Exchanging the values of two variables

The problem of exchanging the values of two variables without using additional variables is very well-known. Most presentations, however, rarely highlight the properties on which the solution depends. In this short note, we will investigate these properties, in order to generalize and better understand the problem.

The traditional solution assumes that the values of the variables can be represented as sequences of bits and exploits the bitwise exclusive-or operation (here denoted by $\not\equiv$). Using the Guarded Command Language, it can be written as

$$
\begin{array}{l}
\{ \quad x = X \quad \wedge \quad y = Y \quad \} \\
x \;:=\; x \not\equiv y \;\; ; \\
y \;:=\; x \not\equiv y \;\; ; \\
x \;:=\; x \not\equiv y \\
\{ \quad x = Y \quad \wedge \quad y = X \quad \} \quad .
\end{array}
$$

In order to determine which properties of $\not\equiv$ are involved and which other operators can be used, let's change $\not\equiv$ to an arbitrary operator $\otimes$ and present all the relevant annotations. Working back from the postcondition to the precondition, we get the following annotated program:

$$
\begin{array}{l}
\{ \quad x = X \quad \wedge \quad y = Y \quad \} \\
\{ \quad (x \otimes y) \otimes ((x \otimes y) \otimes y) \;=\; Y \quad \wedge \quad (x \otimes y) \otimes y = X \quad \} \\
x \;:=\; x \otimes y \\
\{ \quad x \otimes (x \otimes y) \;=\; Y \quad \wedge \quad x \otimes y = X \quad \} \, ; \\
y \;:=\; x \otimes y \\
\{ \quad x \otimes y \;=\; Y \quad \wedge \quad y \;=\; X \quad \} \, ; \\
x \;:=\; x \otimes y \\
\{ \quad x \;=\; Y \quad \wedge \quad y \;=\; X \quad \} \quad .
\end{array}
$$

Now, given the first assertion, we can rewrite the second one as the conjunction of the following two conditions:

$$(x \otimes y) \otimes ((x \otimes y) \otimes y) = y \quad , \quad \text{and}$$

$$(x \otimes y) \otimes y = x \qquad .$$

We want to find properties of the operator $\otimes$ that make these conditions hold. Starting with the simpler condition (i.e., with the second one) and using square brackets to denote universal quantification over all free variables, we calculate:

$$
\begin{array}{ll}
 & (x \otimes y) \otimes y \\
= & \quad \{ \quad \text{assuming that } \otimes \text{ is associative,} \\
 & \qquad \text{in order to isolate } x \quad \} \\
 & x \otimes (y \otimes y) \\
= & \quad \{ \quad \text{assuming that } \otimes \text{ is unitpotent, that is:} \\
 & \qquad [\ z \otimes z \ = \ 1_{\otimes}\ ]\ \text{, where } 1_{\otimes} \text{ is the unit of } \otimes \quad \} \\
 & x \ .
\end{array}
$$

The second condition is thus satisfied by assuming that $\otimes$ is associative and unitpotent. The first condition can be calculated using the same properties:

$$
\begin{array}{ll}
 & (x \otimes y) \otimes ((x \otimes y) \otimes y) \\
= & \quad \{ \quad \otimes \text{ is associative} \quad \} \\
 & ((x \otimes y) \otimes (x \otimes y)) \otimes y \\
= & \quad \{ \quad \otimes \text{ is unitpotent} \quad \} \\
 & y \ .
\end{array}
$$

Thus the correctness of the program presented above follows from the following two properties of $\otimes$:

$\otimes$ is associative , and

$\otimes$ is unitpotent .

Clearly, the bitwise exclusive-or — or, as I prefer to call it, the bitwise inequivalence — is suitable. But note that the bitwise equivalence (usually denoted by $\equiv$) can also be used.!

Generalising $\otimes$

We now generalise $\otimes$ by replacing each occurrence with a separate operator. The new program and corresponding annotations become:

$$
\begin{array}{l}
\{ \quad x = X \quad \wedge \quad y = Y \quad \} \\
\{ \quad (x \otimes y) \ominus ((x \otimes y) \oplus y) = Y \quad \wedge \quad (x \otimes y) \oplus y = X \quad \} \\
x \;:=\; x \otimes y \\
\{ \quad x \ominus (x \oplus y) \;=\; Y \quad \wedge \quad x \oplus y \;=\; X \quad \}\,; \\
y \;:=\; x \oplus y \\
\{ \quad x \ominus y \;=\; Y \quad \wedge \quad y \;=\; X \quad \}\,; \\
x \;:=\; x \ominus y \\
\{ \quad x \;=\; Y \quad \wedge \quad y \;=\; X \quad \} \quad .
\end{array}
$$

Again, from the two initial assertions we get the two following conditions:

$$(x \otimes y) \ominus ((x \otimes y) \oplus y) = y \quad \text{, and}$$

$$(x \otimes y) \oplus y = x \qquad .$$

As before, the goal is to investigate which properties of the operators make these conditions hold. Starting with the second condition, we calculate:

$$
\begin{array}{ll}
 & (x \otimes y) \oplus y \\
= & \quad \{ \quad \text{assuming that } \otimes \text{ associates with } \oplus \quad \} \\
 & x \otimes (y \oplus y) \\
= & \quad \{ \quad \text{assuming that } \oplus \text{ is unitpotent with respect to } \otimes\text{, that is:} \\
 & \qquad [\; z \oplus z = 1_{\otimes} \;]\text{, where } 1_{\otimes} \text{ is the unit of } \otimes \quad \} \\
 & x \quad .
\end{array}
$$

