a twist relation for binary operations

Johan G. F. Belinfante
2012 June 24

In[1]:= SetDirectory["l:"]; << goedel.12jun18a

:Package Title: goedel.12jun18a 2012 June 18 at 2:40 p.m.

Loading takes about seventeen minutes, half that time due to built-in pauses.

It is now: 2012 Jun 24 at 6:41

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 Jun 24 at 6:58

summary

The equivalence relation EQUIDIFF figures prominently in constructing the integers from the natural numbers. In this notebook a small part of that theory is studied in a more general setting. The relation studied here need not in general be an equivalence, and no embedding theorem for general binary operations is derived.

a general twist formula

The general twist formula derived in this section involves two binary operations, as well as their Curried counterparts.

Lemma. Simplification rule.

In[2]:= SubstTest[composite, intersection[composite[inverse[FIRST], flip[t]],
 composite[inverse[SECOND], SECOND]], inverse[FIRST], t -> flip[x]] // Reverse

Out[2]= composite[
 intersection[composite[inverse[FIRST], x], composite[inverse[SECOND], SECOND]],
 inverse[FIRST]] == inverse[rotate[composite[x, SWAP]]]

In[3]:= composite[
 intersection[composite[inverse[FIRST], x_], composite[inverse[SECOND], SECOND]],
 inverse[FIRST]] := inverse[rotate[composite[x, SWAP]]]

Lemma. Simplification rule.
In[4]:= SubstTest[twist, twist[t]], 
   t -> composite[RIF, cross[inverse[rotate[x]], composite[SWAP, inverse[rotate[y]]]]]]
Out[4]= composite[RIF, cross[inverse[rotate[x]], composite[SWAP, inverse[rotate[y]]]]] := 
   inverse[twist[composite[inverse[y], x]]]

In[5]:= composite[RIF, cross[inverse[rotate[x_]], composite[SWAP, inverse[rotate[y_]]]]] := 
   inverse[twist[composite[inverse[y], x]]]

Theorem. A general twist formula.

In[6]:= composite[inverse[E], COMPOSE, cross[composite[INVERSE, APPLY[CURRY, binop[x]]], 
   APPLY[CURRY, binop[y]]]] // FastReifTriNormality
Out[6]= composite[inverse[E], COMPOSE, 
   cross[composite[INVERSE, APPLY[CURRY, binop[x]]], APPLY[CURRY, binop[y]]]] := 
   composite[SWAP, twist[composite[inverse[binop[y]], binop[x]]]]

In[7]:= composite[inverse[E], COMPOSE, 
   cross[composite[INVERSE, APPLY[CURRY, binop[x_]]], APPLY[CURRY, binop[y_]]]] := 
   composite[SWAP, twist[composite[inverse[binop[y]], binop[x]]]]

a special case

In this section the special case x = y = NATADD is studied. The Curried version of NATADD is PLUS.

In[8]:= APPLY[CURRY, NATADD]
Out[8]= PLUS

The left hand side of the twist formula reduces to EQUIDIFF.

In[9]:= composite[inverse[E], COMPOSE, cross[composite[INVERSE, PLUS[x], PLUS]]] 
Out[9]= EQUIDIFF

The right hand side also reduces to EQUIDIFF.

In[10]:= composite[SWAP, twist[composite[inverse[NATADD], NATADD]]] 
Out[10]= EQUIDIFF

a simplification rule

In this section it is shown that the function INVERSE in twist formula derived in the preceding section could be replaced with the function IMAGE[SWAP]. A simplification rule is derived that makes it unnecessary to retain two separate twist formulas.
Theorem. Simplification rule.

\[\text{In}[11]:= \text{Assoc[IMAGE[SWAP], id[P[cart[V, V]]], APPLY[CURRY, binop[x]]]}\]

\[\text{Out}[11]= \text{composite[IMAGE[SWAP], APPLY[CURRY, binop[x]]]} \Rightarrow \text{composite[INVERSE, APPLY[CURRY, binop[x]]]}\]

\[\text{In}[12]:= \text{composite[IMAGE[SWAP], APPLY[CURRY, binop[x_]]]} := \text{composite[INVERSE, APPLY[CURRY, binop[x]]]}\]

The same simplification can be made when the binary operation is replaced with its flip.

