**How I understand context and type information**

The following JAW was originally a series of emails exploring some conventions of our mathematical style. At the time I considered it the first significant contribution I had made to the world, and even expressed that, having written them, I could have died happy. Indeed, the ideas here have blossomed into a fantastic conceptual interface for understanding concepts and mathematics, a new manifesto. In preparation for the writing of that manifesto, I am releasing the original emails now with minor revisions. Enjoy!

* * *

Symbols and properties

The formulae in our calculations involve symbols, and associated with these symbols are **properties** we wish to use when we calculate.

For example: Our calculation may involve the symbol $+$ , which may have associated with it all the properties of real addition. Or it may involve the symbol $1$ , which may have associated with it all the properties of the number $1$ . Or it may involve the symbol $x$ , which may have associated with it all the properties of an integer. (In this last case, the property of being an integer is typically called the 'type' of $x$ . But in my eyes type is no different from any other contextual property, like for instance the property that $+$ is associative.)

The conjunction of these properties constitutes the **context** of a calculation. And a step of a calculation like:

$$\begin{array}{ll} & P \\ \rightarrow & \{ \text{ hint } \} \\ & R \end{array}$$

is no more and no less than a shorthand for:

(0) for all symbols in $P$ and $R$ satisfying their contextual properties, $P \rightarrow R$ holds (with all type information replaced in every quantified expression) .

We need a separate stipulation to put the type information back into quantified expressions, because the context —which contributes all the properties of the symbols— lies outside the quantificational brackets. Here's how to put that information back: Suppose that symbol $i$ is associated with property $f$ in our context. Then replace a quantified expression like:

$$\langle Q i : \ \ldots \ : \ \ldots \ \rangle$$

with

$$\langle Qi: \quad f.i \ \wedge \ \ldots \quad : \quad \ldots \ \rangle \qquad .$$

How painless! And using this convention, one can painlessly derive what the type restrictions need to be on quantifier rules like "one-point" , "instantiation" , "dummy transformation" , etc. And now it should also be clear why we can freely add and remove type information to and from the range of a quantified expression, conjunct-wise.

In the sequel, I refer to (0) by the shorthand: $\langle\!\langle P \rightarrow R \rangle\!\rangle$ .

* * *

## Tweaks

We can of course make minor tweaks to this format. For instance, we often find ourselves calculating with what we call "structures" . Along with structures come an associated everywhere operator [ ] . And so we may make a tweak so that a step like:

$$\begin{array}{ll} & P \\ \rightarrow & \{ \text{ hint } \} \\ & R \end{array}$$

is in fact shorthand for $[\ \langle\!\langle\ P \rightarrow R\ \rangle\!\rangle\ ]$ .

And now we can add in all the other tweaks that Dijkstra and Scholten make to Feijen's proof format in *Predicate Calculus and Program Semantics* .

* * *

## Justification

So why this little note? Don't we all know how to calculate?

Well, yes and no. Over the last few months, there have been some mild and not-so-mild confusions about empty types, empty ranges, symbols being "defined" or "not well defined" , values "existing" or not; etc etc etc. It became clear very quickly that if we were to make sense of this mess, we had to abandon philosophical notions like "assumptions" , "definitions" , "truth" , "existence" , etc.

I take the position now that a variable like $x$ , an operator like $*$ , a constant like $1$ , are from the eyes of our calculational system all the same thing. They are all just symbols, and into a context we put some properties of these symbols. We then calculate in that context, meaning that we can "use" that context to manipulate. A quick glance at (0) shows us why we can use the context: the context is in the range of a universal quantification, while the calculational step is in the term. It's just predicate calculus!

Symbols cannot be "undefined" ; it just may happen that nothing in our context enables to manipulate a expression. From a calculational point of view, then, they are not "absurd" , or "wrong" , or "nonsensical" : they are just useless!

