```
apply_alist_member2  THM

⊢  ∀T:Type, as:(Label×T) List, d1,d2:T, x:Label.
|  (x ∈ 1of(unzip(as))) ⇒ apply_alist(as;x;d1) = apply_alist(as;x;d2)
| by D 0 THENM D 0 THENM ListInd (-1) THENA Auto
|
|\
| 1. T : Type
| 2. (Label×T) List
| ⊢  ∀d1,d2:T, x:Label.
|    (x ∈ 1of(unzip(nil))) ⇒
|    apply_alist(nil;x;d1) = apply_alist(nil;x;d2)
|   by Unfold ‘unzip‘ 0 THEN Reduce 0  THEN Unfold ‘l_member‘ 0
|
|        THEN Reduce 0  THEN Auto THEN ExRepD THEN Auto
|
.\
. 1. T : Type
. 2. (Label×T) List
. 3. u : Label×T
. 4. v : (Label×T) List
. 5. ∀d1,d2:T, x:Label.
. |  (x ∈ 1of(unzip(v))) ⇒ apply_alist(v;x;d1) = apply_alist(v;x;d2)
. ⊢  ∀d1,d2:T, x:Label.
. |  (x ∈ 1of(unzip([u / v]))) ⇒
. |  apply_alist([u / v];x;d1) = apply_alist([u / v];x;d2)
. | by Auto THEN InstLemma ‘apply_alist_cons‘ [T;v;u;x;d1]
. ...
. ...
. |
. |
. .....falsecase .....
. 1. T : Type
. 2. (Label×T) List
. 3. u : Label×T
. 4. v : (Label×T) List
. 5. ∀d1,d2:T, x:Label.
. |  (x ∈ 1of(unzip(v))) ⇒ apply_alist(v;x;d1) = apply_alist(v;x;d2)
. 6. d1 : T
. 7. d2 : T
. 8. x : Label
. 9. (x ∈ 1of(unzip([u / v])))
. 10. ¬1of(u) =1 x
. ⊢  apply_alist(v;x;d1) = apply_alist(v;x;d2)
. | by ...
.
. |    ...
. |
. |
. 1. T : Type
. 2. (Label×T) List
. 3. u : Label×T
. 4. v : (Label×T) List
. 5. ∀d1,d2:T, x:Label.
.    (x ∈ map(λp.1of(p);v)) ⇒
```

```
.    apply_alist(v;x;d1) = apply_alist(v;x;d2)
. 6. T
. 7. T
. 8. x : Label
. 9. x = 1of(u)
. 10. ¬1of(u) =₁ x
. ⊢  (x ∈ map(λp.1of(p);v))
.   by D (-1) THEN HypSubstSq (-1) 0 THEN EqLblReflexive 0 THEN Auto
END
```