

A summary of my work on Euclid's lemma

Prelude on Notation. Throughout, a , b , c , x , and y denote structures of type strictly positive natural, while A , B , C , X , and Y denote structures of type bag. Here are the relative binding powers of our operators, from strongest to weakest:

\cdot , $\#$	function application, bag characteristic
$*$, \diamond	multiplication, unknown operator
$+$	addition
\downarrow , \Downarrow , \cup , \cap	minimum, gcd, bag union, bag intersection
\leq , \sqsubseteq , \subseteq , $=$	at most, divides, subbag, equals
\wedge , \vee	and, or
\Rightarrow	implies
\equiv	equivalences

(End of Prelude on Notation.)

In this note I would like to summarize the work I have done on a problem which arose, in various contexts, in EWD1313, EWD1315, JAW0, JAW3, and JAW9. The tone of this note will be entirely investigatory.

The problem in JAW0 was to find a sufficient condition for

$$(0) \quad [a \sqsubseteq b * c \Rightarrow a \sqsubseteq b]$$

for strictly positive natural structures a , b , and c . First, noting the similarity between left and righthand sides of the implication, in the name of local separation of concerns — see JAW5 —, we investigate the leftmost term:

$$\begin{aligned} & a \sqsubseteq b * c \\ \Leftarrow & \{ \text{transitivity of } \sqsubseteq \} \\ & a \sqsubseteq b \wedge b \sqsubseteq b * c \\ \equiv & \{ \text{property of } \sqsubseteq \} \\ & a \sqsubseteq b \end{aligned}$$

Well, that is not what we were looking for, but it is certainly nice, and allows to rewrite (0) equivalently as

$$(0') \quad [a \sqsubseteq b * c \equiv a \sqsubseteq b]$$

One sufficient condition for (0') now pops out at us:

JAW10-1

$$\begin{aligned} & (0') \\ \Leftarrow & \{ \text{Leibniz} \} \\ & [b * c = b] \\ \equiv & \{ \text{algebra} \} \\ & [c = 1] \quad . \end{aligned}$$

Very clean, but not terribly helpful, since the condition requires us to know the value of c . We might consider this a “trivial” solution of our problem.

In JAW3 and JAW9 I used unique decomposition to press further; we will return to that strategy below, but for now let us pursue the direction of EWD1313 and EWD1315, which considers \sqsubseteq as the partial order underlying a lattice. Towards that end, we write:

$$\begin{aligned} & (0') \\ \equiv & \{ \text{lattice theory} \} \\ & [(a = a \Downarrow b * c) \equiv (a = a \Downarrow b)] \\ \Leftarrow & \{ \text{Leibniz} \} \\ & [a \Downarrow b * c = a \Downarrow b] \quad . \end{aligned}$$

Our investigation thus turns to

$$(1) \quad [a \Downarrow b * c = a \Downarrow b] \quad .$$

Note that (1) is strictly stronger than (0), since under the instantiation $a, b, c := 4, 1, 2$, (0) is **true**, while (1) is **false**.

Calculating with (1) for a bit, we find:

$$\begin{aligned} & (1) \\ \equiv & \{ \Downarrow \text{ preserves } \sqsubseteq : \quad a \Downarrow b \sqsubseteq a \Downarrow b * c \} \\ & [a \Downarrow b * c \sqsubseteq a \Downarrow b] \\ \equiv & \{ \text{absorption: } \quad a \Downarrow b * c \sqsubseteq a \} \\ & [a \Downarrow b * c \sqsubseteq b] \quad , \end{aligned}$$

and we are stuck, since we do not know how to manipulate \Downarrow on the lefthand side of \sqsubseteq ; \sqsubseteq is not a total order. However, we write down this equivalent form of (1), as we will use it in a later investigation:

$$(1') \quad [a \Downarrow b * c \sqsubseteq b] \quad .$$

This last calculation ignored the symmetry in (1), to our peril. Note that there is a great deal of repetition between the two sides of (1), which we might try to exploit

by transforming one side into the other. By a simple syntactic analysis of (1), we see that this amounts to either introducing c and $*$, or removing these elements. Without question, introduction is easier, thanks to the simple property of 1 with respect to multiplication:

$$(2) \quad [x * 1 = x] \quad .$$

Thus we calculate:

