```
quot_rem_exists_n  THM

⊢  ∀a:ℕ, b:ℕ+. ∃q:ℕ, r:ℕb. a = q·b+r  by UnivCD
|
1. a : ℕ
2. b : ℕ+
⊢  ∃q:ℕ, r:ℕb. a = q·b+r
| by Assert (∀c:ℕ. (∃q:ℕ. a = q·b+c) ⇒ (∃q:ℕ, r:ℕb. a = q·b+r))
|    THENA
|    Analyze 0
|
|\
| 3. c : ℕ
| ⊢  (∃q:ℕ. a = q·b+c) ⇒ (∃q:ℕ, r:ℕb. a = q·b+r)
| | by (OnVar ‘c’ CompNatInd) THEN (Analyze 0)
| |
| 4. ∀c1:ℕ. c1<c ⇒ (∃q:ℕ. a = q·b+c1) ⇒ (∃q:ℕ, r:ℕb. a = q·b+r)
| 5. ∃q:ℕ. a = q·b+c
| ⊢  ∃q:ℕ, r:ℕb. a = q·b+r  by Decide (c<b)
| |
| |\
| | 6. c<b
| | ⊢  ∃q:ℕ, r:ℕb. a = q·b+r  by Analyze 5 THEN InstConcl [q;c]
| |
| .\
| . 6. ¬c<b
| . ⊢  ∃q:ℕ, r:ℕb. a = q·b+r  by InstHyp [c-b] 4
| . |
| . .....antecedent .....
| . ⊢  ∃q:ℕ. a = q·b+c-b  by Analyze 5 THEN InstConcl [q+1]
|
.\
. 3. ∀c:ℕ. (∃q:ℕ. a = q·b+c) ⇒ (∃q:ℕ, r:ℕb. a = q·b+r)
. ⊢  ∃q:ℕ, r:ℕb. a = q·b+r  by InstHyp [a] 3
. |
. .....antecedent .....
. ⊢  ∃q:ℕ. a = q·b+a  by InstConcl [0]
END
```