```
dfs_induction  THM

⊢  For any graph
|  ∀the_obj:GraphObject(the_graph), P:(V→Traversal→Traversal→Prop)
|  , s:Traversal, i:V.
|  (∀s1,s2:Traversal, i:V.
|   P(i,s1,s2) ⇒ l_disjoint(V+V;s2;s1) & no_repeats(V+V;s2))
|  ⇒
|  (∀s:Traversal, i:V.
|   member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil))
|  ⇒
|  (∀s1,s2,s3:Traversal, i,j:V.
|   i-the_graph->j
|   ⇒
|   P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2))
|  ⇒
|  (∀s1,s2:Traversal, i:V.
|   ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
|   ⇒
|   P(i,[inr(i) / s1],s2) ⇒ P(i,s1,[inl(i) / (s2 @ [inr(i)])]))
|  ⇒
|  (∃s':Traversal. P(i,s,s') & dfs(the_obj;s;i) = (s' @ s))
| by Auto THEN (Inst Thm: dfs_DASH_measure [the_graph;the_obj])
|    THEN
|    ExRepDHyps
|    THEN
|    (Assert
|     (∀d:ℕ, s:traversal(the_graph), i:Vertices(the_graph).
|      M(s)≤d
|      ⇒
|      (∃s':traversal(the_graph).
|       M(s' @ s)≤M(s) & P(i,s,s') & dfs(the_obj;s;i) = (s' @ s))))
|
|\
| .....assertion .....
| 1. the_graph : Graph
| 2. the_obj : GraphObject(the_graph)
| 3. P :
| |  Vertices(the_graph)→traversal(the_graph)→traversal(the_graph)→
| |  Prop
| 4. traversal(the_graph)
| 5. Vertices(the_graph)
| 6. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| |  P(i,s1,s2)
| |  ⇒
| |  l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| 7. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| |  P(i,s1,s2)
| |  ⇒
| |  no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| 8. ∀s:traversal(the_graph), i:Vertices(the_graph).
| |  member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| 9. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| |  i-the_graph->j
| |  ⇒
```

```
| |  P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| 10. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| |   ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| |   ⇒
| |   P(i,[inr(i) / s1],s2) ⇒ P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| 11. M : traversal(the_graph)→ℕ
| 12. ∀i:Vertices(the_graph), s:traversal(the_graph).
| |   M([inl(i) / s])≤M(s)
| 13. ∀i:Vertices(the_graph), s:traversal(the_graph).
| |   ¬member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ M([inr(i) / s])<M(s)
| ⊢  ∀d:ℕ, s:traversal(the_graph), i:Vertices(the_graph).
| |  M(s)≤d
| |  ⇒
| |  (∃s':traversal(the_graph).
| |   M(s' @ s)≤M(s) & P(i,s,s') & dfs(the_obj;s;i) = (s' @ s))
| | by (Inst Thm: graphobj_DASH_properties [the_graph;the_obj])
| |    THEN
| |    (Analyze -1)
| |    THEN
| |    (Analyze -1)
| |    THEN
| |    (Thin 5)
| |    THEN
| |    (Thin 4)
| |    THEN
| |    CompleteInductionOnNat
| |    THEN
| |    (RecUnfold `dfs` 0)
| |    THEN
| |    SplitOnConclITE
| |    THEN
| |    (Try
| |     ((InstConcl [nil]) THEN (Reduce 0) THEN (Complete EasyHyp)))
| |
| .....falsecase .....
| 1. the_graph : Graph
| 2. the_obj : GraphObject(the_graph)
| 3. P :
| |  Vertices(the_graph)→traversal(the_graph)→traversal(the_graph)→
| |  Prop
| 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| |  P(i,s1,s2)
| |  ⇒
| |  l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| |  P(i,s1,s2)
| |  ⇒

| |  no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| |  member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| |  i-the_graph->j
| |  ⇒
| |  P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
```

```
| |  ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| |  ⇒
| |  P(i,[inr(i) / s1],s2) ⇒ P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| 9. M : traversal(the_graph)→ℕ
| 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| |   M([inl(i) / s])≤M(s)
| 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| |   ¬member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ M([inr(i) / s])<M(s)
| 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| 13. ∀T:Type, s:T, x:Vertices(the_graph)
| |   , f:(T→Vertices(the_graph)→T).
| |   ∃L:Vertices(the_graph) List.
| |   (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| |   & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| |   ∃L:Vertices(the_graph) List.
| |   no_repeats(Vertices(the_graph);L)
| |   & (∀y:Vertices(the_graph). (y ∈ L))
| |   & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| 15. d : ℕ
| 16. ∀d1:ℕ.
| |   d1<d
| |   ⇒
| |   (∀s:traversal(the_graph), i:Vertices(the_graph).
| |    M(s)≤d1
| |    ⇒
| |    (∃s':traversal(the_graph).
| |     M(s' @ s)≤M(s) & P(i,s,s') & dfs(the_obj;s;i) = (s' @ s)))
| 17. s : traversal(the_graph)
| 18. i : Vertices(the_graph)
| 19. M(s)≤d
| 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| ⊢  ∃s':traversal(the_graph).
| |  M(s' @ s)≤M(s)
| |  & P(i,s,s')
| |  & [inl(i) /
| |     (the_obj.eacc((λs',j. dfs(the_obj;s';j)),[inr(i) / s],i))]
| |    =
| |    (s' @ s)
| | by Assert
| |    (∀L:Vertices(the_graph) List.
| |     (∀y:Vertices(the_graph). (y ∈ L) ⇒ i-the_graph->y)
| |     ⇒
| |     (∀s2:traversal(the_graph).
| |      M(s2)<d
| |      ⇒
| |      member-paren(x,y.the_obj.eq(x,y);i;s2)
| |      ⇒
| |      (∃s':traversal(the_graph).
| |       M(s' @ s2)≤M(s2)
| |       & P(i,s2,s')
| |       & list_accum(s',x'.dfs(the_obj;s';x');s2;L) = (s' @ s2))))
| |
| |\
| | .....assertion .....
```

```
| | 1. the_graph : Graph
| | 2. the_obj : GraphObject(the_graph)
| | 3. P :
| | |  Vertices(the_graph)→traversal(the_graph)→traversal(the_graph)
| | |  →Prop
| | 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | |  P(i,s1,s2)
| | |  ⇒
| | |  l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| | 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | |  P(i,s1,s2)
| | |  ⇒
| | |  no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| | 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| | |  member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| | 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| | |  i-the_graph->j
| | |  ⇒
| | |  P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| | 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | |  ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| | |  ⇒
| | |  P(i,[inr(i) / s1],s2) ⇒ P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| | 9. M : traversal(the_graph)→ℕ
| | 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | |   M([inl(i) / s])≤M(s)
| | 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | |   ¬member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ M([inr(i) / s])<M(s)
| | 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| | 13. ∀T:Type, s:T, x:Vertices(the_graph)
| | |   , f:(T→Vertices(the_graph)→T).
| | |   ∃L:Vertices(the_graph) List.
| | |   (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| | |   & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| | 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| | |   ∃L:Vertices(the_graph) List.
| | |   no_repeats(Vertices(the_graph);L)
| | |   & (∀y:Vertices(the_graph). (y ∈ L))
| | |   & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| | 15. d : ℕ
| | 16. ∀d1:ℕ.
| | |   d1<d
| | |   ⇒
| | |   (∀s:traversal(the_graph), i:Vertices(the_graph).
| | |    M(s)≤d1
| | |    ⇒
| | |    (∃s':traversal(the_graph).
| | |     M(s' @ s)≤M(s) & P(i,s,s') & dfs(the_obj;s;i) = (s' @ s)))
| | 17. s : traversal(the_graph)
| | 18. i : Vertices(the_graph)
| | 19. M(s)≤d
| | 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | ⊢  ∀L:Vertices(the_graph) List.
| | |  (∀y:Vertices(the_graph). (y ∈ L) ⇒ i-the_graph->y)
| | |  ⇒
```

```
| | |  (∀s2:traversal(the_graph).
| | |   M(s2)<d
| | |   ⇒
| | |   member-paren(x,y.the_obj.eq(x,y);i;s2)
| | |   ⇒
| | |   (∃s':traversal(the_graph).
| | |    M(s' @ s2)≤M(s2)
| | |    & P(i,s2,s')
| | |    & list_accum(s',x'.dfs(the_obj;s';x');s2;L) = (s' @ s2)))
| | | by InductionOnList THEN (Reduce 0)
| | |
| | |\
| | | 1. the_graph : Graph
| | | 2. the_obj : GraphObject(the_graph)
| | | 3. P :
| | |    Vertices(the_graph)→traversal(the_graph)→
| | |    traversal(the_graph)→Prop
| | | 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | |    P(i,s1,s2)
| | |    ⇒
| | |    l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| | | 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | |    P(i,s1,s2)
| | |    ⇒
| | |    no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| | | 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| | |    member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| | | 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| | |    i-the_graph->j
| | |    ⇒
| | |    P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| | | 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | |    ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| | |    ⇒
| | |    P(i,[inr(i) / s1],s2) ⇒ P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| | | 9. M : traversal(the_graph)→ℕ
| | | 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | |     M([inl(i) / s])≤M(s)
| | | 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | |     ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | |     ⇒
| | |     M([inr(i) / s])<M(s)
| | | 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| | | 13. ∀T:Type, s:T, x:Vertices(the_graph)
| | |     , f:(T→Vertices(the_graph)→T).
| | |     ∃L:Vertices(the_graph) List.
| | |     (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| | |     & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| | | 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| | |     ∃L:Vertices(the_graph) List.
| | |     no_repeats(Vertices(the_graph);L)
| | |     & (∀y:Vertices(the_graph). (y ∈ L))
| | |     & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| | | 15. d : ℕ
| | | 16. ∀d1:ℕ.
