```
rem_quo_unique  THM

⊢  ∀n:ℕ+, r1,r2:ℕn, q1,q2:ℕ. r1+q1·n = r2+q2·n ⇒ r1 = r2 & q1 = q2
| by UnivCD
|
1. n : ℕ+
2. r1 : ℕn
3. r2 : ℕn
4. q1 : ℕ
5. q2 : ℕ
6. r1+q1·n = r2+q2·n
⊢  r1 = r2 & q1 = q2  by Assert (r1 = r2)
|
|\
| .....assertion .....
| ⊢  r1 = r2  by ApFunToHypEquands ‘x’ (x rem n) ℤ 6
| |
| 7. ((r1+q1·n) rem n) = ((r2+q2·n) rem n)
| ⊢
| | by RWH
| |    (LemmaC
| |     Thm* ∀a,b:ℕ, n:ℕ+. ((a+b·n) rem n) = (a rem n)
| |     {rem_invariant})
| |    7
| |
| 7. (r1 rem n) = (r2 rem n)
| ⊢
|   by RWH
|      (LemmaC
|       Thm* ∀a:ℕ, n:ℕ+. a<n ⇒ (a rem n) = a    {rem_base_case})
|      7
|
.\
. 7. r1 = r2
. ⊢   by Analyze 0
. |
. |\
. | ⊢  r1 = r2  by Auto
. |
. .\
. . ⊢  q1 = q2  by RWH (HypC 7) 6
. . |
. . 6. r2+q1·n = r2+q2·n
. . 7. r1 = r2
. . ⊢   by RWH (LemmaC Thm* ∀a,b:ℤ. a+b = b+a    {add_com}) 6
. . |
. . 6. q1·n+r2 = q2·n+r2
. . 7. r1 = r2
. . ⊢
. . | by Inst Thm* ∀a,b,n:ℤ. a+n = b+n ⇒ a = b    {add_cancel_in_eq}
. . |    [q1·n;q2·n;r2]
. . |
. . 8. q1·n = q2·n
. . ⊢   by RWH (LemmaC Thm* ∀a,b:ℤ. a·b = b·a    {mul_com}) 8
. . |
```

```
. . 8. n·q1 = n·q2
. . ⊢
. .   by Inst
. .      Thm* ∀a,b:ℤ, n:ℤ^-0. n·a = n·b ⇒ a = b     {mul_cancel_in_eq}
. .      [q1;q2;n]
END
```