```
dfs-cases  THM

⊢  For any graph
|  ∀the_obj:GraphObject(the_graph), s:Traversal, i:V.
|  ∃s':Traversal.
|  ((inr(i) ∈ s) ∨ (inl(i) ∈ s) ⇒ s' = nil)
|  & (¬(inr(i) ∈ s) & ¬(inl(i) ∈ s)
|      ⇒
|      (∃s2:Traversal. s' = ([inl(i)] @ s2 @ [inr(i)])))
|  & dfs(the_obj;s;i) = (s' @ s)
| by Auto
|    THEN
|    (Inst Thm: graphobj_DASH_properties [the_graph;the_obj])
|    THEN
|    (Analyze -1)
|    THEN
|    (Analyze -1)
|    THEN
|    (RecUnfold `dfs` 0)
|    THEN
|    SplitOnConclITE
|    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}
|     -1)
|
|\
| 1. the_graph : Graph
| 2. the_obj : GraphObject(the_graph)
| 3. s : traversal(the_graph)
| 4. i : Vertices(the_graph)
| 5. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
| 6. ∀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)
| 7. ∀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)
| 8. (inl(i) ∈ s) ∨ (inr(i) ∈ s)
| ⊢  ∃s':traversal(the_graph).
|    ((inr(i) ∈ s) ∨ (inl(i) ∈ s) ⇒ s' = nil)
|    & (¬(inr(i) ∈ s) & ¬(inl(i) ∈ s)
|       ⇒
|       (∃s2:traversal(the_graph). s' = ([inl(i)] @ s2 @ [inr(i)])))
```

```
|    & s = (s' @ s)
|   by (InstConcl [nil]) THEN (Reduce 0) THEN (Analyze -3)
|
.\
. 1. the_graph : Graph
. 2. the_obj : GraphObject(the_graph)
. 3. s : traversal(the_graph)
. 4. i : Vertices(the_graph)
. 5. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
. 6. ∀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)
. 7. ∀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)
. 8. ¬((inl(i) ∈ s) ∨ (inr(i) ∈ s))
. ⊢  ∃s':traversal(the_graph).
. |  ((inr(i) ∈ s) ∨ (inl(i) ∈ s) ⇒ s' = nil)
. |  & (¬(inr(i) ∈ s) & ¬(inl(i) ∈ s)
. |     ⇒
. |     (∃s2:traversal(the_graph). s' = ([inl(i)] @ s2 @ [inr(i)])))
. |  & [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, s:traversal(the_graph).
. |     ∃s2:traversal(the_graph).
. |     list_accum(s',j.dfs(the_obj;s';j);s;L) = (s2 @ s))
. |
. |\
. | .....assertion .....
. | 1. the_graph : Graph
. | 2. the_obj : GraphObject(the_graph)
. | 3. s : traversal(the_graph)
. | 4. i : Vertices(the_graph)
. | 5. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
. | 6. ∀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)
. | 7. ∀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)
. | 8. ¬((inl(i) ∈ s) ∨ (inr(i) ∈ s))
. | ⊢  ∀L:Vertices(the_graph) List, s:traversal(the_graph).
. | |  ∃s2:traversal(the_graph).
. | |  list_accum(s',j.dfs(the_obj;s';j);s;L) = (s2 @ s)
```

```
. | | by InductionOnList THEN (Reduce 0)
. | |
. | |\
. | | 1. the_graph : Graph
. | | 2. the_obj : GraphObject(the_graph)
. | | 3. s : traversal(the_graph)
. | | 4. i : Vertices(the_graph)
. | | 5. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
. | | 6. ∀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)
. | | 7. ∀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)
. | | 8. ¬((inl(i) ∈ s) ∨ (inr(i) ∈ s))
. | | 9. Vertices(the_graph) List
. | | 10. s1 : traversal(the_graph)
. | | ⊢  ∃s2:traversal(the_graph). s1 = (s2 @ s1)
. | |   by (InstConcl [nil]) THEN (Reduce 0)
. | |
. | .\
. | . 1. the_graph : Graph
. | . 2. the_obj : GraphObject(the_graph)
. | . 3. s : traversal(the_graph)
. | . 4. i : Vertices(the_graph)
. | . 5. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
. | . 6. ∀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)
. | . 7. ∀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)
. | . 8. ¬((inl(i) ∈ s) ∨ (inr(i) ∈ s))
. | . 9. Vertices(the_graph) List
. | . 10. u : Vertices(the_graph)
. | . 11. v : Vertices(the_graph) List
. | . 12. ∀s:traversal(the_graph).
