connex strict orders and total orders

Johan G. F. Belinfante
2010 February 8

A relation \( x \) is connex (or connected) if each pair of elements in \( \text{udora}[x] \) are related by either \( x \) or its inverse. This condition can be stated formally as \( \text{udora}[x] \times \text{udora}[x] \subseteq \text{Id} \cup x \cup \text{inverse}[x] \). There is an almost but not quite equivalent condition that avoids explicit mention of \( \text{udora}[x] = \text{domain}[x] \cup \text{range}[x] \), but introduces an additional variable \( y \) that is generally equal to \( \text{udora}[x] \) except in certain trivial situations. This condition is: \( x \subseteq y \times y \subseteq \text{Id} \cup x \cup \text{inverse}[x] \). The main advantage of this alternate description is that it makes replacing \( x \) with a restriction easier to do. In this notebook rewrite rules are derived that relate these two formulations of connectedness. If \( y \) is connected by the relation \( x \), then either \( y \) is a singleton or \( y \subseteq \text{udora}[x] \). If \( x \) is a strict order and \( x \subseteq y \times y \subseteq \text{Id} \cup x \cup \text{inverse}[x] \), then \( x \cup \text{id}[y] \) is a total order. It follows as a corollary that if a class \( y \) is connected by a strict order \( x \) then \( (\text{Id} \cup x) \cap (y \times y) \) is a total order, and consequently the cover relation of \( x \cap (y \times y) \) is a function.

One says that a class \( y \) is connected by a relation \( x \) if \( y \times y \subseteq \text{Id} \cup x \cup \text{inverse}[x] \). When this is the case, it almost follows that the class \( y \) is contained in \( \text{udora}[x] \). The following lemma stops just short of saying this because of the presence of an intersection of \( y \) with \( \text{image}[\text{Di}, y] \).

Lemma.

\[
\text{In}[2]:= \text{SubstTest}[\text{implies}, \text{subclass}[u, v], \text{subclass}[\text{domain}[u], \text{domain}[v]], \\
\{u \rightarrow \text{restrict[Di, y, y]}, v \rightarrow \text{union[x, inverse[x]]}\}] // \text{Reverse}
\]

\[
\text{Out}[2]= \text{or[not[subclass[cart[y, y], union[\text{Id, x, inverse[x]]]], \\
\text{subclass[intersection[y, image[Di, y]], union[domain[x], range[x]]]]} = \text{True}}
\]

\[
\text{In}[3]:= \{\% /. \{x \rightarrow _-, y \rightarrow _-\}\} /. \text{Equal} \rightarrow \text{SetDelayed}
\]
Lemma. This lemma helps eliminate the intersection with image[D, y].

\[
\text{In[4]:=} \text{SubstTest[implies, and[equal[y, t], subclass[t, x]],}
\text{subclass[y, x], t -> intersection[y, image[D, y]]] // Reverse}
\]

\[
\text{Out[4]=} \text{or[member[y, range[SINGLETON]],}
\not[\text{subclass[intersection[y, image[D, y]], x]], subclass[y, x]] = True}
\]

\[
\text{In[5]:=} \text{(% /. \{x \rightarrow x_, y \rightarrow y_\}) /. Equal \rightarrow SetDelayed}
\]

Theorem. If \( y \) is connected by the relation \( x \), then either \( y \) is a singleton or \( y \subset udora[x] \).

\[
\text{In[6]:=} \text{Map[not, SubstTest[and, implies[p1, p2], implies[p2, p3],}
\not[\text{implies[p1, p3]], \{p1 -> subclass[cart[y, y], union[Id, x, inverse[x]]],}
\text{p2 -> subclass[intersection[y, image[D, y]], udora[x]],}
\text{p3 -> or[member[y, range[SINGLETON]], subclass[y, udora[x]]]]] // Reverse}
\]

\[
\text{Out[6]=} \text{or[member[y, range[SINGLETON]],}
\not[\text{subclass[cart[y, y], union[Id, x, inverse[x]]]],}
\text{subclass[y, union[domain[x], range[x]]]] = True}
\]

\[
\text{In[7]:=} \text{or[member[y_, range[SINGLETON]],}
\not[\text{subclass[cart[y_, y_], union[Id, inverse[x], x_]]},
\text{subclass[y_, union[domain[x], range[x]]]] := True}
\]

Lemma. (A corollary of the coextension principle for the case of unions.)

\[
\text{In[8]:=} \text{or[equal[z, union[x, y]], notsubclass[x, z]],}
\not[\text{subclass[z, union[x, y]]] // NotNotTest}
\]

\[
\text{Out[8]=} \text{or[equal[z, union[x, y]], notsubclass[x, z]],}
\not[\text{subclass[y, z]], notsubclass[z, union[x, y]]] = True}
\]

\[
\text{In[9]:=} \text{(% /. \{x \rightarrow x_, y \rightarrow y_\, z \rightarrow z_\}) /. Equal \rightarrow SetDelayed}
\]

A relation \( x \) is connex (or connected) if \( udora[x] \) is connected by \( x \).

