∈ L
| . . . . . . | | |     {l_tricotomy}
| . . . . . . | | |     [Vertices(the_graph);k;i;L])
| . . . . . . | | |    THEN
| . . . . . . | | |    EasyHyp
| . . . . . . | | |    THEN
| . . . . . . | | |    (Analyze -1)
| . . . . . . | | |
| . . . . . . | | |\
| . . . . . . | | | 1. the_graph : Graph
```

```
| . . . . . . | | | 2. the_obj : GraphObject(the_graph)
| . . . . . . | | | 3. L : Vertices(the_graph) List
| . . . . . . | | | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . . . | | | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . . . | | | 6. no_repeats(Vertices(the_graph)+
| . . . . . . | | | |  Vertices(the_graph);dfsl(the_obj;L))
| . . . . . . | | | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . . . | | | 8. ∀i:Vertices(the_graph).
| . . . . . . | | | |  (i ∈ L)
| . . . . . . | | | |  ⇒
| . . . . . . | | | |  (inr(i) ∈ dfsl(the_obj;L))
| . . . . . . | | | |  & (inl(i) ∈ dfsl(the_obj;L))
| . . . . . . | | | 9. non-trivial-loop-free(the_graph)
| . . . . . . | | | 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . . . | | | |   L = (L1 @ L2)
| . . . . . . | | | |   ⇒
| . . . . . . | | | |   (∃s1,s2:Vertices(the_graph) List.
| . . . . . . | | | |    topsortl(the_obj;L) = (s2 @ s1)
| . . . . . . | | | |    & (∀j:Vertices(the_graph).
| . . . . . . | | | |        ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . . . | | | |        & ((j ∈ s2)
| . . . . . . | | | |           ⟺
| . . . . . . | | | |           L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . . . | | | 11. ∀i:Vertices(the_graph)
| . . . . . . | | | |   , s1,s2:Vertices(the_graph) List.
| . . . . . . | | | |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . . . | | | |   ⇒
| . . . . . . | | | |   (∀j:Vertices(the_graph).
| . . . . . . | | | |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . . . | | | 12. ∀i:Vertices(the_graph).
| . . . . . . | | | |   (i ∈ topsortl(the_obj;L))
| . . . . . . | | | 13. ∀i,j:Vertices(the_graph).
| . . . . . . | | | |   ¬j = i
| . . . . . . | | | |   ⇒
| . . . . . . | | | |   i-the_graph->*j
| . . . . . . | | | |   ⇒
| . . . . . . | | | |   i before j ∈ topsortl(the_obj;L)
| . . . . . . | | | 14. ∀i,j:Vertices(the_graph).
| . . . . . . | | | |   j before i ∈ L
| . . . . . . | | | |   ⇒
| . . . . . . | | | |   ¬j-the_graph->*i
| . . . . . . | | | |   ⇒
| . . . . . . | | | |   (∀k:Vertices(the_graph).
| . . . . . . | | | |    k before j ∈ L ⇒ ¬k-the_graph->*i)
| . . . . . . | | | |   ⇒
| . . . . . . | | | |   i before j ∈ topsortl(the_obj;L)
| . . . . . . | | | 15. i : Vertices(the_graph)
| . . . . . . | | | 16. j : Vertices(the_graph)
| . . . . . . | | | 17. k : Vertices(the_graph)
| . . . . . . | | | 18. k-the_graph->*j
| . . . . . . | | | 19. ¬k-the_graph->*i
| . . . . . . | | | 20. ∀k':Vertices(the_graph).
| . . . . . . | | | |   k' before k ∈ L ⇒ ¬k'-the_graph->*i
| . . . . . . | | | 21. k = i ∨ k before i ∈ L
| . . . . . . | | | ⊢  k before i ∈ L
```

```
| . . . . . . | | | | by (Analyze -1) THEN (Try Trivial)
| . . . . . . | | | |
| . . . . . . | | | 1. the_graph : Graph
| . . . . . . | | | 2. the_obj : GraphObject(the_graph)
| . . . . . . | | | 3. L : Vertices(the_graph) List
| . . . . . . | | | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . . . | | | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . . . | | | 6. no_repeats(Vertices(the_graph)+
| . . . . . . | | |    Vertices(the_graph);dfsl(the_obj;L))
| . . . . . . | | | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . . . | | | 8. ∀i:Vertices(the_graph).