. | . |   ∃s2:traversal(the_graph).
. | . |   list_accum(s',j.dfs(the_obj;s';j);s;v) = (s2 @ s)
. | . 13. s1 : traversal(the_graph)
. | . ⊢  ∃s2:traversal(the_graph).
. | . |  list_accum(s',j.dfs(the_obj;s';j);dfs(the_obj;s1;u);v)
. | . |  =
. | . |  (s2 @ s1)
. | . | by (InstHyp [dfs(the_obj;s1;u)] -2) THEN ExRepD
. | . |    THEN
. | . |    (HypSubst -1 0)
```

```
. | . |    THEN
. | . |    (Inst Thm: dfs_member [the_graph;the_obj;s1;u])
. | . |    THEN
. | . |    ExRepD
. | . |    THEN
. | . |    (HypSubst -1 0)
. | . |
. | . 1. the_graph : Graph
. | . 2. the_obj : GraphObject(the_graph)
. | . 3. s : traversal(the_graph)
. | . 4. i : Vertices(the_graph)
. | . 5. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
. | . 6. ∀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)
. | . 7. ∀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)
. | . 8. ¬((inl(i) ∈ s) ∨ (inr(i) ∈ s))
. | . 9. Vertices(the_graph) List
. | . 10. u : Vertices(the_graph)
. | . 11. v : Vertices(the_graph) List
. | . 12. ∀s:traversal(the_graph).
. | .     ∃s2:traversal(the_graph).
. | .     list_accum(s',j.dfs(the_obj;s';j);s;v) = (s2 @ s)
. | . 13. s1 : traversal(the_graph)
. | . 14. s2 : traversal(the_graph)
. | . 15. list_accum(s',j.dfs(the_obj;s';j);dfs(the_obj;s1;u);v)
. | .     =
. | .     (s2 @ dfs(the_obj;s1;u))
. | . 16. s' : traversal(the_graph)
. | . 17. (inl(u) ∈ s') ∨ (inl(u) ∈ s1) ∨ (inr(u) ∈ s1)
. | . 18. l_disjoint(Vertices(the_graph)+Vertices(the_graph);s';s1)
. | . 19. no_repeats(Vertices(the_graph)+Vertices(the_graph);s')
. | . 20. paren(Vertices(the_graph);s')
. | . 21. dfs(the_obj;s1;u) = (s' @ s1)
. | . ⊢  ∃s2@0:traversal(the_graph). (s2 @ s' @ s1) = (s2@0 @ s1)
. | .   by (InstConcl [s2 @ s'])
. | .      THEN
. | .      (RW0
. | .       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. s : traversal(the_graph)
. . 4. i : Vertices(the_graph)
. . 5. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
. . 6. ∀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)
. . 7. ∀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)
. . 8. ¬((inl(i) ∈ s) ∨ (inr(i) ∈ s))
. . 9. ∀L:Vertices(the_graph) List, s:traversal(the_graph).
. . |  ∃s2:traversal(the_graph).
. . |  list_accum(s',j.dfs(the_obj;s';j);s;L) = (s2 @ s)
. . ⊢  ∃s':traversal(the_graph).