A final point. If our context implies **false** , then our context is not very interesting, because then by predicate calculus, (0) is just **true** , regardless of $P$ and $R$ . In some recent conversations, people were afraid of discussing an element $g$ of the empty set, because "it didn't exist" . They had the vague feeling they could not even write down the symbol $g$ , let alone prove a theorem about it. (I implore you to immediately dispel this feeling by grabbing a sheet of paper and writing down $g$ . If you're feeling very daring, continue with: "is an element of the empty set" . )

But this is just letting philosophy confuse us. We can certainly write down $g$ , and not only can we prove *some* theorem about $g$ , but we can prove *any* theorem about *anything*, when our context contains " $g$ is an element of the empty set" . Whether this consequence is desirable or undesirable depends on the situation. In the situation we were dealing with, it was highly desirable; but often, it is not desirable to use contexts that imply **false** , since then everything equivales **true** .

*  *
*

## First postscript: Contexts and nesting

In the little story I told above, there was one grand context, which stood in the range of a huge universal quantification. Thanks to "nesting" , we are welcome to view this one context as several nested contexts, and in practice we do. For instance, the properties of the boolean operators are usually in the outermost context; the properties of domain-particular operators are in the next innermost context, as well as the types of dummies and constants; then we have calculation-specific contexts; then step-specific (or "local" ) contexts; then the ranges of quantified formulae; and last but not least, contexts within formulae: for instance, in the formulae $C \wedge X$ and $C \Rightarrow X$ we may view $X$ as being in the context of $C$ .

This comment may be filed under: techniques of being orderly and effecting separations of concerns.

*  *
*

## Second postscript: When ranges are **false**

One of my correspondents asked me to clarify how it could be useful to allow a context which implies **false** .

Remembering that the context constitutes the range of a universal quantification, we ought to reflect on how we usually make use of universal quantifications with ranges equal to **false** . Such quantifications always equivale **true** , so we do not write such quantifications down, for instance, to capture properties of symbols. And so analogously, if we are setting up a fixed context in preparation for a series of calculational investigations, we would not want that context to imply **false** : in that context, everything equivales **true** .

Often, a range equivalent to **false** arises because we are considering a class of ranges, one of which just happens to equivale **false** . For example, in the naturals, a range like the one in:

(1) $\langle\forall x:\ 0 \le x < S\ :\ \ldots\ \rangle$

equivales **false** when $S = 0$ . Now, if we wished to prove (1) for all $S$ , we could make a case distinction, since the case $S = 0$ comes for free. But often we don't: often we find we can prove (1) without such a distinction; it all depends on what properties of $S$ are needed for the calculation! (For example, what if we needed to perform a range split in our calculation?)

Analogously, we may be asked to prove a theorem for a class of contexts. For example, where $x$ is an element of a set $S$ . When $S$ is the empty set, our context implies **false** . We could make a case distinction, since the case $S =$ empty set comes for free. But often we don't: often we find we can prove the theorem without such a distinction; it all depends on what properties of $S$ are needed for the calculation! (For example, what if we needed to choose an element of $S$ in our proof?)

* * *

For some reason, no one ever gets confused by (1) when $S = 0$ . No one worries that in such a case "but $x$ doesn't exist!" . They just recognize that in such a case, the range equivales **false** , and so the whole quantification equivales **true** .

But in my experience, a lot of confusion has arisen when a context implies **false** . Suddenly, calculations are thrown to the wind, and philosophy comes in. People worry about even writing down $x$ , and start saying things like "but $x$ doesn't exist!" .

It's clear what's happened here: Because the range of a quantification is written down explicitly in a formula, it's treated symbolically. But because the context of a calculation is not written down, and because it is of type boolean , it gets merged in the mind with philosophical notions like "truth" , "assumptions" , or "prior validities" , or other things of this nature. And when "truths" , "assumptions" , or "validities" are false, logical thought breaks down!

But there is no difference: The context of a calculation is no more and no less than a range of a quantification. It does not reflect "truth" , or "reality" , or "existence" , or anything like this.

Many of us have made this sort of mistake. (For example, recently someone said that an element of the empty set cannot be of type natural, because it doesn't exist. This is only the mildest mistake in this vein.) It should be sobering for us to realize that, although we are all rather well-trained in the calculational style, we have all been misled at one point or another. And why? Simply because we didn't write something down! We should have been more suspicious of our brains. Unfortunately, in 2006 we have no excuse: we have been so admonishing ourselves and our pupils for decades.

Eindhoven, 5 May 2006

Jeremy Weissmann
11260 Overland Ave. #21A
Culver City, CA 90230
USA

jeremy@mathmeth.com