```
counter_append  THM

⊢  ∀e:ℤ, L,M:ℤ*. ||e:L @ M|| = ||e:L||+||e:M|| ∈ ℕ  by UnivCD
|
1. e : ℤ
2. L : ℤ*
3. M : ℤ*
⊢  ||e:L @ M|| = ||e:L||+||e:M||  by ListInd 2
|
|\
| ⊢  ||e:nil @ M|| = ||e:nil||+||e:M||  by RecCaseSplit ‘append‘
| |
| ⊢  ||e:M|| = ||e:nil||+||e:M||
| | by Inst Thm* ∀e:ℤ. ||e:nil|| = 0 ∈ ℕ    {counter_of_nil} [e]
| |
| 4. ||e:nil|| = 0
| ⊢   by Auto
|
.\
. 4. u : ℤ
. 5. v : ℤ*
. 6. ||e:v @ M|| = ||e:v||+||e:M||
. ⊢  ||e:(u.v) @ M|| = ||e:u.v||+||e:M||
. | by RW (AddrC [2] (RecUnfoldC ‘el_counter‘)) 0
. |
. ⊢  if
. |   null((u.v) @ M)→0
. |  ;e=₂hd(((u.v) @ M))→1+||e:tl(((u.v) @ M))||
. |  else ||e:tl(((u.v) @ M))|| fi
. |  =
. |  ||e:u.v||+||e:M||  by Reduce 0
. |
. ⊢  if e=₂u→ 1+||e:v @ M|| else ||e:v @ M|| fi = ||e:u.v||+||e:M||
. | by SplitOnConclITE
. |
. |\
. | .....truecase .....
. | 7. e = u
. | ⊢  1+||e:v @ M|| = ||e:u.v||+||e:M||
. | | by RW (AddrC [3;1] (RecUnfoldC ‘el_counter‘)) 0
. | |
. | ⊢  1+||e:v @ M||
. | |  =
. | |  if
. | |   null(u.v)→0
. | |  ;e=₂hd((u.v))→1+||e:tl((u.v))||
. | |  else ||e:tl((u.v))|| fi+||e:M||  by Reduce 0
. | |
. | ⊢  1+||e:v @ M|| = if e=₂u→ 1+||e:v|| else ||e:v|| fi+||e:M||
. | | by SplitOnConclITE
. | |
. | |\
. | | .....truecase .....
. | | 8. e = u
. | | ⊢  1+||e:v @ M|| = 1+||e:v||+||e:M||  by Auto
```

```
. | |
. | .\
. | . .....falsecase .....
. | . 8. ¬e = u
. | . ⊢  1+||e:v @ M|| = ||e:v||+||e:M||  by Auto
. |
. .\
. . .....falsecase .....
. . 7. ¬e = u
. . ⊢  ||e:v @ M|| = ||e:u.v||+||e:M||
. . | by RW (AddrC [3;1] (RecUnfoldC ‘el_counter‘)) 0
. . |
. . ⊢  ||e:v @ M||
. . |  =
. . |  if
. . |   null(u.v)→0
. . |  ;e=₂hd((u.v))→1+||e:tl((u.v))||
. . |  else ||e:tl((u.v))|| fi+||e:M||  by Reduce 0
. . |
. . ⊢  ||e:v @ M|| = if e=₂u→ 1+||e:v|| else ||e:v|| fi+||e:M||
. .   by SplitOnConclITE
END
```