| . . . . . . | | |    (i ∈ L)
| . . . . . . | | |    ⇒
| . . . . . . | | |    (inr(i) ∈ dfsl(the_obj;L))
| . . . . . . | | |    & (inl(i) ∈ dfsl(the_obj;L))
| . . . . . . | | | 9. non-trivial-loop-free(the_graph)
| . . . . . . | | | 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . . . | | |     L = (L1 @ L2)
| . . . . . . | | |     ⇒
| . . . . . . | | |     (∃s1,s2:Vertices(the_graph) List.
| . . . . . . | | |      topsortl(the_obj;L) = (s2 @ s1)
| . . . . . . | | |      & (∀j:Vertices(the_graph).
| . . . . . . | | |         ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . . . | | |         & ((j ∈ s2)
| . . . . . . | | |            ⟺
| . . . . . . | | |            L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . . . | | | 11. ∀i:Vertices(the_graph)
| . . . . . . | | |     , s1,s2:Vertices(the_graph) List.
| . . . . . . | | |     topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . . . | | |     ⇒
| . . . . . . | | |     (∀j:Vertices(the_graph).
| . . . . . . | | |      ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . . . | | | 12. ∀i:Vertices(the_graph).
| . . . . . . | | |     (i ∈ topsortl(the_obj;L))
| . . . . . . | | | 13. ∀i,j:Vertices(the_graph).
| . . . . . . | | |     ¬j = i
| . . . . . . | | |     ⇒
| . . . . . . | | |     i-the_graph->*j
| . . . . . . | | |     ⇒
| . . . . . . | | |     i before j ∈ topsortl(the_obj;L)
| . . . . . . | | | 14. ∀i,j:Vertices(the_graph).
| . . . . . . | | |     j before i ∈ L
| . . . . . . | | |     ⇒
| . . . . . . | | |     ¬j-the_graph->*i
| . . . . . . | | |     ⇒
| . . . . . . | | |     (∀k:Vertices(the_graph).
| . . . . . . | | |      k before j ∈ L ⇒ ¬k-the_graph->*i)
| . . . . . . | | |     ⇒
| . . . . . . | | |     i before j ∈ topsortl(the_obj;L)
| . . . . . . | | | 15. i : Vertices(the_graph)
| . . . . . . | | | 16. j : Vertices(the_graph)
| . . . . . . | | | 17. k : Vertices(the_graph)
| . . . . . . | | | 18. k-the_graph->*j
| . . . . . . | | | 19. ¬k-the_graph->*i
| . . . . . . | | | 20. ∀k':Vertices(the_graph).
```

```
| . . . . . | | |     k' before k ∈ L ⇒ ¬k'-the_graph->*i
| . . . . . | | | 21. k = i
| . . . . . | | | ⊢  k before i ∈ L
| . . . . . | | |   by ((Analyze -3) THEN (HypSubst -1 0))
| . . . . . | | |      THEN
| . . . . . | | |      Connect
| . . . . . | | |
| . . . . . | | .\
| . . . . . | | . 1. the_graph : Graph
| . . . . . | | . 2. the_obj : GraphObject(the_graph)
| . . . . . | | . 3. L : Vertices(the_graph) List
| . . . . . | | . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . . | | . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . . | | . 6. no_repeats(Vertices(the_graph)+
| . . . . . | | .    Vertices(the_graph);dfsl(the_obj;L))
| . . . . . | | . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . . | | . 8. ∀i:Vertices(the_graph).
| . . . . . | | .    (i ∈ L)
| . . . . . | | .    ⇒
| . . . . . | | .    (inr(i) ∈ dfsl(the_obj;L))
| . . . . . | | .    & (inl(i) ∈ dfsl(the_obj;L))
| . . . . . | | . 9. non-trivial-loop-free(the_graph)
| . . . . . | | . 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . . | | .     L = (L1 @ L2)
| . . . . . | | .     ⇒
| . . . . . | | .     (∃s1,s2:Vertices(the_graph) List.
