```
adjl-edge-accum-properties  THM

⊢  ∀A:AdjList, T:Type, s:T, x:Vertices(adjl-graph(A))
|   , f:(T→Vertices(adjl-graph(A))→T).
|   ∃L:Vertices(adjl-graph(A)) List.
|   (∀y:Vertices(adjl-graph(A)). x-adjl-graph(A)->y ⟺ (y ∈ L))
|   & adjl-edge-accum(A;s',x'.f(s',x');s;x)
|     =
|     list_accum(s',x'.f(s',x');s;L)
| by (Unfolds [`adjl-graph`;`adjl-edge-accum`] 0) THEN (Reduce 0)
|     THEN
|     (InstConcl [A.out(x)])
|     THEN
|     (All (Unfold `edge`))
|     THEN
|     (All Reduce)
|     THEN
|     ExRepD
|     THEN
|     (Try ((Analyze -1) THEN (Reduce 0)))
|
|\
| 1. A : AdjList
| 2. T : Type
| 3. T
| 4. x : ℕA.size
| 5. T→ℕA.size→T
| 6. y : ℕA.size
| 7. e : x:ℕA.size×ℕ||A.out(x)||
| 8. <1of(e),(A.out(1of(e)))[2of(e)]> = <x,y>
| ⊢  (y ∈ A.out(x))
|   by (SplitPair -1) THEN (Unfold `l_member` 0)
|      THEN
|      (InstConcl [2of(e)])
|      THEN
|      (SubstFor x 0)
|
.\
. 1. A : AdjList
. 2. T : Type
. 3. T
. 4. x : ℕA.size
. 5. T→ℕA.size→T
. 6. y : ℕA.size
. 7. i : ℕ
. 8. i<||A.out(x)|| & y = (A.out(x))[i]
. ⊢  ∃e:(x:ℕA.size×ℕ||A.out(x)||).
. |  <1of(e),(A.out(1of(e)))[2of(e)]> = <x,y>
. | by (Analyze -1) THEN (InstConcl [<x,i>]) THEN (Reduce 0)
. |
. 1. A : AdjList
. 2. T : Type
. 3. T
. 4. x : ℕA.size
. 5. T→ℕA.size→T
```

```
. 6. y : ℕA.size
. 7. i : ℕ
. 8. i<||A.out(x)||
. 9. y = (A.out(x))[i]
. ⊢  <x,(A.out(x))[i]> = <x,y>  by Analyze
END
```