pairset rules

Johan G. F. Belinfante
2004 April 10

In[1]:= << goedel56.09a << tools.m;

Package Title: goedel56.09a 2004 April 9 at 3:30 p.m.
It is now: 2004 Apr 12 at 8:20
Loading Simplification Rules
TOOLS.M Revised 2004 April 6
weightlimit = 40

summary

The rules for pairset derived in this notebook were inspired by various theorems that had previously been proved using Otter.

three equality substitution rules for pairset

In this section, three equality substitution rules for pairset are derived. The substitution rule which is easiest to derive is the following, and it is also the most general of the three:

In[2]:= SubstTest[implies, and[equal[s, t], equal[w, z]], equal[union[s, w], union[t, z]], 
{s -> singleton[u], t -> singleton[x], w -> singleton[v], z -> singleton[y]}]

Out[2]= or[equal[pairset[u, v], pairset[x, y]]], not[equal[singleton[u], singleton[x]]], 
not[equal[singleton[v], singleton[y]]] := True

In[3]:= or[equal[pairset[u_, v_], pairset[x_, y_]]], not[equal[singleton[u_], singleton[x_]]], 
not[equal[singleton[v_], singleton[y_]]] := True

Corollary UP−SS−4 in the UP−2 group is a special case of the following mixed substitution rule, and was the inspiration for this more general substitution rule.

In[4]:= Map[not, SubstTest[and, implies[p1, p3], 
implies[and[p2, p3], p4], not[implies[and[p1, p2], p4]], 
{p1 -> equal[u, x], p2 -> equal[singleton[v], singleton[y]], 
p3 -> equal[singleton[u], singleton[x]], 
p4 -> equal[pairset[u, v], pairset[x, y]]}]]

Out[4]= or[equal[pairset[u, v], pairset[x, y]]], 
not[equal[u, x]], not[equal[singleton[v], singleton[y]]] := True

In[5]:= or[equal[pairset[u_, v_], pairset[x_, y_]], 
not[equal[u_, x_]], not[equal[singleton[v_], singleton[y_]]]] := True

The final substitution rule has no singleton wrappers in the hypotheses.
a lemma

In this section a general result is derived that will be needed in the next section.

\[
\text{equiv[andsubclass[w, x], subclass[w, union[x, y]], subclass[w, x]]}
\]

\text{Lemma.}

\[
\text{subclass[w, union[x, intersection[y, image[V, z]]]] // AssertTest}
\]

\text{results inspired by Theorem CP–UP–SU}

Theorem CP–UP–SU in the CP–SS group was the inspiration for the results derived in this section, but the only specific fact about \text{pairset} that will be needed is the following lemma:

\[
\text{image[pairset[x, y], z] // Normality}
\]

The next result derived in the preceding section does not say anything in particular about pairsets, but note that the quantities that appear in it resemble those appearing on the right hand side of the preceding rewrite rule. Nothing further is needed for the GOEDEL program to recognize the truth of Theorem CP–UP–SU:

\[
\text{or[not[member[pair[x, y], cart[V, V]]], subclass[cart[singleton[z], pairset[x, y]], pairset[pair[z, x], pair[z, y]]]}
\]
functions with one or two points

Any relation with just one point is a function. For functions with two points, the following variable-free result holds:

```
In[15]:= image[inverse[PAIRSET], FUNS] // ReinNormality
```

In[16]:= image[inverse[PAIRSET], FUNS] :=
    composite[id[cart[V, V]], complement[cross[Id, Di]], id[cart[V, V]]]

regular pairs

The singleton of a regular set is regular.

```
In[17]:= image[inverse[SINGLETON], REGULAR] // Normality
```

In[18]:= image[inverse[SINGLETON], REGULAR] := REGULAR

A similar result hold for pairsets. This is Theorem REG-13 in the REG-2 group:

```
In[19]:= Map[implies[member[PAIR[x, y], cart[REGULAR, REGULAR]], #] &,
    SubstTest[member, union[u, v], REGULAR, {u -> singleton[x], v -> singleton[y]}]]
Out[19]= or[member[PAIR[x, y], REGULAR],
    not[member[x, REGULAR]], not[member[y, REGULAR]]] := True
```

In[20]:= or[member[PAIR[x_, y_], REGULAR],
    not[member[x_, REGULAR]], not[member[y_, REGULAR]]] := True

A variable-free version of this can be derived:

```
In[21]:= IminComp[CUP, cross[SINGLETON, SINGLETON], REGULAR]
```

In[22]:= image[inverse[PAIRSET], REGULAR] := cart[REGULAR, REGULAR]

a new vertical section rule

The GOEDEL program contains two ordered pairs, called pair[x,y] and PAIR[x,y] which coincide only when x and y are both sets. Since pair[x,y] is in the domain of the function PAIRSET only when both x and y are sets, it follows that the vertical sections of PAIRSET must be identical for pair[x,y] and PAIR[x,y]. This fact can be explicitly verified using assert:

```
up-rules.nb 3
```
This fact will be added as a new rewrite rule.

With the addition of this new rewrite rule, the GOEDEL program recognizes the truth of Theorem PRS–AP2 in the PRS group: