```
decls_mng_rename_member  THM

⊢  ∀ds1,ds2:Collection(dec()), x,y:Label, rho:Decl, v:[[ds1]] rho(x).
|  (∀d:dec(). d ∈ ds2 ⇒ d.lbl = y ⇒ mk_dec(x, d.typ) ∈ ds1) ⇒
|  v ∈ [[ds2]] rho(y)
| by Unfold `decls_mng` 0 THEN Auto

|    THEN Try (Fold `decl` 0 THEN Auto)
|
|
1. ds1 : Collection(dec())
2. ds2 : Collection(dec())
3. x : Label
4. y : Label
5. rho : Decl
6. v : [[d]] rho for d ∈ {d:dec()| d ∈ ds1 }(x)
7. ∀d:dec(). d ∈ ds2 ⇒ d.lbl = y ⇒ mk_dec(x, d.typ) ∈ ds1
⊢  v ∈ [[d]] rho for d ∈ {d:dec()| d ∈ ds2 }(y)
| by All (Unfold `dall`) THEN All Reduce
|
1. ds1 : Collection(dec())
2. ds2 : Collection(dec())
3. x : Label
4. y : Label
5. rho : Decl
6. v : ∩d:{d:dec()| d ∈ ds1 }. [[d]] rho(x)
7. ∀d:dec(). d ∈ ds2 ⇒ d.lbl = y ⇒ mk_dec(x, d.typ) ∈ ds1
⊢  v ∈ (∩d:{d:dec()| d ∈ ds2 }. [[d]] rho(y))
| by MemTypeCD THENA Auto
|
1. ds1 : Collection(dec())
2. ds2 : Collection(dec())
3. x : Label
4. y : Label
5. rho : Decl
6. v : ∩d:{d:dec()| d ∈ ds1 }. [[d]] rho(x)
7. ∀d:dec(). d ∈ ds2 ⇒ d.lbl = y ⇒ mk_dec(x, d.typ) ∈ ds1
8. d : {d:dec()| d ∈ ds2 }
⊢  v ∈ [[d]] rho(y)  by Decide  d.lbl =1 y THENA Auto
|
|\
| 1. ds1 : Collection(dec())
| 2. ds2 : Collection(dec())
| 3. x : Label
| 4. y : Label
| 5. rho : Decl
| 6. v : ∩d:{d:dec()| d ∈ ds1 }. [[d]] rho(x)
| 7. ∀d:dec(). d ∈ ds2 ⇒ d.lbl = y ⇒ mk_dec(x, d.typ) ∈ ds1
| 8. d : {d:dec()| d ∈ ds2 }
| 9. d.lbl =1 y
| ⊢  v ∈ [[d]] rho(y)
| | by AllHyps (IsectHD mk_dec(x, d.typ)) THEN Auto
| |
| |\
| | 1. ds1 : Collection(dec())
```

```
| | 2. ds2 : Collection(dec())
| | 3. x : Label
| | 4. y : Label
| | 5. rho : Decl
| | 6. ∩d:{d:dec()| d ∈ ds1 }. [[d]] rho(x)
| | 7. ∀d:dec(). d ∈ ds2 ⇒ d.lbl = y ⇒ mk_dec(x, d.typ) ∈ ds1
| | 8. d : {d:dec()| d ∈ ds2 }
| | 9. d.lbl =1 y
| | ⊢  mk_dec(x, d.typ) = mk_dec(x, d.typ)
| | | by Fold ‘member‘ 0 THEN D (-2)
| | THEN MemTypeCD THEN Auto
| | |
| | .....set predicate .....