. . |  ((inr(i) ∈ s) ∨ (inl(i) ∈ s) ⇒ s' = nil)
. . |  & (¬(inr(i) ∈ s) & ¬(inl(i) ∈ s)
. . |     ⇒
. . |     (∃s2:traversal(the_graph). s' = ([inl(i)] @ s2 @ [inr(i)])))
. . |  & [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)]
. . |     6)
. . |    THEN
. . |    (ReduceSOAps -1)
. . |    THEN
. . |    ExRepD
. . |    THEN
. . |    (HypSubst -1 0)
. . |    THEN
. . |    (Thin -1)
. . |    THEN
. . |    (InstHyp [L;[inr(i) / s]] 9)
. . |    THEN
. . |    ExRepD
. . |    THEN
. . |    (HypSubst -1 0)
. . |    THEN
. . |    (Thin -1)
. . |    THEN
. . |    (InstConcl [[inl(i)] @ s2 @ [inr(i)]])
. . |
. . |\
. . | 1. the_graph : Graph
. . | 2. the_obj : GraphObject(the_graph)
. . | 3. s : traversal(the_graph)
. . | 4. i : Vertices(the_graph)
. . | 5. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
. . | 6. ∀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)
. . | 7. ∀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)
. . | 8. ¬((inl(i) ∈ s) ∨ (inr(i) ∈ s))
. . | 9. ∀L:Vertices(the_graph) List, s:traversal(the_graph).
. . |    ∃s2:traversal(the_graph).
. . |    list_accum(s',j.dfs(the_obj;s';j);s;L) = (s2 @ s)
. . | 10. L : Vertices(the_graph) List
. . | 11. ∀y:Vertices(the_graph). i-the_graph->y ⟺ (y ∈ L)
. . | 12. s2 : traversal(the_graph)
. . | 13. (inr(i) ∈ s) ∨ (inl(i) ∈ s)
. . | ⊢  ([inl(i)] @ s2 @ [inr(i)]) = nil
. . |   by (Analyze 8) THEN (ObviousFrom [-1])
. . |
. . |\
. . | 1. the_graph : Graph
. . | 2. the_obj : GraphObject(the_graph)
. . | 3. s : traversal(the_graph)
. . | 4. i : Vertices(the_graph)
. . | 5. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
. . | 6. ∀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)
. . | 7. ∀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)
. . | 8. ¬((inl(i) ∈ s) ∨ (inr(i) ∈ s))
. . | 9. ∀L:Vertices(the_graph) List, s:traversal(the_graph).
. . |    ∃s2:traversal(the_graph).
. . |    list_accum(s',j.dfs(the_obj;s';j);s;L) = (s2 @ s)
. . | 10. L : Vertices(the_graph) List
. . | 11. ∀y:Vertices(the_graph). i-the_graph->y ⟺ (y ∈ L)
. . | 12. s2 : traversal(the_graph)
. . | 13. ¬(inr(i) ∈ s)
. . | 14. ¬(inl(i) ∈ s)
. . | ⊢  ∃s2@0:traversal(the_graph).
. . |    ([inl(i)] @ s2 @ [inr(i)]) = ([inl(i)] @ s2@0 @ [inr(i)])
. . |   by InstConcl [s2]
. . |
. . .\
. . . 1. the_graph : Graph
. . . 2. the_obj : GraphObject(the_graph)
. . . 3. s : traversal(the_graph)
. . . 4. i : Vertices(the_graph)
. . . 5. ∀x,y:Vertices(the_graph). the_obj.eq(x,y) ⟺ x = y
. . . 6. ∀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)
. . . 7. ∀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)
. . . 8. ¬((inl(i) ∈ s) ∨ (inr(i) ∈ s))
. . . 9. ∀L:Vertices(the_graph) List, s:traversal(the_graph).
. . .    ∃s2:traversal(the_graph).
. . .    list_accum(s',j.dfs(the_obj;s';j);s;L) = (s2 @ s)
. . . 10. L : Vertices(the_graph) List
. . . 11. ∀y:Vertices(the_graph). i-the_graph->y ⟺ (y ∈ L)
. . . 12. s2 : traversal(the_graph)
. . . ⊢  [inl(i) / (s2 @ [inr(i) / s])]
. . .    =
. . .    (([inl(i)] @ s2 @ [inr(i)]) @ s)
. . .   by (RWW
. . .       Thm* ∀a,b,c:Top List. ((a @ b) @ c) ∼ (a @ b @ c)
. . .       {append_assoc_sq}
. . .       0)
. . .      THEN
. . .      (Reduce 0)
END
```