RECIPE

Alessio Guglielmi (TU Dresden)
28.9.2002 - updated on 27.10.2002

A family of tautologies that hold in most logics can be used, following a simple recipe, to produce deductive systems where all rules are local, including cut.

In the following, CL stands for classical logic, LL for linear logic and ML for (some flavour of) modal logic. Consider the following tautologies:

(1) $((A \Rightarrow B) \wedge (C \Rightarrow D)) \Rightarrow (A \wedge C) \Rightarrow (B \wedge D)$ (in CL and ML);
(2) $((A \multimap B) \otimes (C \multimap D)) \multimap (A \otimes C) \multimap (B \otimes D)$ (in LL);
(3) $((A \multimap B) \,\&\, (C \multimap D)) \multimap (A \,\&\, C) \multimap (B \,\&\, D)$ (in LL);
(4) $\forall x.(A \Rightarrow B) \Rightarrow \forall x.A \Rightarrow \forall x.B$ (in CL and ML);
(5) $\forall x.(A \multimap B) \multimap \forall x.A \multimap \forall x.B$ (in LL);
(6) $!(A \multimap B) \multimap\, !A \multimap\, !B$ (in LL);
(7) $\Box(A \Rightarrow B) \Rightarrow \Box A \Rightarrow \Box B$ (in ML).

It is remarkable that all these formulae are valid: they all have the same `distributive´ shape of a logical relation over implication (remember that implication associates to the right). The scheme is always the same: (1), (2) and (3) are binary cases, the others are unary. In pictures:

$$\begin{array}{c}A\\ \Downarrow\\ B\end{array} \wedge \begin{array}{c}C\\ \Downarrow\\ D\end{array} \Rightarrow \begin{array}{c}A \wedge C\\ \Downarrow\\ B \wedge D\end{array} \quad \text{and} \quad \forall x.\begin{array}{c}A\\ \Downarrow\\ B\end{array} \Rightarrow \begin{array}{c}\forall x.A\\ \Downarrow\\ \forall x.B\end{array}\ .$$

These laws lead straightforwardly to atomicity of cut in the calculus of structures [WS], when we have involutive negation. If they hold (and they always do), we can reduce identity and cut to atomic form by following a simple recipe which requires no creativity. The recipe also holds for exotic systems like BV [AG] and NEL [GS2].

I believe we can also prove a general splitting theorem for all subfragments of the structural core + interaction fragment. This would imply cut elimination and admissibility of the up fragment as easy corollaries.

The Recipe for the Structural Core of a System (Binary Case)

Ingredients

1) An implication $\rightarrow$;
2) an involutive negation $\neg$ and a disjunction $+$ related to $\rightarrow$ by $(A \rightarrow B) \equiv (\neg A + B)$ (so, using $\neg$ and $+$ is just an equivalent way of writing $\rightarrow$);

3) a conjunction $\otimes$ and its dual $\oplus$ (so far totally independent from $\to$), such that $\neg(A \otimes B) \equiv (\neg A \oplus \neg B)$.

This recipe corresponds to the law

(RB) $((A \to B) \otimes (C \to D)) \to (A \otimes C) \to (B \otimes D)$,

which we assume valid.

<u>Execution</u>

Build a (sound) inference rule corresponding to (RB) as follows:

$$\frac{(A \to B) \otimes (C \to D)}{(A \otimes C) \to (B \otimes D)} \equiv \frac{(\neg A + B) \otimes (\neg C + D)}{(\neg A \oplus \neg C) + (B \otimes D)}$$

$$\equiv \frac{(B + A) \otimes (D + C)}{(B \otimes D) + (A \oplus C)} .$$

The last rule corresponds to the core structural down rule for the couple $\otimes$-$\oplus$; the corresponding up rule is obtained by duality.

Examples

<u>With and Plus in LL</u>

The recipe above gives us

$$\frac{(B \# A) \& (D \# C)}{(B \& D) \# (A \oplus C)} ,$$

where # is par, & is with and $\oplus$ is plus. This rule and its dual belong to linear logic as presented in the calculus of structures [LS], of course.

