```
df-traversal-cons2  THM

⊢  For any graph
|  ∀i:V, s:Traversal.
|  (∀j:V. (inr(j) ∈ s) ⇒ ¬(inl(j) ∈ s) ⇒ j-the_graph->*i)
|  ⇒
|  df-traversal(the_graph;s) ⇒ df-traversal(the_graph;[inr(i) / s])
| by Unfold ‘df-traversal‘ 0
|
|\
| 1. the_graph : Graph
| 2. i : Vertices(the_graph)
| 3. s : traversal(the_graph)
| 4. ∀j:Vertices(the_graph).
| |  (inr(j) ∈ s) ⇒ ¬(inl(j) ∈ s) ⇒ j-the_graph->*i
| 5. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| |  s = (s1 @ [inr(i)] @ s2)
| |  ⇒
| |  (∀j:Vertices(the_graph).
| |   (inr(j) ∈ s2) ⇒ ¬(inl(j) ∈ s2) ⇒ j-the_graph->*i)
| 6. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| |  (∀j:Vertices(the_graph).
| |   i-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j))
| |  ⇒
| |  s = (s1 @ [inl(i)] @ s2)
| |  ⇒
| |  (∀j:Vertices(the_graph). i-the_graph->*j ⇒ (inr(j) ∈ s2))
| 7. i@0 : Vertices(the_graph)
| 8. s1 : traversal(the_graph)
| 9. s2 : traversal(the_graph)
| 10. [inr(i) / s] = (s1 @ [inr(i@0)] @ s2)
| 11. j : Vertices(the_graph)
| 12. (inr(j) ∈ s2)
| 13. ¬(inl(j) ∈ s2)
| ⊢  j-the_graph->*i@0  by (Analyze -6) THEN (All Reduce)
| |
| |\
| | 1. the_graph : Graph
| | 2. i : Vertices(the_graph)
| | 3. s : traversal(the_graph)
| | 4. ∀j:Vertices(the_graph).
| | |  (inr(j) ∈ s) ⇒ ¬(inl(j) ∈ s) ⇒ j-the_graph->*i
| | 5. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| | |  s = (s1 @ [inr(i) / s2])
| | |  ⇒
| | |  (∀j:Vertices(the_graph).
| | |   (inr(j) ∈ s2) ⇒ ¬(inl(j) ∈ s2) ⇒ j-the_graph->*i)
| | 6. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| | |  (∀j:Vertices(the_graph).
| | |   i-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j))
| | |  ⇒
| | |  s = (s1 @ [inl(i) / s2])
| | |  ⇒
| | |  (∀j:Vertices(the_graph). i-the_graph->*j ⇒ (inr(j) ∈ s2))
| | 7. i@0 : Vertices(the_graph)
```

```
| | 8. s2 : traversal(the_graph)
| | 9. [inr(i) / s] = [inr(i@0) / s2]
| | 10. j : Vertices(the_graph)
| | 11. (inr(j) ∈ s2)
| | 12. ¬(inl(j) ∈ s2)
| | ⊢  j-the_graph->*i@0  by (SplitCons -4) THEN (Subst (i@0 = i) 0)
| | |
| | |\
| | | .....equality .....
| | | 1. the_graph : Graph
| | | 2. i : Vertices(the_graph)
| | | 3. s : traversal(the_graph)
| | | 4. ∀j:Vertices(the_graph).
| | |    (inr(j) ∈ s) ⇒ ¬(inl(j) ∈ s) ⇒ j-the_graph->*i
| | | 5. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| | |    s = (s1 @ [inr(i) / s2])
| | |    ⇒
| | |    (∀j:Vertices(the_graph).
| | |     (inr(j) ∈ s2) ⇒ ¬(inl(j) ∈ s2) ⇒ j-the_graph->*i)
| | | 6. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| | |    (∀j:Vertices(the_graph).
| | |     i-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j))
| | |    ⇒
| | |    s = (s1 @ [inl(i) / s2])
| | |    ⇒
| | |    (∀j:Vertices(the_graph). i-the_graph->*j ⇒ (inr(j) ∈ s2))
| | | 7. i@0 : Vertices(the_graph)
| | | 8. s2 : traversal(the_graph)
| | | 9. j : Vertices(the_graph)
| | | 10. (inr(j) ∈ s2)
| | | 11. ¬(inl(j) ∈ s2)
| | | 12. inr(i) = inr(i@0)
| | | 13. s = s2
| | | ⊢  i@0 = i  by ObviousFrom [-2]
| | |
| | .\
| | . 1. the_graph : Graph
| | . 2. i : Vertices(the_graph)
| | . 3. s : traversal(the_graph)
| | . 4. ∀j:Vertices(the_graph).
