unions of chains of binary operations

Johan G. F. Belinfante
2007 September 10

```
In[1]:= SetDirectory["l:"]; << goedel97.10a; << tools.m

        :Package Title: goedel97.10a          2007 September 10 at 12:05 noon

        It is now:  2007 Sep 10 at 15:0

        Loading Simplification Rules

        TOOLS.M                                      Revised 2007 June 25

        weightlimit = 40
```

summary

The union of a chain of binary operations is a binary operation. This theorem is derived in this notebook as a corollary of a more general theorem about transforms of chains.

transforms

Lemma. (Corollary of an existing rewrite rule.)

```
In[2]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
          {u → image[IMAGE[IMAGE[thinpart[x]]], intersection[chains[S], y]],
           v → image[IMAGE[IMAGE[thinpart[x]]], chains[S]], w → chains[S]}] // Reverse

Out[2]= subclass[image[IMAGE[IMAGE[thinpart[x]]], intersection[y, chains[S]]], chains[S]] == True

In[3]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed

Lemma.

In[4]:= SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
          {t → BIGCUP, u → image[IMAGE[IMAGE[thinpart[x]]], intersection[chains[S], P[y]]],
           v -> intersection[chains[S], P[image[IMAGE[thinpart[x]], y]]]}] // Reverse

Out[4]= subclass[
          image[BIGCUP, image[IMAGE[IMAGE[thinpart[x]]], intersection[chains[S], P[y]]]],
          Uchains[image[IMAGE[thinpart[x]], y]]] == True

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

Theorem.
```

```
In[6]:= Map[subclass[#, Uchains[image[IMAGE[thinpart[x]], y]]] &,
          ImageComp[BIGCUP, IMAGE[IMAGE[thinpart[x]]], intersection[chains[S], P[y]]]]

Out[6]= subclass[image[IMAGE[thinpart[x]], Uchains[y]],
          Uchains[image[IMAGE[thinpart[x]], y]]] == True

In[7]:= subclass[image[IMAGE[thinpart[x_]], Uchains[y_]],
          Uchains[image[IMAGE[thinpart[x_]], y_]]] := True
```

inverse transforms

Lemma.

```
In[8]:= SubstTest[implies, subclass[u, v], subclass[Uchains[u], Uchains[v]],
          {u → intersection[x, y], v → x}] // Reverse

Out[8]= subclass[Uchains[intersection[x, y]], Uchains[x]] == True

In[9]:= subclass[Uchains[intersection[x_, y_]], Uchains[x_]] := True
```

Lemma.

```
In[10]:= SubstTest[subclass, image[IMAGE[thinpart[x]], Uchains[t]],
           Uchains[image[IMAGE[thinpart[x]], t]],
           t → image[inverse[IMAGE[thinpart[x]]], y]] // Reverse

Out[10]= subclass[image[IMAGE[thinpart[x]], Uchains[image[inverse[IMAGE[thinpart[x]]], y]]],
           Uchains[intersection[y, range[IMAGE[x]]]]] == True

In[11]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Corollary.

```
In[12]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
           {u -> Uchains[image[inverse[IMAGE[thinpart[x]]], y]], v ->
             image[inverse[IMAGE[thinpart[x]]], Uchains[intersection[y, range[IMAGE[x]]]]],
            w -> image[inverse[IMAGE[thinpart[x]]], Uchains[y]]}] // Reverse

Out[12]= subclass[image[IMAGE[thinpart[x]],
            Uchains[image[inverse[IMAGE[thinpart[x]]], y]]], Uchains[y]] == True

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

Uchains[BINOPS]

Application.

```
In[14]:= SubstTest[subclass, Uchains[image[inverse[IMAGE[thinpart[x]]], y]],
           image[inverse[IMAGE[thinpart[x]]], Uchains[y]],
           {x → FIRST, y → image[CART, Id]}] // Reverse

Out[14]= subclass[image[IMAGE[FIRST], Uchains[image[inverse[IMAGE[FIRST]], image[CART, Id]]]],
           image[CART, Id]] == True

In[15]:= % /. Equal → SetDelayed
```

