```
graph-isomorphic_transitivity  THM

⊢  ∀G,H,K:Graph. G ≡ H ⇒ H ≡ K ⇒ G ≡ K
| by Auto THEN (All (Unfold `graph-isomorphic`)) THEN ExRepD
|
1. G : Graph
2. H : Graph
3. K : Graph
4. v1 : Vertices(G)→Vertices(H)
5. e1 : Edges(G)→Edges(H)
6. Bij(Vertices(G); Vertices(H); v1)
7. Bij(Edges(G); Edges(H); e1)
8. (v1,v1) o Incidence(G) = Incidence(H) o e1
9. vmap : Vertices(H)→Vertices(K)
10. emap : Edges(H)→Edges(K)
11. Bij(Vertices(H); Vertices(K); vmap)
12. Bij(Edges(H); Edges(K); emap)
13. (vmap,vmap) o Incidence(H) = Incidence(K) o emap
⊢  ∃vmap:(Vertices(G)→Vertices(K)), emap:(Edges(G)→Edges(K)).
|  Bij(Vertices(G); Vertices(K); vmap)
|  & Bij(Edges(G); Edges(K); emap)
|  & (vmap,vmap) o Incidence(G) = Incidence(K) o emap
| by InstConcl [vmap o v1;emap o e1]
|
1. G : Graph
2. H : Graph
3. K : Graph
4. v1 : Vertices(G)→Vertices(H)
5. e1 : Edges(G)→Edges(H)
6. Bij(Vertices(G); Vertices(H); v1)
7. Bij(Edges(G); Edges(H); e1)
8. (v1,v1) o Incidence(G) = Incidence(H) o e1
9. vmap : Vertices(H)→Vertices(K)
10. emap : Edges(H)→Edges(K)
11. Bij(Vertices(H); Vertices(K); vmap)
12. Bij(Edges(H); Edges(K); emap)
13. (vmap,vmap) o Incidence(H) = Incidence(K) o emap
⊢  Bij(Vertices(G); Vertices(K); vmap o v1)
|  & Bij(Edges(G); Edges(K); emap o e1)
|  & (vmap o v1,vmap o v1) o Incidence(G) = Incidence(K) o emap o e1
| by (Reduce 0)
|    THEN
|    (Try
|     (BLemma'
|      Thm* ∀f:(A→B), g:(B→C).
|           Bij(A; B; f) ⇒ Bij(B; C; g) ⇒ Bij(A; C; g o f)
|      {compose_DASH_biject}))
|
1. G : Graph
2. H : Graph
3. K : Graph
4. v1 : Vertices(G)→Vertices(H)
5. e1 : Edges(G)→Edges(H)
6. Bij(Vertices(G); Vertices(H); v1)
7. Bij(Edges(G); Edges(H); e1)
```

```
8. (v1,v1) o Incidence(G) = Incidence(H) o e1
9. vmap : Vertices(H)→Vertices(K)
10. emap : Edges(H)→Edges(K)
11. Bij(Vertices(H); Vertices(K); vmap)
12. Bij(Edges(H); Edges(K); emap)
13. (vmap,vmap) o Incidence(H) = Incidence(K) o emap
⊢  (vmap o v1,vmap o v1) o Incidence(G) = Incidence(K) o emap o e1
  by (RWO
      Thm* ∀f:(A→B), g:(B→C), h:(C→D).
           h o (g o f) = (h o g) o f ∈ A→D
      {comp_assoc}
      0)
     THEN
     (RevHypSubst -1 0)
     THEN
     (RWO "comp2_comp_assoc<" 0)
     THEN
     (RevHypSubst -6 0)
     THEN
     (RWO
      Thm* ∀h:(A'→A1×A2), g1:(A1→B), g2:(A2→C), f1:(B→B')
           , f2:(C→C'). (f1,f2) o (g1,g2) o h = (f1 o g1,f2 o g2) o h
      {comp2_comp2_assoc}
      0)
END
```