| . . . . . | | .      topsortl(the_obj;L) = (s2 @ s1)
| . . . . . | | .      & (∀j:Vertices(the_graph).
| . . . . . | | .          ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . . | | .          & ((j ∈ s2)
| . . . . . | | .              ⟺
| . . . . . | | .              L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . . | | . 11. ∀i:Vertices(the_graph)
| . . . . . | | .     , s1,s2:Vertices(the_graph) List.
| . . . . . | | .     topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . . | | .     ⇒
| . . . . . | | .     (∀j:Vertices(the_graph).
| . . . . . | | .      ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . . | | . 12. ∀i:Vertices(the_graph).
| . . . . . | | .     (i ∈ topsortl(the_obj;L))
| . . . . . | | . 13. ∀i,j:Vertices(the_graph).
| . . . . . | | .     ¬j = i
| . . . . . | | .     ⇒
| . . . . . | | .     i-the_graph->*j
| . . . . . | | .     ⇒
| . . . . . | | .     i before j ∈ topsortl(the_obj;L)
| . . . . . | | . 14. ∀i,j:Vertices(the_graph).
| . . . . . | | .     j before i ∈ L
| . . . . . | | .     ⇒
| . . . . . | | .     ¬j-the_graph->*i
| . . . . . | | .     ⇒
| . . . . . | | .     (∀k:Vertices(the_graph).
| . . . . . | | .      k before j ∈ L ⇒ ¬k-the_graph->*i)
| . . . . . | | .     ⇒
| . . . . . | | .     i before j ∈ topsortl(the_obj;L)
```

```
| . . . . . . | | . 15. i : Vertices(the_graph)
| . . . . . . | | . 16. j : Vertices(the_graph)
| . . . . . . | | . 17. k : Vertices(the_graph)
| . . . . . . | | . 18. k-the_graph->*j
| . . . . . . | | . 19. ¬k-the_graph->*i
| . . . . . . | | . 20. ∀k':Vertices(the_graph).
| . . . . . . | | .     k' before k ∈ L ⇒ ¬k'-the_graph->*i
| . . . . . . | | . 21. i before k ∈ L
| . . . . . . | | . ⊢  k before i ∈ L
| . . . . . . | | .   by (InstHyp [i] -2) THEN (Analyze -1)
| . . . . . . | | .      THEN
| . . . . . . | | .      Connect
| . . . . . . | |
| . . . . . . | .\
| . . . . . . | . 1. the_graph : Graph
| . . . . . . | . 2. the_obj : GraphObject(the_graph)
| . . . . . . | . 3. L : Vertices(the_graph) List
| . . . . . . | . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . . . | . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . . . | . 6. no_repeats(Vertices(the_graph)+
| . . . . . . | . |  Vertices(the_graph);dfsl(the_obj;L))
| . . . . . . | . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . . . | . 8. ∀i:Vertices(the_graph).
| . . . . . . | . |  (i ∈ L)
| . . . . . . | . |  ⇒
| . . . . . . | . |  (inr(i) ∈ dfsl(the_obj;L))
| . . . . . . | . |  & (inl(i) ∈ dfsl(the_obj;L))
| . . . . . . | . 9. non-trivial-loop-free(the_graph)
| . . . . . . | . 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . . . | . |   L = (L1 @ L2)
| . . . . . . | . |   ⇒
| . . . . . . | . |   (∃s1,s2:Vertices(the_graph) List.
| . . . . . . | . |    topsortl(the_obj;L) = (s2 @ s1)
| . . . . . . | . |    & (∀j:Vertices(the_graph).
| . . . . . . | . |        ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . . . | . |        & ((j ∈ s2)
| . . . . . . | . |            ⟺
| . . . . . . | . |           L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . . . | . 11. ∀i:Vertices(the_graph)
| . . . . . . | . |   , s1,s2:Vertices(the_graph) List.
| . . . . . . | . |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . . . | . |   ⇒
| . . . . . . | . |   (∀j:Vertices(the_graph).
