The contingent transition problem

Chris Martens

September 15, 2012

The problem is this: you're representing a state transition system in linear logic, but some state transitions require a contingent "check" before they fire. Such rules might be written this way:

$$[C]A \multimap B$$

Read "on the condition C, we can take a transition from state A to state B."

The question is how to express the uninterpreted syntax – meaning to check C without consuming it – either in standard linear logic or through some other meaningful proof theory.

Attempt 1: Consume and Produce

At first cut, this problem may seem trivial: just express the rule as

$$C \otimes A \multimap C \otimes B$$

But this interpretation is unsatisfying for a number of reasons. First, consider the special case where C is a disjunction:

$$(C_1 \oplus C_2) \otimes A \multimap (C_1 \oplus C_2) \otimes B$$

To fire the transition, we have the knowledge that *a specific one of* C_1 and C_2 is true, but the resulting state loses that knowledge.

Second, consider when C may share resources with A:

$$(c \otimes a_1) \otimes (a_1 \otimes a_2) \multimap (c \otimes a_1) \otimes B$$

In this case, naively copying the precondition to the conclusion doesn't fly, because we actually wanted *part* of it to be consumed:

$$(c \otimes a_1) \otimes (a_1 \otimes a_2) \multimap c \otimes B$$

The generalization of these failures is that it's not valid to interpret *checking a precondition* as deconstructing it and rebuilding it. We really want to leave it entirely untouched, in "read-only" mode. This is why Frank Pfenning refers to it as the "read-only access to resources problem."

Observation: Additivity

For each of the two failure modes above, there seem to be local solutions.

$$[C_1 \oplus C_2]A \multimap B$$

can be interpreted as

$$(C_1 \otimes A \multimap C_1 \otimes B)\&(C_2 \otimes A \multimap C_2 \otimes B)$$

And for any particular *known* overlap between C and A, we can interpret the rule by simply not writing that part on the right of the $\multimap$.

A potential generalization of this idea is that we can *share the context* used to check the precondition with that used to generate the consumed resources.

Attempt 2: Restricted Implication

I was discussing this problem with Taus Brock-Nannestad, who suggested the following connective:

$$\frac{\Delta \vdash A \quad B \vdash C}{\Delta, A \rightsquigarrow B \vdash C} \rightsquigarrow L \qquad \frac{\Delta, A \vdash B}{\Delta \vdash A \rightsquigarrow B} \rightsquigarrow R$$

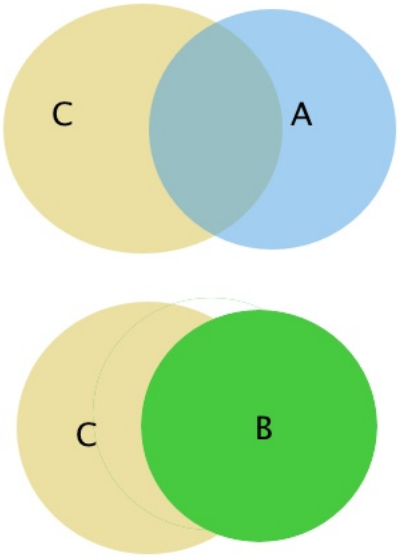
This connective is a special case of $\multimap$ where to use such a rule on the left, the *entire remainder of the context* must go toward proving the argument and only the consequent B may be used to continue. The idea was to represent $[C]A \multimap B$ as $(C \otimes \top) \rightsquigarrow A \multimap B$.

Unfortunately, this connective fails the commutative cases of cut.

Furthermore, it should't really be expected to adequately represent read-only access. It correctly captures the idea of matching the current context against $C \otimes \top$, but there is no reason why the argument A shouldn't have access to those resources. Indeed, this fails to capture the overlapping resources example.

Observation: overlapping resources

The idea of using $C \otimes \top$ to match the context against the condition seems to make sense, but really what's going on is that we want something like the image below, where the top set of overlapping resources is replaced by the bottom one, B's opacity meant to indicate that it subsumes the overlap.



Attempt 3: Additive implication

Noam suggested that at least on the left, we might replace $[C]A \multimap B$ with $(C \multimap 0) \oplus (A \multimap B)$ (or indeed in general, define $[C]A$ as $(C \multimap 0) \oplus A$. This gives us the sort-of additive property we need: in one copy of the context, we "check" C by assuming it to imply a successful sequent. This already has the problem that we might get the sequent to succeed in some other way with some other rules; possibly those could be dodged by forcing focusing. This avenue seems worth exploring, but it seems unsatisfying to need to change the proof search semantics (also, potentially in violation of completeness of focusing).

Attempt 4: Partially Additive Implication

The last thing I came up with is the following left rule for yet another funny form of implication:

$$\frac{\Delta_1 \vdash A \quad \Delta_1, \Delta_2, B \vdash C}{\Delta_1, \Delta_2, A \twoheadrightarrow B \vdash C}$$

...but a lot of fiddling with potential right rules yields nothing that satisfies both identity and (principal) cut. Two thoughts:

1. Maybe it can only appear on the left, like a validity judgment? When is this, er, valid?

2. Maybe an affinity judgment or context is the right judgmental machinery to augment with. but so far I haven't gotten either to work.