Mismatch II

Robert J. Simmons (Carnegie Mellon)
October 2, 2009

The mismatch between meta and object levels can be partially dealt with by understanding much of the context of an inference rule as ambient. However, this generalization is not sufficient for describing calculi like ordered logic or the logic of bunched implications. We briefly discuss two ways around this problem, one which is connected to polarized logic [NZ], the other of which is connected to hybrid logic [JR].

This note borrows extensively from the style and content of Alessio Guglielmi's 2003 manuscript [AG].

1 The Mismatch Between Meta and Object Level In Classic Sequent Calculi

Alessio Guglielmi [AG] describes a problematic mismatch between the meta and object level in many calculi. His example of a calculus that does not have an issue with mismatch is Gentzen's sequent calculus:

$$\wedge\ \frac{\vdash A, \Gamma \qquad \vdash B, \Delta}{\vdash A \wedge B, \Gamma, \Delta}, \qquad \vee\ \frac{\vdash A, B, \Gamma}{\vdash A \vee B, \Gamma}$$

This is because, if we give the conjunction between branches the meaning of meta-level `and' and give the horizontal stroke the meaning of meta-level `implies,' we get the correct meaning of the rules:

$$(A \vee \Delta) \wedge (B \vee \Delta) \Rightarrow ((A \wedge B) \vee \Gamma \vee \Delta)$$
$$(A \vee B \vee \Delta) \Rightarrow ((A \vee B) \vee \Delta)$$

However, in a classical linear sequent calculus, there is a mismatch, because in the rule for multiplicative conjunction and the rule for additive conjunction, the conjunction between the branches absolutely must have different meanings for the two definitions of conjunction.

$$\wp\ \frac{\vdash A, B, \Gamma}{\vdash A \wp B, \Gamma} \qquad \otimes\ \frac{\vdash A, \Gamma \qquad \vdash B, \Delta}{\vdash A \otimes B, \Gamma, \Delta} \qquad \&\ \frac{\vdash A, \Gamma \qquad \vdash B, \Gamma}{\vdash A \& B, \Gamma}$$

$$(A \wp B \wp \Gamma) \multimap ((A \wp B) \wp \Gamma)$$
$$(A \wp \Gamma) \otimes (B \wp \Delta) \multimap ((A \otimes B) \wp \Gamma \wp \Delta)$$
$$(A \wp \Gamma) \& (B \wp \Gamma) \multimap ((A \& B) \wp \Gamma)$$

$$(A \wp \Gamma) \& (B \wp \Delta) \text{ -/o } ((A \otimes B) \wp \Gamma \wp \Delta)$$
$$(A \wp \Gamma) \otimes (B \wp \Gamma) \text{ -/o } ((A \& B) \wp \Gamma)$$

2 Resolving Mismatch In Intuitionstic Logic

We can get a little bit more pre-mismatch mileage out of the more-or-less standard way of interpreting intuitionstic systems of sequent calculus and natural deduction. Critically, the context is usually not explicitly mentioned in these presentations; therefore when we talk about "what does conjunction between branches mean" the consideration of exactly how the branches are split naturally follows. This addresses the aforementioned problem in linear logic, as more choices for how to split the branches should naturally lead to more varieties of conjunction. However, it also brings up more questions about how the judgment stroke "⊢" and the horizontal line should be represented.

We often describe this problem as being one of wanting to correctly encode the propositional rules of our language, for instance encoding the and-introduction rule as "pf(A) ∧ pf(B) ⇒ pf(A ∧ B)", but we will mostly neglect this view in this section.

2.1 Reconsidering Propositional Logic

These are the usual rules for conjunction in a natural deduction presentation of propositional logic:

```
   Γ ⊢ A    Γ ⊢ B          Γ ⊢ A ∧ B         Γ ⊢ A ∧ B
∧I ===============,  ∧E1 ===========,  ∨E2 ===========
     Γ ⊢ A ∧ B               Γ ⊢ A             Γ ⊢ B
```

If we define propositional logic so that it is never necessary to split the context, it is natural that we only need one notion of conjunction, and the rules are obvious:

```
A ∧ B ⇒ A ∧ B
A ∧ B ⇒ A
A ∧ B ⇒ B
```

Both possible sequent calculus left rules also make sense:

```
   Γ, A, B ⊢ C            Γ, A ⊢ C             Γ, B ⊢ C
∧L =============,   ∧L1 =============,   ∧L2 =============
   Γ, A ∧ B ⊢ C         Γ, A ∧ B ⊢ C         Γ, A ∧ B ⊢ C
```

```
(A ⇒ B ⇒ C) ⇒ (A ∧ B ⇒ C)
(A ⇒ C) ⇒ (A ∧ B ⇒ C)
(B ⇒ C) ⇒ (A ∧ B ⇒ C)
```