| . . . . . . | . |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . . . | . 12. ∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L))
| . . . . . . | . 13. ∀i,j:Vertices(the_graph).
| . . . . . . | . |   ¬j = i
| . . . . . . | . |   ⇒
| . . . . . . | . |   i-the_graph->*j
| . . . . . . | . |   ⇒
| . . . . . . | . |   i before j ∈ topsortl(the_obj;L)
| . . . . . . | . 14. ∀i,j:Vertices(the_graph).
| . . . . . . | . |   j before i ∈ L
| . . . . . . | . |   ⇒
| . . . . . . | . |   ¬j-the_graph->*i
```

```
| . . . . . . | . |   ⇒
| . . . . . . | . |   (∀k:Vertices(the_graph).
| . . . . . . | . |    k before j ∈ L ⇒ ¬k-the_graph->*i)
| . . . . . . | . |   ⇒
| . . . . . . | . |   i before j ∈ topsortl(the_obj;L)
| . . . . . . | . 15. i : Vertices(the_graph)
| . . . . . . | . 16. j : Vertices(the_graph)
| . . . . . . | . 17. k : Vertices(the_graph)
| . . . . . . | . 18. k-the_graph->*j
| . . . . . . | . 19. ¬k-the_graph->*i
| . . . . . . | . 20. ∀k':Vertices(the_graph).
| . . . . . . | . |   k' before k ∈ L ⇒ ¬k'-the_graph->*i
| . . . . . . | . 21. i before k ∈ topsortl(the_obj;L)
| . . . . . . | . ⊢  i before j ∈ topsortl(the_obj;L)
| . . . . . . | . | by Assert Dec(k = j)
| . . . . . . | . |
| . . . . . . | . |\
| . . . . . . | . | .....assertion .....
| . . . . . . | . | 1. the_graph : Graph
| . . . . . . | . | 2. the_obj : GraphObject(the_graph)
| . . . . . . | . | 3. L : Vertices(the_graph) List
| . . . . . . | . | 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . . . | . | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . . . | . | 6. no_repeats(Vertices(the_graph)+
| . . . . . . | . |    Vertices(the_graph);dfsl(the_obj;L))
| . . . . . . | . | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . . . | . | 8. ∀i:Vertices(the_graph).
| . . . . . . | . |    (i ∈ L)
| . . . . . . | . |    ⇒
| . . . . . . | . |    (inr(i) ∈ dfsl(the_obj;L))
| . . . . . . | . |    & (inl(i) ∈ dfsl(the_obj;L))
| . . . . . . | . | 9. non-trivial-loop-free(the_graph)
| . . . . . . | . | 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . . . | . |     L = (L1 @ L2)
| . . . . . . | . |     ⇒
| . . . . . . | . |     (∃s1,s2:Vertices(the_graph) List.
| . . . . . . | . |      topsortl(the_obj;L) = (s2 @ s1)
| . . . . . . | . |      & (∀j:Vertices(the_graph).
| . . . . . . | . |          ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . . . | . |          & ((j ∈ s2)
| . . . . . . | . |             ⟺
| . . . . . . | . |             L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . . . | . | 11. ∀i:Vertices(the_graph)
| . . . . . . | . |     , s1,s2:Vertices(the_graph) List.
| . . . . . . | . |     topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . . . | . |     ⇒
| . . . . . . | . |     (∀j:Vertices(the_graph).
| . . . . . . | . |      ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . . . | . | 12. ∀i:Vertices(the_graph).
| . . . . . . | . |     (i ∈ topsortl(the_obj;L))
| . . . . . . | . | 13. ∀i,j:Vertices(the_graph).
| . . . . . . | . |     ¬j = i
| . . . . . . | . |     ⇒
| . . . . . . | . |     i-the_graph->*j
| . . . . . . | . |     ⇒
```

```
| . . . . . . | . |     i before j ∈ topsortl(the_obj;L)
| . . . . . . | . | 14. ∀i,j:Vertices(the_graph).
