

Abstraction, and Kaldewaij's “**mod**” exercises

When we are faced with a problem to solve, it can be useful and sometimes even crucial to generalize the problem, to abstract away from its details. In doing so, we give ourselves the opportunity to solve a whole class of problems in one go, including the original problem. But also, by suppressing distracting or irrelevant details, the problem may become easier to solve, despite being more general.

In this note, I will use abstraction in various ways to aid in solving some problems about the binary **mod** operator on integers, as given in Anne Kaldewaij's book *Programming*. Some additional results will be included, for completeness's sake.

* *

 *

In what follows, all variables are of type integer, and in particular, $m \neq 0$, unless otherwise mentioned.

Kaldewaij defines the binary infix operator **mod** together with **div**, by:

$$(0) \quad a \mathbf{div} m = q \quad \wedge \quad a \mathbf{mod} m = r$$

$$\equiv$$

$$a = m * q + r \quad \wedge \quad 0 \leq r < |m| \quad ,$$

where, recall, we have $m \neq 0$. So that **mod** and **div** are properly defined by (0), Kaldewaij invites us to note that equation:

$$(1) \quad (q, r) : \quad a = m * q + r \quad \wedge \quad 0 \leq r < |m|$$

has, for $m \neq 0$, precisely one solution. A simple program will settle the existence of a solution; as for the uniqueness, suppose (q_0, r_0) and (q_1, r_1) are solutions to (1), so that we have:

$$(2) \quad a = m * q_0 + r_0$$

$$(3) \quad 0 \leq r_0 < |m|$$

$$(4) \quad a = m * q_1 + r_1$$

$$(5) \quad 0 \leq r_1 < |m| \quad .$$

From (2) and (4) we calculate:

$$a = m * q_0 + r_0 \quad \wedge \quad a = m * q_1 + r_1$$

$$\Rightarrow \quad \{ \text{transitivity of equality} \}$$

$$m * q_0 + r_0 = m * q_1 + r_1$$

$$\equiv \quad \{ \text{algebra, collecting like variables} \}$$

$$m * (q_0 - q_1) = r_1 - r_0 \quad ,$$

hence we conclude:

$$(6) \quad m * (q_0 - q_1) = r_1 - r_0 \quad .$$

Exploring now (3) and (5) , we calculate:

$$\begin{aligned} & 0 \leq r_0 < |m| \quad \wedge \quad 0 \leq r_1 < |m| \\ \Rightarrow & \{ \text{ a few steps of arithmetic } \} \\ & -|m| < r_1 - r_0 < |m| \\ \equiv & \{ (6) \} \\ & -|m| < m * (q_0 - q_1) < |m| \\ \equiv & \{ m \neq 0 \} \\ & -1 < q_0 - q_1 < 1 \\ \equiv & \{ q_0 - q_1 \text{ is integer} \} \\ & q_0 - q_1 = 0 \\ \equiv & \{ \text{ algebra} \} \\ & q_0 = q_1 \quad . \end{aligned}$$

From (6) we then conclude $r_0 = r_1$, hence $(q_0, r_0) = (q_1, r_1)$, as desired.

* * *

Because in this note we will not be concerned with **div** , we use predicate calculus to help us “eliminate” it from definition (0) :

$$\begin{aligned} & a \mathbf{mod} m = r \\ \equiv & \{ \text{ one-point rule} \} \\ & \langle \exists q : a \mathbf{div} m = q : a \mathbf{mod} m = r \rangle \\ \equiv & \{ \text{ shunting} \} \\ & \langle \exists q :: a \mathbf{div} m = q \quad \wedge \quad a \mathbf{mod} m = r \rangle \\ \equiv & \{ (0) \} \\ & \langle \exists q :: a = m * q + r \quad \wedge \quad 0 \leq r < |m| \rangle \\ \equiv & \{ \wedge \text{ over } \exists \} \\ & \langle \exists q :: a = m * q + r \rangle \quad \wedge \quad 0 \leq r < |m| \\ \equiv & \{ \text{ definition of } \sqsubseteq , \text{ “divides” ; see Appendix} \} \\ & m \sqsubseteq a - r \quad \wedge \quad 0 \leq r < |m| \quad ; \end{aligned}$$