Now, the first condition:

$$
\begin{array}{ll}
 & (x \otimes y) \ominus ((x \otimes y) \oplus y) \\
= & \quad \{ \quad \text{previous calculation} \quad \} \\
 & (x \otimes y) \ominus x \\
= & \quad \{ \quad \text{assuming that } \otimes \text{ is symmetric} \quad \} \\
 & (y \otimes x) \ominus x \\
= & \quad \{ \quad \text{assuming that } \otimes \text{ associates with } \ominus \quad \} \\
 & y \otimes (x \ominus x)
\end{array}
$$

$$
\begin{array}{ll}
= & \{ \quad \text{assuming that } \ominus \text{ is unitpotent with respect to } \otimes \quad \} \\
& y \ .
\end{array}
$$

Note that the choices made in this calculation could be different. The final section of this note deals with a calculation that leads to different properties.

We thus conclude from the two previous calculations that our new program is correct if the following properties hold:

$\otimes$ is symmetric ,

$\otimes$ associates with $\oplus$,

$\otimes$ associates with $\ominus$,

$\oplus$ is unitpotent with respect to $\otimes$, and

$\ominus$ is unitpotent with respect to $\otimes$.

As we can see, operations $\oplus$ and $\ominus$ are identical with respect to these conditions. In fact, we can prove that these five properties imply that $\oplus$ and $\ominus$ are equal:

$$
\begin{array}{ll}
& x \oplus y \\
= & \{ \quad \text{unitpotency of } \ominus \text{ with respect to } \otimes, \\
& \quad \text{in order to introduce } \ominus \quad \} \\
& (x \oplus y) \otimes (y \ominus y) \\
= & \{ \quad \otimes \text{ associates with } \ominus \quad \} \\
& ((x \oplus y) \otimes y) \ominus y \\
= & \{ \quad \text{deferred proof obligation of } [\ (x \oplus y) \otimes y = x\ ]; \\
& \quad \text{see below} \quad \} \\
& x \ominus y \ .
\end{array}
$$

The assumption in the last step can be easily proved from the other properties as follows:

$$
\begin{array}{ll}
& (x \oplus y) \otimes y \\
= & \{ \quad \otimes \text{ is symmetric} \quad \} \\
& y \otimes (x \oplus y)
\end{array}
$$

$$
\begin{array}{ll}
= & \{ \quad \otimes \text{ associates with } \oplus \quad \} \\
& (y \otimes x) \oplus y \\
= & \{ \quad \otimes \text{ is symmetric} \quad \} \\
& (x \otimes y) \oplus y \\
= & \{ \quad \otimes \text{ associates with } \oplus \quad \} \\
& x \otimes (y \oplus y) \\
= & \{ \quad \oplus \text{ is unitpotent with respect to } \otimes \quad \} \\
& x \quad .
\end{array}
$$

Thus we write both $\oplus$ and $\ominus$ as $\oplus$ and our program becomes:

$$
\begin{array}{l}
\{ \quad x = X \quad \wedge \quad y = Y \quad \} \\
x \; := \; x \otimes y \; ; \\
y \; := \; x \oplus y \; ; \\
x \; := \; x \oplus y \\
\{ \quad x = Y \quad \wedge \quad y = X \quad \} \quad .
\end{array}
$$

Recall that this program is correct if the following properties hold:

$\otimes$ is symmetric ,

$\otimes$ associates with $\oplus$, and

$\oplus$ is unitpotent with respect to $\otimes$.

A simple refinement

An immediate corollary is that if we have a group with a symmetric operation $\otimes$, and if we define the operator $\oplus$ as

$$x \oplus y = x \otimes y^{-1} \quad ,$$

where y^{-1} is the inverse of y , then the above properties will hold, as the reader can verify. If we take, for instance, real addition for $\otimes$ and real subtraction for $\oplus$, we get the following program:

$$
\begin{array}{l}
\{ \quad x = X \quad \wedge \quad y = Y \quad \} \\
x \;:=\; x \;+\; y \;; \\
y \;:=\; x \;-\; y \;; \\
x \;:=\; x \;-\; y \\
\{ \quad x = Y \quad \wedge \quad y = X \quad \} \quad .
\end{array}
$$

Eliminating the symmetry requirement

As said before, this section presents an alternative choice for the calculation presented in page 2. This choice leads to a different set of assumptions:

$$
\begin{array}{ll}
 & (x \otimes y) \ominus ((x \otimes y) \oplus y) \\
= & \quad \{ \quad \ominus \text{ associates with } \oplus \quad \} \\
 & ((x \otimes y) \ominus (x \otimes y)) \oplus y \\
= & \quad \{ \quad \ominus \text{ is unitpotent with respect to } \oplus \quad \} \\
 & y \quad .
\end{array}
$$

We then conclude that $\otimes$ does not need to be symmetric and that the program with three operations is correct if

- $\otimes$ associates with $\oplus$,
- $\ominus$ associates with $\oplus$,
- $\oplus$ is unitpotent with respect to $\otimes$, and
- $\ominus$ is unitpotent with respect to $\oplus$.

Acknowledgments

I'd like to thank to my colleagues in the Nottingham Tuesday Morning Club, to Jeremy Weissmann and to Apurva Mehta for their comments, suggestions and corrections on earlier versions of this note.

João Fernando Ferreira
July 26, 2007

School of Computer Science, University of Nottingham
NG8 1BB, UK
joao@joaoferreira.org