```

```
| | |     d1<d
| | |     ⇒
| | |     (∀s:traversal(the_graph), i:Vertices(the_graph).
| | |      M(s)≤d1
| | |      ⇒
| | |      (∃s':traversal(the_graph).
| | |       M(s' @ s)≤M(s) & P(i,s,s') & dfs(the_obj;s;i) = (s' @ s)))
| | | 17. s : traversal(the_graph)
| | | 18. i : Vertices(the_graph)
| | | 19. M(s)≤d
| | | 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | | 21. Vertices(the_graph) List
| | | ⊢  (∀y:Vertices(the_graph). (y ∈ nil) ⇒ i-the_graph->y)
| | |    ⇒
| | |    (∀s2:traversal(the_graph).
| | |     M(s2)<d
| | |     ⇒
| | |     member-paren(x,y.the_obj.eq(x,y);i;s2)
| | |     ⇒
| | |     (∃s':traversal(the_graph).
| | |      M(s' @ s2)≤M(s2) & P(i,s2,s') & s2 = (s' @ s2)))
| | |   by Auto THEN (InstConcl [nil]) THEN (Reduce 0) THEN EasyHyp
| | |
| | .\
| | . 1. the_graph : Graph
| | . 2. the_obj : GraphObject(the_graph)
| | . 3. P :

| | . |  Vertices(the_graph)→traversal(the_graph)→
| | . |  traversal(the_graph)→Prop
| | . 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . |  P(i,s1,s2)
| | . |  ⇒
| | . |  l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| | . 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . |  P(i,s1,s2)
| | . |  ⇒
| | . |  no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| | . 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| | . |  member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| | . 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| | . |  i-the_graph->j
| | . |  ⇒
| | . |  P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| | . 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . |  ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| | . |  ⇒
| | . |  P(i,[inr(i) / s1],s2) ⇒ P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| | . 9. M : traversal(the_graph)→ℕ
| | . 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . |   M([inl(i) / s])≤M(s)
| | . 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . |   ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . |   ⇒
| | . |   M([inr(i) / s])<M(s)
| | . 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
```

```
| | . 13. ∀T:Type, s:T, x:Vertices(the_graph)
| | . |    , f:(T→Vertices(the_graph)→T).
| | . |   ∃L:Vertices(the_graph) List.
| | . |   (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| | . |   & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| | . 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| | . |   ∃L:Vertices(the_graph) List.
| | . |   no_repeats(Vertices(the_graph);L)
| | . |   & (∀y:Vertices(the_graph). (y ∈ L))
| | . |   & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| | . 15. d : ℕ
| | . 16. ∀d1:ℕ.
| | . |   d1<d
| | . |   ⇒
| | . |   (∀s:traversal(the_graph), i:Vertices(the_graph).
| | . |    M(s)≤d1
| | . |    ⇒
| | . |    (∃s':traversal(the_graph).
| | . |     M(s' @ s)≤M(s) & P(i,s,s') & dfs(the_obj;s;i) = (s' @ s)))
| | . 17. s : traversal(the_graph)
| | . 18. i : Vertices(the_graph)
| | . 19. M(s)≤d
| | . 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . 21. Vertices(the_graph) List
| | . 22. u : Vertices(the_graph)
| | . 23. v : Vertices(the_graph) List
| | . 24. (∀y:Vertices(the_graph). (y ∈ v) ⇒ i-the_graph->y)
| | . |   ⇒
| | . |   (∀s2:traversal(the_graph).
| | . |    M(s2)<d
| | . |    ⇒
| | . |    member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . |    ⇒
| | . |    (∃s':traversal(the_graph).
| | . |     M(s' @ s2)≤M(s2)
| | . |     & P(i,s2,s')
| | . |     & list_accum(s',x'.dfs(the_obj;s';x');s2;v) = (s' @ s2)))
| | . ⊢  (∀y:Vertices(the_graph). (y ∈ [u / v]) ⇒ i-the_graph->y)
| | . |  ⇒
| | . |  (∀s2:traversal(the_graph).
| | . |   M(s2)<d
| | . |   ⇒
| | . |   member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . |   ⇒
| | . |   (∃s':traversal(the_graph).
| | . |    M(s' @ s2)≤M(s2)
| | . |    & P(i,s2,s')
| | . |    & list_accum(s',x'.dfs(the_obj;s';x');dfs(the_obj;s2;u);v)
| | . |      =
| | . |      (s' @ s2)))  by ParallelOp -1
| | . |
| | . |\
| | . | .....antecedent .....
| | . | 1. the_graph : Graph
| | . | 2. the_obj : GraphObject(the_graph)
```

```
| | . | 3. P :
| | . | |  Vertices(the_graph)→traversal(the_graph)→
| | . | |  traversal(the_graph)→Prop
| | . | 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . | |  P(i,s1,s2)
| | . | |  ⇒
| | . | |  l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| | . | 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . | |  P(i,s1,s2)
| | . | |  ⇒
| | . | |  no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| | . | 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| | . | |  member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| | . | 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| | . | |  i-the_graph->j
| | . | |  ⇒
| | . | |  P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| | . | 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . | |  ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| | . | |  ⇒
| | . | |  P(i,[inr(i) / s1],s2) ⇒ P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| | . | 9. M : traversal(the_graph)→ℕ
| | . | 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . | |   M([inl(i) / s])≤M(s)
| | . | 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . | |   ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . | |   ⇒
| | . | |   M([inr(i) / s])<M(s)
| | . | 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| | . | 13. ∀T:Type, s:T, x:Vertices(the_graph)
| | . | |   , f:(T→Vertices(the_graph)→T).
| | . | |   ∃L:Vertices(the_graph) List.
| | . | |   (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| | . | |   & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| | . | 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| | . | |   ∃L:Vertices(the_graph) List.
| | . | |   no_repeats(Vertices(the_graph);L)
| | . | |   & (∀y:Vertices(the_graph). (y ∈ L))
| | . | |   & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| | . | 15. d : ℕ
| | . | 16. ∀d1:ℕ.
| | . | |   d1<d
| | . | |   ⇒
| | . | |   (∀s:traversal(the_graph), i:Vertices(the_graph).
| | . | |    M(s)≤d1
| | . | |    ⇒
| | . | |    (∃s':traversal(the_graph).
| | . | |     M(s' @ s)≤M(s)
| | . | |     & P(i,s,s')
| | . | |     & dfs(the_obj;s;i) = (s' @ s)))
| | . | 17. s : traversal(the_graph)
| | . | 18. i : Vertices(the_graph)
| | . | 19. M(s)≤d
| | . | 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . | 21. Vertices(the_graph) List
```

```
| | . | 22. u : Vertices(the_graph)
| | . | 23. v : Vertices(the_graph) List
| | . | 24. ∀y:Vertices(the_graph). (y ∈ [u / v]) ⇒ i-the_graph->y
| | . | ⊢  ∀y:Vertices(the_graph). (y ∈ v) ⇒ i-the_graph->y
| | . | | by RepeatFor 2 (ParallelOp -1)
| | . | |
| | . | .....antecedent .....
| | . | 1. the_graph : Graph
| | . | 2. the_obj : GraphObject(the_graph)
| | . | 3. P :
| | . |    Vertices(the_graph)→traversal(the_graph)→
| | . |    traversal(the_graph)→Prop
| | . | 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . |    P(i,s1,s2)
| | . |    ⇒
| | . |    l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| | . | 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . |    P(i,s1,s2)
| | . |    ⇒
| | . |    no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| | . | 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| | . |    member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| | . | 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| | . |    i-the_graph->j
| | . |    ⇒
| | . |    P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| | . | 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . |    ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| | . |    ⇒
| | . |    P(i,[inr(i) / s1],s2) ⇒ P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| | . | 9. M : traversal(the_graph)→ℕ
| | . | 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . |     M([inl(i) / s])≤M(s)
| | . | 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . |     ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . |     ⇒
| | . |     M([inr(i) / s])<M(s)
| | . | 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| | . | 13. ∀T:Type, s:T, x:Vertices(the_graph)
| | . |     , f:(T→Vertices(the_graph)→T).
