binhom[x, y] ⊆ domain[eval[z]]

Johan G. F. Belinfante
2013 July 22

In[1]:= SetDirectory["1:"]; << goedel.13jul20a

:Package Title: goedel.13jul20a 2013 July 20 at 1:00 p.m.

Loading takes about seventeen minutes, half that time due to builtin pauses.

It is now: 2013 Jul 22 at 13:17

Loading Simplification Rules

TOOLS.M is now incorporated in the GOEDEL program as of 2010 September 3

weightlimit = 40

Loading completed.

It is now: 2013 Jul 22 at 13:34

summary

A rewrite rule is derived for the inclusion \texttt{binhom}[x, y] \subseteq \texttt{domain}[eval[z]] that subsumes several existing rules for special cases.

derivation

Lemma. An implication in one direction.

\textbf{In[2]:=} \texttt{Map[implies[member[z, fix[domain[x]]], \# \$],}

\texttt{SubstTest[implies, and[subclass[u, v], subclass[v, w], subclass[u, w]],}

\texttt{(u \rightarrow \texttt{binhom}[x, y], v \rightarrow \texttt{map[fix[domain[x]], fix[domain[y]]], w \rightarrow domain[eval[z]]])] //}

\texttt{MapNotNot // Reverse}

\textbf{Out[2]=} \texttt{or[not[member[z, fix[domain[x]]]], subclass[binhom[x, y], domain[eval[z]]]] = True}

\textbf{In[3]:=} \texttt{(# /\{x \rightarrow x\_, y \rightarrow y\_, z \rightarrow z\_\}) /\ Equal \rightarrow SetDelayed}

A weaker implication in the reverse direction will next be derived.

Lemma. (Starting point for an implication in the reverse direction.)
Corollary. Main result.

In[12] := equiv[subclass[binhom[x, y], domain[eval[z]]],
     or[empty[binhom[x, y]], member[z, fix[domain[x]]]]] = True

Out[12] = True

In[13] := subclass[binhom[x_, y_], domain[eval[z_]]] :=
     or[equal[0, binhom[x, y]], member[z, fix[domain[x]]]]