lexicographic product of well-founded relations

Johan G. F. Belinfante
2004 September 23

It is shown in this notebook that the lexicographic product of two well-founded relations is well-founded.

a basic lemma

The lemma derived in this section just says that if the domain of a relation is empty, then the relation itself is empty.

```wolfram
In[2]:= SubstTest[implies, and[equal[x, y], equal[y, z]], equal[x, z], \{y \rightarrow \text{composite[Id, x]}, z \rightarrow 0\}]
```

```wolfram
Out[2]= \text{or[equal[0, x], not[equal[0, domain[x]]], not[subclass[x, cart[V, V]]]]} = \text{True}
```

The following is a slight generalization of this fact.

```wolfram
In[4]:= Map[not, SubstTest[and, implies[p1, p3], implies[and[p2, p3], p4], not[implies[and[p1, p2], p4]], \{p1 \rightarrow \text{subclass[x, cart[y, z]]}, p2 \rightarrow \text{equal[0, domain[x]]}, p3 \rightarrow \text{subclass[x, cart[V, V]]}, p4 \rightarrow \text{equal[0, x]]}\}]
```

```wolfram
Out[4]= \text{or[equal[0, x], not[equal[0, domain[x]]], not[subclass[x, cart[y, z]]]]} = \text{True}
```
sethood lemmas

Lemma: If the domain of $x$ is contained in $y$, then $\text{image}[x, \text{complement}[y]]$ is empty, and is therefore a set.

A similar result:

The **lexicographic product** of two relations is defined as follows:

The following temporary abbreviation will be introduced to increase readability:

Note that the lexicographic product is a union of two cross-products:

When $x$ and $y$ are well-founded, each of these two cross-products is individually well-founded, but in general the union of two well-founded relations need not be well-founded.
founded. Nonetheless, it will be shown below that for the particular case of the lexicographic product, the union is well-founded.

derivation

Lemma.

```
In[13]:= SubstTest[implies, subclass[w, z], subclass[image[w, t], image[z, t]],
   {t -> complement[x], z -> union[cart[x, V], composite[y, w]]}]
```

```
Out[13]= or[not[subclass[w, union[cart[x, V], composite[y, w]]]],
   subclass[image[w, complement[x]], image[y, image[w, complement[x]]]]] = True
```

```
In[14]:= (% /. {w -> w_, x -> x_, y -> y_}) /. Equal -> SetDelayed
```

The definition of well-foundedness implies the following:

```
In[15]:= SubstTest[member, x, subvar[t], t -> wf[y]] // Reverse
```

```
Out[15]= and[member[x, V], subclass[x, image[wf[y], x]]] = equal[0, x]
```

```
In[16]:= and[member[x_, V], subclass[x_, image[wf[y_], x_]]] := equal[0, x]
```

The following corollary follows because \( \text{wf}[y] \) is well-founded.

```
In[17]:= (implies[member[w, subvar[lex[wf[x], wf[y]]]],
   member[image[w, complement[lex[wf[x], domain[w]]]], subvar[t]]] //
   NotNotTest) /. t -> wf[y]
```

```
Out[17]= or[not[member[w, V]], not[
   subclass[w, union[cart[image[wf[x], domain[w]], V], composite[wf[y], w]]]],
   subclass[domain[w], image[wf[x], domain[w]]]] = True
```

```
In[18]:= (% /. {w -> w_, x -> x_, y -> y_}) /. Equal -> SetDelayed
```

Using also the well-foundedness of \( \text{wf}[x] \), one now deduces:

```
In[19]:= Map[not,
   SubstTest[and, implies[p1, p2], implies[p1, p3], implies[and[p2, p3], p4],
   not[implies[p1, p4]], {p1 -> member[w, subvar[lex[wf[x], wf[y]]]]},
   p2 -> subvariant[wf[x], domain[w]],
   p3 -> member[domain[w], V], p4 -> equal[0, domain[w]]]]
```

```
Out[19]= or[equal[0, domain[w]], not[member[w, V]], not[subclass[w,
   union[cart[image[wf[x], domain[w]], V], composite[wf[y], w]]]]] = True
```

```
In[20]:= (% /. {w -> w_, x -> x_, y -> y_}) /. Equal -> SetDelayed
```
Since \( w \) is a relation, the conclusion \( \text{equal}[0, \text{domain}[w]] \) can be sharpened to \( \text{equal}[0, w] \).

\[
\begin{align*}
\text{In}[21] & := \text{Map}[\text{not}, \text{SubstTest}[\text{and}, \text{implies}[p1, p2], \text{implies}[p1, p3], \\
& \quad \text{implies}[p1, p4], \text{implies}[\text{and}[p2, p3], p5], \text{implies}[\text{and}[p4, p5], p6], \\
& \quad \text{not}[\text{implies}[p1, p6]], [p1 \to \text{member}[w, \text{subvar}[\text{lex}[w, x], w[y]]]), \\
& \quad p2 \to \text{subvariant}[w[x], \text{domain}[w]], p3 \to \text{member}[\text{domain}[w], V], \\
& \quad p4 \to \text{subclass}[w, \text{cart}[V, V]], p5 \to \text{equal}[0, \text{domain}[w]], p6 \to \text{equal}[0, w]])
\end{align*}
\]

\[
\text{Out}[21] := \text{or}[\text{equal}[0, w], \text{not}[\text{member}[w, V]], \text{not}[\text{subclass}[w, \\
& \quad \text{union}[\text{cart}[\text{image}[w[x], \text{domain}[w]], V], \text{composite}[w[y], w[\_]])]] \Rightarrow \text{True}
\]

\[
\text{In}[22] := (\% / \{w \mapsto w[\_], x \mapsto x[\_], y \mapsto y[\_]\}) / \text{Equal} \Rightarrow \text{SetDelayed}
\]