| | . |     ∃L:Vertices(the_graph) List.
| | . |     (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| | . |     & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| | . | 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| | . |     ∃L:Vertices(the_graph) List.
| | . |     no_repeats(Vertices(the_graph);L)
| | . |     & (∀y:Vertices(the_graph). (y ∈ L))
| | . |     & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| | . | 15. d : ℕ
| | . | 16. ∀d1:ℕ.
| | . |     d1<d
| | . |     ⇒
| | . |     (∀s:traversal(the_graph), i:Vertices(the_graph).
| | . |      M(s)≤d1
| | . |      ⇒
```

```
| | . |      (∃s':traversal(the_graph).
| | . |       M(s' @ s)≤M(s)
| | . |       & P(i,s,s')
| | . |       & dfs(the_obj;s;i) = (s' @ s)))
| | . | 17. s : traversal(the_graph)
| | . | 18. i : Vertices(the_graph)
| | . | 19. M(s)≤d
| | . | 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . | 21. Vertices(the_graph) List
| | . | 22. u : Vertices(the_graph)
| | . | 23. v : Vertices(the_graph) List
| | . | 24. ∀y:Vertices(the_graph). (y ∈ [u / v]) ⇒ i-the_graph->y
| | . | 25. y : Vertices(the_graph)
| | . | 26. (y ∈ v)
| | . | ⊢  (y ∈ [u / v])  by ObviousFrom [-1]
| | . |
| | . .\
| | . . 1. the_graph : Graph
| | . . 2. the_obj : GraphObject(the_graph)
| | . . 3. P :
| | . . |  Vertices(the_graph)→traversal(the_graph)→
| | . . |  traversal(the_graph)→Prop
| | . . 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . |  P(i,s1,s2)
| | . . |  ⇒
| | . . |  l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| | . . 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . |  P(i,s1,s2)
| | . . |  ⇒
| | . . |  no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| | . . 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . |  member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| | . . 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| | . . |  i-the_graph->j
| | . . |  ⇒
| | . . |  P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| | . . 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . |  ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| | . . |  ⇒
| | . . |  P(i,[inr(i) / s1],s2) ⇒ P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| | . . 9. M : traversal(the_graph)→ℕ
| | . . 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . |   M([inl(i) / s])≤M(s)
| | . . 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . |   ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . |   ⇒
| | . . |   M([inr(i) / s])<M(s)
| | . . 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| | . . 13. ∀T:Type, s:T, x:Vertices(the_graph)
| | . . |   , f:(T→Vertices(the_graph)→T).
| | . . |   ∃L:Vertices(the_graph) List.
| | . . |   (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| | . . |   & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| | . . 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| | . . |   ∃L:Vertices(the_graph) List.
```

```
| | . . |   no_repeats(Vertices(the_graph);L)
| | . . |   & (∀y:Vertices(the_graph). (y ∈ L))
| | . . |   & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| | . . 15. d : ℕ
| | . . 16. ∀d1:ℕ.
| | . . |   d1<d
| | . . |   ⇒
| | . . |   (∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . |    M(s)≤d1
| | . . |    ⇒
| | . . |    (∃s':traversal(the_graph).
| | . . |     M(s' @ s)≤M(s)
| | . . |     & P(i,s,s')
| | . . |     & dfs(the_obj;s;i) = (s' @ s)))
| | . . 17. s : traversal(the_graph)
| | . . 18. i : Vertices(the_graph)
| | . . 19. M(s)≤d
| | . . 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . 21. Vertices(the_graph) List
| | . . 22. u : Vertices(the_graph)
| | . . 23. v : Vertices(the_graph) List
| | . . 24. ∀y:Vertices(the_graph). (y ∈ [u / v]) ⇒ i-the_graph->y
| | . . 25. ∀s2:traversal(the_graph).
| | . . |   M(s2)<d
| | . . |   ⇒
| | . . |   member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . . |   ⇒
| | . . |   (∃s':traversal(the_graph).
| | . . |    M(s' @ s2)≤M(s2)
| | . . |    & P(i,s2,s')
| | . . |    & list_accum(s',x'.dfs(the_obj;s';x');s2;v) = (s' @ s2))
| | . . ⊢  ∀s2:traversal(the_graph).
| | . . |  M(s2)<d
| | . . |  ⇒
| | . . |  member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . . |  ⇒
| | . . |  (∃s':traversal(the_graph).
| | . . |   M(s' @ s2)≤M(s2)
| | . . |   & P(i,s2,s')
| | . . |   & list_accum(s',x'.dfs(the_obj;s';x');dfs(the_obj;s2;u);v)
| | . . |     =
| | . . |     (s' @ s2))  by Auto
| | . . |
| | . . 1. the_graph : Graph
| | . . 2. the_obj : GraphObject(the_graph)
| | . . 3. P :
| | . . |  Vertices(the_graph)→traversal(the_graph)→
| | . . |  traversal(the_graph)→Prop
| | . . 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . |  P(i,s1,s2)
| | . . |  ⇒
| | . . |  l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| | . . 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . |  P(i,s1,s2)
| | . . |  ⇒
```

```
| | . . |  no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| | . . 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . |  member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| | . . 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| | . . |  i-the_graph->j
| | . . |  ⇒
| | . . |  P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| | . . 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . |  ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| | . . |  ⇒
| | . . |  P(i,[inr(i) / s1],s2) ⇒ P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| | . . 9. M : traversal(the_graph)→ℕ
| | . . 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . |   M([inl(i) / s])≤M(s)
| | . . 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . |   ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . |   ⇒
| | . . |   M([inr(i) / s])<M(s)
| | . . 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| | . . 13. ∀T:Type, s:T, x:Vertices(the_graph)
| | . . |   , f:(T→Vertices(the_graph)→T).
| | . . |   ∃L:Vertices(the_graph) List.
| | . . |   (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| | . . |   & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| | . . 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| | . . |   ∃L:Vertices(the_graph) List.
| | . . |   no_repeats(Vertices(the_graph);L)
| | . . |   & (∀y:Vertices(the_graph). (y ∈ L))
| | . . |   & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| | . . 15. d : ℕ
| | . . 16. ∀d1:ℕ.
| | . . |   d1<d
| | . . |   ⇒
| | . . |   (∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . |    M(s)≤d1
| | . . |    ⇒
| | . . |    (∃s':traversal(the_graph).
| | . . |     M(s' @ s)≤M(s)
| | . . |     & P(i,s,s')
| | . . |     & dfs(the_obj;s;i) = (s' @ s)))
| | . . 17. s : traversal(the_graph)
| | . . 18. i : Vertices(the_graph)
| | . . 19. M(s)≤d
| | . . 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . 21. Vertices(the_graph) List
| | . . 22. u : Vertices(the_graph)
| | . . 23. v : Vertices(the_graph) List
| | . . 24. ∀y:Vertices(the_graph). (y ∈ [u / v]) ⇒ i-the_graph->y
| | . . 25. ∀s2:traversal(the_graph).
| | . . |   M(s2)<d
| | . . |   ⇒
| | . . |   member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . . |   ⇒
| | . . |   (∃s':traversal(the_graph).
| | . . |    M(s' @ s2)≤M(s2)
```

```
| | . . |    & P(i,s2,s')
| | . . |    & list_accum(s',x'.dfs(the_obj;s';x');s2;v) = (s' @ s2))
| | . . 26. s2 : traversal(the_graph)
| | . . 27. M(s2)<d
| | . . 28. member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . . ⊢  ∃s':traversal(the_graph).
| | . . |  M(s' @ s2)≤M(s2)
| | . . |  & P(i,s2,s')
| | . . |  & list_accum(s',x'.dfs(the_obj;s';x');dfs(the_obj;s2;u);v)
| | . . |    =
| | . . |    (s' @ s2)  by InstHyp [d-1;s2;u] 16
| | . . |
| | . . 1. the_graph : Graph
| | . . 2. the_obj : GraphObject(the_graph)
| | . . 3. P :
| | . . |  Vertices(the_graph)→traversal(the_graph)→
| | . . |  traversal(the_graph)→Prop
| | . . 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . |  P(i,s1,s2)
| | . . |  ⇒
| | . . |  l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| | . . 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . |  P(i,s1,s2)
| | . . |  ⇒
| | . . |  no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| | . . 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . |  member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| | . . 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| | . . |  i-the_graph->j
| | . . |  ⇒
| | . . |  P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| | . . 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . |  ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| | . . |  ⇒
| | . . |  P(i,[inr(i) / s1],s2) ⇒ P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| | . . 9. M : traversal(the_graph)→ℕ
| | . . 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . |   M([inl(i) / s])≤M(s)
| | . . 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . |   ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . |   ⇒
| | . . |   M([inr(i) / s])<M(s)
| | . . 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| | . . 13. ∀T:Type, s:T, x:Vertices(the_graph)
| | . . |   , f:(T→Vertices(the_graph)→T).
| | . . |   ∃L:Vertices(the_graph) List.
| | . . |   (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| | . . |   & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| | . . 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| | . . |   ∃L:Vertices(the_graph) List.
| | . . |   no_repeats(Vertices(the_graph);L)
| | . . |   & (∀y:Vertices(the_graph). (y ∈ L))
| | . . |   & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| | . . 15. d : ℕ
| | . . 16. ∀d1:ℕ.