| | .    (inr(j) ∈ s) ⇒ ¬(inl(j) ∈ s) ⇒ j-the_graph->*i
| | . 5. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| | .    s = (s1 @ [inr(i) / s2])
| | .    ⇒
| | .    (∀j:Vertices(the_graph).
| | .     (inr(j) ∈ s2) ⇒ ¬(inl(j) ∈ s2) ⇒ j-the_graph->*i)
| | . 6. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| | .    (∀j:Vertices(the_graph).
| | .     i-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j))
| | .    ⇒
| | .    s = (s1 @ [inl(i) / s2])
| | .    ⇒
| | .    (∀j:Vertices(the_graph). i-the_graph->*j ⇒ (inr(j) ∈ s2))
| | . 7. i@0 : Vertices(the_graph)
| | . 8. s2 : traversal(the_graph)
```

```
| | . 9. j : Vertices(the_graph)
| | . 10. (inr(j) ∈ s2)
| | . 11. ¬(inl(j) ∈ s2)
| | . 12. inr(i) = inr(i@0)
| | . 13. s = s2
| | . ⊢  j-the_graph->*i  by BackThruSomeHyp THEN (HypSubst -1 0)
| |
| .\
| . 1. the_graph : Graph
| . 2. i : Vertices(the_graph)
| . 3. s : traversal(the_graph)
| . 4. ∀j:Vertices(the_graph).
| . |  (inr(j) ∈ s) ⇒ ¬(inl(j) ∈ s) ⇒ j-the_graph->*i
| . 5. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| . |  s = (s1 @ [inr(i) / s2])
| . |  ⇒
| . |  (∀j:Vertices(the_graph).
| . |   (inr(j) ∈ s2) ⇒ ¬(inl(j) ∈ s2) ⇒ j-the_graph->*i)
| . 6. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| . |  (∀j:Vertices(the_graph).
| . |   i-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j))
| . |  ⇒
| . |  s = (s1 @ [inl(i) / s2])
| . |  ⇒
| . |  (∀j:Vertices(the_graph). i-the_graph->*j ⇒ (inr(j) ∈ s2))
| . 7. i@0 : Vertices(the_graph)
| . 8. u : Vertices(the_graph)+Vertices(the_graph)
| . 9. v : (Vertices(the_graph)+Vertices(the_graph)) List
| . 10. s2 : traversal(the_graph)
| . 11. [inr(i) / s] = [u / (v @ [inr(i@0) / s2])]
| . 12. j : Vertices(the_graph)
| . 13. (inr(j) ∈ s2)
| . 14. ¬(inl(j) ∈ s2)
| . ⊢  j-the_graph->*i@0  by InstHyp [i@0;v;s2;j] 5
| . |
| . .....antecedent .....
| . 1. the_graph : Graph
| . 2. i : Vertices(the_graph)
| . 3. s : traversal(the_graph)
| . 4. ∀j:Vertices(the_graph).
| .    (inr(j) ∈ s) ⇒ ¬(inl(j) ∈ s) ⇒ j-the_graph->*i
| . 5. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| .    s = (s1 @ [inr(i) / s2])
| .    ⇒
| .    (∀j:Vertices(the_graph).
| .     (inr(j) ∈ s2) ⇒ ¬(inl(j) ∈ s2) ⇒ j-the_graph->*i)
| . 6. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
| .    (∀j:Vertices(the_graph).
| .     i-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j))
| .    ⇒
| .    s = (s1 @ [inl(i) / s2])
| .    ⇒
| .    (∀j:Vertices(the_graph). i-the_graph->*j ⇒ (inr(j) ∈ s2))
| . 7. i@0 : Vertices(the_graph)
| . 8. u : Vertices(the_graph)+Vertices(the_graph)
```

```
| . 9. v : (Vertices(the_graph)+Vertices(the_graph)) List
| . 10. s2 : traversal(the_graph)
| . 11. [inr(i) / s] = [u / (v @ [inr(i@0) / s2])]
| . 12. j : Vertices(the_graph)
| . 13. (inr(j) ∈ s2)
| . 14. ¬(inl(j) ∈ s2)
| . ⊢  s = (v @ [inr(i@0) / s2])  by SplitCons -4
|
.\
. 1. the_graph : Graph
. 2. i : Vertices(the_graph)
. 3. s : traversal(the_graph)
. 4. ∀j:Vertices(the_graph).
