```
list-connect-append  THM

⊢  For any graph
|  ∀A,B:V List, i:V. A @ B-->*i ⟺ A-->*i ∨ B-->*i
| by (Unfold ‘list-connect‘ 0) THEN (Unfold ‘l_exists‘ 0)
|    THEN
|    (RWO
|     Thm* ∀x:T, l1,l2:T List. (x ∈ l1 @ l2) ⟺ (x ∈ l1) ∨ (x ∈ l2)
|     {member_append}
|     0)
|
|\
| 1. the_graph : Graph
| 2. A : Vertices(the_graph) List
| 3. B : Vertices(the_graph) List
| 4. i : Vertices(the_graph)
| 5. ∃y:Vertices(the_graph). (y ∈ A) ∨ (y ∈ B) & y-the_graph->*i
| ⊢  (∃y:Vertices(the_graph). (y ∈ A) & y-the_graph->*i)
|    ∨ (∃y:Vertices(the_graph). (y ∈ B) & y-the_graph->*i)
|   by ExRepD THEN (ParallelOp -2) THEN (InstConcl [y])
|
.\
. 1. the_graph : Graph
. 2. A : Vertices(the_graph) List
. 3. B : Vertices(the_graph) List
. 4. i : Vertices(the_graph)
. 5. (∃y:Vertices(the_graph). (y ∈ A) & y-the_graph->*i)
.    ∨ (∃y:Vertices(the_graph). (y ∈ B) & y-the_graph->*i)
. ⊢  ∃y:Vertices(the_graph). (y ∈ A) ∨ (y ∈ B) & y-the_graph->*i
.   by (Analyze -1) THEN (ParallelOp -1) THEN Obvious
END
```