```
paren_balance2  THM

⊢  ∀T:Type, s':(T+T) List.
|  paren(T;s') ⇒ (∀i:T. (inr(i) ∈ s') ⇒ (inl(i) ∈ s'))
| by (Analyze 0) THEN (BackThru Thm: paren_induction)
|
|\
| 1. T : Type
| 2. i : T
| 3. (inr(i) ∈ nil)
| ⊢  (inl(i) ∈ nil)  by Obvious
|
|\
| 1. T : Type
| 2. s1 : (T+T) List
| 3. s2 : (T+T) List
| 4. ∀i:T. (inr(i) ∈ s1) ⇒ (inl(i) ∈ s1)
| 5. ∀i:T. (inr(i) ∈ s2) ⇒ (inl(i) ∈ s2)
| 6. paren(T;s1)
| 7. paren(T;s2)
| 8. i : T
| 9. (inr(i) ∈ s1 @ s2)
| ⊢  (inl(i) ∈ s1 @ s2)
| | by All
| |    (RWO
| |     Thm* ∀x:T, l1,l2:T List. (x ∈ l1 @ l2) ⟺ (x ∈ l1) ∨ (x ∈ l2)
| |     {member_append})
| |
| 1. T : Type
| 2. s1 : (T+T) List
| 3. s2 : (T+T) List
| 4. ∀i:T. (inr(i) ∈ s1) ⇒ (inl(i) ∈ s1)
| 5. ∀i:T. (inr(i) ∈ s2) ⇒ (inl(i) ∈ s2)
| 6. paren(T;s1)
| 7. paren(T;s2)
| 8. i : T
| 9. (inr(i) ∈ s1) ∨ (inr(i) ∈ s2)
| ⊢  (inl(i) ∈ s1) ∨ (inl(i) ∈ s2)  by Obvious
|
.\
. 1. T : Type
. 2. s' : (T+T) List
. 3. t : T
. 4. ∀i:T. (inr(i) ∈ s') ⇒ (inl(i) ∈ s')
. 5. paren(T;s')
. 6. i : T
. 7. (inr(i) ∈ [inl(t)] @ s' @ [inr(t)])
. ⊢  (inl(i) ∈ [inl(t)] @ s' @ [inr(t)])
.   by (All
.       (λi.
.        (RWW
.         Thm* ∀x:T, l1,l2:T List.
.              (x ∈ l1 @ l2) ⟺ (x ∈ l1) ∨ (x ∈ l2)
.         {member_append}
.         i)
```

```
.        THEN
.        (RWO Thm* ∀a,x:T. (x ∈ [a]) ⟺ x = a    {member_singleton}
.         i)))
.      THEN
.      Obvious
END
```