```
term_typing2  THM

⊢  ∀ds,daa:Collection(dec()), da:Collection(SimpleType), de:sig()
|  , rho:Decl, t:Term, s,s':{[[ds]] rho}, e:{1of([[de]] rho)}
|  , a:SimpleType, v:[[da]] rho, tr:trace_env([[daa]] rho).
|  trace_consistent(rho;daa;tr.proj;t) ⇒
|  a ∈ term_types(ds;da;de;t) ⇒ [[t]] e s s' v tr ∈ [[a]] rho
| by RepeatFor 6 (D 0 THENA Auto)
|
1. ds : Collection(dec())
2. daa : Collection(dec())
3. da : Collection(SimpleType)
4. de : sig()
5. rho : Decl
6. t : Term
⊢  ∀s,s':{[[ds]] rho}, e:{1of([[de]] rho)}, a:SimpleType, v:[[da]] rho
|  , tr:trace_env([[daa]] rho).
|  trace_consistent(rho;daa;tr.proj;t) ⇒
|  a ∈ term_types(ds;da;de;t) ⇒ [[t]] e s s' v tr ∈ [[a]] rho
| by ...
THEN Try (Folds ``term_types term_mng2`` 0)

|    THEN Try (Auto THEN Try (Fold `r_select` 0) THEN

|             BackThruLemma `record_select_wf_decls_mng2`

|             THEN Auto)

|    ...
|                  Unfold `r_select` 0 THEN

|    ...

|    ...

|    THEN Try (Auto THEN BackThruLemma `sts_mng_subtype` THEN Auto)
|    •
|
|\
| 1. ds : Collection(dec())
| 2. daa : Collection(dec())
| 3. da : Collection(SimpleType)
| 4. de : sig()
| 5. rho : Decl
| 6. Term
| 7. u : Term→Type
| 8. t:{v:Term| u(v) }→∀s,s':{[[ds]] rho}, e:{1of([[de]] rho)}
| |  , a:SimpleType, v:[[da]] rho, tr:trace_env([[daa]] rho).
| |  trace_consistent(rho;daa;tr.proj;t) ⇒
| |  a ∈ term_types(ds;da;de;t) ⇒ [[t]] e s s' v tr ∈ [[a]] rho
| 9. y1 : Label
| ⊢  ∀s,s':{[[ds]] rho}, e:{1of([[de]] rho)}, a:SimpleType
| |  , v:[[da]] rho, tr:trace_env([[daa]] rho).
| |  trace_consistent(rho;daa;tr.proj;trace(y1)) ⇒
| |  a ∈ <lbl_pr(<Trace, y1>)> ⇒ tr.y1 ∈ [[a]] rho
```

```
| | by  Auto THEN RW ColMemberC (-1) THEN HypSubstSq (-1) 0 ·
| |
| 1. ds : Collection(dec())
| 2. daa : Collection(dec())
| 3. da : Collection(SimpleType)
| 4. de : sig()
| 5. rho : Decl
| 6. Term
| 7. u : Term→Type
| 8. t:{v:Term| u(v) }→∀s,s':{[[ds]] rho}, e:{1of([[de]] rho)}
|    , a:SimpleType, v:[[da]] rho, tr:trace_env([[daa]] rho).
|    trace_consistent(rho;daa;tr.proj;t) ⇒
|    a ∈ term_types(ds;da;de;t) ⇒ [[t]] e s s' v tr ∈ [[a]] rho
| 9. y1 : Label
| 10. {[[ds]] rho}
| 11. {[[ds]] rho}
| 12. {1of([[de]] rho)}
| 13. a : SimpleType
| 14. [[da]] rho
| 15. tr : trace_env([[daa]] rho)
| 16. trace_consistent(rho;daa;tr.proj;trace(y1))
| 17. a = lbl_pr(<Trace, y1>)
| ⊢  tr.y1 ∈ [[lbl_pr(<Trace, y1>)]] rho
|   by BackThruLemma `tproj_w_f2` THEN Auto
|
.\
. 1. ds : Collection(dec())
. 2. daa : Collection(dec())
. 3. da : Collection(SimpleType)
. 4. de : sig()
. 5. rho : Decl
. 6. Term
. 7. u : Term→Type
. 8. t:{v:Term| u(v) }→∀s,s':{[[ds]] rho}, e:{1of([[de]] rho)}
.    , a:SimpleType, v:[[da]] rho, tr:trace_env([[daa]] rho).
.    trace_consistent(rho;daa;tr.proj;t) ⇒
.    a ∈ term_types(ds;da;de;t) ⇒ [[t]] e s s' v tr ∈ [[a]] rho
. 9. y1 : {v:Term| u(v) }
. 10. y2 : {v:Term| u(v) }
. ⊢  ∀s,s':{[[ds]] rho}, e:{1of([[de]] rho)}, a:SimpleType
.    , v:[[da]] rho, tr:trace_env([[daa]] rho).
.    trace_consistent(rho;daa;tr.proj;y1 y2) ⇒
.    a ∈ st_app(term_types(ds;da;de;y1);term_types(ds;da;de;y2)) ⇒
.    [[y1]] e s s' v tr([[y2]] e s s' v tr) ∈ [[a]] rho
.   by UnivCD THENA Auto THEN RWW "member_st_app" (-1) THENA Auto
.
.      THEN ExRepD THEN
. AllHyps (\i. InstHyp [y2;s;s';e;a1;v;tr
.      ] i THENA Auto) THEN
. AllHyps (\i. InstHyp [y1;s;s';e;a1→a;v;
.      tr] i THENA Auto)
. THEN Reduce (-1) THEN Auto
. THEN
.      AllHyps (\h. RepeatFor 2 (ParallelOp h) THEN ParallelOp (-1)
.
```

```
.       THEN Reduce 0 THEN RW assert_pushdownC 0 THENA Auto
.
.       THEN Complete SimpConcl)
END
```