weak vs strong inequalities

Johan G. F. Belinfante
2002 October 14

```
<< goedel52.p88; << tools.m
```

```
:Package Title: goedel52.p88          2002 October 14 at 9:48 a.m.

It is now:  2002 Oct 14 at 13:9

Loading Simplification Rules

TOOLS.M                   Revised 2002 September 16

weightlimit = 40
```

■ Summary

Any weak inequality for natural numbers can be converted to a strong inequality by negation: A natural number **m** is less than or equal to a natural number **n** if and only if **n** is not strictly greater than **m**. Quaife chose to write most arithmetical inequalities as strict ones.

```
Art Quaife, Automated Development of Fundamental Mathematical Theories,
Kluwer Academic Publishers, Dordrecht, 1992. (See page 71.)
```

When working with the **GOEDEL** program, working with weak inequalities is generally easier, especially when relational variants are studied. The presence of the membership relation **E** often produces expressions involving **IMAGE**, whereas this is not the case for the subset relation **S**.

```
composite[E, NATADD]
```

```
composite[inverse[IMAGE[inverse[NATADD]]], E]
```

```
composite[NATADD, inverse[E]]
```

```
composite[inverse[E], IMAGE[NATADD]]
```

Nonetheless there may be an advantage to using strict inequalities because doing so automatically adds set−hoodhypotheses that may simplify some formulas. In this notebook we study one example, related to Quaife's theorem **(O14)**. (See op.cit., page 189.) The statement of this theorem using weak inequalities is this:

```
subclass[x, natadd[x, y]]
```

```
True
```

■ conversion to strong form

The conversion to strong form via negation uses this rewrite rule:

```
composite[id[omega], complement[E], id[omega]]
```

```
composite[id[omega], inverse[S], id[omega]]
```

Note that using this rule does add extra hypotheses, which will shortly be eliminated:

```
SubstTest[member, pair[u, v], intersection[cart[w, w], complement[E]],
    {u -> natadd[x, y], v -> x, w -> omega}]
```

```
and[member[x, omega], member[y, omega]] ==
 and[member[x, omega], member[y, omega], not[member[natadd[x, y], x]]]
```

Transpose everything to one side:

```
Map[implies[First[%], #] &, %] // Reverse
```

```
or[not[member[x, omega]], not[member[y, omega]], not[member[natadd[x, y], x]]] == True
```

This temporary rule will be improved below:

```
or[not[member[x_, omega]],
  not[member[y_, omega]], not[member[natadd[x_, y_], x_]]] := True
```

Note that the following set−hoodrule also holds:

```
SubstTest[implies, member[w, z], member[w, V],
    w -> natadd[x, y]]
```

```
or[and[member[x, omega], member[y, omega]], not[member[natadd[x, y], z]]] == True
```

```
or[and[member[x_, omega], member[y_, omega]], not[member[natadd[x_, y_], z_]]] := True
```

Combining these two facts produces a simpler result that is identical to Quaife's theorem **(O14)**:

```
Map[not, SubstTest[and, implies[p1, or[not[p2], not[p3]]], implies[p1, and[p2, p3]],
{p1 -> member[natadd[x, y], x], p2 -> member[x, omega], p3 -> member[y, omega]}]] // Reverse
```

```
member[natadd[x, y], x] == False
```

```
member[natadd[x_, y_], x_] := False
```

■ relational version

One way to obtain a relational version is as follows. First we replace **natadd** with **NATADD** using this trick:

```
SubstTest[class, y, member[A[image[n, singleton[PAIR[x, y]]]], x], n -> NATADD]
```

```
0 == image[image[inverse[NATADD], x], singleton[x]]
```

```
image[image[inverse[NATADD], x_], singleton[x_]] := 0
```

```
image[fix[composite[inverse[E], IMAGE[inverse[NATADD]], FIRST]], singleton[x]] //
 Normality
```

```
image[fix[composite[inverse[E], IMAGE[inverse[NATADD]], FIRST]], singleton[x]] == 0
```

```
image[fix[composite[inverse[E], IMAGE[inverse[NATADD]], FIRST]], singleton[x_]] := 0
```

Next, the variable **x** can be eliminated:

```
fix[composite[inverse[E], IMAGE[inverse[NATADD]], FIRST]] // VSNormality
```

```
fix[composite[inverse[E], IMAGE[inverse[NATADD]], FIRST]] == 0
```

```
fix[composite[inverse[E], IMAGE[inverse[NATADD]], FIRST]] := 0
```

■ a related rule

Another way to eliminate the variable **x** is like this:

```
Map[complement,
  SubstTest[class, x, equal[0, image[image[inverse[n], x], singleton[x]]],
    n -> NATADD]] // Reverse
```

```
intersection[omega,
  fix[composite[inverse[E], IMAGE[inverse[S]], IMAGE[id[omega]]]]] == 0
```

```
intersection[omega,
  fix[composite[inverse[E], IMAGE[inverse[S]], IMAGE[id[omega]]]]] := 0
```

This can be rewritten using:

```
Map[VERTSECT, Assoc[FIRST, inverse[NATADD], inverse[E]]]
```

```
composite[IMAGE[FIRST], IMAGE[inverse[NATADD]]] ==
 composite[IMAGE[id[omega]], IMAGE[inverse[S]], IMAGE[id[omega]]]
```

The above formula can also be derived another way as follows:

```
SubstTest[IMAGE, composite[u, inverse[NATADD]], u -> FIRST] // Reverse
```

```
composite[IMAGE[FIRST], IMAGE[inverse[NATADD]]] ==
 composite[IMAGE[id[omega]], IMAGE[inverse[S]], IMAGE[id[omega]]]
```

The following rule is a consequence:

```
Map[fix[composite[inverse[E], #]] &, %]
```

```
fix[composite[inverse[E], IMAGE[FIRST], IMAGE[inverse[NATADD]]]] == 0
```