```

```
| | . . |   d1<d
| | . . |   ⇒
| | . . |   (∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . |    M(s)≤d1
| | . . |    ⇒
| | . . |    (∃s':traversal(the_graph).
| | . . |     M(s' @ s)≤M(s)
| | . . |     & P(i,s,s')
| | . . |     & dfs(the_obj;s;i) = (s' @ s)))
| | . . 17. s : traversal(the_graph)
| | . . 18. i : Vertices(the_graph)
| | . . 19. M(s)≤d
| | . . 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . 21. Vertices(the_graph) List
| | . . 22. u : Vertices(the_graph)
| | . . 23. v : Vertices(the_graph) List
| | . . 24. ∀y:Vertices(the_graph). (y ∈ [u / v]) ⇒ i-the_graph->y
| | . . 25. ∀s2:traversal(the_graph).
| | . . |   M(s2)<d
| | . . |   ⇒
| | . . |   member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . . |   ⇒
| | . . |   (∃s':traversal(the_graph).
| | . . |    M(s' @ s2)≤M(s2)
| | . . |    & P(i,s2,s')
| | . . |    & list_accum(s',x'.dfs(the_obj;s';x');s2;v) = (s' @ s2))
| | . . 26. s2 : traversal(the_graph)
| | . . 27. M(s2)<d
| | . . 28. member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . . 29. ∃s':traversal(the_graph).
| | . . |   M(s' @ s2)≤M(s2)
| | . . |   & P(u,s2,s')
| | . . |   & dfs(the_obj;s2;u) = (s' @ s2)
| | . . ⊢  ∃s':traversal(the_graph).
| | . . |  M(s' @ s2)≤M(s2)
| | . . |  & P(i,s2,s')
| | . . |  & list_accum(s',x'.dfs(the_obj;s';x');dfs(the_obj;s2;u);v)
| | . . |    =
| | . . |    (s' @ s2)
| | . . | by ExRepD THEN (HypSubst -1 0) THEN (InstHyp [s' @ s2] -8)
| | . . |
| | . . |\
| | . . | .....antecedent .....
| | . . | 1. the_graph : Graph
| | . . | 2. the_obj : GraphObject(the_graph)
| | . . | 3. P :
| | . . | |  Vertices(the_graph)→traversal(the_graph)→
| | . . | |  traversal(the_graph)→Prop
| | . . | 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . | |  P(i,s1,s2)
| | . . | |  ⇒
| | . . | |  l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| | . . | 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . | |  P(i,s1,s2)
| | . . | |  ⇒
```

```
| | . . | |  no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| | . . | 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . | |  member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| | . . | 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| | . . | |  i-the_graph->j
| | . . | |  ⇒
| | . . | |  P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| | . . | 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . | |  ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| | . . | |  ⇒
| | . . | |  P(i,[inr(i) / s1],s2)
| | . . | |  ⇒
| | . . | |  P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| | . . | 9. M : traversal(the_graph)→ℕ
| | . . | 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . | |   M([inl(i) / s])≤M(s)
| | . . | 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . | |   ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . | |   ⇒
| | . . | |   M([inr(i) / s])<M(s)
| | . . | 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| | . . | 13. ∀T:Type, s:T, x:Vertices(the_graph)
| | . . | |   , f:(T→Vertices(the_graph)→T).
| | . . | |   ∃L:Vertices(the_graph) List.
| | . . | |   (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| | . . | |   & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| | . . | 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| | . . | |   ∃L:Vertices(the_graph) List.
| | . . | |   no_repeats(Vertices(the_graph);L)
| | . . | |   & (∀y:Vertices(the_graph). (y ∈ L))
| | . . | |   & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| | . . | 15. d : ℕ
| | . . | 16. ∀d1:ℕ.
| | . . | |   d1<d
| | . . | |   ⇒
| | . . | |   (∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . | |    M(s)≤d1
| | . . | |    ⇒
| | . . | |    (∃s':traversal(the_graph).
| | . . | |     M(s' @ s)≤M(s)
| | . . | |     & P(i,s,s')
| | . . | |     & dfs(the_obj;s;i) = (s' @ s)))
| | . . | 17. s : traversal(the_graph)
| | . . | 18. i : Vertices(the_graph)
| | . . | 19. M(s)≤d
| | . . | 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . | 21. Vertices(the_graph) List
| | . . | 22. u : Vertices(the_graph)
| | . . | 23. v : Vertices(the_graph) List
| | . . | 24. ∀y:Vertices(the_graph). (y ∈ [u / v]) ⇒ i-the_graph->y
| | . . | 25. ∀s2:traversal(the_graph).
| | . . | |   M(s2)<d
| | . . | |   ⇒
| | . . | |   member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . . | |   ⇒
```

```
| | . . | |   (∃s':traversal(the_graph).
| | . . | |    M(s' @ s2)≤M(s2)
| | . . | |    & P(i,s2,s')
| | . . | |    & list_accum(s',x'.dfs(the_obj;s';x');s2;v)
| | . . | |      =
| | . . | |      (s' @ s2))
| | . . | 26. s2 : traversal(the_graph)
| | . . | 27. M(s2)<d
| | . . | 28. member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . . | 29. s' : traversal(the_graph)
| | . . | 30. M(s' @ s2)≤M(s2)
| | . . | 31. P(u,s2,s')
| | . . | 32. dfs(the_obj;s2;u) = (s' @ s2)
| | . . | ⊢  member-paren(x,y.the_obj.eq(x,y);i;s' @ s2)
| | . . | | by (MoveToConcl -5)
| | . . | |    THEN
| | . . | |    (RWO
| | . . | |     Thm* ∀E:(T→T→𝔹).
| | . . | |          (∀x,y:T. E(x,y) ⟺ x = y)
| | . . | |          ⇒
| | . . | |          (∀i:T, s:(T+T) List.
| | . . | |           member-paren(x,y.E(x,y);i;s)
| | . . | |           ⟺
| | . . | |           (inl(i) ∈ s) ∨ (inr(i) ∈ s))
| | . . | |     {assert_DASH_member_DASH_paren}
| | . . | |     0)
| | . . | |
| | . . | 1. the_graph : Graph
| | . . | 2. the_obj : GraphObject(the_graph)
| | . . | 3. P :
| | . . |    Vertices(the_graph)→traversal(the_graph)→
| | . . |    traversal(the_graph)→Prop
| | . . | 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . |    P(i,s1,s2)
| | . . |    ⇒
| | . . |    l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| | . . | 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . |    P(i,s1,s2)
| | . . |    ⇒
| | . . |    no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| | . . | 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . |    member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| | . . | 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| | . . |    i-the_graph->j
| | . . |    ⇒
| | . . |    P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| | . . | 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . |    ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| | . . |    ⇒
| | . . |    P(i,[inr(i) / s1],s2)
| | . . |    ⇒
| | . . |    P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| | . . | 9. M : traversal(the_graph)→ℕ
| | . . | 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . |     M([inl(i) / s])≤M(s)
```

```
| | . . | 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . |     ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . |     ⇒
| | . . |     M([inr(i) / s])<M(s)
| | . . | 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| | . . | 13. ∀T:Type, s:T, x:Vertices(the_graph)
| | . . |     , f:(T→Vertices(the_graph)→T).
| | . . |     ∃L:Vertices(the_graph) List.
| | . . |     (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| | . . |     & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| | . . | 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| | . . |     ∃L:Vertices(the_graph) List.
| | . . |     no_repeats(Vertices(the_graph);L)
| | . . |     & (∀y:Vertices(the_graph). (y ∈ L))
| | . . |     & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| | . . | 15. d : ℕ
| | . . | 16. ∀d1:ℕ.