Theorem. If \( x \subset y \times y \subset Id \cup x \cup \text{inverse}[x] \), then either \( y \) is a singleton or \( y = udora[x] \).

\[
\text{In[10]:=} \text{Map[not, SubstTest[and, implies[and[p2, p3, p5], p6], not[implies[p1, or[p4, p6]]],}
\{p1 -> andsubclass[x, cart[y, y]], subclass[cart[y, y], union[Id, x, inverse[x]]],}
\text{p2 -> subclass[domain[x], y], p3 -> subclass[range[x], y],}
\text{p4 -> member[y, range[SINGLETON]], p5 -> subclass[y, union[domain[x], range[x]]],}
\text{p6 -> equal[y, union[domain[x], range[x]]]]] // Reverse}
\]

\[
\text{Out[10]=} \text{or[equal[y, union[domain[x], range[x]]],}
\text{member[y, range[SINGLETON]], notsubclass[x, cart[y, y]]],}
\not[\text{subclass[cart[y, y], union[Id, x, inverse[x]]]]] = True}
\]

\[
\text{In[11]:=} \text{or[equal[y_, union[domain[x_], range[x_]]],}
\text{member[y_, range[SINGLETON]], notsubclass[x_, cart[y_, y_]]],}
\not[\text{subclass[cart[y_, y_], union[Id, x_, inverse[x_]]]]] := True}
\]
the case that $y$ is a singleton

When $x$ is irreflexive, then the case that $y$ is a singleton can only arise if $x = 0$.

**Lemma.**

\[
\text{In[12]} := \text{SubstTest[implies, subclass[x, set[t]], or[empty[x], equal[x, set[t]]], t \to \text{PAIR}[y, y]] // Reverse}
\]

\[
\text{Out[12]} = \text{or[equal[0, x], equal[x, cart[set[y], set[y]]], not[subclass[x, cart[set[y], set[y]]]] == True}
\]

\[
\text{In[13]} := (\% /. \{x \to x_, y \to y_\}) /. \text{Equal} \to \text{SetDelayed}
\]

**Lemma.**

\[
\text{In[14]} := \text{Map[not, SubstTest[and, implies[p1, or[p3, p4]], implies[and[p2, p3], p4], not[implies[and[p1, p2], p4], \{p1 \to \text{subclass[x, cartsq[set[y]]]}, p2 \to \text{empty[fix[x]]}, p3 \to \text{equal[x, cart[set[y], set[y]]]}, p4 \to \text{empty[x]}\}]] // Reverse}
\]

\[
\text{Out[14]} = \text{or[equal[0, x], not[equal[0, fix[x]]], not[subclass[x, cart[set[y], set[y]]]] == True}
\]

\[
\text{In[15]} := (\% /. \{x \to x_, y \to y_\}) /. \text{Equal} \to \text{SetDelayed}
\]

**Corollary.** An irreflexive relation contained in the cartesian square of a singleton is empty.

\[
\text{In[16]} := \text{Map[equal[V, \#] \&, SubstTest[class, t, or[equal[0, x], not[equal[0, fix[x]]]], not[equal[y, set[t]]], not[subclass[x, z]], z \to \text{cart[y, y]]}]
\]

\[
\text{Out[16]} = \text{or[equal[0, x], not[equal[0, fix[x]]]], not[member[y, range[SINGLETON]]], not[subclass[x, cart[y, y]]]] == True}
\]

\[
\text{In[17]} := \text{or[equal[0, x_], not[equal[0, fix[x_]]], not[member[y_, range[SINGLETON]]], not[subclass[x_, cart[y_, y_]]]]} := \text{True}
\]

**connex strict relation**

A **strict order** is an irreflexive transitive order.

**Theorem.** If $x$ is a strict order and $x \subset y \times y \subset \text{Id} \cup x \cup \text{inverse}[x]$, then $x \cup \text{id}[y]$ is a total order.
Lemma. A simplification rule.

In[20]:= SubstTest[or, not[equal[0, fix[t]]], not[subclass[t, cart[y, y]]],
not[subclass[cart[y, y], union[Id, t, inverse[t]]]],
TOTALORDER[union[t, id[y]], t -> restrict[x, y, y]]] // AssertTest

Out[20]= or[not[equal[0, union[Id, t, inverse[t]]]],
not[subclass[cart[y, y], union[Id, t, inverse[t]]]],
TOTALORDER[union[t, id[y]], t -> restrict[x, y, y]]]

In[21]:= subclass[cart[y, y], union[Id, composite[id[y]], x, id[y]]],
composite[id[y], inverse[x, id[y]]]] // AssertTest

Out[21]= subclass[cart[y, y],
union[Id, composite[id[y]], x, id[y]],
composite[id[y], inverse[x, id[y]]]]

Lemma.

In[22]:= SubstTest[or, not[equal[0, intersection[y, fix[x]]]]],
not[subclass[cart[y, y], union[Id, x, inverse[x]]]],
TOTALORDER[union[composite[cart[y, y], x, id[y]]],
inverse[x, id[y]]]] // Reverse

Out[22]= or[not[union[Id, x, inverse[x]]]],
not[subclass[cart[y, y], union[Id, x, inverse[x]]]],
TOTALORDER[union[composite[cart[y, y], x, id[y]]],
inverse[x, id[y]]]]

In[23]:= (% /. {x -> x_, y -> y_}) /. Equal -> SetDelayed

Corollary. If a class \(y\) is connected by a strict order \(x\), then \(id[x] \cup restrict[x, y, y]\) is a total order.

In[24]:= Map[not, SubstTest[and, implies[and[p1, p3],
implies[and[p2, p3], or[implies[and[p1, p2], p4]]],
tp1 -> STRICTORDER[x], p2 -> subclass[cart[y, y], union[Id, x, inverse[x]]]],
p3 -> STRICTORDER[restrict[x, y, y]],
p4 -> TOTALORDER[union[composite[cart[y, y], x, id[y]]], id[y]]]]] // Reverse

Out[24]= or[not[equal[0, fix[x]]]],
not[subclass[cart[y, y], union[Id, x, inverse[x]]]],
TOTALORDER[union[composite[cart[y, y], x, id[y]]], id[y]]]]]
Corollary. If a class $y$ is connected by a strict order $x$, then the cover relation of $x \cap (y \times y)$ is a function.