These left rules have forced us to explain how we understand not only the judgment stroke "⊢" but the relationship between it and the context formation operator ",". Here we have chosen to see both "⊢" and "," as being described by implication, so that writing "Γ, A, B

$\vdash C$" is essentially like writing "$\Gamma \vdash (A \vdash (B \vdash C))$". We could have chosen the alternative of understanding "," as conjunction, in which case the translation of the first rule would have been an instance of identity:

$$(A \land B \Rightarrow C) \Rightarrow (A \land B \Rightarrow C)$$

2.2 Mismatch-Free Linear Logic

In linear logic, we need two be able to describe kinds of conjunction:

```
   Γ ⊢ A    Δ ⊢ B          Γ ⊢ A ⊗ B    Δ, A, B ⊢ C
⊗I ===============,  ∧E ==========================
     Γ, Δ ⊢ A ⊗ B                 Γ, Δ ⊢ C

   Γ ⊢ A    Γ ⊢ B          Γ ⊢ A & B          Γ ⊢ A & B
&I ===============,   &E1 ==========,   &E2 ==========
     Γ ⊢ A & B               Γ ⊢ A              Γ ⊢ B
```

These can be captured just fine, because a "context splitting" (multiplicative) conjunction is represented by $\otimes$ and a "context duplicating" (additive) conjunction is represented by &:

$$A \otimes B \multimap A \otimes B.$$
$$(A \otimes B) \otimes (A \multimap B \multimap C) \multimap C$$
$$A \,\&\, B \multimap A \,\&\, B$$
$$A \,\&\, B \multimap A$$
$$A \,\&\, B \multimap B$$

The rule for disjunction elimination is neat: it uses both context splitting and context duplicating conjunction:

```
   Γ ⊢ A ⊕ B    Δ, A ⊢ C    Δ, B ⊢ C
∧E ===================================
           Γ, Δ ⊢ C
```

$$(A \oplus B) \otimes ((A \multimap C) \,\&\, (B \multimap C)) \multimap C$$

Again, the left rules for the sequent calculus presentation all work without fuss:

```
    Γ, A, B ⊢ C           Γ, A ⊢ C            Γ, A ⊢ C   Γ, B ⊢ C
⊗L ==============,   &L1 =============,   ∨L ====================
    Γ, A ⊗ B ⊢ C          Γ, A & B ⊢ C           Γ, A ⊕ B ⊢ C
```

$$(A \multimap B \multimap C) \multimap (A \otimes B \multimap C)$$
$$(A \multimap C) \multimap (A \,\&\, B \multimap C)$$
$$(A \multimap C) \,\&\, (B \multimap C) \multimap (A \oplus B \multimap C)$$

3 Encountering Mismatch Again In Ordered Logic

In ordered logic, we re-encounter what seems to be a new kind of mismatch when we look at some of the sequent calculus rules

$$\bullet R\ \frac{\Omega_1 \vdash A \qquad \Omega_2 \vdash B}{\Omega_1,\ \Omega_2 \vdash A \bullet B}, \qquad \bullet L\ \frac{\Omega_1,\ A,\ B,\ \Omega_2 \vdash C}{\Omega_1,\ A \bullet B,\ \Omega_2 \vdash C}$$

The •R rule can be perfectly well represented:

(A • B) -• (A • B)

Interestingly, the implication -• can equally well represent left implication ↣ or right implication ↠. But how to represent •L?

As a slight digression, the pattern continues with the rules for those connectives:

$$\twoheadrightarrow R\ \frac{\Omega,\ A \vdash B}{\Omega \vdash A \twoheadrightarrow B}, \qquad \twoheadrightarrow L\ \frac{\Omega A \vdash A \qquad \Omega_1,\ B,\ \Omega_2 \vdash C}{\Omega_1,\ A \twoheadrightarrow B,\ \Omega A,\ \Omega_2 \vdash C}$$

$$\rightarrowtail R\ \frac{A,\ \Omega \vdash B}{\Omega \vdash A \rightarrowtail B}, \qquad \rightarrowtail L\ \frac{\Omega A \vdash A \qquad \Omega_1,\ B,\ \Omega_2 \vdash C}{\Omega_1,\ \Omega A,\ A \rightarrowtail B,\ \Omega_2 \vdash C}$$

The right rules are representable, and use the ambivalent -• for the horizontal bar but use the corresponding connective in the premise.

