```
select-map  THM

⊢  ∀f:Top, T:Type, L:T List, i:ℕ||L||. map(f;L)[i] ~ (f(L[i]))
| by InductionOnList THEN Reduce 0 THEN Analyze 0
|
|\
| 1. f : Top
| 2. T : Type
| 3. T List
| 4. i : ℕ0
| ⊢  nil[i] ~ (f(nil[i]))  by Auto
|
.\
. 1. f : Top
. 2. T : Type
. 3. T List
. 4. u : T
. 5. v : T List
. 6. ∀i:ℕ||v||. map(f;v)[i] ~ (f(v[i]))
. 7. i : ℕ(||v||+1)
. ⊢  [(f(u)) / map(f;v)][i] ~ (f([u / v][i]))
. | by CaseNat 0 ‘i’ THEN Reduce 0 THEN Try Trivial
. |
. 1. f : Top
. 2. T : Type
. 3. T List
. 4. u : T
. 5. v : T List
. 6. ∀i:ℕ||v||. map(f;v)[i] ~ (f(v[i]))
. 7. i : ℕ(||v||+1)
. 8. ¬i = 0
. ⊢  [(f(u)) / map(f;v)][i] ~ (f([u / v][i]))
. | by Unfold ‘select‘ 0 THEN RecUnfold ‘nth_tl‘ 0 THEN Reduce 0
. |    THEN
. |    SplitOnConclITE
. |
. |\
. | .....truecase .....
. | 1. f : Top
. | 2. T : Type
. | 3. T List
. | 4. u : T
. | 5. v : T List
. | 6. ∀i:ℕ||v||. map(f;v)[i] ~ (f(v[i]))
. | 7. i : ℕ(||v||+1)
. | 8. ¬i = 0
. | 9. i≤0
. | ⊢  hd([(f(u)) / map(f;v)]) ~ (f(hd([u / v])))  by Auto
. |
. .\
. . .....falsecase .....
. . 1. f : Top
. . 2. T : Type
. . 3. T List
. . 4. T
```

```
. . 5. v : T List
. . 6. ∀i:ℕ||v||. map(f;v)[i] ~ (f(v[i]))
. . 7. i : ℕ(||v||+1)
. . 8. ¬i = 0
. . 9. 0<i
. . ⊢ hd(nth_tl(i-1;map(f;v))) ~ (f(hd(nth_tl(i-1;v))))
. .   by Fold `select` 0 THEN BackThruSomeHyp
END
```