```
graph-isomorphic_inversion  THM

⊢  ∀G,H:Graph. G ≡ H ⇒ H ≡ G
| by (Unfold `graph-isomorphic` 0) THEN ExRepD
|    THEN
|    (AllHyps
|     (λh.
|      FwdThru
|      Thm* ∀f:(A→B).
|           Bij(A; B; f)
|           ⇒
|           (∃g:(B→A). Bij(B; A; g) & InvFuns(A; B; f; g))
|      {inverse_DASH_biject}
|      [h]))
|    THEN
|    ExRepD
|    THEN
|    (InstConcl [g;g1])
|
1. G : Graph
2. H : Graph
3. vmap : Vertices(G)→Vertices(H)
4. emap : Edges(G)→Edges(H)
5. Bij(Vertices(G); Vertices(H); vmap)
6. Bij(Edges(G); Edges(H); emap)
7. (vmap,vmap) o Incidence(G) = Incidence(H) o emap
8. g1 : Edges(H)→Edges(G)
9. Bij(Edges(H); Edges(G); g1)
10. InvFuns(Edges(G); Edges(H); emap; g1)
11. g : Vertices(H)→Vertices(G)
12. Bij(Vertices(H); Vertices(G); g)
13. InvFuns(Vertices(G); Vertices(H); vmap; g)
⊢  (g,g) o Incidence(H) = Incidence(G) o g1
| by All (Unfold `inv_funs`)
|
1. G : Graph
2. H : Graph
3. vmap : Vertices(G)→Vertices(H)
4. emap : Edges(G)→Edges(H)
5. Bij(Vertices(G); Vertices(H); vmap)
6. Bij(Edges(G); Edges(H); emap)
7. (vmap,vmap) o Incidence(G) = Incidence(H) o emap
8. g1 : Edges(H)→Edges(G)
9. Bij(Edges(H); Edges(G); g1)
10. g1 o emap = Id
11. emap o g1 = Id
12. g : Vertices(H)→Vertices(G)
13. Bij(Vertices(H); Vertices(G); g)
14. g o vmap = Id
15. vmap o g = Id
⊢  (g,g) o Incidence(H) = Incidence(G) o g1
  by (RW
      (AddrC [2]
       (RevLemmaC Thm* ∀f:(A→B). f o Id = f    {comp_id_r}))
      0)
```

```
    THEN
    (SubstFor Id 0)
    THEN
    (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
    (RWO "comp2_comp_assoc<" 0)
    THEN
    (SubstFor (Incidence(H) o emap) 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)
    THEN
    (SubstFor (g o vmap) 0)
    THEN
    (RWO Thm* ∀g:(A→B×C). (Id,Id) o g = g    {comp2_id_l} 0)
END
```