| | . . |     d1<d
| | . . |     ⇒
| | . . |     (∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . |      M(s)≤d1
| | . . |      ⇒
| | . . |      (∃s':traversal(the_graph).
| | . . |       M(s' @ s)≤M(s)
| | . . |       & P(i,s,s')
| | . . |       & dfs(the_obj;s;i) = (s' @ s)))
| | . . | 17. s : traversal(the_graph)
| | . . | 18. i : Vertices(the_graph)
| | . . | 19. M(s)≤d
| | . . | 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . | 21. Vertices(the_graph) List
| | . . | 22. u : Vertices(the_graph)
| | . . | 23. v : Vertices(the_graph) List
| | . . | 24. ∀y:Vertices(the_graph). (y ∈ [u / v]) ⇒ i-the_graph->y
| | . . | 25. ∀s2:traversal(the_graph).
| | . . |     M(s2)<d
| | . . |     ⇒
| | . . |     member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . . |     ⇒
| | . . |     (∃s':traversal(the_graph).
| | . . |      M(s' @ s2)≤M(s2)
| | . . |      & P(i,s2,s')
| | . . |      & list_accum(s',x'.dfs(the_obj;s';x');s2;v)
| | . . |        =
| | . . |        (s' @ s2))
| | . . | 26. s2 : traversal(the_graph)
| | . . | 27. M(s2)<d
| | . . | 28. s' : traversal(the_graph)
| | . . | 29. M(s' @ s2)≤M(s2)
| | . . | 30. P(u,s2,s')
| | . . | 31. dfs(the_obj;s2;u) = (s' @ s2)
| | . . | ⊢  (inl(i) ∈ s2) ∨ (inr(i) ∈ s2)
| | . . |    ⇒
| | . . |    (inl(i) ∈ s' @ s2) ∨ (inr(i) ∈ s' @ s2)  by ObviousConcl
| | . . |
```

```
| | . . .\
| | . . . 1. the_graph : Graph
| | . . . 2. the_obj : GraphObject(the_graph)
| | . . . 3. P :
| | . . . |  Vertices(the_graph)→traversal(the_graph)→
| | . . . |  traversal(the_graph)→Prop
| | . . . 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . . |  P(i,s1,s2)
| | . . . |  ⇒
| | . . . |  l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| | . . . 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . . |  P(i,s1,s2)
| | . . . |  ⇒
| | . . . |  no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| | . . . 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . . |  member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| | . . . 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| | . . . |  i-the_graph->j
| | . . . |  ⇒
| | . . . |  P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| | . . . 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . . |  ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| | . . . |  ⇒
| | . . . |  P(i,[inr(i) / s1],s2)
| | . . . |  ⇒
| | . . . |  P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| | . . . 9. M : traversal(the_graph)→ℕ
| | . . . 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . . |   M([inl(i) / s])≤M(s)
| | . . . 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . . |   ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . . |   ⇒
| | . . . |   M([inr(i) / s])<M(s)
| | . . . 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| | . . . 13. ∀T:Type, s:T, x:Vertices(the_graph)
| | . . . |   , f:(T→Vertices(the_graph)→T).
| | . . . |   ∃L:Vertices(the_graph) List.
| | . . . |   (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| | . . . |   & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| | . . . 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| | . . . |   ∃L:Vertices(the_graph) List.
| | . . . |   no_repeats(Vertices(the_graph);L)
| | . . . |   & (∀y:Vertices(the_graph). (y ∈ L))
| | . . . |   & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| | . . . 15. d : ℕ
| | . . . 16. ∀d1:ℕ.
| | . . . |   d1<d
| | . . . |   ⇒
| | . . . |   (∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . . |    M(s)≤d1
| | . . . |    ⇒
| | . . . |    (∃s':traversal(the_graph).
| | . . . |     M(s' @ s)≤M(s)
| | . . . |     & P(i,s,s')
| | . . . |     & dfs(the_obj;s;i) = (s' @ s)))
```

```
| | . . . 17. s : traversal(the_graph)
| | . . . 18. i : Vertices(the_graph)
| | . . . 19. M(s)≤d
| | . . . 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . . 21. Vertices(the_graph) List
| | . . . 22. u : Vertices(the_graph)
| | . . . 23. v : Vertices(the_graph) List
| | . . . 24. ∀y:Vertices(the_graph). (y ∈ [u / v]) ⇒ i-the_graph->y
| | . . . 25. ∀s2:traversal(the_graph).
| | . . . |   M(s2)<d
| | . . . |   ⇒
| | . . . |   member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . . . |   ⇒
| | . . . |   (∃s':traversal(the_graph).
| | . . . |    M(s' @ s2)≤M(s2)
| | . . . |    & P(i,s2,s')
| | . . . |    & list_accum(s',x'.dfs(the_obj;s';x');s2;v)
| | . . . |      =
| | . . . |      (s' @ s2))
| | . . . 26. s2 : traversal(the_graph)
| | . . . 27. M(s2)<d
| | . . . 28. member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . . . 29. s' : traversal(the_graph)
| | . . . 30. M(s' @ s2)≤M(s2)
| | . . . 31. P(u,s2,s')
| | . . . 32. dfs(the_obj;s2;u) = (s' @ s2)
| | . . . 33. ∃s'@0:traversal(the_graph).
| | . . . |   M(s'@0 @ s' @ s2)≤M(s' @ s2)
| | . . . |   & P(i,s' @ s2,s'@0)
| | . . . |   & list_accum(s',x'.dfs(the_obj;s';x');s' @ s2;v)
| | . . . |     =
| | . . . |     (s'@0 @ s' @ s2)
| | . . . ⊢  ∃s'@0:traversal(the_graph).
| | . . . |  M(s'@0 @ s2)≤M(s2)
| | . . . |  & P(i,s2,s'@0)
| | . . . |  & list_accum(s',x'.dfs(the_obj;s';x');s' @ s2;v)
| | . . . |    =
| | . . . |    (s'@0 @ s2)
| | . . . | by ExRepD THEN (HypSubst -1 0) THEN (Thin -1)
| | . . . |
| | . . . 1. the_graph : Graph
| | . . . 2. the_obj : GraphObject(the_graph)
| | . . . 3. P :
| | . . . |  Vertices(the_graph)→traversal(the_graph)→
| | . . . |  traversal(the_graph)→Prop
| | . . . 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . . |  P(i,s1,s2)
| | . . . |  ⇒
| | . . . |  l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| | . . . 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . . |  P(i,s1,s2)
| | . . . |  ⇒
| | . . . |  no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| | . . . 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . . |  member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
```

```
| | . . . 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| | . . . |  i-the_graph->j
| | . . . |  ⇒
| | . . . |  P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| | . . . 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . . |  ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| | . . . |  ⇒
| | . . . |  P(i,[inr(i) / s1],s2)
| | . . . |  ⇒
| | . . . |  P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| | . . . 9. M : traversal(the_graph)→ℕ
| | . . . 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . . |   M([inl(i) / s])≤M(s)
| | . . . 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . . |   ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . . |   ⇒
| | . . . |   M([inr(i) / s])<M(s)
| | . . . 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| | . . . 13. ∀T:Type, s:T, x:Vertices(the_graph)
| | . . . |   , f:(T→Vertices(the_graph)→T).
| | . . . |   ∃L:Vertices(the_graph) List.
| | . . . |   (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| | . . . |   & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| | . . . 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| | . . . |   ∃L:Vertices(the_graph) List.
| | . . . |   no_repeats(Vertices(the_graph);L)
| | . . . |   & (∀y:Vertices(the_graph). (y ∈ L))
| | . . . |   & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| | . . . 15. d : ℕ
| | . . . 16. ∀d1:ℕ.
| | . . . |   d1<d
| | . . . |   ⇒
| | . . . |   (∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . . |    M(s)≤d1
| | . . . |    ⇒
| | . . . |    (∃s':traversal(the_graph).
| | . . . |     M(s' @ s)≤M(s)
| | . . . |     & P(i,s,s')
| | . . . |     & dfs(the_obj;s;i) = (s' @ s)))
| | . . . 17. s : traversal(the_graph)
| | . . . 18. i : Vertices(the_graph)
| | . . . 19. M(s)≤d
| | . . . 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . . 21. Vertices(the_graph) List
| | . . . 22. u : Vertices(the_graph)
| | . . . 23. v : Vertices(the_graph) List
| | . . . 24. ∀y:Vertices(the_graph). (y ∈ [u / v]) ⇒ i-the_graph->y
| | . . . 25. ∀s2:traversal(the_graph).
| | . . . |   M(s2)<d
| | . . . |   ⇒
| | . . . |   member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . . . |   ⇒
| | . . . |   (∃s':traversal(the_graph).