. |  (inr(j) ∈ s) ⇒ ¬(inl(j) ∈ s) ⇒ j-the_graph->*i
. 5. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
. |  s = (s1 @ [inr(i)] @ s2)
. |  ⇒
. |  (∀j:Vertices(the_graph).
. |   (inr(j) ∈ s2) ⇒ ¬(inl(j) ∈ s2) ⇒ j-the_graph->*i)
. 6. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
. |  (∀j:Vertices(the_graph).
. |   i-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j))
. |  ⇒
. |  s = (s1 @ [inl(i)] @ s2)
. |  ⇒
. |  (∀j:Vertices(the_graph). i-the_graph->*j ⇒ (inr(j) ∈ s2))
. 7. i@0 : Vertices(the_graph)
. 8. s1 : traversal(the_graph)
. 9. s2 : traversal(the_graph)
. 10. ∀j:Vertices(the_graph).
. |   i@0-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j)
. 11. [inr(i) / s] = (s1 @ [inl(i@0)] @ s2)
. 12. j : Vertices(the_graph)
. 13. i@0-the_graph->*j
. ⊢  (inr(j) ∈ s2)
. | by (Analyze -6) THEN (All Reduce) THEN (SplitCons -3)
. |

. |\
. | 1. the_graph : Graph
. | 2. i : Vertices(the_graph)
. | 3. s : traversal(the_graph)
. | 4. ∀j:Vertices(the_graph).
. |    (inr(j) ∈ s) ⇒ ¬(inl(j) ∈ s) ⇒ j-the_graph->*i
. | 5. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
. |    s = (s1 @ [inr(i) / s2])
. |    ⇒
. |    (∀j:Vertices(the_graph).
. |     (inr(j) ∈ s2) ⇒ ¬(inl(j) ∈ s2) ⇒ j-the_graph->*i)
. | 6. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
. |    (∀j:Vertices(the_graph).
. |     i-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j))
. |    ⇒
. |    s = (s1 @ [inl(i) / s2])
. |    ⇒
. |    (∀j:Vertices(the_graph). i-the_graph->*j ⇒ (inr(j) ∈ s2))
```

```
. | 7. i@0 : Vertices(the_graph)
. | 8. s2 : traversal(the_graph)
. | 9. ∀j:Vertices(the_graph).
. |    i@0-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j)
. | 10. j : Vertices(the_graph)
. | 11. i@0-the_graph->*j
. | 12. inr(i) = inl(i@0)
. | 13. s = s2
. | ⊢  (inr(j) ∈ s2)  by ObviousContradiction [-2]
. |
. .\
. . 1. the_graph : Graph
. . 2. i : Vertices(the_graph)
. . 3. s : traversal(the_graph)
. . 4. ∀j:Vertices(the_graph).
. .    (inr(j) ∈ s) ⇒ ¬(inl(j) ∈ s) ⇒ j-the_graph->*i
. . 5. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
. .    s = (s1 @ [inr(i) / s2])
. .    ⇒
. .    (∀j:Vertices(the_graph).
. .     (inr(j) ∈ s2) ⇒ ¬(inl(j) ∈ s2) ⇒ j-the_graph->*i)
. . 6. ∀i:Vertices(the_graph), s1,s2:traversal(the_graph).
. .    (∀j:Vertices(the_graph).
. .     i-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j))
. .    ⇒
. .    s = (s1 @ [inl(i) / s2])
. .    ⇒
. .    (∀j:Vertices(the_graph). i-the_graph->*j ⇒ (inr(j) ∈ s2))
. . 7. i@0 : Vertices(the_graph)
. . 8. u : Vertices(the_graph)+Vertices(the_graph)
. . 9. v : (Vertices(the_graph)+Vertices(the_graph)) List
. . 10. s2 : traversal(the_graph)
. . 11. ∀j:Vertices(the_graph).
. .     i@0-the_graph->*j ⇒ ¬non-trivial-loop(the_graph;j)
. . 12. j : Vertices(the_graph)
. . 13. i@0-the_graph->*j
. . 14. inr(i) = u
. . 15. s = (v @ [inl(i@0) / s2])
. . ⊢  (inr(j) ∈ s2)  by AutoInst [i@0;v;s2;j]
END
```