summarizing:

$$(7) \quad a \mathbf{mod} m = r$$

$$\equiv$$

$$m \sqsubseteq a - r \quad \wedge \quad 0 \leq r < |m| \quad .$$

Instantiating (7) with $r := a \mathbf{mod} m$, we conclude:

$$(8) \quad m \sqsubseteq a - (a \mathbf{mod} m) \quad \text{and}$$

$$(9) \quad 0 \leq a \mathbf{mod} m < |m| \quad ,$$

which we might call the “defining properties” of \mathbf{mod} .

So much for the definition of \mathbf{mod} and its defining properties.

* * *

We turn now to a “warm-up” exercise, one that will not require any abstraction, but will allow us to practice with our definition of \mathbf{mod} . The theorem to be proved is:

$$(10) \quad a \mathbf{mod} m = a \mathbf{mod} (-m) \quad .$$

(This exercise also comes from Kaldewaij.) A calculational proof of (10) is straightforward:

$$a \mathbf{mod} (-m) = r$$

$$\equiv \{ (7) \}$$

$$-m \sqsubseteq a - r \quad \wedge \quad 0 \leq r < |-m|$$

$$\equiv \{ \text{property of } \sqsubseteq ; \text{ property of absolute value } \}$$

$$m \sqsubseteq a - r \quad \wedge \quad 0 \leq r < |m|$$

$$\equiv \{ (7) \}$$

$$a \mathbf{mod} m = r \quad ,$$

from which we can conclude (10) on account of “indirect equality”.

* * *

Well, that was simple enough, but already the next exercise is much more challenging with its embedded \mathbf{mod} ’s:

$$(11) \quad (a \mathbf{mod} m) \mathbf{mod} m = a \mathbf{mod} m \quad .$$

With **mod**'s all over the place, application of (7) is uninviting, because we have many choices for where to apply it, and we may have to apply it multiple times.

In such an instance, abstraction becomes invaluable. Notice that (11) is of the form:

$$(12) \quad r \mathbf{mod} m = r \quad ,$$

which is much more inviting to manipulate. We calculate with (12) to discover when it holds:

$$\begin{aligned} & r \mathbf{mod} m = r \\ \equiv & \{ (7) \} \\ & m \sqsubseteq r - r \quad \wedge \quad 0 \leq r < |m| \\ \equiv & \{ \text{property of } \sqsubseteq \text{ and predicate calculus} \} \\ & 0 \leq r < |m| \quad . \end{aligned}$$

So (12) holds precisely when $0 \leq r < |m|$. The relevant instantiation for (11) is $r := a \mathbf{mod} m$, so the question now is whether we have $0 \leq a \mathbf{mod} m < |m|$, and by defining property (9) of **mod**, the answer is a resounding yes.

Problem (11) was a mess, but solving it via the more general (12) couldn't have been easier. Three cheers for abstraction.

* * *

Finally, we turn to the last problem from Kaldewaij:

$$(13) \quad ((a \mathbf{mod} m) + (b \mathbf{mod} m)) \mathbf{mod} m = (a + b) \mathbf{mod} m \quad .$$