| | . . . |    M(s' @ s2)≤M(s2)
| | . . . |    & P(i,s2,s')
```

```
| | . . . |    & list_accum(s',x'.dfs(the_obj;s';x');s2;v)
| | . . . |      =

| | . . . |      (s' @ s2))
| | . . . 26. s2 : traversal(the_graph)
| | . . . 27. M(s2)<d
| | . . . 28. member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . . . 29. s' : traversal(the_graph)
| | . . . 30. M(s' @ s2)≤M(s2)
| | . . . 31. P(u,s2,s')
| | . . . 32. dfs(the_obj;s2;u) = (s' @ s2)
| | . . . 33. s'@0 : traversal(the_graph)
| | . . . 34. M(s'@0 @ s' @ s2)≤M(s' @ s2)
| | . . . 35. P(i,s' @ s2,s'@0)
| | . . . ⊢  ∃s'@1:traversal(the_graph).
| | . . . |  M(s'@1 @ s2)≤M(s2)
| | . . . |  & P(i,s2,s'@1)
| | . . . |  & (s'@0 @ s' @ s2) = (s'@1 @ s2)
| | . . . | by (InstConcl [s'@0 @ s'])
| | . . . |    THEN
| | . . . |    (RWO
| | . . . |     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. P :
| | . . . |  Vertices(the_graph)→traversal(the_graph)→
| | . . . |  traversal(the_graph)→Prop
| | . . . 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . . |  P(i,s1,s2)
| | . . . |  ⇒
| | . . . |  l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| | . . . 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . . |  P(i,s1,s2)
| | . . . |  ⇒
| | . . . |  no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| | . . . 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . . |  member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| | . . . 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| | . . . |  i-the_graph->j
| | . . . |  ⇒
| | . . . |  P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| | . . . 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . . |  ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| | . . . |  ⇒
| | . . . |  P(i,[inr(i) / s1],s2)
| | . . . |  ⇒
| | . . . |  P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| | . . . 9. M : traversal(the_graph)→ℕ
| | . . . 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . . |   M([inl(i) / s])≤M(s)
| | . . . 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . . |   ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . . |   ⇒
```

```
| | . . . |   M([inr(i) / s])<M(s)
| | . . . 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| | . . . 13. ∀T:Type, s:T, x:Vertices(the_graph)
| | . . . |   , f:(T→Vertices(the_graph)→T).
| | . . . |   ∃L:Vertices(the_graph) List.
| | . . . |   (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| | . . . |   & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| | . . . 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| | . . . |   ∃L:Vertices(the_graph) List.
| | . . . |   no_repeats(Vertices(the_graph);L)
| | . . . |   & (∀y:Vertices(the_graph). (y ∈ L))
| | . . . |   & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| | . . . 15. d : ℕ
| | . . . 16. ∀d1:ℕ.
| | . . . |   d1<d
| | . . . |   ⇒
| | . . . |   (∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . . |    M(s)≤d1
| | . . . |    ⇒
| | . . . |    (∃s':traversal(the_graph).
| | . . . |     M(s' @ s)≤M(s)
| | . . . |     & P(i,s,s')
| | . . . |     & dfs(the_obj;s;i) = (s' @ s)))
| | . . . 17. s : traversal(the_graph)
| | . . . 18. i : Vertices(the_graph)
| | . . . 19. M(s)≤d
| | . . . 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . . 21. Vertices(the_graph) List
| | . . . 22. u : Vertices(the_graph)
| | . . . 23. v : Vertices(the_graph) List
| | . . . 24. ∀y:Vertices(the_graph). (y ∈ [u / v]) ⇒ i-the_graph->y
| | . . . 25. ∀s2:traversal(the_graph).
| | . . . |   M(s2)<d
| | . . . |   ⇒
| | . . . |   member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . . . |   ⇒
| | . . . |   (∃s':traversal(the_graph).
| | . . . |    M(s' @ s2)≤M(s2)
| | . . . |    & P(i,s2,s')
| | . . . |    & list_accum(s',x'.dfs(the_obj;s';x');s2;v)
| | . . . |      =
| | . . . |      (s' @ s2))
| | . . . 26. s2 : traversal(the_graph)
| | . . . 27. M(s2)<d
| | . . . 28. member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . . . 29. s' : traversal(the_graph)
| | . . . 30. M(s' @ s2)≤M(s2)

| | . . . 31. P(u,s2,s')
| | . . . 32. dfs(the_obj;s2;u) = (s' @ s2)
| | . . . 33. s'@0 : traversal(the_graph)
| | . . . 34. M(s'@0 @ s' @ s2)≤M(s' @ s2)
| | . . . 35. P(i,s' @ s2,s'@0)
| | . . . ⊢  P(i,s2,s'@0 @ s')
| | . . . | by (InstHyp [u] -12) THEN (Try (BackThruHyp' 7))
| | . . . |
```

```
| | . . . .....antecedent .....
| | . . . 1. the_graph : Graph
| | . . . 2. the_obj : GraphObject(the_graph)
| | . . . 3. P :
| | . . .    Vertices(the_graph)→traversal(the_graph)→
| | . . .    traversal(the_graph)→Prop
| | . . . 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . .    P(i,s1,s2)
| | . . .    ⇒
| | . . .    l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| | . . . 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . .    P(i,s1,s2)
| | . . .    ⇒
| | . . .    no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| | . . . 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . .    member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| | . . . 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| | . . .    i-the_graph->j
| | . . .    ⇒
| | . . .    P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| | . . . 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| | . . .    ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| | . . .    ⇒
| | . . .    P(i,[inr(i) / s1],s2)
| | . . .    ⇒
| | . . .    P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| | . . . 9. M : traversal(the_graph)→ℕ
| | . . . 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . .     M([inl(i) / s])≤M(s)
| | . . . 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| | . . .     ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . .     ⇒
| | . . .     M([inr(i) / s])<M(s)
| | . . . 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⇐⇒ x = y
| | . . . 13. ∀T:Type, s:T, x:Vertices(the_graph)
| | . . .     , f:(T→Vertices(the_graph)→T).
| | . . .     ∃L:Vertices(the_graph) List.
| | . . .     (∀y:Vertices(the_graph). x-the_graph->y ⇐⇒ (y ∈ L))
| | . . .     & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| | . . . 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| | . . .     ∃L:Vertices(the_graph) List.
| | . . .     no_repeats(Vertices(the_graph);L)
| | . . .     & (∀y:Vertices(the_graph). (y ∈ L))
| | . . .     & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| | . . . 15. d : ℕ
| | . . . 16. ∀d1:ℕ.
| | . . .     d1<d
| | . . .     ⇒
| | . . .     (∀s:traversal(the_graph), i:Vertices(the_graph).
| | . . .      M(s)≤d1
| | . . .      ⇒
| | . . .      (∃s':traversal(the_graph).
| | . . .       M(s' @ s)≤M(s)
| | . . .       & P(i,s,s')
| | . . .       & dfs(the_obj;s;i) = (s' @ s)))
```

```
| | . . . 17. s : traversal(the_graph)
| | . . . 18. i : Vertices(the_graph)
| | . . . 19. M(s)≤d
| | . . . 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| | . . . 21. Vertices(the_graph) List
| | . . . 22. u : Vertices(the_graph)
| | . . . 23. v : Vertices(the_graph) List
| | . . . 24. ∀y:Vertices(the_graph). (y ∈ [u / v]) ⇒ i-the_graph->y
| | . . . 25. ∀s2:traversal(the_graph).
| | . . .     M(s2)<d
| | . . .     ⇒
| | . . .     member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . . .     ⇒
| | . . .     (∃s':traversal(the_graph).
| | . . .      M(s' @ s2)≤M(s2)
| | . . .      & P(i,s2,s')
| | . . .      & list_accum(s',x'.dfs(the_obj;s';x');s2;v)
| | . . .        =
| | . . .        (s' @ s2))
| | . . . 26. s2 : traversal(the_graph)
| | . . . 27. M(s2)<d
| | . . . 28. member-paren(x,y.the_obj.eq(x,y);i;s2)
| | . . . 29. s' : traversal(the_graph)
| | . . . 30. M(s' @ s2)≤M(s2)
| | . . . 31. P(u,s2,s')
| | . . . 32. dfs(the_obj;s2;u) = (s' @ s2)
| | . . . 33. s'@0 : traversal(the_graph)
| | . . . 34. M(s'@0 @ s' @ s2)≤M(s' @ s2)
| | . . . 35. P(i,s' @ s2,s'@0)
| | . . . ⊢  (u ∈ [u / v])  by ObviousConcl
| |
| .\
| . 1. the_graph : Graph
| . 2. the_obj : GraphObject(the_graph)
| . 3. P :
| . |  Vertices(the_graph)→traversal(the_graph)→traversal(the_graph)
| . |  →Prop
| . 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| . |  P(i,s1,s2)
| . |  ⇒
| . |  l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| . 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| . |  P(i,s1,s2)
| . |  ⇒
| . |  no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| . 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| . |  member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| . 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| . |  i-the_graph->j
| . |  ⇒
| . |  P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| . 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| . |  ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| . |  ⇒
| . |  P(i,[inr(i) / s1],s2) ⇒ P(i,s1,[inl(i) / (s2 @ [inr(i)])])
```

```
| . 9. M : traversal(the_graph)→ℕ
| . 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| . |   M([inl(i) / s])≤M(s)
| . 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| . |   ¬member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ M([inr(i) / s])<M(s)
| . 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| . 13. ∀T:Type, s:T, x:Vertices(the_graph)
| . |   , f:(T→Vertices(the_graph)→T).
