```
term_types_functionality  THM

⊢  ∀ds1,ds2:Collection(dec()), da1,da2:Collection(SimpleType)
|  , de:sig(), t:Term.
|  ds1 = ds2 ⇒
|  da1 = da2 ⇒ term_types(ds1;da1;de;t) = term_types(ds2;da2;de;t)
| by UnivCD THENA Auto THEN TermInd (-3)
THEN Unfold `term_types` 0

|    THEN Reduce 0
THEN Try (RW (SweepDnC (HypC 7 ORELSEC HypC 8)) 0)
|
THEN Try (Fold `term_types` 0)
THENM Try RelRST
THENA Auto
|
1. ds1 : Collection(dec())
2. ds2 : Collection(dec())
3. da1 : Collection(SimpleType)
4. da2 : Collection(SimpleType)
5. de : sig()
6. Term
7. ds1 = ds2
8. da1 = da2
9. u : Term→Type
10. t:{v:Term| u(v) }→
|   term_types(ds1;da1;de;t) = term_types(ds2;da2;de;t)
11. y1 : {v:Term| u(v) }
12. y2 : {v:Term| u(v) }
⊢  st_app(term_types(ds1;da1;de;y1);term_types(ds1;da1;de;y2)) = ...
| by InstHyp [y1] (-3)
THENM InstHyp [y2] (-4)
THENA Auto
|
1. ds1 : Collection(dec())
2. ds2 : Collection(dec())
3. da1 : Collection(SimpleType)
4. da2 : Collection(SimpleType)
5. de : sig()
6. Term
7. ds1 = ds2
8. da1 = da2
9. u : Term→Type
10. t:{v:Term| u(v) }→
    term_types(ds1;da1;de;t) = term_types(ds2;da2;de;t)
11. y1 : {v:Term| u(v) }
12. y2 : {v:Term| u(v) }
13. term_types(ds1;da1;de;y1) = term_types(ds2;da2;de;y1)
14. term_types(ds1;da1;de;y2) = term_types(ds2;da2;de;y2)
⊢  st_app(term_types(ds1;da1;de;y1);term_types(ds1;da1;de;y2)) = ...
  by  RW (SweepDnC (HypC (-1) ORELSEC (HypC (-2)))) 0
THENM RelRST

     THEN Auto·
END
```