compatible collections of functions

Johan G. F. Belinfante
2010 October 1

In[1]:= SetDirectory["l:"]; << goedel.10sep30a

:Package Title: goedel.10sep30a 2010 September 30 at 6:20 p.m.

It is now: 2010 Oct 1 at 7:43

Loading Simplification Rules

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

weightlimit = 40

---

summary

A collection of functions is **compatible** if their union is a function. In this notebook a new criterion for compatibility is derived. It will be shown that a collection of functions is compatible if and only if they are all restrictions of their union.

---

**general results**

The new criterion for compatibility can be most succinctly formulated in terms of the class \( RS[x] \) of restrictions of a class \( x \). Recall that the membership rule for the class of restrictions is the following.

In[2]:= member[u, RS[x]]

Out[2]= and[equal[u, composite[x, id[domain[u]]]], member[u, V]]

An important fact needed in the sequel is that a class is a function if and only if every subset is a restriction. That is, a class is a function if and only if its class of restrictions is equal to its power class.

In[3]:= equal[P[x], RS[x]]

Out[3]= FUNCTION[x]

If the union of a class of sets is a function, then every member of the class is a function. This fact is also already available in the GOEDEL program:

In[4]:= implies[FUNCTION[U[x]], subclass[x, FUNS]]


The following general result will also be needed later.
Theorem. Every member of a class of functions is a function.

\[
\text{In[5]} := \text{SubstTest[implies, and\{member[x, y], subclass[y, z]\}, member[x, z], z \rightarrow \text{FUNS}] // Reverse}
\]
\[
\text{Out[5]} = \text{or[FUNCTION[x], not\{member[x, y]\], not\{subclass[y, \text{FUNS}]\}] = True}
\]
\[
\text{In[6]} := \text{or[FUNCTION[x\_,], not\{member[x\_, y\_]\], not\{subclass[y\_, \text{FUNS}]\}] := True}
\]

---

**derivation**

Theorem. If the union of a collection of sets is a function, then every member of the collection is a restriction of their union.

\[
\text{In[7]} := \text{SubstTest[implies, and\{subclass[u, v], equal[v, w]\], subclass[u, w], \{u \rightarrow x, v \rightarrow \text{P}[U[x]], w \rightarrow \text{RS}[U[x]]\}] // Reverse}
\]
\[
\text{Out[7]} = \text{or[not[FUNCTION[U[x]]], subclass[x, RS[U[x]]]] = True}
\]
\[
\text{In[8]} := \text{or[not[FUNCTION[U[x\_]]], subclass[x\_, RS[U[x\_]]]] := True}
\]

To derive an implication in the reverse direction, a new variable \( u \) will be introduced, and later eliminated.

Lemma. If \( u \in x \) and \( x \subset \text{RS[U[x]]} \), then \( u \) is a restriction of \( U[x] \).

\[
\text{In[9]} := \text{SubstTest[implies, and\{member[u, x], subclass[x, v]\], member[u, v], v \rightarrow \text{RS[U[x]]}] // Reverse}
\]
\[
\text{Out[9]} = \text{or[equal[u, \text{composite}[U[x], \text{id}[\text{domain}[u]]]], not[member[u, x]], not[subclass[x, RS[U[x]]]]] = True}
\]
\[
\text{In[10]} := \text{or[equal[u\_, \text{composite}[U[x\_], \text{id}[\text{domain}[u\_]]]], not[member[u\_, x\_]], not[subclass[x\_, RS[U[x\_]]]]] := True}
\]

The **funpart** of any class \( x \) is its restriction the class of points where its vertical sections are singletons. A class is a function if it is its own funpart.

Lemma. If \( x \subset \text{FUNS} \) and \( x \subset \text{RS[U[x]]} \), then every member of \( \text{RS[U[x]]} \) is a subset of **funpart**[U[x]].

\[
\text{In[11]} := \text{Map[not, SubstTest[and, implies[and\{p0, p1, p3\], implies[and\{p0, p2\], p4\], implies[and\{p3, p4\], p5\], not[implies[and\{p0, p1, p2\], p5\], p3 \rightarrow \text{FUNCTION}[u], p4 \rightarrow \text{equal}[u, \text{composite}[U[x], \text{id}[\text{domain}[u]]]], p5 \rightarrow \text{equal}[u, \text{composite}[\text{funpart}[U[x]], \text{id}[\text{domain}[u]]]]] // Reverse}
\]
\[
\text{Out[11]} = \text{or[not[member[u, x]], not[subclass[x, \text{FUNS}]], not[subclass[x, RS[U[x]]]], subclass[u, \text{funpart[U[x]]}]] = True}
\]
\[
\text{In[12]} := (\% /. \{u \rightarrow u\_, x \rightarrow x\_\}) /. \text{Equal} \rightarrow \text{SetDelayed}
\]

Theorem. If every member of a class of functions is a restriction of their union, then their union is a function.
In[13]:= SubstTest[class, u, imlies[and[member[u, x], subclass[x, y]], member[u, v]], 
    {v -> P[funpart[U[x]]], y -> intersection[FUNS, RS[U[x]]]}]

Out[13]= or[FUNCTION[U[x]], not[subclass[x, FUNS]], not[subclass[x, RS[U[x]]]]] := True

In[14]:= or[FUNCTION[U[x_]], not[subclass[x_, FUNS]], not[subclass[x_, RS[U[x_]]]]] := True

Corollary. A condition for a class of functions to be compatible.

In[15]:= equiv[and[subclass[x, FUNS], subclass[x, RS[U[x]]]], FUNCTION[U[x]]] // not // not


In[16]:= and[subclass[x_, FUNS], subclass[x_, RS[U[x_]]]] := FUNCTION[U[x]]

serendipity

The following simple fact was encountered in the course of this study, but was not actually needed.

Theorem.

In[17]:= SubstTest[and, subclass[u, v], subclass[v, u], {u -> P[x], v -> RS[x]}] // Reverse

Out[17]= subclass[P[x], RS[x]] = FUNCTION[x]

In[18]:= subclass[P[x_], RS[x_]] := FUNCTION[x]