| | 1. ds1 : Collection(dec())
| | 2. ds2 : Collection(dec())
| | 3. x : Label
| | 4. y : Label
| | 5. rho : Decl
| | 6. ∩d:{d:dec()| d ∈ ds1 }. [[d]] rho(x)
| | 7. ∀d:dec(). d ∈ ds2 ⇒ d.lbl = y ⇒ mk_dec(x, d.typ) ∈ ds1
| | 8. d : dec()
| | 9. d ∈ ds2
| | 10. d.lbl =1 y
| | ⊢  mk_dec(x, d.typ) ∈ ds1  by BackThruSomeHyp THEN Auto
| | |
| | 1. ds1 : Collection(dec())
| | 2. ds2 : Collection(dec())
| | 3. x : Label
| | 4. y : Label
| | 5. rho : Decl
| | 6. ∩d:{d:dec()| d ∈ ds1 }. [[d]] rho(x)
| | 7. ∀d:dec(). d ∈ ds2 ⇒ d.lbl = y ⇒ mk_dec(x, d.typ) ∈ ds1
| | 8. d : dec()
| | 9. d ∈ ds2
| | 10. d.lbl =1 y
| | ⊢  d.lbl = y  by EqLblFwd THEN Auto
| |
| .\
| . 1. ds1 : Collection(dec())
| . 2. ds2 : Collection(dec())
| . 3. x : Label
| . 4. y : Label
| . 5. rho : Decl
| . 6. v : ∩d:{d:dec()| d ∈ ds1 }. [[d]] rho(x)
| . 7. ∀d:dec(). d ∈ ds2 ⇒ d.lbl = y ⇒ mk_dec(x, d.typ) ∈ ds1
| . 8. d : {d:dec()| d ∈ ds2 }
| . 9. d.lbl =1 y
| . 10. v ∈ [[mk_dec(x, d.typ)]] rho(x)
| . ⊢  v ∈ [[d]] rho(y)
| . | by Unfold ‘dec_mng‘ (-1) THEN Unfold ‘dbase‘ (-1)
| .
| . |    THEN Reduce (-1) THEN EqLblReflexive (-1)
| . |
| . 1. ds1 : Collection(dec())
| . 2. ds2 : Collection(dec())
```

```
| . 3. x : Label
| . 4. y : Label
| . 5. rho : Decl
| . 6. v : ∩d:{d:dec()| d ∈ ds1 }. [[d]] rho(x)
| . 7. ∀d:dec(). d ∈ ds2 ⇒ d.lbl = y ⇒ mk_dec(x, d.typ) ∈ ds1
| . 8. d : {d:dec()| d ∈ ds2 }
| . 9. d.lbl =₁ y
| . 10. v ∈ [[d.typ]] rho
| . ⊢  v ∈ [[d]] rho(y)
| .   by D (-3) THEN D (-4) THEN Unfold ‘dec_mng‘ 0
| .
| .      THEN Unfold ‘dbase‘ 0 THEN All Reduce
| .
| .      THEN SplitOnConclITE THEN Auto
|
.\
. 1. ds1 : Collection(dec())
. 2. ds2 : Collection(dec())
. 3. x : Label
. 4. y : Label
. 5. rho : Decl
. 6. v : ∩d:{d:dec()| d ∈ ds1 }. [[d]] rho(x)
. 7. ∀d:dec(). d ∈ ds2 ⇒ d.lbl = y ⇒ mk_dec(x, d.typ) ∈ ds1
. 8. d : {d:dec()| d ∈ ds2 }
. 9. ¬d.lbl =₁ y
. ⊢  v ∈ [[d]] rho(y)
. | by D (-2) THEN D (-3) THEN
. Unfold ‘dec_mng‘ 0 THEN
.
. |    Unfold ‘dbase‘ 0 THEN
. All Reduce
. |
. 1. ds1 : Collection(dec())
. 2. ds2 : Collection(dec())
. 3. x : Label
. 4. y : Label
. 5. rho : Decl
. 6. v : ∩d:{d:dec()| d ∈ ds1 }. [[d]] rho(x)
. 7. ∀d:dec(). d ∈ ds2 ⇒ d.lbl = y ⇒ mk_dec(x, d.typ) ∈ ds1
. 8. lbl : Label
. 9. d1 : SimpleType
. 10. <lbl,d1> ∈ ds2
. 11. ¬lbl =₁ y
. ⊢  v ∈ if y =₁ lbl→ [[d1]] rho else Top fi
. | by SplitOnConclITE THEN Auto
. |
. .....truecase .....
. 1. ds1 : Collection(dec())
. 2. ds2 : Collection(dec())
. 3. x : Label
. 4. y : Label
. 5. rho : Decl
. 6. v : ∩d:{d:dec()| d ∈ ds1 }. [[d]] rho(x)
. 7. ∀d:dec(). d ∈ ds2 ⇒ d.lbl = y ⇒ mk_dec(x, d.typ) ∈ ds1
. 8. lbl : Label
```

```
. 9. d1 : SimpleType
. 10. <lbl,d1> ∈ ds2
. 11. ¬lbl =₁ y
. 12. y =₁ lbl
. ⊢  v ∈ [[d1]] rho
.   by D (-2) THEN EqLblFwd THEN HypSubst (-1) 0
.
.      THEN Auto THEN EqLblReflexive 0  THEN Auto
END
```