apply_alist_non_member THM

```
⊢  ∀T:Type, as:(Label×T) List, d:T, x:Label.
|  ¬(x ∈ 1of(unzip(as))) ⇒ apply_alist(as;x;d) = d
| by D 0 THENM D 0 THENM ListInd (-1) THENA Auto
...
|
|\
| ...
|
.\
. 1. T : Type
. 2. (Label×T) List
. 3. u : Label×T
. 4. v : (Label×T) List
. 5. ∀d:T, x:Label. ¬(x ∈ map(λp.1of(p);v)) ⇒ apply_alist(v;x;d) = d
. ⊢  ∀d:T, x:Label.
. |  ¬(x ∈ [1of(u) / map(λp.1of(p);v)]) ⇒
. |  apply_alist([u / v];x;d) = d  by RWW "cons_member" 0
. THEN Auto
. |
. 1. T : Type
. 2. (Label×T) List
. 3. u : Label×T
. 4. v : (Label×T) List
. 5. ∀d:T, x:Label. ¬(x ∈ map(λp.1of(p);v)) ⇒ apply_alist(v;x;d) = d
. 6. d : T
. 7. x : Label
. 8. ¬(x = 1of(u) ∨ (x ∈ map(λp.1of(p);v)))
. ⊢  apply_alist([u / v];x;d) = d
. | by InstLemma ‘apply_alist_cons‘ [T;v;u;x;d]
. ...
. THENA Auto
. |
. 1. T : Type
. 2. (Label×T) List
. 3. u : Label×T
. 4. v : (Label×T) List
. 5. ∀d:T, x:Label. ¬(x ∈ map(λp.1of(p);v)) ⇒ apply_alist(v;x;d) = d
. 6. d : T
. 7. x : Label
. 8. ¬(x = 1of(u) ∨ (x ∈ map(λp.1of(p);v)))
. 9. apply_alist([u / v];x;d) ~ if 1of(u) =1 x→ 2of(u)
. |                               else apply_alist(v;x;d) fi
. ⊢  if 1of(u) =1 x→ 2of(u) else apply_alist(v;x;d) fi = d
. | by SplitITE 0 THENA Auto THEN Try TrivialOr
. |
. |\
. | .....truecase .....
. | 1. T : Type
. | 2. (Label×T) List
. | 3. u : Label×T
. | 4. v : (Label×T) List
. | 5. ∀d:T, x:Label.
. | |  ¬(x ∈ map(λp.1of(p);v)) ⇒ apply_alist(v;x;d) = d
```

```
. | 6. d : T
. | 7. x : Label
. | 8. ¬(x = 1of(u) ∨ (x ∈ map(λp.1of(p);v)))
. | 9. apply_alist([u / v];x;d) ∼ if 1of(u) =_1 x→ 2of(u)
. | |                             else apply_alist(v;x;d) fi
. | 10. 1of(u) = x
. | ⊢  2of(u) = d
. | | by D (-3) THEN Sel 1 (D 0) THEN  Auto
. | ...
. | |
. | 1. T : Type
. | 2. (Label×T) List
. | 3. u1 : Pattern
. | 4. ground_ptn(u1)
. | 5. u2 : T
. | 6. v : (Label×T) List
. | 7. ∀d:T, x:Label.
. |    ¬(x ∈ map(λp.1of(p);v)) ⇒ apply_alist(v;x;d) = d
. | 8. d : T
. | 9. x : Pattern
. | 10. ground_ptn(x)
. | 11. apply_alist([<u1,u2> / v];x;d) ∼ if u1 =_1 x→ u2
. |                                    else apply_alist(v;x;d) fi
. | 12. u1 = x
. | ⊢  x = u1  by MemTypeCD THEN Auto
. |
. .\
. . .....falsecase .....
. . 1. T : Type
. . 2. (Label×T) List
. . 3. u : Label×T
. . 4. v : (Label×T) List
. . 5. ∀d:T, x:Label.
. . |  ¬(x ∈ map(λp.1of(p);v)) ⇒ apply_alist(v;x;d) = d
. . 6. d : T
. . 7. x : Label
. . 8. ¬(x = 1of(u) ∨ (x ∈ map(λp.1of(p);v)))
. . 9. apply_alist([u / v];x;d) ∼ if 1of(u) =_1 x→ 2of(u)
. . |                             else apply_alist(v;x;d) fi
. . 10. ¬1of(u) = x
. . ⊢  apply_alist(v;x;d) = d
. . | by BackThruSomeHyp THEN D 0  THENA Auto
. . |
. . 1. T : Type
. . 2. (Label×T) List
. . 3. u : Label×T
. . 4. v : (Label×T) List
. . 5. ∀d:T, x:Label.
. .    ¬(x ∈ map(λp.1of(p);v)) ⇒ apply_alist(v;x;d) = d
. . 6. d : T
. . 7. x : Label
. . 8. ¬(x = 1of(u) ∨ (x ∈ map(λp.1of(p);v)))
. . 9. apply_alist([u / v];x;d) ∼ if 1of(u) =_1 x→ 2of(u)
. .                               else apply_alist(v;x;d) fi
. . 10. ¬1of(u) = x
```

```
. . 11. (x ∈ map(λp.1of(p);v))
. . ⊢  False  by D (-4) THEN TrivialOr
END
```