If (11) was a mess, (13) is a nightmare. And unfortunately, abstraction (12) is of no help, because (13) is not of that form. That is all right: we just make a new abstraction! Formula (13) is of the shape:

$$(14) \quad c \mathbf{mod} m = d \mathbf{mod} m \quad ,$$

so let us investigate when (14) holds:

$$\begin{aligned} & c \mathbf{mod} m = d \mathbf{mod} m \\ \equiv & \{ (7) \text{ with } a := c \text{ and } r := d \mathbf{mod} m \} \\ & m \sqsubseteq c - (d \mathbf{mod} m) \quad \wedge \quad 0 \leq d \mathbf{mod} m < |m| \\ \equiv & \{ (9) \text{ with } a := d ; \text{ predicate calculus} \} \end{aligned}$$

$$\begin{aligned}
& m \sqsubseteq c - (d \mathbf{mod} m) \\
\equiv & \{ \text{by (8)}, m \sqsubseteq d - (d \mathbf{mod} m) ; \text{subtractive property of } \sqsubseteq \} \\
& m \sqsubseteq c - d \quad .
\end{aligned}$$

Terrific! So making the appropriate instantiations, we see that (13) holds precisely when:

$$\begin{aligned}
& m \sqsubseteq ((a \mathbf{mod} m) + (b \mathbf{mod} m)) - (a + b) \\
\equiv & \{ \text{algebra} \} \\
& m \sqsubseteq ((a \mathbf{mod} m) - a) + ((b \mathbf{mod} m) - b) \\
\Leftarrow & \{ \text{additive property of } \sqsubseteq \} \\
& m \sqsubseteq (a \mathbf{mod} m) - a \quad \wedge \quad m \sqsubseteq (b \mathbf{mod} m) - b \\
\equiv & \{ \text{property of } \sqsubseteq, \text{ twice} \} \\
& m \sqsubseteq a - (a \mathbf{mod} m) \quad \wedge \quad m \sqsubseteq b - (b \mathbf{mod} m) \\
\equiv & \{ (8) \text{ twice; predicate calculus} \} \\
& \mathbf{true} \quad ,
\end{aligned}$$

which settles (13) .

Generalization (14) is quite general, and hence quite powerful. Note that our earlier problem (11) is also of form (14) , with $c := a \mathbf{mod} m$ and $d := a$. By the above calculation, the relevant check is simply whether $m \sqsubseteq (a \mathbf{mod} m) - a$, which it does by (8) —and a theorem about \sqsubseteq — .

Similarly, we can use our result about (14) to prove (a generalization of) a theorem given in *Programming*, namely:

$$(a + k * m) \mathbf{mod} m = a \mathbf{mod} m$$

by observing:

$$\begin{aligned}
& m \sqsubseteq (a + k * m) - a \\
\equiv & \{ \text{algebra} \} \\
& m \sqsubseteq k * m \\
\equiv & \{ \text{property of } \sqsubseteq \} \\
& \mathbf{true} \quad .
\end{aligned}$$

Et cetera!

* *

 *

To reiterate in conclusion: three cheers for abstraction!

Appendix

Integer relation \sqsubseteq (“divides”) can be defined by:

$$a \sqsubseteq b \equiv \langle \exists q :: b = a * q \rangle .$$

From this definition it follows that \sqsubseteq is a partial order, as the reader can verify. Also in this document we have used the following properties of \sqsubseteq , which are easily verified:

$$a \sqsubseteq 0$$

$$a \sqsubseteq k * a$$

$$a \sqsubseteq b \equiv -a \sqsubseteq b$$

$$a \sqsubseteq b \equiv a \sqsubseteq -b$$

$$a \sqsubseteq (b + c) \Leftarrow a \sqsubseteq b \wedge a \sqsubseteq c$$

$$a \sqsubseteq b \Rightarrow (a \sqsubseteq c \equiv a \sqsubseteq (c - b))$$

(As I have learned recently from Netty van Gasteren’s thesis, this last property of \sqsubseteq the driving force behind most implementations of Euclid’s algorithm for computing the greatest common divisor of two positive integers.)

(End of Appendix .)

Written: Zurich (Starbucks Cafe & Train to Salzburg), 21 December 2005

Typeset: Westward o’er the Atlantic, 4 January 2006

Jeremy Weissmann
 11260 Overland Ave. #21A
 Culver City, CA 90230
 USA
 jeremy@mathmeth.com