| . |   ∃L:Vertices(the_graph) List.
| . |   (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| . |   & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| . 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| . |   ∃L:Vertices(the_graph) List.
| . |   no_repeats(Vertices(the_graph);L)
| . |   & (∀y:Vertices(the_graph). (y ∈ L))
| . |   & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| . 15. d : ℕ
| . 16. ∀d1:ℕ.
| . |   d1<d
| . |   ⇒
| . |   (∀s:traversal(the_graph), i:Vertices(the_graph).
| . |    M(s)≤d1
| . |    ⇒
| . |    (∃s':traversal(the_graph).
| . |     M(s' @ s)≤M(s) & P(i,s,s') & dfs(the_obj;s;i) = (s' @ s)))
| . 17. s : traversal(the_graph)
| . 18. i : Vertices(the_graph)
| . 19. M(s)≤d
| . 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| . 21. ∀L:Vertices(the_graph) List.
| . |   (∀y:Vertices(the_graph). (y ∈ L) ⇒ i-the_graph->y)
| . |   ⇒
| . |   (∀s2:traversal(the_graph).
| . |    M(s2)<d
| . |    ⇒
| . |    member-paren(x,y.the_obj.eq(x,y);i;s2)
| . |    ⇒
| . |    (∃s':traversal(the_graph).
| . |     M(s' @ s2)≤M(s2)
| . |     & P(i,s2,s')
| . |     & list_accum(s',x'.dfs(the_obj;s';x');s2;L) = (s' @ s2)))
| . ⊢  ∃s':traversal(the_graph).
| . |  M(s' @ s)≤M(s)
| . |  & P(i,s,s')
| . |  & [inl(i) /
| . |     (the_obj.eacc((λs',j. dfs(the_obj;s';j)),[inr(i) / s],i))]
| . |    =
| . |    (s' @ s)
| . | by ((InstHyp
| . |      [traversal(the_graph)
| . |      ;[inr(i) / s]
| . |      ;i
| . |      ;λs',j. dfs(the_obj;s';j)]
| . |      13)
| . |     THEN
```

```
| . |     ExRepD
| . |     THEN
| . |     (HypSubst -1 0)
| . |     THEN
| . |     (Thin -1)
| . |     THEN
| . |     (InstHyp [L;[inr(i) / s]] -3))
| . |    THENA
| . |    (Auto THEN EasyHyp
| . |     THEN
| . |     (Try
| . |      (RWO
| . |       Thm* ∀E:(T→T→𝔹).
| . |            (∀x,y:T. E(x,y) ⟺ x = y)
| . |            ⇒
| . |            (∀i:T, s:(T+T) List.
| . |             member-paren(x,y.E(x,y);i;s)
| . |             ⟺
| . |             (inl(i) ∈ s) ∨ (inr(i) ∈ s))
| . |       {assert_DASH_member_DASH_paren}
| . |       0))
| . |     THEN
| . |     (Try (AutoInst [i;s])))
| . |
| . |\
| . | 1. the_graph : Graph
| . | 2. the_obj : GraphObject(the_graph)
| . | 3. P :
| . |    Vertices(the_graph)→traversal(the_graph)→
| . |    traversal(the_graph)→Prop
| . | 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| . |    P(i,s1,s2)
| . |    ⇒
| . |    l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| . | 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| . |    P(i,s1,s2)
| . |    ⇒
| . |    no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| . | 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| . |    member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| . | 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| . |    i-the_graph->j
| . |    ⇒
| . |    P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| . | 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| . |    ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| . |    ⇒
| . |    P(i,[inr(i) / s1],s2) ⇒ P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| . | 9. M : traversal(the_graph)→ℕ
| . | 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| . |     M([inl(i) / s])≤M(s)
| . | 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| . |     ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| . |     ⇒
| . |     M([inr(i) / s])<M(s)
```

```
| . | 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| . | 13. ∀T:Type, s:T, x:Vertices(the_graph)
| . |     , f:(T→Vertices(the_graph)→T).
| . |     ∃L:Vertices(the_graph) List.
| . |     (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| . |     & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| . | 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| . |     ∃L:Vertices(the_graph) List.
| . |     no_repeats(Vertices(the_graph);L)
| . |     & (∀y:Vertices(the_graph). (y ∈ L))
| . |     & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| . | 15. d : ℕ
| . | 16. ∀d1:ℕ.
| . |     d1<d
| . |     ⇒
| . |     (∀s:traversal(the_graph), i:Vertices(the_graph).
| . |      M(s)≤d1
| . |      ⇒
| . |      (∃s':traversal(the_graph).
| . |       M(s' @ s)≤M(s) & P(i,s,s') & dfs(the_obj;s;i) = (s' @ s)))
| . | 17. s : traversal(the_graph)
| . | 18. i : Vertices(the_graph)
| . | 19. M(s)≤d
| . | 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| . | 21. ∀L:Vertices(the_graph) List.
| . |     (∀y:Vertices(the_graph). (y ∈ L) ⇒ i-the_graph->y)
| . |     ⇒
| . |     (∀s2:traversal(the_graph).
| . |      M(s2)<d
| . |      ⇒
| . |      member-paren(x,y.the_obj.eq(x,y);i;s2)
| . |      ⇒
| . |      (∃s':traversal(the_graph).
| . |       M(s' @ s2)≤M(s2)
| . |       & P(i,s2,s')
| . |       & list_accum(s',x'.dfs(the_obj;s';x');s2;L) = (s' @ s2)))
| . | 22. L : Vertices(the_graph) List
| . | 23. ∀y:Vertices(the_graph). i-the_graph->y ⟺ (y ∈ L)
| . | ⊢  (inl(i) ∈ [inr(i) / s]) ∨ (inr(i) ∈ [inr(i) / s])
| . |   by ObviousConcl
| . |
| . .\
| . . 1. the_graph : Graph
| . . 2. the_obj : GraphObject(the_graph)
| . . 3. P :
| . . |  Vertices(the_graph)→traversal(the_graph)→
| . . |  traversal(the_graph)→Prop
| . . 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| . . |  P(i,s1,s2)
| . . |  ⇒
| . . |  l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| . . 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| . . |  P(i,s1,s2)
| . . |  ⇒
| . . |  no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
```

```
| . . 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| . . |  member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| . . 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| . . |  i-the_graph->j
| . . |  ⇒
| . . |  P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| . . 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| . . |  ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| . . |  ⇒
| . . |  P(i,[inr(i) / s1],s2) ⇒ P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| . . 9. M : traversal(the_graph)→ℕ
| . . 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| . . |   M([inl(i) / s])≤M(s)
| . . 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| . . |   ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| . . |   ⇒
| . . |   M([inr(i) / s])<M(s)
| . . 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| . . 13. ∀T:Type, s:T, x:Vertices(the_graph)
| . . |   , f:(T→Vertices(the_graph)→T).
| . . |   ∃L:Vertices(the_graph) List.
| . . |   (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| . . |   & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| . . 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| . . |   ∃L:Vertices(the_graph) List.
| . . |   no_repeats(Vertices(the_graph);L)
| . . |   & (∀y:Vertices(the_graph). (y ∈ L))
| . . |   & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| . . 15. d : ℕ
| . . 16. ∀d1:ℕ.
| . . |   d1<d
| . . |   ⇒
| . . |   (∀s:traversal(the_graph), i:Vertices(the_graph).
| . . |    M(s)≤d1
| . . |    ⇒
| . . |    (∃s':traversal(the_graph).
| . . |     M(s' @ s)≤M(s) & P(i,s,s') & dfs(the_obj;s;i) = (s' @ s)))
| . . 17. s : traversal(the_graph)
| . . 18. i : Vertices(the_graph)
| . . 19. M(s)≤d
| . . 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| . . 21. ∀L:Vertices(the_graph) List.
| . . |   (∀y:Vertices(the_graph). (y ∈ L) ⇒ i-the_graph->y)
| . . |   ⇒
| . . |   (∀s2:traversal(the_graph).