| . . . . . . | . |     j before i ∈ L
| . . . . . . | . |     ⇒
| . . . . . . | . |     ¬j-the_graph->*i
| . . . . . . | . |     ⇒
| . . . . . . | . |     (∀k:Vertices(the_graph).
| . . . . . . | . |      k before j ∈ L ⇒ ¬k-the_graph->*i)
| . . . . . . | . |     ⇒
| . . . . . . | . |     i before j ∈ topsortl(the_obj;L)
| . . . . . . | . | 15. i : Vertices(the_graph)
| . . . . . . | . | 16. j : Vertices(the_graph)
| . . . . . . | . | 17. k : Vertices(the_graph)
| . . . . . . | . | 18. k-the_graph->*j
| . . . . . . | . | 19. ¬k-the_graph->*i
| . . . . . . | . | 20. ∀k':Vertices(the_graph).
| . . . . . . | . |     k' before k ∈ L ⇒ ¬k'-the_graph->*i
| . . . . . . | . | 21. i before k ∈ topsortl(the_obj;L)
| . . . . . . | . | ⊢  Dec(k = j)
| . . . . . . | . |   by Using [`the_obj',the_obj]
| . . . . . . | . |      (BackThru
| . . . . . . | . |       Thm* For any graph
| . . . . . . | . |             ∀the_obj:GraphObject(the_graph), x,y:V.
| . . . . . . | . |             Dec(x = y)
| . . . . . . | . |       {decidable__equal_gr_v})
| . . . . . . | . |
| . . . . . . | . .\
| . . . . . . | . . 1. the_graph : Graph
| . . . . . . | . . 2. the_obj : GraphObject(the_graph)
| . . . . . . | . . 3. L : Vertices(the_graph) List
| . . . . . . | . . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . . . | . . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . . . | . . 6. no_repeats(Vertices(the_graph)+
| . . . . . . | . . |  Vertices(the_graph);dfsl(the_obj;L))
| . . . . . . | . . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . . . | . . 8. ∀i:Vertices(the_graph).
| . . . . . . | . . |  (i ∈ L)
| . . . . . . | . . |  ⇒
| . . . . . . | . . |  (inr(i) ∈ dfsl(the_obj;L))
| . . . . . . | . . |  & (inl(i) ∈ dfsl(the_obj;L))
| . . . . . . | . . 9. non-trivial-loop-free(the_graph)
| . . . . . . | . . 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . . . | . . |   L = (L1 @ L2)
| . . . . . . | . . |   ⇒
| . . . . . . | . . |   (∃s1,s2:Vertices(the_graph) List.
| . . . . . . | . . |    topsortl(the_obj;L) = (s2 @ s1)
| . . . . . . | . . |    & (∀j:Vertices(the_graph).
| . . . . . . | . . |       ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . . . | . . |       & ((j ∈ s2)
| . . . . . . | . . |          ⟺
| . . . . . . | . . |          L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . . . | . . 11. ∀i:Vertices(the_graph)
| . . . . . . | . . |   , s1,s2:Vertices(the_graph) List.
| . . . . . . | . . |   topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . . . | . . |   ⇒
```

```
| . . . . . . | . . |   (∀j:Vertices(the_graph).
| . . . . . . | . . |    ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . . . | . . 12. ∀i:Vertices(the_graph).
| . . . . . . | . . |   (i ∈ topsortl(the_obj;L))
| . . . . . . | . . 13. ∀i,j:Vertices(the_graph).
| . . . . . . | . . |   ¬j = i
| . . . . . . | . . |   ⇒
| . . . . . . | . . |   i-the_graph->*j
| . . . . . . | . . |   ⇒
| . . . . . . | . . |   i before j ∈ topsortl(the_obj;L)
| . . . . . . | . . 14. ∀i,j:Vertices(the_graph).
| . . . . . . | . . |   j before i ∈ L
| . . . . . . | . . |   ⇒
| . . . . . . | . . |   ¬j-the_graph->*i
| . . . . . . | . . |   ⇒
| . . . . . . | . . |   (∀k:Vertices(the_graph).
