```
sigma_decls_mng2  THM

⊢  ∀da:Collection(dec()), rho:Decl, k:Label
|  , w:[[dec_lookup(da;k)]] rho. <k,w> ∈ (Σ[[da]] rho)  by Auto
|
1. da : Collection(dec())
2. rho : Decl
3. k : Label
4. w : [[dec_lookup(da;k)]] rho
⊢  <k,w> ∈ (Σ[[da]] rho)
| by Unfold ‘sigma‘ 0 THEN Auto THEN Unfold ‘decl_type‘ 0
|
1. da : Collection(dec())
2. rho : Decl
3. k : Label
4. w : [[dec_lookup(da;k)]] rho
⊢  w ∈ [[da]] rho(k)  by Unfold ‘decls_mng‘ 0
|
1. da : Collection(dec())
2. rho : Decl
3. k : Label
4. w : [[dec_lookup(da;k)]] rho
⊢  w ∈ [[d]] rho for d ∈ {d:dec()| d ∈ da }(k)
| by Unfold ‘dall‘ 0 THEN Reduce 0
|
1. da : Collection(dec())
2. rho : Decl
3. k : Label
4. w : [[dec_lookup(da;k)]] rho
⊢  w ∈ (∩d:{d:dec()| d ∈ da }. [[d]] rho(k))  by MemTypeCD THENA Auto
|
1. da : Collection(dec())
2. rho : Decl
3. k : Label
4. w : [[dec_lookup(da;k)]] rho
5. d : {d:dec()| d ∈ da }
⊢  w ∈ [[d]] rho(k)  by D (-1)
|
1. da : Collection(dec())
2. rho : Decl
3. k : Label
4. w : [[dec_lookup(da;k)]] rho
5. d : dec()
6. d ∈ da
⊢  w ∈ [[d]] rho(k)
| by ...
| THEN Reduce 0 THEN SplitOnConclITE THEN Auto
|
.....truecase .....
1. da : Collection(dec())
2. rho : Decl
3. k : Label
4. w : [[dec_lookup(da;k)]] rho
5. lbl : Label
6. d1 : SimpleType
```

```
7. <lbl,d1> ∈ da
8. k =₁ lbl
⊢  w ∈ [[d1]] rho
| by EqLblFwd THEN Using [‘sts’,dec_lookup(da;k)
|    ] (BackThruLemma ‘sts_mng_subtype‘) THEN Auto
|
1. da : Collection(dec())
2. rho : Decl
3. k : Label
4. [[dec_lookup(da;k)]] rho
5. lbl : Label
6. d1 : SimpleType
7. <lbl,d1> ∈ da
8. k =₁ lbl
9. k = lbl
⊢  d1 ∈ dec_lookup(da;k)
  by RWW "member_dec_lookup" 0 THEN Auto THEN HypSubstSq (-1) 0 THEN

        Unfold ‘mk_dec‘ 0 THEN Auto
END
```