Corollary. A dual simplification rule.

\[\text{In}[13]:= \text{SubstTest[composite, IMAGE[SWAP], APPLY[CURRY, binop[t]], t \to \text{flip[binop[x]]] // Reverse}\}

\[\text{Out}[13]= \text{composite[IMAGE[SWAP], APPLY[CURRY, composite[binop[x], SWAP]]]} \Rightarrow \text{composite[INVERSE, APPLY[CURRY, composite[binop[x], SWAP]]]}\]

\[\text{In}[14]:= \text{composite[IMAGE[SWAP], APPLY[CURRY, composite[binop[x_], SWAP]]]} := \text{composite[INVERSE, APPLY[CURRY, composite[binop[x], SWAP]]]}\]

\section*{an inclusion}

In this section an inclusion is derived for the special case that the two binary operations are related by \textit{flip}.

Lemma. Simplification rule.

\[\text{In}[18]:= \text{(fix[composite[SWAP, twist[composite[inverse[binop[x]], binop[y]]]]]} // \text{RelnNormality} /. y \to \text{flip[binop[x]]}\]

\[\text{Out}[18]= \text{fix[composite[SWAP, twist[composite[inverse[binop[x]], binop[x], SWAP]]]} \Rightarrow \text{domain[binop[x]]}\]

\[\text{In}[19]= \text{domain[binop[x]]}\]

Theorem. An inclusion.

\[\text{In}[20]:= \text{SubstTest[subclass, domain[funpart[t]], fix[composite[inverse[E], funpart[t]]], t \to \text{composite[COMPOSE, cross[composite[INVERSE, APPLY[CURRY, binop[t]]]]]} \Rightarrow \text{composite[INVERSE, APPLY[CURRY, binop[x]]]} // . t \to \text{flip[binop[x]]}\}

\[\text{Out}[20]= \text{subclass[composite[COMPOSE, cross[composite[INVERSE, APPLY[CURRY, composite[binop[x], SWAP]]]], APPLY[CURRY, binop[x]]], E] = True}\]

\[\text{In}[21]= \text{subclass[composite[COMPOSE, cross[composite[INVERSE, APPLY[CURRY, composite[binop[x_], SWAP]]], APPLY[CURRY, binop[x_]]], E] = True}\]
Corollary. A dual result.

\[
\text{In[22]:=} \quad \text{SubstTest[\text{subclass, composite}]\{\text{COMPOSE, cross[composite[\text{INVERSE, APPLY[CURRY, composite[binop[t], SWAP]]}], APPLY[CURRY, binop[t]]]], E, t \to \text{flip[binop[x]]}}} \quad \text{// Reverse}
\]

\[
\text{Out[22]=} \quad \text{subclass[composite[COMPOSE, cross[composite[\text{INVERSE, APPLY[CURRY, binop[x]]}], APPLY[CURRY, composite[binop[x], SWAP]]]], E] \Rightarrow \text{True}}
\]

\[
\text{In[23]:=} \quad \text{SubstTest[\text{subclass, composite}]\{\text{COMPOSE, cross[composite[\text{INVERSE, APPLY[CURRY, binop[x]]}], APPLY[CURRY, composite[binop[x], SWAP]]]], E] \Rightarrow \text{True}}
\]

Corollary. A special case of interest.

\[
\text{In[24]:=} \quad \text{SubstTest[\text{subclass, composite}]\{\text{COMPOSE, cross[composite[\text{INVERSE, APPLY[CURRY, composite[binop[t], SWAP]]}], APPLY[CURRY, binop[t]]]], E, t \to \text{NATADD}}} \quad \text{// Reverse}
\]

\[
\text{Out[24]=} \quad \text{subclass[composite[VERTSECT[EQUIDIFF], id[cart[omega, omega]]]], E] \Rightarrow \text{True}}
\]

\[
\text{In[25]:=} \quad \text{subclass[composite[VERTSECT[EQUIDIFF], id[cart[omega, omega]]]], E] \Rightarrow \text{True}}
\]

Comment. The above result could also have been deduced from the following.

\[
\text{In[26]:=} \quad \text{composite[\text{id[Z]}, E]}
\]

\[
\text{Out[26]=} \quad \text{composite[VERTSECT[EQUIDIFF], id[cart[omega, omega]]]}
\]