```
Tarski  THM

⊢  ¬(∃Tr:(Term→Prop), tr:Term, L:(Term→Prop). RepsTruth(L; Tr; tr))
| by Analyze 0
|
1. ∃Tr:(Term→Prop), tr:Term, L:(Term→Prop). RepsTruth(L; Tr; tr)
⊢  False  by Unfold ‘RepsTruth‘ 1 THEN ExRepD
|
1. Tr : Term→Prop
2. tr : Term
3. L : Term→Prop
4. ∀S:Term. (∃t:Term. S ↓= t) ⇒ L(SUBX(tr; S))
5. RespectsNot(Tr; L)
6. ReflectsProp(Tr; tr; Tr)
⊢  False  by Let (S = s(NOT(tr)))
|
7. S : Term
8. S = s(NOT(tr))
⊢  False  by Assert (S ↓= NOT(SUBX(tr; S)))
|
|\
| .....assertion .....
| ⊢  S ↓= NOT(SUBX(tr; S))  by HypSubst 8 0
| |
| ⊢  s(NOT(tr)) ↓= NOT(SUBX(tr; s(NOT(tr))))
| | by Rewrite←
| |    Thm* ∀t,e:Term. SUBX(NOT(t); e) = NOT(SUBX(t; e))
| |    {qnot_subx}
| |    0
| |
| ⊢  s(NOT(tr)) ↓= SUBX(NOT(tr); s(NOT(tr)))
|   by BackThru Thm* ∀t:Term. s(t) ↓= SUBX(t; s(t))    {s_reps}
|
.\
. 9. S ↓= NOT(SUBX(tr; S))
. ⊢  False
. | by InstHyp [S] 4 THENA Witness NOT(SUBX(tr; S)) THEN Thin 4
. |
. 4. RespectsNot(Tr; L)
. 5. ReflectsProp(Tr; tr; Tr)
. 6. S : Term
. 7. S = s(NOT(tr))
. 8. S ↓= NOT(SUBX(tr; S))
. 9. L(SUBX(tr; S))
. ⊢  False
. | by Unfold ‘ReflectsProp‘ 5 THEN InstHyp [NOT(SUBX(tr; S));S] 5
. |    THEN
. |    Thin 5
. |
. 5. S : Term
. 6. S = s(NOT(tr))
. 7. S ↓= NOT(SUBX(tr; S))
. 8. L(SUBX(tr; S))
. 9. Tr(SUBX(tr; S)) ⟺ Tr(NOT(SUBX(tr; S)))
. ⊢  False
```

```
. | by Unfold `RespectsNot` 4 THEN InstHyp [SUBX(tr; S)] 4
. |    THEN
. |    Thin 4
. |
. 4. S : Term
. 5. S = s(NOT(tr))
. 6. S ↓= NOT(SUBX(tr; S))
. 7. L(SUBX(tr; S))
. 8. Tr(SUBX(tr; S)) ⟺ Tr(NOT(SUBX(tr; S)))
. 9. Tr(NOT(SUBX(tr; S))) ⟺ L(SUBX(tr; S)) & ¬Tr(SUBX(tr; S))
. ⊢  False
. | by Assert (Tr(NOT(SUBX(tr; S))) ⟺ ¬Tr(SUBX(tr; S)))
. |    THENA
. |    Using [`B',(L(SUBX(tr; S)))]
. |    (BackThru Thm* B ⇒ (A ⟺ B & C) ⇒ (A ⟺ C)    {prop_and_iff})
. |    THEN
. |    OnHyps [9;7] Thin
. |
. 7. Tr(SUBX(tr; S)) ⟺ Tr(NOT(SUBX(tr; S)))
. 8. Tr(NOT(SUBX(tr; S))) ⟺ ¬Tr(SUBX(tr; S))
. ⊢  False
. | by FwdThru
. |    Thm* (A ⟺ B) ⇒ (B ⟺ C) ⇒ (A ⟺ C)    {prop_iff_trans}
. |    [7;8]
. |    THEN
. |    OnHyps [8;7] Thin
. |
. 7. Tr(SUBX(tr; S)) ⟺ ¬Tr(SUBX(tr; S))
. ⊢  False
.   by FwdThru Thm* (P ⟺ ¬P) ⇒ False    {prop_iff_contra} [7]
END
```