(A ↠ B) -• (A ↠ B)
(A ↣ B) -• (A ↣ B)

Since all these rules essentially just look like the identity, it's probably easier to explain all of these right rules in terms of encoding the propositional fragment in the first order fragment, where we would see something like this:

(pf(A) • pf(B)) -• pf(A • B)
(pf(A) ↠ pf(B)) -• pf(A ↠ B)
(pf(A) ↣ pf(B)) -• pf(A ↣ B)

So we have two problems. The seemingly small issue is that we no longer have an implication that really represents the vertical bar faithfully in the right rules. The seemingly large issue is that there appears to be no way to correctly represent the left rules without mentioning the context, which has been our entire program up to this point.

Similar issues arise in the logic of bunched implications, which has a context represented by nested bunches of additive (persistent predicate logic-like) and multiplicative (linear logic-like) contexts. These again show up primarily in left or elimination rules, where we have to "frame out" a part of the context in order to examine or modify it.

```
    Γ(A, B) ⊢ C            Δ ⊢ A * B    Γ(A, B) ⊢ C
 *L =============,     *E ==========================
    Γ(A * B) ⊢ C                  Γ(Δ) ⊢ C
```

The logic of bunched implications is also similar to ordered logic in that there are two implications "-*" and "⇒" and the interpretation of the "⊢" is ambiguous between the two.

4 Dealing With Mismatch

We discuss two ways of dealing with the problems described above. The first method is to give a different interpretation to left rules, one that is justified by polarized logic.

4.1 Polarized Metalogic

Let's consider the linear logic rule ⊸L:

```
     Δ ⊢ A     Γ, B ⊢ C
 ⊸L ====================
      Γ, Δ, A ⊸ B ⊢ C
```

If we remember that our first step before was to ignore the excess context Γ and Δ, we can consider ignoring the right-hand side C as well as an unnecessary part of the "context." In that case, we interpret the rule in a bottom-up way as 'If we can frame off a part of the context that is the formula "A ⊸ B" and another part of the context that implies A, then we can replace that entire framed-off part of the context with B.' Since we're working entirely on the left, treating implication in that bottom-up direction makes a lot of sense, and this reading naturally corresponds to the formula:

$$A \otimes (A \multimap B) \multimap B$$

This is a little clearer from the perspective of polarized logic, where we would describe the rule as follows:

$$conc^{-}(A) \otimes hyp^{+}(A \multimap B) \multimap hyp^{+}(B)$$

This strategy lets us give polarized encodings to the left rules in ordered logic that we were unable to give before.

•L: $hyp^{+}(A \bullet B)$ -• $(hyp^{+}(A) \bullet hyp^{+}(B))$
↣L: $conc^{-}(A) \bullet hyp^{+}(A \rightarrowtail B)$ -• $hyp^{+}(B)$
↠L: $hyp^{+}(A \twoheadrightarrow B) \bullet conc^{-}(A)$ -• $hyp^{+}(B)$

4.2 Hybrid Metalogic

The solution that is in line with the tools Jason Reed explored in his thesis advocate a hybrid approach where computation remains "bound" to the right-hand side [JR]. In this setting, we give up on a close correspondence between the metalogic and the logic, and instead use equational properties of the unrestricted logic to encode the appropriate context discipline:

$$\bullet L: (\Pi\beta,\gamma.\ hyp(A)@\beta \to hyp(B)@\gamma \to conc(C)@(w\cdot\beta\cdot\gamma\cdot w')) \to (hyp(A \bullet B)@\alpha \to conc(C)@(w\cdot\alpha\cdot w'))$$

$$\rightarrowtail L: (\Pi\beta.\ conc(C)@wa \to hyp(B)@\beta \to conc(C)@(w\cdot\beta\cdot w')) \to (hyp(A \rightarrowtail B)@\alpha \to conc(C)@(w\cdot wa\cdot\alpha\cdot w'))$$

$$\twoheadrightarrow L: (\Pi\beta.\ conc(C)@wa \to hyp(B)@\beta \to conc(C)@(w\cdot\beta\cdot w')) \to (hyp(A \twoheadrightarrow B)@\alpha \to conc(C)@(w\cdot\alpha\cdot wa\cdot w'))$$

References

[AG] Alessio Guglielmi. Mismatch. Manuscript, 2003. http://cs.bath.ac.uk/ag/p/AG9.pdf

[JR] Jason C. Reed. A Hybrid Metalogical Framework. Ph.D. Thesis, Carnegie Mellon University. July 2009. Available as Technical Report CMU-CS-09-155.

[NZ] Noam Zeilberger. The Logical Basis of Evaluation Order and Pattern-Matching. Ph.D. Thesis, Carnegie Mellon University. April 2009. Available as Technical Report CMU-CS-09-122.