invariant classes and transitive closure

Johan G. F. Belinfante
2004 August 27

```
In[1]:= SetDirectory["i:"]; << goedel60.26a; << tools.m;
```

```
:Package Title: goedel60.26a          2004 August 26 at 3:33 p.m.

It is now:  2004 Aug 27 at 2:24

Loading Simplification Rules

TOOLS.M                       Revised 2004 August 11

weightlimit = 40
```

---

summary

Let us say that **x** conducts **y** to a class **z** if **subclass[image[x, y], z]**. A class is **invariant** under **x** if **x** conducts the class to itself. It is shown in this notebook that if **x** conducts **y** to a class **z** which is invariant under **x**, then the transitive closure of **x** also conducts **y** to **z**. The main obstacle to deriving this intuitively obvious fact is that the images of the transitive closure of a relation **x** are closely related to the range of **iterate[x,y]**. Some new simplification rules are derived to cope with this connection.

---

derivation

Lemma 1.

```
In[2]:= ImageComp[union[Id, trv[x]], x, y] // Reverse
```

```
Out[2]= image[x, range[iterate[x, y]]] == image[trv[x], y]
```

```
In[3]:= image[x_, range[iterate[x_, y_]]] := image[trv[x], y]
```

Lemma 2.

```
In[4]:= SubstTest[subclass, union[u, v], z, {u → z, v → image[trv[x], z]}]
```

```
Out[4]= subclass[range[iterate[x, z]], z] == subclass[image[x, z], z]
```

```
In[5]:= subclass[range[iterate[x_, z_]], z_] := subclass[image[x, z], z]
```

Temporary result.

```
In[6]:= SubstTest[implies, subclass[u, v], subclass[image[w, u], image[w, v]],
         {u → image[x, y], v → z, w → union[Id, trv[x]]}]

Out[6]= or[not[subclass[image[x, y], z]],
          subclass[image[trv[x], y], range[iterate[x, z]]]] == True

In[7]:= (% /. {x → x_, y → y_, z → z_}) /. Equal → SetDelayed
```

Temporary result.

```
In[8]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
         {u → image[trv[x], y], v → range[iterate[x, z]], w → z}]

Out[8]= or[not[subclass[image[x, z], z]],
          not[subclass[image[trv[x], y], range[iterate[x, z]]]],
          subclass[image[trv[x], y], z]] == True

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

Main theorem.

```
In[10]:= Map[not, SubstTest[and, implies[p1, p3],
           implies[and[p2, p3], p4], not[implies[and[p1, p2], p4]],
           {p1 → subclass[image[x, y], z], p2 → subclass[image[x, z], z],
            p3 → subclass[image[trv[x], y], range[iterate[x, z]]],
            p4 → subclass[image[trv[x], y], z]}]]

Out[10]= or[not[subclass[image[x, y], z]],
           not[subclass[image[x, z], z]], subclass[image[trv[x], y], z]] == True

In[11]:= or[not[subclass[image[x_, y_], z_]],
           not[subclass[image[x_, z_], z_]], subclass[image[trv[x_], y_], z_]] := True
```

---

serendipity

In the course of deriving the main result, the following additional fact was derived.

```
In[12]:= SubstTest[implies, subclass[u, v], subclass[image[w, u], image[w, v]],
          {u → complement[singleton[0]], v → V, w → iterate[x, y]}]

Out[12]= subclass[image[trv[x], y], range[iterate[x, y]]] == True

In[13]:= subclass[image[trv[x_], y_], range[iterate[x_, y_]]] := True
```