```
mn_quo_append_assoc  THM

⊢  ∀R:(Alph*→Alph*→Prop).
|  (EquivRel x,y:Alph*. x R y) ⇒
|  (∀x,y,z:Alph*. (x R y) ⇒ ((z @ x) R (z @ y))) ⇒
|  (∀z1,z2:Alph*, y:x,y:Alph*//(x R y). z1 @ z2@_q y = z1@_q z2@_q y)
| by UnivCD
|
1. Alph : Type
2. R : Alph*→Alph*→Prop
3. EquivRel x,y:Alph*. x R y
4. ∀x,y,z:Alph*. (x R y) ⇒ ((z @ x) R (z @ y))
5. z1 : Alph*
6. z2 : Alph*
7. y : x,y:Alph*//(x R y)
⊢  z1 @ z2@_q y = z1@_q z2@_q y  by RWH (UnfoldC ‘mn_quo_append‘) 0
|
⊢  ((z1 @ z2) @ y) = (z1 @ z2 @ y)  by Analyze 7
|
7. y1 : Alph*
8. y2 : Alph*
9. y1 R y2
⊢  ((z1 @ z2) @ y1) = (z1 @ z2 @ y2)  by Analyze
|
⊢  ((z1 @ z2) @ y1) R (z1 @ z2 @ y2)
| by RWH
|    (RevLemmaC
|     Thm* ∀as,bs,cs:T*. ((as @ bs) @ cs) = (as @ (bs @ cs))
|     {append_assoc})
|    0
|
⊢  ((z1 @ z2) @ y1) R ((z1 @ z2) @ y2)  by BackThru 4
END
```