```
derive_refl  THM

⊢  ∀G:Grammar(V;T), l:(V+T)*. l ⇒ l
| by Unfold `derive` 0 THEN Reduce 0
|
1. V : Type
2. T : Type
3. G : Grammar(V;T)
4. l : (V+T)*
⊢  ∃a:((V+T)* List+).
|  l = a[0] & (∀i:ℕ(||a||-1). a[i] ⇒ a[(i+1)]) & a[(||a||-1)] = l
| by Witness [l]
|
|\
| .....wf .....
| ⊢  [l] ∈ ((V+T)* List+)  by Analyze
| |
| .....set predicate .....
| ⊢  ||[l]||>0
| | by RWH
| |    (LemmaC Thm* ∀a:A, as:A*. ||a.as|| = ||as||+1    {length_cons})
| |    0
| |
| ⊢  ||nil||+1>0  by RWH (LemmaC Thm* ||nil|| = 0    {length_nil}) 0
|
|\
| ⊢  l = [l][0]
|    & (∀i:ℕ(||[l]||-1). [l][i] ⇒ [l][(i+1)])
|    & [l][(||[l]||-1)] = l  by Reduce 0
|
.\
. .....wf .....
. 5. a : (V+T)* List+
. ⊢  (l = a[0] & (∀i:ℕ(||a||-1). a[i] ⇒ a[(i+1)]) & a[(||a||-1)] = l)
.    ∈ Prop  by Analyze 5
END
```