```
dfsl-traversal-nil  THM

⊢  For any graph
|  dfsl-traversal(the_graph;nil;nil)  by Unfold `dfsl-traversal` 0
|
|\
| 1. the_graph : Graph
| ⊢  df-traversal(the_graph;nil)
| | by (Unfold `df-traversal` 0) THEN (Unfold `traversal` 0)
| |
| |\
| | 1. the_graph : Graph
| | 2. i : Vertices(the_graph)
| | 3. s1 : (Vertices(the_graph)+Vertices(the_graph)) List
| | 4. s2 : (Vertices(the_graph)+Vertices(the_graph)) List
| | 5. nil = (s1 @ [inr(i)] @ s2)
| | 6. j : Vertices(the_graph)
| | 7. (inr(j) ∈ s2)
| | 8. ¬(inl(j) ∈ s2)
| | ⊢  j-the_graph->*i  by ObviousContradiction [5]
| |
| .\
| . 1. the_graph : Graph
| . 2. i : Vertices(the_graph)
| . 3. s1 : (Vertices(the_graph)+Vertices(the_graph)) List
| . 4. s2 : (Vertices(the_graph)+Vertices(the_graph)) List
| . 5. ∀j:Vertices(the_graph).
| .    i-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j)
| . 6. nil = (s1 @ [inl(i)] @ s2)
| . 7. j : Vertices(the_graph)
| . 8. i-the_graph->*j
| . ⊢  (inr(j) ∈ s2)  by ObviousContradiction [6]
|
|\
| 1. the_graph : Graph
| 2. i : Vertices(the_graph)
| 3. (inl(i) ∈ nil)
| ⊢  nil-the_graph->*i  by ObviousContradiction [-1]
|
.\
. 1. the_graph : Graph
. 2. ∀i:Vertices(the_graph).
. |  nil-the_graph->*i ⇒ ¬non-trivial-loop(the_graph;i)
. 3. L1 : Vertices(the_graph) List
. 4. L2 : Vertices(the_graph) List
. 5. nil = (L1 @ L2)
. ⊢  ∃s1,s2:traversal(the_graph).
. |  nil = (s2 @ s1)
. |  & paren(Vertices(the_graph);s1) & paren(Vertices(the_graph);s2)
. |  & (∀j:Vertices(the_graph).
. |     ((inl(j) ∈ s1) ⟺ L1-the_graph->*j)
. |     & ((inl(j) ∈ s2) ⟺ L2-the_graph->*j & ¬L1-the_graph->*j))
. | by (Thin 2)
. |    THEN
. |    (Inst
```

```
. |     Thm* ∀l1,l2:T List. (l1 @ l2) = nil ⟺ l1 = nil & l2 = nil
. |     {append_is_nil}
. |     [Vertices(the_graph);L1;L2])
. |    THEN
. |    (Analyze -1)
. |    THEN
. |    (Thin -1)
. |    THEN
. |    (Analyze -1)
. |    THEN
. |    (Analyze -1)
. |    THEN
. |    (RepeatFor 2 ((HypSubst -1 0) THEN (Thin -1)))
. |    THEN
. |    (AllHyps Thin)
. |    THEN
. |    (InstConcl [nil;nil])
. |    THEN
. |    (Reduce 0)
. |
. |\
. | 1. the_graph : Graph
. | ⊢  paren(Vertices(the_graph);nil)  by Obvious
. |
. |\
. | 1. the_graph : Graph
. | ⊢  paren(Vertices(the_graph);nil)  by Obvious
. |
. |\
. | 1. the_graph : Graph
. | 2. j : Vertices(the_graph)
. | 3. (inl(j) ∈ nil)
. | ⊢  nil-the_graph->*j  by Obvious
. |
. |\
. | 1. the_graph : Graph
. | 2. j : Vertices(the_graph)
. | 3. nil-the_graph->*j
. | ⊢  (inl(j) ∈ nil)  by (Analyze -1) THEN (Analyze -1)
. | |
. | 1. the_graph : Graph
. | 2. j : Vertices(the_graph)
. | 3. y : Vertices(the_graph)
. | 4. (y ∈ nil)
. | 5. y-the_graph->*j
. | ⊢  (inl(j) ∈ nil)  by ObviousContradiction [-2]
. |

. |\
. | 1. the_graph : Graph
. | 2. j : Vertices(the_graph)
. | 3. (inl(j) ∈ nil)
. | ⊢  nil-the_graph->*j  by Obvious
. |
. .\
. . 1. the_graph : Graph
```

```
. . 2. j : Vertices(the_graph)
. . 3. (inl(j) ∈ nil)
. . ⊢  ¬nil-the_graph->*j  by Obvious
END
```