

# RIF, ASSOC and reify

*Johan G. F. Belinfante*  
2002 July 9

```
<< goedel52.o92; << tools.m
:Package Title: goedel52.o92          2002 July 5 at 10:15 a.m.

It is now: 2002 Jul 9 at 0:42

Loading Simplification Rules

TOOLS.M                               Revised 2002 June 12

weightlimit = 40
```

## ■ Introduction

In this notebook a new formula for **RIF** is derived to facilitate removing variables in formulas involving the associative law. The associative law is viewed as the fact that left–addition and right–addition commute.

## ■ LEFT and RIGHT

Suppose that **n** is an associative binary operation:

```
associative[n_] := Equal[composite[n, cross[n, Id]], composite[n, cross[Id, n], ASSOC]]
```

From the associative law one can derive this commutativity statement:

```
Map[composite[#, RIGHT[x], LEFT[y]] &, associative[n]]
composite[n, RIGHT[x], n, LEFT[y]] == composite[n, LEFT[y], n, RIGHT[x]]
```

How can one go back? The idea is to do this using **reify**. The basic observation is this:

```
rotate[inverse[reify[x, composite[n, LEFT[x]]]]]
composite[n, id[cart[V, V]]]
```

A similar result holds for **RIGHT[x]** if one adds a **flip**.

```
flip[rotate[inverse[reify[x, composite[n, RIGHT[x]]]]]
composite[n, id[cart[V, V]]]
```

The only hitch is that the **reify** formulas introduce **RIF** instead of **ASSOC**.