| . . . . . . | . . |    k before j ∈ L ⇒ ¬k-the_graph->*i)
| . . . . . . | . . |   ⇒
| . . . . . . | . . |   i before j ∈ topsortl(the_obj;L)
| . . . . . . | . . 15. i : Vertices(the_graph)
| . . . . . . | . . 16. j : Vertices(the_graph)
| . . . . . . | . . 17. k : Vertices(the_graph)
| . . . . . . | . . 18. k-the_graph->*j
| . . . . . . | . . 19. ¬k-the_graph->*i
| . . . . . . | . . 20. ∀k':Vertices(the_graph).
| . . . . . . | . . |   k' before k ∈ L ⇒ ¬k'-the_graph->*i
| . . . . . . | . . 21. i before k ∈ topsortl(the_obj;L)
| . . . . . . | . . 22. Dec(k = j)
| . . . . . . | . . ⊢  i before j ∈ topsortl(the_obj;L)
| . . . . . . | . . | by (Analyze -1) THEN (Try (RevHypSubst -1 0))
| . . . . . . | . . |    THEN
| . . . . . . | . . |    (Using [`y',k]
| . . . . . . | . . |     (BackThru
| . . . . . . | . . |      Thm* ∀l:T List, x,y,z:T.
| . . . . . . | . . |           no_repeats(T;l)
| . . . . . . | . . |           ⇒
| . . . . . . | . . |           x before y ∈ l

| . . . . . . | . . |           ⇒
| . . . . . . | . . |           y before z ∈ l ⇒ x before z ∈ l
| . . . . . . | . . |      {l_before_transitivity}))
| . . . . . . | . . |    THEN
| . . . . . . | . . |    (Try ((BackThru 13) THEN (ParallelOp -1)))
| . . . . . . | . . |
| . . . . . . | . . 1. the_graph : Graph
| . . . . . . | . . 2. the_obj : GraphObject(the_graph)
| . . . . . . | . . 3. L : Vertices(the_graph) List
| . . . . . . | . . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . . . | . . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . . . | . . 6. no_repeats(Vertices(the_graph)+
| . . . . . . | . .    Vertices(the_graph);dfsl(the_obj;L))
| . . . . . . | . . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . . . | . . 8. ∀i:Vertices(the_graph).
| . . . . . . | . .    (i ∈ L)
| . . . . . . | . .    ⇒
| . . . . . . | . .    (inr(i) ∈ dfsl(the_obj;L))
```

```
| . . . . . . | . .    & (inl(i) ∈ dfsl(the_obj;L))
| . . . . . . | . . 9. non-trivial-loop-free(the_graph)
| . . . . . . | . . 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . . . | . .     L = (L1 @ L2)
| . . . . . . | . .     ⇒
| . . . . . . | . .     (∃s1,s2:Vertices(the_graph) List.
| . . . . . . | . .      topsortl(the_obj;L) = (s2 @ s1)
| . . . . . . | . .      & (∀j:Vertices(the_graph).
| . . . . . . | . .         ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . . . | . .         & ((j ∈ s2)
| . . . . . . | . .            ⟺
| . . . . . . | . .            L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . . . | . . 11. ∀i:Vertices(the_graph)
| . . . . . . | . .     , s1,s2:Vertices(the_graph) List.
| . . . . . . | . .     topsortl(the_obj;L) = (s1 @ [i] @ s2)
| . . . . . . | . .     ⇒
| . . . . . . | . .     (∀j:Vertices(the_graph).
| . . . . . . | . .      ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . . . | . . 12. ∀i:Vertices(the_graph).
| . . . . . . | . .     (i ∈ topsortl(the_obj;L))
| . . . . . . | . . 13. ∀i,j:Vertices(the_graph).
| . . . . . . | . .     ¬j = i
| . . . . . . | . .     ⇒
| . . . . . . | . .     i-the_graph->*j
| . . . . . . | . .     ⇒
| . . . . . . | . .     i before j ∈ topsortl(the_obj;L)
| . . . . . . | . . 14. ∀i,j:Vertices(the_graph).
