```
dfsl-induction1  THM

⊢  For any graph
|  ∀the_obj:GraphObject(the_graph), P:((V List)→Traversal→Prop).
|  P(nil,nil)
|  ⇒
|  (∀L:V List, s:Traversal, i:V.
|   P(L,s) ⇒ P(L @ [i],dfs(the_obj;s;i)))
|  ⇒
|  (∀L:V List. P(L,dfsl(the_obj;L)))
| by (Auto THEN (Unfold ‘dfsl‘ 0) THEN (MoveToConcl -1)
|     THEN
|     InductionOnLength
|     THEN
|     (AutoBoolCase null(L)))
|    THENL
|    [(HypSubst -1 0) THEN (Reduce 0)
|    ;((Inst
|       Thm* ∀L:T List. 0<||L|| ⇒ (∃x:T, L’:T List. L = (L’ @ [x]))
|       {list_decomp_reverse}
|       [Vertices(the_graph);L])
|      THENA
|      (Auto
|       THEN
|       (BackThru
|        Thm* ∀L:T List. ¬L = nil ⇒ 0<||L||    {non_nil_length})))
|     THEN
|     ExRepD
|     THEN
|     (HypSubst -1 0)]
|
1. the_graph : Graph
2. the_obj : GraphObject(the_graph)
3. P : (Vertices(the_graph) List)→traversal(the_graph)→Prop
4. P(nil,nil)
5. ∀L:Vertices(the_graph) List, s:traversal(the_graph)
|  , i:Vertices(the_graph). P(L,s) ⇒ P(L @ [i],dfs(the_obj;s;i))
6. ℕ
7. L : Vertices(the_graph) List
8. ∀L1:Vertices(the_graph) List.
|  ||L1||<||L|| ⇒ P(L1,list_accum(s,i.dfs(the_obj;s;i);nil;L1))
9. ¬L = nil
10. x : Vertices(the_graph)
11. L’ : Vertices(the_graph) List
12. L = (L’ @ [x])
⊢  P(L’ @ [x],list_accum(s,i.dfs(the_obj;s;i);nil;L’ @ [x]))
| by (InstHyp [L’] -5)
|    THENA
|    ((Auto THEN (HypSubst -1 0))
|     THEN
|     (RWO
|      Thm* ∀as,bs:T List. ||as @ bs|| = ||as||+||bs|| ∈ ℤ
|      {length_append}
|      0))
|
```

```
1. the_graph : Graph
2. the_obj : GraphObject(the_graph)
3. P : (Vertices(the_graph) List)→traversal(the_graph)→Prop
4. P(nil,nil)
5. ∀L:Vertices(the_graph) List, s:traversal(the_graph)
   , i:Vertices(the_graph). P(L,s) ⇒ P(L @ [i],dfs(the_obj;s;i))
6. ℕ
7. L : Vertices(the_graph) List
8. ∀L1:Vertices(the_graph) List.
   ||L1||<||L|| ⇒ P(L1,list_accum(s,i.dfs(the_obj;s;i);nil;L1))
9. ¬L = nil
10. x : Vertices(the_graph)
11. L' : Vertices(the_graph) List
12. L = (L' @ [x])
13. P(L',list_accum(s,i.dfs(the_obj;s;i);nil;L'))
⊢  P(L' @ [x],list_accum(s,i.dfs(the_obj;s;i);nil;L' @ [x]))
  by (RWO
      Thm* ∀l1,l2:Top List, f,y:Top.
           list_accum(x,a.f(x,a);y;l1
           @ l2) ~ list_accum(x,a.f(x,a);list_accum(x,a.f(x
                                                          ,a);y;l1);l2)
      {list_accum_append}
      0)
     THEN
     (Reduce 0)
     THEN
     BackThruSomeHyp
END
```