Removing the variable \( w \) yields the main result.

\[
\begin{align*}
\text{In}[23] & := \text{Map}[\text{equal}[V, \#] \&, \text{SubstTest}[\text{class}, w, \text{implies}[\text{member}[w, z], \text{equal}[0, w]], \\
& \quad z \to \text{subvar}[\text{lex}[w[x], w[y]]])] // \text{Reverse}
\end{align*}
\]

\[
\text{Out}[23] := \text{WELLO\textsc{d\textregistered}N\textsc{d\textregistered}ED}[ \\
& \quad \text{union}[\text{composite}[\text{inverse}[\text{FIRST}], w[x], \text{FIRST}, \text{cross}[\text{Id}, w[y]]]] \Rightarrow \text{True}
\]

\[
\text{In}[24] := \text{WELLO\textsc{d\textregistered}N\textsc{d\textregistered}ED}[ \\
& \quad \text{union}[\text{composite}[\text{inverse}[\text{FIRST}], w[x\_], \text{FIRST}, \text{cross}[\text{Id}, w[y\_]]]] := \text{True}
\]

The wrappers can be removed, if desired:

\[
\begin{align*}
\text{In}[25] & := \text{SubstTest}[\text{implies}, \text{and}[\text{equal}[u, w[x]], \text{equal}[v, w[y]]], \text{WELLO\textsc{d\textregistered}N\textsc{d\textregistered}ED}[ \\
& \quad \text{union}[\text{composite}[\text{inverse}[\text{FIRST}], u, \text{FIRST}, \text{cross}[\text{Id}, v]]], (u \mapsto x, v \mapsto y)]
\end{align*}
\]

\[
\text{Out}[25] := \text{or}[\text{not}[\text{WELLO\textsc{d\textregistered}N\textsc{d\textregistered}ED}[x]], \text{not}[\text{WELLO\textsc{d\textregistered}N\textsc{d\textregistered}ED}[y]], \text{WELLO\textsc{d\textregistered}N\textsc{d\textregistered}ED}[ \\
& \quad \text{union}[\text{composite}[\text{inverse}[\text{FIRST}], x, \text{FIRST}, \text{cross}[\text{Id}, y]]]]]] := \text{True}
\]

\[
\text{In}[26] := \text{or}[\text{not}[\text{WELLO\textsc{d\textregistered}N\textsc{d\textregistered}ED}[x\_]], \text{not}[\text{WELLO\textsc{d\textregistered}N\textsc{d\textregistered}ED}[y\_]], \text{WELLO\textsc{d\textregistered}N\textsc{d\textregistered}ED}[ \\
& \quad \text{union}[\text{composite}[\text{inverse}[\text{FIRST}], x\_, \text{FIRST}, \text{cross}[\text{Id}, y\_]]]]]] := \text{True}
\]

---

some comments

When \( x \) and \( y \) are sets, the lexicographic product, as we have defined it, generally fails to be a set because of the presence of the global identity relation \( \text{Id} \).

\[
\begin{align*}
\text{In}[27] & := \text{member}[\text{lex}[\text{setpart}[x], \text{setpart}[y]], V] // \text{assert}
\end{align*}
\]

\[
\text{Out}[27] := \text{and}[\text{equal}[0, \text{domain}[\text{setpart}[x]]], \text{equal}[0, \text{domain}[\text{setpart}[y]]])]
\]
Since any subclass of a well-founded relation is well-founded, one can replace the global identity by any restriction of it, and the theorem remains true.

In[28]:=
\text{subclass[cross[id[z], y], union[x, cross[Id, y]]]} \quad \text{// AssertTest}

Out[28]= \text{subclass[cross[id[z], y], union[x, cross[Id, y]]]} = \text{True}

In[29]:=
\text{(% /. \{x \rightarrow x\_, y \rightarrow y\_, z \rightarrow z\_\}) \quad \text{// Equal} \rightarrow \text{SetDelayed}

In[30]:=
\text{SubstTest[implies, and[subclass[u, v], WELLFOUNDED[v]], WELLFOUNDED[u],}
\quad \{u \rightarrow \text{union[composite[inverse[FIRST], wf[x], FIRST], cross[id[z], wf[y]]]},
\quad v \rightarrow \text{union[composite[inverse[FIRST], wf[x], FIRST], cross[Id, wf[y]]]})\}}

Out[30]= \text{WELLFOUNDED[}

\quad \text{union[composite[inverse[FIRST], wf[x], FIRST], cross[id[z], wf[y]]]} = \text{True}

In[31]:=
\text{WELLFOUNDED[union[}
\quad \text{composite[inverse[FIRST], wf[x\_], FIRST], cross[id[z\_], wf[y\_]]]]} = \text{True}

Further such generalizations are possible: one could, for example, replace \text{composite[inverse[FIRST], wf[x], FIRST]} with \text{cross[wf[x], t]}, where \text{t} is arbitrary. The theorem itself is then the special case \text{t = V}.