$$\begin{aligned}
 & a \Downarrow b \\
 = & \quad \{ \text{As stated above, our task is to introduce } c \text{ and } * ; \\
 & \quad \text{we can introduce } * \text{ at no cost via (2).} \} \\
 & a \Downarrow b * 1 \\
 = & \quad \{ \text{In order to introduce } c, \text{ we postulate some relation} \\
 & \quad \text{holding between } c \text{ and } 1; \text{ this will presumably be} \\
 & \quad \text{the sufficient condition we are looking for. We take} \\
 & \quad \text{this relation in the form } [x \diamond c = 1] \text{ for as yet} \\
 & \quad \text{undetermined } \diamond \text{ and } x; \text{ we introduce } x \text{ so as to} \\
 & \quad \text{give us the possibility of introducing } a \text{ or } b \text{ into this} \\
 & \quad \text{relation, thereby avoiding trivial conditions like the} \\
 & \quad \text{one we derived above.} \} \\
 & a \Downarrow b * (x \diamond c) \\
 = & \quad \{ \text{Our goal has } b \text{ and } c \text{ as co-arguments of } * . \text{ Here} \\
 & \quad \text{instead of } c \text{ we have a } \diamond \text{-term involving } c, \text{ hence} \\
 & \quad \text{distributivity is called for. We thus require of } \diamond \\
 & \quad \text{that } * \text{ distributes over it.} \} \\
 & a \Downarrow (b * x) \diamond (b * c) \\
 = & \quad \{ \text{Similarly, we assume distributivity of } \Downarrow \text{ over } \diamond . \} \\
 & (a \Downarrow b * x) \diamond (a \Downarrow b * c) \\
 = & \quad \{ \text{We have created our goal! However, we have a bit of} \\
 & \quad \text{garbage to clean up. Let us first aim to reduce the} \\
 & \quad \text{first term here; we can do this by choosing } x := a, \\
 & \quad \text{thanks to the absorption properties of } \Downarrow . \} \\
 & a \diamond (a \Downarrow b * c)
 \end{aligned}$$

= { The need to eliminate the repeated a suggests we
 exploit idempotence, and thus we choose \Downarrow for \diamond . }
 $a \Downarrow b * c$.

So we have derived the condition $[a \Downarrow c = 1]$ and the proof of its sufficiency hand in hand... provided of course our assumptions in the proof about \diamond/\Downarrow are valid. Thankfully they are: \Downarrow enjoys auto-distributivity —like any associative, symmetric, and idempotent operator—, and $*$ distributes over \Downarrow , the proof of which is beyond the scope of this note. (See EWD1315 .)

Our first observation is that the heuristics are much more satisfying if we apply this proof strategy to the original demonstrandum (0) :

$a \sqsubseteq b$
 \equiv { (2) }
 $a \sqsubseteq b * 1$
 \equiv { assuming $[x \diamond c = 1]$ }
 $a \sqsubseteq b * (x \diamond c)$
 \equiv { assuming $*$ over \diamond }
 $a \sqsubseteq (b * x) \diamond (b * c)$
 \equiv { The desire to distribute \sqsubseteq over \diamond immediately
 suggests \Downarrow for \diamond , thanks to lattice theory. }
 $a \sqsubseteq b * x \wedge a \sqsubseteq b * c$
 \equiv { And now the choice $x := a$ is practically forced
 upon us. }
 $a \sqsubseteq b * c$.

At least for me, the choices in the last two steps are far more dictated than in the previous proof. Perhaps this is a reflection of the fact that (1) is strictly stronger than (0) .

Our second observation is that when the sufficient condition $[a \Downarrow c = 1]$ is known already, the same heuristics work even better for the design of the proofs of (0) and (1) .

* *
 *

In the last part of this note, we will retread some of the ground covered in JAW9 , for completeness, and also because I have found a way to refine the heuristics. The strategy is to start with

(1') $[a \Downarrow b * c \sqsubseteq b]$,

the dead end from our previous analysis, and use unique factorization to probe further. Towards this end we introduce a function f from positive natural structures to bag structures, defined by:

$$[f.x = \text{“the bag of primes in } x \text{’s decomposition”}] \quad .$$

We then have lovely translation theorems like:

$$\begin{aligned} (3) \quad & [x = y] \equiv [f.x = f.y] \\ & [x \sqsubseteq y] \equiv [f.x \subseteq f.y] \\ & [f.(x \downarrow y) = f.x \cap f.y] \\ & [f.(x * y) = f.x \cup f.y] \quad , \\ & [f.1 = \emptyset] \quad , \quad \text{where } \emptyset \text{ is the empty bag} \quad , \end{aligned}$$

so that we can calculate:

$$\begin{aligned} (1') \\ \equiv \quad & \{ (3) , \text{ several times} \} \\ & [f.a \cap (f.b \cup f.c) \subseteq f.b] \quad . \end{aligned}$$

For simplicity’s sake, we gamble that it is irrelevant that the elements of these bag structures are primes, and thus we investigate:

$$(4) \quad [A \cap (B \cup C) \subseteq B]$$

for arbitrary bag structures A , B , and C .

As in JAW9 , we translate bag structures to natural structures via an operator $\#$. To do this, we define the state space for these natural structures to be the “domain of discourse” for the elements of the bag structures—in the case in question, this would be the set of all primes—, and define $\#X$ pointwise to be the number of times that element occurs in X . Thus we have:

$$\begin{aligned} (5) \quad & [X = Y] \equiv [\#X = \#Y] \\ & [X \subseteq Y] \equiv [\#X \leq \#Y] \\ & [\#(X \cap Y) = \#X \downarrow \#Y] \\ & [\#(X \cup Y) = \#X + \#Y] \\ & [\#\emptyset = 0] \quad , \end{aligned}$$

and we calculate:

$$\begin{aligned}
& (4) \\
\equiv & \{ (5), \text{ several times} \} \\
& [\#A \downarrow \#B + \#C \leq \#B] \\
\equiv & \{ \text{lattice theory for total orders} \} \\
& [\#A \leq \#B \vee \#B + \#C \leq \#B] \\
\equiv & \{ \text{algebra} \} \\
& [\#A \leq \#B \vee \#C \leq 0] \quad .
\end{aligned}$$

