```
term_types_closed  THM

⊢  ∀t:Term, ds:Collection(dec()), da1,da2:Collection(SimpleType)
|  , de:sig().
|  closed_term(t) ⇒ term_types(ds;da1;de;t) = term_types(ds;da2;de;t)
| by D 0 THENA Auto THEN TermInd (-1) THEN Unfold ‘term_types‘ 0

|    THEN Reduce 0 THEN Try (Fold ‘term_types‘ 0)

|    THEN Auto THEN Try (RelRST THEN Auto)
|
|\
| 1. Term
| 2. u : Term→Type
| 3. t:{v:Term| u(v) }→∀ds:Collection(dec())
| |  , da1,da2:Collection(SimpleType), de:sig().
| |  closed_term(t) ⇒
| |  term_types(ds;da1;de;t) = term_types(ds;da2;de;t)
| 4. x : Label
| 5. Collection(dec())
| 6. da1 : Collection(SimpleType)
| 7. da2 : Collection(SimpleType)
| 8. sig()
| 9. closed_term(x)
| ⊢  da1 = da2
| | by Unfold ‘closed_term‘ (-1) THEN Reduce (-1) THEN
| Auto
| |
| 1. Term
| 2. u : Term→Type
| 3. t:{v:Term| u(v) }→∀ds:Collection(dec())
| |  , da1,da2:Collection(SimpleType), de:sig().
| |  closed_term(t) ⇒
| |  term_types(ds;da1;de;t) = term_types(ds;da2;de;t)
| 4. x : Label
| 5. Collection(dec())
| 6. da1 : Collection(SimpleType)
| 7. da2 : Collection(SimpleType)
| 8. sig()
| 9. term_free_vars(x) = nil
| ⊢  da1 = da2  by Unfold ‘term_free_vars‘ (-1) THEN Reduce (-1)
| |
| 1. Term
| 2. u : Term→Type
| 3. t:{v:Term| u(v) }→∀ds:Collection(dec())
|    , da1,da2:Collection(SimpleType), de:sig().
|    closed_term(t) ⇒
|    term_types(ds;da1;de;t) = term_types(ds;da2;de;t)
| 4. x : Label
| 5. Collection(dec())
| 6. da1 : Collection(SimpleType)
| 7. da2 : Collection(SimpleType)
| 8. sig()
| 9. [x] = nil
| ⊢  da1 = da2  by SimpHyp (-1)•
```

```
|
.\
. 1. Term
. 2. u : Term→Type
. 3. t:{v:Term| u(v) }→∀ds:Collection(dec())
. |  , da1,da2:Collection(SimpleType), de:sig().
. |  closed_term(t) ⇒
. |  term_types(ds;da1;de;t) = term_types(ds;da2;de;t)
. 4. y1 : {v:Term| u(v) }
. 5. y2 : {v:Term| u(v) }
. 6. ds : Collection(dec())
. 7. da1 : Collection(SimpleType)
. 8. da2 : Collection(SimpleType)
. 9. de : sig()
. 10. closed_term(y1 y2)
. ⊢  st_app(term_types(ds;da1;de;y1);term_types(ds;da1;de;y2)) = ...
. | by InstHyp [y1 ;ds;da1;da2;de] 3 THENA Try (Complete Auto)
. |
. |\
. | .....antecedent .....
. | 1. Term
. | 2. u : Term→Type
. | 3. t:{v:Term| u(v) }→∀ds:Collection(dec())
. |    , da1,da2:Collection(SimpleType), de:sig().
. |    closed_term(t) ⇒
. |    term_types(ds;da1;de;t) = term_types(ds;da2;de;t)
. | 4. y1 : {v:Term| u(v) }
. | 5. y2 : {v:Term| u(v) }
. | 6. Collection(dec())
. | 7. Collection(SimpleType)
. | 8. Collection(SimpleType)
. | 9. sig()
. | 10. closed_term(y1 y2)
. | ⊢  closed_term(y1)
. |   by FwdThruLemma ‘closed_tapp‘ [(-1)] THEN Auto
. |
. .\
. . 1. Term
. . 2. u : Term→Type
. . 3. t:{v:Term| u(v) }→∀ds:Collection(dec())
. . |  , da1,da2:Collection(SimpleType), de:sig().
. . |  closed_term(t) ⇒
. . |  term_types(ds;da1;de;t) = term_types(ds;da2;de;t)
. . 4. y1 : {v:Term| u(v) }
. . 5. y2 : {v:Term| u(v) }
. . 6. ds : Collection(dec())
. . 7. da1 : Collection(SimpleType)
. . 8. da2 : Collection(SimpleType)
. . 9. de : sig()
. . 10. closed_term(y1 y2)
. . 11. term_types(ds;da1;de;y1) = term_types(ds;da2;de;y1)
. . ⊢  st_app(term_types(ds;da1;de;y1);term_types(ds;da1;de;y2)) = ...
. . | by InstHyp [y2 ;ds;da1;da2;de] 3 THENA Try (Complete Auto)
. . |
. . |\
```

```
. . | .....antecedent .....
. . | 1. Term
. . | 2. u : Term→Type
. . | 3. t:{v:Term| u(v) }→∀ds:Collection(dec())
. . |    , da1,da2:Collection(SimpleType), de:sig().
. . |    closed_term(t) ⇒
. . |    term_types(ds;da1;de;t) = term_types(ds;da2;de;t)
. . | 4. y1 : {v:Term| u(v) }
. . | 5. y2 : {v:Term| u(v) }
. . | 6. ds : Collection(dec())
. . | 7. da1 : Collection(SimpleType)
. . | 8. da2 : Collection(SimpleType)
. . | 9. de : sig()
. . | 10. closed_term(y1 y2)
. . | 11. term_types(ds;da1;de;y1) = term_types(ds;da2;de;y1)
. . | ⊢  closed_term(y2)
. . |   by InstLemma `closed_tapp` [y1;y2] THEN Auto
. . |
. . |      THEN ThinTrivial THEN Auto
. . |
. . .\
. . . 1. Term
. . . 2. u : Term→Type
. . . 3. t:{v:Term| u(v) }→∀ds:Collection(dec())
. . .    , da1,da2:Collection(SimpleType), de:sig().
. . .    closed_term(t) ⇒
. . .    term_types(ds;da1;de;t) = term_types(ds;da2;de;t)
. . . 4. y1 : {v:Term| u(v) }
. . . 5. y2 : {v:Term| u(v) }
. . . 6. ds : Collection(dec())
. . . 7. da1 : Collection(SimpleType)
. . . 8. da2 : Collection(SimpleType)
. . . 9. de : sig()
. . . 10. closed_term(y1 y2)
. . . 11. term_types(ds;da1;de;y1) = term_types(ds;da2;de;y1)
. . . 12. term_types(ds;da1;de;y2) = term_types(ds;da2;de;y2)
. . . ⊢  st_app(term_types(ds;da1;de;y1);...) = st_app(...;...)
. . .   by RW (SweepDnC (FirstC (map HypC [(-1);(-2)]))) 0
. . .
. . .      THENM RelRST
. . . THEN Auto
END
```