| . . . . . . | . .     j before i ∈ L
| . . . . . . | . .     ⇒
| . . . . . . | . .     ¬j-the_graph->*i
| . . . . . . | . .     ⇒
| . . . . . . | . .     (∀k:Vertices(the_graph).
| . . . . . . | . .      k before j ∈ L ⇒ ¬k-the_graph->*i)
| . . . . . . | . .     ⇒
| . . . . . . | . .     i before j ∈ topsortl(the_obj;L)
| . . . . . . | . . 15. i : Vertices(the_graph)
| . . . . . . | . . 16. j : Vertices(the_graph)
| . . . . . . | . . 17. k : Vertices(the_graph)
| . . . . . . | . . 18. k-the_graph->*j
| . . . . . . | . . 19. ¬k-the_graph->*i
| . . . . . . | . . 20. ∀k':Vertices(the_graph).
| . . . . . . | . .     k' before k ∈ L ⇒ ¬k'-the_graph->*i
| . . . . . . | . . 21. i before k ∈ topsortl(the_obj;L)
| . . . . . . | . . 22. ¬k = j
| . . . . . . | . . ⊢  no_repeats(Vertices(the_graph);topsortl(...;L))
| . . . . . . | . .   by (Unfold `topsortl` 0)
| . . . . . . | . .      THEN
| . . . . . . | . .      (BackThru
| . . . . . . | . .       Thm* ∀s:(A+B) List.
| . . . . . . | . .            no_repeats(A+B;s)
| . . . . . . | . .            ⇒
| . . . . . . | . .            no_repeats(A;mapoutl(s))
| . . . . . . | . .       {no_repeats_mapoutl})
| . . . . . . |
| . . . . . . .\
```

```
| . . . . . . . 1. the_graph : Graph
| . . . . . . . 2. the_obj : GraphObject(the_graph)
| . . . . . . . 3. L : Vertices(the_graph) List
| . . . . . . . 4. ∀i:Vertices(the_graph). (i ∈ L)
| . . . . . . . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
| . . . . . . . 6. no_repeats(Vertices(the_graph)+
| . . . . . . .    Vertices(the_graph);dfsl(the_obj;L))
| . . . . . . . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
| . . . . . . . 8. ∀i:Vertices(the_graph).
| . . . . . . .    (i ∈ L)
| . . . . . . .    ⇒
| . . . . . . .    (inr(i) ∈ dfsl(the_obj;L))
| . . . . . . .    & (inl(i) ∈ dfsl(the_obj;L))
| . . . . . . . 9. non-trivial-loop-free(the_graph)
| . . . . . . . 10. ∀L1,L2:Vertices(the_graph) List.
| . . . . . . .     L = (L1 @ L2)
| . . . . . . .     ⇒
| . . . . . . .     (∃s1,s2:Vertices(the_graph) List.
| . . . . . . .      topsortl(the_obj;L) = (s2 @ s1)
| . . . . . . .      & (∀j:Vertices(the_graph).
| . . . . . . .         ((j ∈ s1) ⟺ L1-the_graph->*j)
| . . . . . . .         & ((j ∈ s2)
| . . . . . . .            ⟺
| . . . . . . .            L2-the_graph->*j & ¬L1-the_graph->*j)))
| . . . . . . . 11. ∀i:Vertices(the_graph)
| . . . . . . .     , s1,s2:Vertices(the_graph) List.
| . . . . . . .     topsortl(the_obj;L) = (s1 @ [i] @ s2)

| . . . . . . .     ⇒
| . . . . . . .     (∀j:Vertices(the_graph).
| . . . . . . .      ¬j = i ⇒ i-the_graph->*j ⇒ (j ∈ s2))
| . . . . . . . 12. ∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L))
| . . . . . . . 13. ∀i,j:Vertices(the_graph).
| . . . . . . .     ¬j = i
| . . . . . . .     ⇒
| . . . . . . .     i-the_graph->*j ⇒ i before j ∈ topsortl(the_obj;L)
| . . . . . . . 14. ∀i,j:Vertices(the_graph).
| . . . . . . .     j before i ∈ L
| . . . . . . .     ⇒
| . . . . . . .     ¬j-the_graph->*i
| . . . . . . .     ⇒
| . . . . . . .     (∀k:Vertices(the_graph).