Theorem.

```
In[16]:= SubstTest[and, subclass[u, v], subclass[v, u],
           {u -> Uchains[image[inverse[IMAGE[FIRST]], image[CART, Id]]],
            v -> image[inverse[IMAGE[FIRST]], image[CART, Id]]}]

Out[16]= equal[image[inverse[IMAGE[FIRST]], image[CART, Id]],
           Uchains[image[inverse[IMAGE[FIRST]], image[CART, Id]]]] == True

In[17]:= Uchains[image[inverse[IMAGE[FIRST]], image[CART, Id]]] :=
           image[inverse[IMAGE[FIRST]], image[CART, Id]]
```

Lemma.

```
In[18]:= Map[complement, image[inverse[DORA],
             composite[inverse[S], IMAGE[inverse[DUP]]]] // Normality] // Reverse

Out[18]= image[inverse[DORA], composite[complement[inverse[S]], IMAGE[inverse[DUP]]]] ==
           complement[image[inverse[DORA], composite[inverse[S], IMAGE[inverse[DUP]]]]]

In[19]:= image[inverse[DORA], composite[complement[inverse[S]], IMAGE[inverse[DUP]]]] :=
           complement[image[inverse[DORA], composite[inverse[S], IMAGE[inverse[DUP]]]]]
```

Theorem.

```
In[20]:= transvar[SECOND, composite[inverse[DUP], FIRST]] // Normality

Out[20]= transvar[SECOND, composite[inverse[DUP], FIRST]] ==
           image[inverse[DORA], composite[inverse[S], IMAGE[inverse[DUP]]]]

In[21]:= transvar[SECOND, composite[inverse[DUP], FIRST]] :=
           image[inverse[DORA], composite[inverse[S], IMAGE[inverse[DUP]]]]

In[22]:= SubstTest[Uclosure, transvar[x, y],
           {x → SECOND, y → composite[inverse[DUP], FIRST]}] // Reverse

Out[22]= Uclosure[image[inverse[DORA], composite[inverse[S], IMAGE[inverse[DUP]]]]] ==
           image[inverse[DORA], composite[inverse[S], IMAGE[inverse[DUP]]]]

In[23]:= Uclosure[image[inverse[DORA], composite[inverse[S], IMAGE[inverse[DUP]]]]] :=
           image[inverse[DORA], composite[inverse[S], IMAGE[inverse[DUP]]]]
```

Lemma.

```
In[24]:= SubstTest[implies, and[equal[u, Uchains[u]], equal[v, Uchains[v]]],
           equal[intersection[u, v], Uchains[intersection[u, v]]],
           {u → FUNS, v -> image[inverse[IMAGE[FIRST]], image[CART, Id]]}] // Reverse

Out[24]= equal[intersection[FUNS, image[inverse[IMAGE[FIRST]], image[CART, Id]]],
           Uchains[intersection[FUNS, image[inverse[IMAGE[FIRST]], image[CART, Id]]]]] == True

In[25]:= % /. Equal → SetDelayed
```

Lemma.

```
In[26]:= SubstTest[Uchains, Uclosure[t],
           t -> image[inverse[DORA], composite[inverse[S], IMAGE[inverse[DUP]]]]] // Reverse

Out[26]= Uchains[image[inverse[DORA], composite[inverse[S], IMAGE[inverse[DUP]]]]] ==
          image[inverse[DORA], composite[inverse[S], IMAGE[inverse[DUP]]]]

In[27]:= % /. Equal → SetDelayed
```

Main Theorem. The class of binary operations is closed under unions of chains.

```
In[28]:= SubstTest[implies, and[equal[u, Uchains[u]], equal[v, Uchains[v]]],
           equal[intersection[u, v], Uchains[intersection[u, v]]],
           {u → intersection[FUNS, image[inverse[IMAGE[FIRST]], image[CART, Id]]],
            v -> image[inverse[DORA], composite[inverse[S], IMAGE[inverse[DUP]]]]}] // Reverse

Out[28]= equal[BINOPS, Uchains[BINOPS]] == True

In[29]:= Uchains[BINOPS] := BINOPS
```