```
paren_induction  THM

⊢  ∀T:Type, P:(((T+T) List)→Prop).
|  P(nil)
|  ⇒
|  (∀s1,s2:(T+T) List.
|   P(s1) ⇒ P(s2) ⇒ paren(T;s1) ⇒ paren(T;s2) ⇒ P(s1 @ s2))
|  ⇒
|  (∀s:(T+T) List, t:T.
|   P(s) ⇒ paren(T;s) ⇒ P([inl(t)] @ s @ [inr(t)]))
|  ⇒
|  (∀s:(T+T) List. paren(T;s) ⇒ P(s))
| by (Unfold `guard` 0) THEN (MoveToConcl -1) THEN (LengthInd -1)
|    THEN
|    (RecUnfold `paren` -1)
|    THEN
|    SplitOrHyps
|    THEN
|    ExRepD
|
|\
| 1. T : Type
| 2. P : ((T+T) List)→Prop
| 3. P(nil)
| 4. ∀s1,s2:(T+T) List.
|    P(s1) ⇒ P(s2) ⇒ paren(T;s1) ⇒ paren(T;s2) ⇒ P(s1 @ s2)
| 5. ∀s:(T+T) List, t:T.
|    P(s) ⇒ paren(T;s) ⇒ P([inl(t)] @ s @ [inr(t)])
| 6. ℕ
| 7. s : (T+T) List
| 8. ∀s1:(T+T) List. ||s1||<||s|| ⇒ paren(T;s1) ⇒ P(s1)
| 9. s = nil
| ⊢  P(s)  by Obvious
|
|\
| 1. T : Type
| 2. P : ((T+T) List)→Prop
| 3. P(nil)
| 4. ∀s1,s2:(T+T) List.
| |  P(s1) ⇒ P(s2) ⇒ paren(T;s1) ⇒ paren(T;s2) ⇒ P(s1 @ s2)
| 5. ∀s:(T+T) List, t:T.
| |  P(s) ⇒ paren(T;s) ⇒ P([inl(t)] @ s @ [inr(t)])
| 6. ℕ
| 7. s : (T+T) List
| 8. ∀s1:(T+T) List. ||s1||<||s|| ⇒ paren(T;s1) ⇒ P(s1)
| 9. t : T
| 10. s' : (T+T) List
| 11. s = ([inl(t)] @ s' @ [inr(t)])
| 12. paren(T;s')
| ⊢  P(s)  by (HypSubst -2 0) THEN (BackThru 5) THEN BackThruSomeHyp
| |
| 1. T : Type
| 2. P : ((T+T) List)→Prop
| 3. P(nil)
| 4. ∀s1,s2:(T+T) List.
```

```
|    P(s1) ⇒ P(s2) ⇒ paren(T;s1) ⇒ paren(T;s2) ⇒ P(s1 @ s2)
| 5. ∀s:(T+T) List, t:T.
|    P(s) ⇒ paren(T;s) ⇒ P([inl(t)] @ s @ [inr(t)])
| 6. ℕ
| 7. s : (T+T) List
| 8. ∀s1:(T+T) List. ||s1||<||s|| ⇒ paren(T;s1) ⇒ P(s1)
| 9. t : T
| 10. s' : (T+T) List
| 11. s = ([inl(t)] @ s' @ [inr(t)])
| 12. paren(T;s')
| ⊢  ||s'||<||s||
|   by (HypSubst -2 0)
|      THEN
|      (RWW
|       Thm* ∀as,bs:T List. ||as @ bs|| = ||as||+||bs|| ∈ ℤ
|       {length_append}
|       0)
|      THEN
|      (Reduce 0)
|
.\
. 1. T : Type
. 2. P : ((T+T) List)→Prop
. 3. P(nil)
. 4. ∀s1,s2:(T+T) List.
.    P(s1) ⇒ P(s2) ⇒ paren(T;s1) ⇒ paren(T;s2) ⇒ P(s1 @ s2)
. 5. ∀s:(T+T) List, t:T.
.    P(s) ⇒ paren(T;s) ⇒ P([inl(t)] @ s @ [inr(t)])
. 6. ℕ
. 7. s : (T+T) List
. 8. ∀s1:(T+T) List. ||s1||<||s|| ⇒ paren(T;s1) ⇒ P(s1)
. 9. s' : (T+T) List
. 10. s'' : (T+T) List
. 11. ||s'||<||s||
. 12. ||s''||<||s||
. 13. s = (s' @ s'')
. 14. paren(T;s')
. 15. paren(T;s'')
. ⊢  P(s)  by (HypSubst -3 0) THEN (BackThru 4) THEN BackThruSomeHyp
END
```