| . . . . . . .      k before j ∈ L ⇒ ¬k-the_graph->*i)
| . . . . . . .     ⇒
| . . . . . . .     i before j ∈ topsortl(the_obj;L)
| . . . . . . . 15. ∀i,j,k:Vertices(the_graph).
| . . . . . . .     k-the_graph->*j
| . . . . . . .     ⇒
| . . . . . . .     ¬k-the_graph->*i
| . . . . . . .     ⇒
| . . . . . . .     (∀k':Vertices(the_graph).
| . . . . . . .      k' before k ∈ L ⇒ ¬k'-the_graph->*i)
| . . . . . . .     ⇒
| . . . . . . .     i before j ∈ topsortl(the_obj;L)
| . . . . . . . ⊢  topsortedl(the_graph;L;topsortl(the_obj;L))
| . . . . . . .   by (Analyze 0) THEN (Try Trivial)
```

```
|
.\
. 1. the_graph : Graph
. 2. the_obj : GraphObject(the_graph)
. 3. L : Vertices(the_graph) List
. 4. ∀i:Vertices(the_graph). (i ∈ L)
. 5. paren(Vertices(the_graph);dfsl(the_obj;L))
. 6. no_repeats(Vertices(the_graph)+
. |  Vertices(the_graph);dfsl(the_obj;L))
. 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
. 8. ∀i:Vertices(the_graph).
. |  (i ∈ L)
. |  ⇒
. |  (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
. ⊢  (∀i:Vertices(the_graph). (i ∈ topsortl(the_obj;L)))
. |  & no_repeats(Vertices(the_graph);topsortl(the_obj;L))  by Auto
. |
. |\
. | 1. the_graph : Graph
. | 2. the_obj : GraphObject(the_graph)
. | 3. L : Vertices(the_graph) List
. | 4. ∀i:Vertices(the_graph). (i ∈ L)
. | 5. paren(Vertices(the_graph);dfsl(the_obj;L))
. | 6. no_repeats(Vertices(the_graph)+
. |    Vertices(the_graph);dfsl(the_obj;L))
. | 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
. | 8. ∀i:Vertices(the_graph).
. |    (i ∈ L)
. |    ⇒
. |    (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
. | 9. i : Vertices(the_graph)
. | ⊢  (i ∈ topsortl(the_obj;L))
. |   by (Unfold `topsortl` 0) THEN (InstHyp [i] -2) THEN EasyHyp
. |      THEN
. |      (RWO
. |       Thm* ∀s:(A+B) List, x:A.
. |            (x ∈ mapoutl(s))
. |            ⟺
. |            (∃y:A+B. (y ∈ s) & isl(y) & x = outl(y))
. |       {member_mapoutl}
. |       0)
. |      THEN
. |      (InstConcl [inl(i)])
. |      THEN
. |      (Reduce 0)
. |
. .\
. . 1. the_graph : Graph
. . 2. the_obj : GraphObject(the_graph)
. . 3. L : Vertices(the_graph) List
. . 4. ∀i:Vertices(the_graph). (i ∈ L)
. . 5. paren(Vertices(the_graph);dfsl(the_obj;L))
. . 6. no_repeats(Vertices(the_graph)+
. .    Vertices(the_graph);dfsl(the_obj;L))
. . 7. dfsl-traversal(the_graph;L;dfsl(the_obj;L))
```

```
. . 8. ∀i:Vertices(the_graph).
. .     (i ∈ L)
. .     ⇒
. .     (inr(i) ∈ dfsl(the_obj;L)) & (inl(i) ∈ dfsl(the_obj;L))
. . ⊢  no_repeats(Vertices(the_graph);topsortl(the_obj;L))
. .   by (Unfold ‘topsortl‘ 0)
. .      THEN
. .      (BackThru
. .       Thm* ∀s:(A+B) List.
. .            no_repeats(A+B;s) ⇒ no_repeats(A;mapoutl(s))
. .       {no_repeats_mapoutl})
END
```