<u>And and Or in CL</u>

This is a very common case: the dual of the logical relation which \`distributes´ with implication corresponds to the disjunction associated to the same implication. I'll use here the usual notation we adopt in the calculus of structures (see for example [BT]): [A,B,C] stands for $A \vee B \vee C$ and (A,B,C) stands for $A \wedge B \wedge C$. The recipe yelds the dual rules:

$$s{\downarrow}\frac{S([B,A],[D,C])}{S[(B,D),A,C]} \quad \text{and} \quad s{\uparrow}\frac{S([B,D],A,C)}{S[(B,A),(D,C)]}$$

(called *switch down* and *switch up*). In this case we can adopt a self-dual rule which can equivalently replace both of them, the usual *switch*:

```
 S([R,U],T)
s---------- .
 S[(R,T),U]
```

<u>Of-course and Why-not in LL</u>

The unary version of the recipe gives rise to the following *local* version of promotion (here shown with copromotion) [GS1]:

```
  ![B,A]           (?B,!A)
p↓-------   and   p↑------- .
  [!B,?A]           ?(B,A)
```

Comments and Conjectures

Remember that the *structural core* of a system is defined as the set of rules necessary for reducing identity and cut to atomic form without loss of derivability (see [GS1], for example).

The recipe provides all (and only) the rules of the structural core. (This can be proved easily, see for example [GS1]). Interaction rules are always a given, so the recipe provides a large fragment of any given sought-for system. In classical logic, only contraction and weakening are left out and have to be `invented´.

All the rules produced by the recipe are *local*, in the sense discussed, for example, in [BT]. This means that the systems provided by the recipe are particularly good, since locality is not always possible in the sequent calculus.

As decomposition theorems show (see for example [AG], [BT] and [GS1]), the non-core rules can always be pushed to the extremes of a derivation, and the non-core up rules are always trivially admissible. Then a cut elimination theorem for the core suffices for proving cut elimination in the entire system, in the presence of decomposition (which apparently always works, I don't know why). We used this technique in [GS2].

Since our simple recipe can generate very complex core fragments, it could be convenient to prove cut elimination once and for all for any generic system given by the recipe. I think a splitting argument should work [AG]:

Conjecture *Every system composed by interaction + structural core fragments enjoys a splitting theorem, and then it enjoys cut elimination and in general admissibility of the up fragment.*

If this works, it should then be straightforward to extend the result to include those non-core rules that do not create redexes while going up in a derivation, like weakening and atomic contraction.

This would move the remaining difficulty for very powerful *automatic* cut elimination upon medial rules (remember that contraction = atomic contraction + medial).

But then, also medial rules appear to obey a recipe. We still have to investigate, but if this turns out to be true, then there is hope of having an extremely general splitting theorem, which would encompass classical logic and a number of different logics, some of them truly exotic.

References

[AG] Alessio Guglielmi. A system of interaction and structure. Technical Report WV-02-10, Dresden University of Technology, 2002, URL: http://www.ki.inf.tu-dresden.de/~guglielm/Research/Gug/Gug.pdf, submitted.

[BT] Kai Brünnler and Alwen Tiu. A local system for classical logic. In R. Nieuwenhuis and A. Voronkov, editors, *LPAR 2001*, volume 2250 of *Lecture Notes in Artificial Intelligence*, pages 347-361. Springer-Verlag, 2001. URL: http://www.ki.inf.tu-dresden.de/~kai/LocalClassicalLogic-lpar.pdf.

[GS1] Alessio Guglielmi and Lutz Straßburger. Non-commutativity and MELL in the calculus of structures. In L. Fribourg, editor, *CSL 2001*, volume 2142 of *Lecture Notes in Computer Science*, pages 54-68. Springer-Verlag, 2001. URL: http://www.ki.inf.tu-dresden.de/~guglielm/Research/GugStra/GugStra.pdf.

[GS2] Alessio Guglielmi and Lutz Straßburger. A non-commutative extension of MELL. In M. Baaz and A. Voronkov, editors, *LPAR 2002*, volume 2514 of *Lecture Notes in Artificial Intelligence*, pages 231-246. Springer-Verlag, 2002. URL: http://www.ki.inf.tu-dresden.de/~lutz/NEL.pdf.

[LS] Lutz Straßburger. A local system for linear logic. In M. Baaz and A. Voronkov, editors, *LPAR 2002*, volume 2514 of *Lecture Notes in Artificial Intelligence*, pages 388-402. Springer-Verlag, 2002. URL: http://www.ki.inf.tu-dresden.de/~lutz/lls-lpar.pdf.

Web Site

[WS] http://www.ki.inf.tu-dresden.de/~guglielm/Research.