| . . |    M(s2)<d
| . . |    ⇒
| . . |    member-paren(x,y.the_obj.eq(x,y);i;s2)
| . . |    ⇒
| . . |    (∃s':traversal(the_graph).
| . . |     M(s' @ s2)≤M(s2)
| . . |     & P(i,s2,s')
| . . |     & list_accum(s',x'.dfs(the_obj;s';x');s2;L) = (s' @ s2)))
| . . 22. L : Vertices(the_graph) List
| . . 23. ∀y:Vertices(the_graph). i-the_graph->y ⟺ (y ∈ L)
```

```
| . . 24. ∃s':traversal(the_graph).
| . . |   M(s' @ [inr(i) / s])≤M([inr(i) / s])
| . . |   & P(i,[inr(i) / s],s')
| . . |   & list_accum(s',x'.dfs(the_obj;s';x');[inr(i) / s];L)
| . . |     =
| . . |     (s' @ [inr(i) / s])
| . . ⊢  ∃s':traversal(the_graph).
| . . |  M(s' @ s)≤M(s)
| . . |  & P(i,s,s')
| . . |  & [inl(i) /
| . . |     list_accum(s',x'.(λs',j. dfs(the_obj;s';j))(s'
| . . |                                                  ,x');[inr(i) /
| . . |                                                        s];L)]
| . . |    =
| . . |    (s' @ s)
| . . | by (ReduceSOAps 0) THEN ExRepD
| . . |    THEN
| . . |    (InstConcl [[inl(i)] @ s' @ [inr(i)]])
| . . |    THEN
| . . |    (Reduce 0)
| . . |    THEN
| . . |    EasyHyp
| . . |    THEN
| . . |    (Try
| . . |     ((RW0
| . . |       Thm* ∀a,b,c:Top List. ((a @ b) @ c) ~ (a @ b @ c)
| . . |       {append_assoc_sq}
| . . |       0)
| . . |      THEN
| . . |      (Reduce 0)
| . . |      THEN
| . . |      Analyze))
| . . |
| . . 1. the_graph : Graph
| . . 2. the_obj : GraphObject(the_graph)
| . . 3. P :
| . .    Vertices(the_graph)→traversal(the_graph)→
| . .    traversal(the_graph)→Prop
| . . 4. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| . .    P(i,s1,s2)
| . .    ⇒
| . .    l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
| . . 5. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| . .    P(i,s1,s2)
| . .    ⇒
| . .    no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
| . . 6. ∀s:traversal(the_graph), i:Vertices(the_graph).
| . .    member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
| . . 7. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
| . .    i-the_graph->j
| . .    ⇒
| . .    P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
| . . 8. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
| . .    ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
| . .    ⇒
```

```
| . .    P(i,[inr(i) / s1],s2) ⇒ P(i,s1,[inl(i) / (s2 @ [inr(i)])])
| . . 9. M : traversal(the_graph)→ℕ
| . . 10. ∀i:Vertices(the_graph), s:traversal(the_graph).
| . .     M([inl(i) / s])≤M(s)
| . . 11. ∀i:Vertices(the_graph), s:traversal(the_graph).
| . .     ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| . .     ⇒
| . .     M([inr(i) / s])<M(s)
| . . 12. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| . . 13. ∀T:Type, s:T, x:Vertices(the_graph)
| . .     , f:(T→Vertices(the_graph)→T).
| . .     ∃L:Vertices(the_graph) List.
| . .     (∀y:Vertices(the_graph). x-the_graph->y ⟺ (y ∈ L))
| . .     & the_obj.eacc(f,s,x) = list_accum(s',x'.f(s',x');s;L)
| . . 14. ∀T:Type, s:T, f:(T→Vertices(the_graph)→T).
| . .     ∃L:Vertices(the_graph) List.
| . .     no_repeats(Vertices(the_graph);L)
| . .     & (∀y:Vertices(the_graph). (y ∈ L))
| . .     & the_obj.vacc(f,s) = list_accum(s',x'.f(s',x');s;L)
| . . 15. d : ℕ
| . . 16. ∀d1:ℕ.
| . .     d1<d
| . .     ⇒
| . .     (∀s:traversal(the_graph), i:Vertices(the_graph).
| . .      M(s)≤d1
| . .      ⇒
| . .      (∃s':traversal(the_graph).
| . .       M(s' @ s)≤M(s) & P(i,s,s') & dfs(the_obj;s;i) = (s' @ s)))
| . . 17. s : traversal(the_graph)
| . . 18. i : Vertices(the_graph)
| . . 19. M(s)≤d
| . . 20. ¬member-paren(x,y.the_obj.eq(x,y);i;s)
| . . 21. ∀L:Vertices(the_graph) List.
| . .     (∀y:Vertices(the_graph). (y ∈ L) ⇒ i-the_graph->y)
| . .     ⇒
| . .     (∀s2:traversal(the_graph).
| . .      M(s2)<d
| . .      ⇒
| . .      member-paren(x,y.the_obj.eq(x,y);i;s2)
| . .      ⇒
| . .      (∃s':traversal(the_graph).
| . .       M(s' @ s2)≤M(s2)
| . .       & P(i,s2,s')
| . .       & list_accum(s',x'.dfs(the_obj;s';x');s2;L) = (s' @ s2)))
| . . 22. L : Vertices(the_graph) List
| . . 23. ∀y:Vertices(the_graph). i-the_graph->y ⟺ (y ∈ L)
| . . 24. s' : traversal(the_graph)
| . . 25. M(s' @ [inr(i) / s])≤M([inr(i) / s])
| . . 26. P(i,[inr(i) / s],s')
| . . 27. list_accum(s',x'.dfs(the_obj;s';x');[inr(i) / s];L)
| . .     =
| . .     (s' @ [inr(i) / s])
| . . ⊢  M([inl(i) / ((s' @ [inr(i)]) @ s)])≤M(s)
| . .   by (RWO
| . .       Thm* ∀a,b,c:Top List. ((a @ b) @ c) ∼ (a @ b @ c)
```

```
| . .       {append_assoc_sq}
| . .       0)
| . .      THEN
| . .      (Reduce 0)
| . .      THEN
| . .      (InstHyp [i;(s' @ [inr(i)]) @ s] 10)
| . .      THEN
| . .      (RWO
| . .       Thm* ∀a,b,c:Top List. ((a @ b) @ c) ~ (a @ b @ c)
| . .       {append_assoc_sq}
| . .       -1)
| . .      THEN
| . .      (Reduce -1)
| . .      THEN
| . .      (InstHyp [i;s] 11)
|
.\
. 1. the_graph : Graph
. 2. the_obj : GraphObject(the_graph)
. 3. P :
.    Vertices(the_graph)→traversal(the_graph)→traversal(the_graph)→
.    Prop
. 4. s : traversal(the_graph)
. 5. i : Vertices(the_graph)
. 6. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
.    P(i,s1,s2)
.    ⇒
.    l_disjoint(Vertices(the_graph)+Vertices(the_graph);s2;s1)
. 7. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
.    P(i,s1,s2)
.    ⇒
.    no_repeats(Vertices(the_graph)+Vertices(the_graph);s2)
. 8. ∀s:traversal(the_graph), i:Vertices(the_graph).
.    member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ P(i,s,nil)
. 9. ∀s1,s2,s3:traversal(the_graph), i,j:Vertices(the_graph).
.    i-the_graph->j
.    ⇒
.    P(j,s1,s2) ⇒ P(i,s2 @ s1,s3) ⇒ P(i,s1,s3 @ s2)
. 10. ∀s1,s2:traversal(the_graph), i:Vertices(the_graph).
.     ¬member-paren(x,y.the_obj.eq(x,y);i;s1)
.     ⇒
.     P(i,[inr(i) / s1],s2) ⇒ P(i,s1,[inl(i) / (s2 @ [inr(i)])])
. 11. M : traversal(the_graph)→ℕ
. 12. ∀i:Vertices(the_graph), s:traversal(the_graph).
.     M([inl(i) / s])≤M(s)
. 13. ∀i:Vertices(the_graph), s:traversal(the_graph).
.     ¬member-paren(x,y.the_obj.eq(x,y);i;s) ⇒ M([inr(i) / s])<M(s)
. 14. ∀d:ℕ, s:traversal(the_graph), i:Vertices(the_graph).
.     M(s)≤d
.     ⇒
.     (∃s':traversal(the_graph).
.      M(s' @ s)≤M(s) & P(i,s,s') & dfs(the_obj;s;i) = (s' @ s))

. ⊢  ∃s':traversal(the_graph). P(i,s,s') & dfs(the_obj;s;i) = (s' @ s)
.   by (InstHyp [M(s);s;i] -1) THEN (ParallelOp -1)
END
```