iterate and image

Johan G. F. Belinfante
2012 December 18

```
In[1]:= SetDirectory["l:"]; << goedel.12dec16a

        :Package Title: goedel.12dec16a            2012 December 16 at 2:30 a.m.

        Loading takes about sixteen minutes, half that time due to builtin pauses.

        It is now:  2012 Dec 18 at 16:25

        Loading Simplification Rules

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

        weightlimit = 40

        Loading completed.

        It is now:  2012 Dec 18 at 16:42
```

summary

An intertwine equation is derived involving **image** and **iterate**.

derivation

Lemma.

```
In[4]:= SubstTest[implies, equal[u, v],
          equal[image[u, z], image[v, z]], {u -> composite[image[power[w], t], x],
           v -> composite[x, image[power[y], t]]}] // Reverse

Out[4]= or[equal[image[x, image[iterate[y, z], t]], image[iterate[w, image[x, z]], t]],
          not[equal[composite[x, image[power[y], t]], composite[image[power[w], t], x]]]] == True

In[5]:= (% /. {t → t_, w → w_, x → x_, y → y_, z → z_}) /. Equal → SetDelayed
```

Lemma.

```
In[6]:=  Map[not, SubstTest[and, implies[p1, p2], implies[p2, p3],
            not[implies[p1, p3]], {p1 -> equal[composite[w, x], composite[x, y]],
             p2 → equal[composite[image[power[w], t], x], composite[x, image[power[y], t]]],
             p3 → equal[image[x, image[iterate[y, z], t]],
               image[iterate[w, image[x, z]], t]]}]] // Reverse

Out[6]=  or[equal[image[x, image[iterate[y, z], t]], image[iterate[w, image[x, z]], t]],
           not[equal[composite[w, x], composite[x, y]]]] == True

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

The variable **t** can be eliminated using **reify** and **case**.

Theorem. An intertwine equation involving **image** and **iterate**.

```
In[8]:=  Map[equal[V, domain[#]] &,
          SubstTest[reify, t, case[implies[equal[u, v], equal[image[x, image[iterate[y, z], t]],
              image[iterate[w, image[x, z]], t]]]], {u → composite[w, x], v → composite[x, y]}]]

Out[8]=  or[equal[composite[x, iterate[y, z]], iterate[w, image[x, z]]],
           not[equal[composite[w, x], composite[x, y]]]] == True

In[10]:=  or[equal[composite[x_, iterate[y_, z_]], iterate[w_, image[x_, z_]]],
            not[equal[composite[w_, x_], composite[x_, y_]]]] := True
```