```
compose-biject  THM

⊢  ∀A,B,C:Type, f:(A→B), g:(B→C).
|  Bij(A; B; f) ⇒ Bij(B; C; g) ⇒ Bij(A; C; g o f)
| by (Unfold `biject` 0) THEN (Unfolds [`inject`;`surject`] 0)
|    THEN
|    (Reduce 0)
|
|\
| 1. A : Type
| 2. B : Type
| 3. C : Type
| 4. f : A→B
| 5. g : B→C
| 6. ∀a1,a2:A. f(a1) = f(a2) ⇒ a1 = a2
| 7. ∀b:B. ∃a:A. f(a) = b
| 8. ∀a1,a2:B. g(a1) = g(a2) ⇒ a1 = a2
| 9. ∀b:C. ∃a:B. g(a) = b
| 10. a1 : A
| 11. a2 : A
| 12. g(f(a1)) = g(f(a2))
| ⊢  a1 = a2  by Obvious
|
.\
. 1. A : Type
. 2. B : Type
. 3. C : Type
. 4. f : A→B
. 5. g : B→C
. 6. ∀a1,a2:A. f(a1) = f(a2) ⇒ a1 = a2
. 7. ∀b:B. ∃a:A. f(a) = b
. 8. ∀a1,a2:B. g(a1) = g(a2) ⇒ a1 = a2
. 9. ∀b:C. ∃a:B. g(a) = b
. 10. b : C
. ⊢  ∃a:A. g(f(a)) = b
. | by (InstHyp [b] -2) THEN ExRepD THEN (InstHyp [a] -6)
. |    THEN
. |    (ParallelOp -1)
. |
. 1. A : Type
. 2. B : Type
. 3. C : Type
. 4. f : A→B
. 5. g : B→C
. 6. ∀a1,a2:A. f(a1) = f(a2) ⇒ a1 = a2
. 7. ∀b:B. ∃a:A. f(a) = b
. 8. ∀a1,a2:B. g(a1) = g(a2) ⇒ a1 = a2
. 9. ∀b:C. ∃a:B. g(a) = b
. 10. b : C
. 11. a : B
. 12. g(a) = b
. 13. a@0 : A
. 14. f(a@0) = a
. ⊢  g(f(a@0)) = b  by Obvious
END
```