So far, great, but (5) does not allow us to translate a disjunction. That is all right, however, because we can strengthen the disjunction of two inequalities into a single inequality, via a \leq -preserving operator. And strengthening is all right because we are looking for a sufficient condition! All that remains is to choose a translatable \leq -preserving operator. Unluckily, both \downarrow and $+$ are monotonic, so that we can proceed in two directions. First we try \downarrow , which has the potential advantage of eliminating $\#B$:

$$\begin{aligned}
& [\#A \leq \#B \vee \#C \leq 0] \\
\Leftarrow & \{ \downarrow \text{ preserves } \leq \} \\
& [\#A \downarrow \#C \leq \#B \downarrow 0] \\
\equiv & \{ \#B \downarrow 0 = 0 \} \\
& [\#A \downarrow \#C \leq 0] \\
\equiv & \{ (5) \} \\
& [\#(A \cap C) \leq \#\emptyset] \\
\equiv & \{ (3) \} \\
& [A \cap C \subseteq \emptyset] \quad ,
\end{aligned}$$

which yields in the original context:

$$\begin{aligned}
& [f.a \cap f.c \subseteq \emptyset] \\
\equiv & \{ (3) \} \\
& [a \downarrow c \sqsubseteq 1] \\
\equiv & \{ \text{arithmetic} \} \\
& [a \downarrow c = 1] \quad .
\end{aligned}$$

That was nice, and painless too, considering we were able to capture all the relevant information about prime factorization in a tidy calculus.

If we had tried to exploit the fact that $+$ preserves \leq , we would have gotten:

$$\begin{aligned}
& [\#A \leq \#B \quad \vee \quad \#C \leq 0] \\
\Leftarrow & \{ + \text{ preserves } \leq \} \\
& [\#A + \#C \leq \#B] \\
\equiv & \{ (5) \} \\
& [\#(A \cup C) \leq \#B] \\
\equiv & \{ (3) \} \\
& [A \cup C \subseteq B] \quad ,
\end{aligned}$$

which yields in the original context:

$$\begin{aligned}
& [f.a \cup f.c \subseteq f.b] \\
\equiv & \{ (3) \} \\
& [a * c \sqsubseteq b] \quad .
\end{aligned}$$

At first glance, it seems as if we have discovered an alternative to Euclid's famous lemma! But in fact this is just a "degenerate" solution: $[a * c \sqsubseteq b]$ implies $[a \sqsubseteq b]$, in which case (0) and (1) are trivially satisfied.

Afterword. When I said earlier that the lattice theory approach got stuck at (1'), I was quite mistaken. Some reflection and whisky led me to the following beautiful calculation starting from (1') :

$$\begin{aligned}
& [a \Downarrow b * c \sqsubseteq b] \\
\equiv & \{ \text{lattice theory} \} \\
& [b \Uparrow (a \Downarrow b * c) = b] \\
\equiv & \{ \Uparrow \text{ over } \Downarrow \} \\
& [(b \Uparrow a) \Downarrow (b \Uparrow b * c) = b] \\
\equiv & \{ \text{absorption} \} \\
& [(b \Uparrow a) \Downarrow b * c = b] \\
\equiv & \{ * \text{ over } \Downarrow ; b \sqsubseteq b \Uparrow a \} \\
& [b * ((b \Uparrow a) / b \Downarrow c) = b] \\
\equiv & \{ \text{cancelling } b \} \\
& [(b \Uparrow a) / b \Downarrow c = 1] \\
\equiv & \{ (x = 1) \equiv (x \sqsubseteq 1) \} \\
& [(b \Uparrow a) / b \Downarrow c \sqsubseteq 1] \\
\Leftarrow & \{ \text{shrinking: } b \Uparrow a \sqsubseteq b * a \text{ and monotonicity of } (/b) \text{ and } (\Downarrow c) \}
\end{aligned}$$

JAW10-7

$$\begin{aligned} & [(b * a) / b \Downarrow c \sqsubseteq 1] \\ \equiv & \{ \text{cancelling } b \} \\ & [a \Downarrow c \sqsubseteq 1] \\ \equiv & \{ (x = 1) \equiv (x \sqsubseteq 1) \} \\ & [a \Downarrow c = 1] \quad . \end{aligned}$$

I have perhaps included more detail than is absolutely necessary; I am just coming to grips with using the monotonicity of \Downarrow and \Uparrow , and found it valuable to work through the calculation thoroughly.

This effectively solves the entire problem without the slightest hint of a rabbit. The only conceivable rabbit is the first step; however, you will find that if you want to push forward by massaging (1') via lattice theory, this is the only manipulation that doesn't run you in circles!

Culver City, 22 August 2005

Afterword, 27 August 2005

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