```
tree_leaf_one_one  THM

⊢  ∀E:Type, x1,x2:E. tree_leaf(x1) = tree_leaf(x2) ⇒ x1 = x2
| by Auto
|
1. E : Type
2. x1 : E
3. x2 : E
4. tree_leaf(x1) = tree_leaf(x2)
⊢  x1 = x2
| by ApFunToHypEquands ‘z’ if is_leaf(z)→ leaf_value(z) else x1 fi E
|     (-1)
|
|
|\
| .....fun wf .....
| 1. E : Type
| 2. x1 : E
| 3. x2 : E
| 4. tree_leaf(x1) = tree_leaf(x2)
| 5. z : Tree(E)
| ⊢  if is_leaf(z)→ leaf_value(z) else x1 fi
|    =
|    if is_leaf(z)→ leaf_value(z) else x1 fi
|   by SplitOnConclITE THEN Auto
|
|\
| .....wf .....
| 1. E : Type
| 2. x1 : E
| 3. x2 : E
| 4. tree_leaf(x1) = tree_leaf(x2)
| ⊢  Tree(E) = Tree(E)  by Auto
|
.\
. 1. E : Type
. 2. x1 : E
. 3. x2 : E
. 4. tree_leaf(x1) = tree_leaf(x2)
. 5. if is_leaf(tree_leaf(x1))→ leaf_value(tree_leaf(x1)) else x1 fi
.    =
.    if is_leaf(tree_leaf(x2))→ leaf_value(tree_leaf(x2)) else x1 fi
. ⊢  x1 = x2  by Reduce (-1) THEN Auto
END
```