NORMALISATION WITHOUT CUT ELIMINATION

Alessio Guglielmi (TU Dresden)
10.5.2002 - updated on 25.2.2003

Inside the calculus of structures [WS], I propose a possible notion of equivalence for proofs in classical logic, sort of a normal form for case analysis.

Let's consider the following formula (not intuitionistically valid):

$$F = \exists x.\forall y.(p(x) \Rightarrow p(y)) \; .$$

We can prove its validity by knowing the two lemmas:

(L_1) If $\forall z.p(z)$ then F is true because its matrix's conclusion is always true.

(L_2) If $\neg\forall z.p(z)$ then F is true because its matrix's premise can be falsified.

Since the lemmas have dual hypotheses, we can build the following proof in the sequent calculus:

```
                ------          ------
                 \Π_1/           \Π_2/
                  \/              \/
        ∀z.p(z) |- F    ¬∀z.p(z) |- F
     cut---------------------------- ,
                     |- F, F
                  c_R-------
                      |- F
```

where Π_1 and Π_2 are easy cut-free proofs corresponding to L_1 and L_2.

What happens when we eliminate the cut? We get:

```
                     -------------------------
                      p(x), p(z) |- p(z), p(y)
       ⇒_R; ⇒_R-------------------------------------
                |- p(x) ⇒ p(z) ,        p(z) ⇒ p(y)
            ∀_R----------------------------------------
                |- p(x) ⇒ p(z) ,    ∀y.(p(z) ⇒ p(y))
            ∃_R----------------------------------------
                |- p(x) ⇒ p(z) , ∃x.∀y.(p(x) ⇒ p(y))
        ∀_R--------------------------------------------
            |- ∀y.(p(x) ⇒ p(y)), ∃x.∀y.(p(x) ⇒ p(y))
     ∃_R-----------------------------------------------
          |- ∃x.∀y.(p(x) ⇒ p(y)), ∃x.∀y.(p(x) ⇒ p(y))
     c_R----------------------------------------------- .
                   |- ∃x.∀y.(p(x) ⇒ p(y))
```

The point is that, according to me, this proof is intuitively *essentially different* than the other one corresponding to the case analysis above. Of course, if I want to be convincing, I have to make this \`essential difference´ into some sort of technical notion.

If the only normalisation tool we have is cut elimination (in the sequent calculus), we are tempted to try to convince ourselves that the two proofs *must* be the same, in some denotational sense. On the other hand, given a more refined normalisation tool than cut elimination in the sequent calculus, perhaps:
1) we *don't want* to do cut elimination anymore, and
2) we *don't need* to do it (in its full generality)!

We Don't Want To Eliminate Cuts: In the calculus of structures we have decomposition theorems. All proofs can be reduced to normal forms of the kind:

```
              identities ===
                          T
      all the other rules ===
                          S
                    cuts ===
                          R
              weakenings === .
                          Q
            contractions ===
                          P
```

This fits the scheme of the first proof above. Kai Brünnler and Alwen Tiu showed that this is impossible, in general, in the sequent calculus, but possible in the calculus of structures [KB,BT].

So, why not considering *this* the right normal form? It must be noted that we have to do some work in order to eliminate certain evil cuts (see below) and in order to equate several derivations that only differ for trivial permutations (this is non-trivial).

We Don't Need To Eliminate Cuts: But isn't the cut evil? Why do we retain some cuts? The short answer is that, in fact, there are two kinds of cut, the good and the evil, and you can discriminate between them when they are atomic.

The cut in the first proof above is good, because its eigenformula is $\forall z.p(z)$, which is an atom already present in F: one doesn't really have to invent anything, such a \`guess´ can be made finitarily.

An evil cut is a cut that introduces, going up, atoms that are not present before. But these cuts can trivially be removed, as Kai Brünnler and I show in [BG].

* * *

So, in summary, I propose the following idea:

1) reduce the cuts to atomic form (trademark feature of the calculus of structures);
2) eliminate those cuts that introduce new stuff (which correspond to detours in natural deduction) and keep the others;
3) push all weakenings and contractions down;
4) push all cuts immediately above weakenings;
5) quotient under permuting of inference rules.

Given a normal form of this kind, we can read several phases in the bottom up construction of the proof:

1 - Contraction: Generate as many copies of the formula (or of some of its subformulae) as there are branches in the tree of the case analysis. In the example there are two.

2 - Weakening: Eliminate all unnecessary material from each branch. In the example, the matrix's premise in branch Π_1 and the matrix's conclusion in branch Π_2.

3 - Finitary Cut: Generate all (atomic) dual hypotheses, chosen among the finite amount of atoms available. In the example we generate $\forall z.p(z)$ and $\neg\forall z.p(z)$.

4 - Linear Phase: Organize in branches the formulae you got from 1 and 2 and combine the stuff until you generate identities. This is done in some linear extension of predicate *multiplicative linear logic*, since all the additive rules are no more necessary.

5 - Identities: Just check logical axioms.

In all phases, and most notably in 1 and 2, one needs to apply rules deep inside formulae, and here is where the sequent calculus fails.

The existence of the above decomposition has been proven semantically with atomic contraction. It is still open whether this can be the outcome of a normalising rewriting procedure.

Conjecture *It is possible to decompose a proof the way above by a rewriting procedure, and it is possible to do so in a (graphic) representation of proofs where normal forms are unique.*

If we manage to do this, we have a notion of equivalence classes of proofs, which could be suitable denotations for the associated computational process. The question, perhaps, is *what is* this computational process.

On the other hand, even if the computational notion is not very meaningful, as I suspect, we still can get a good notion of what a proof really is *in essence*: it would be its case analysis plus a linear combinatorial phase.

The proof of the example, in the calculus of structures, is:

```
                 t
  i↓------------------------ (identities)
    [   ∀z.p(z) , ∃x.¬p(x) ]
i↓----------------------------------------------------- (ident.)
  ( [   ∀z.p(z) , ∃x.¬p(x) ] , [ ∃z.¬p(z) , ∀y.p(y) ] )
 s----------------------------------------------------- (lin. ph.)
  [ ( [ ∀z.p(z) , ∃x.¬p(x) ] , ∃z.¬p(z) ) , ∀y.p(y) ]
 s--------------------------------------------------- (linear ph.)
  [ (   ∀z.p(z) , ∃z.¬p(z) ) , ∀y.p(y) , ∃x.¬p(x) ]
i↑------------------------------------------------- (cut)
             [ ∀y.p(y) , ∃x.¬p(x) ]
w↓;w↓------------------------------------------ (2 weakenings)
     [ ∃x.∀y.[¬p(x),p(y)] , ∃x.∀y.[¬p(x),p(y)] ]
   c↓------------------------------------------ (contraction) .
       ∃x.∀y.[¬p(x),p(y)]
```

I didn't use local rules for identities and cut, for simplicity, but we know that we can use them [BT]. If one performs full cut elimination, this proof becomes the more streamlined (but *different*, in my opinion):

```
                        t
              i↓----------------------
                 [ ∀y.p(y) , ∃x.¬p(x) ]
      w↓;w↓------------------------------------------
           [ ∃x.∀y.[¬p(x),p(y)] , ∃x.∀y.[¬p(x),p(y)] ]
         c↓------------------------------------------ .
             ∃x.∀y.[¬p(x),p(y)]
```

By the way, please notice how happily we manage quantifiers in the calculus of structures.

References

[BG] Kai Brünnler and Alessio Guglielmi. Consistency without cut elimination. Technical Report WV-02-16, Dresden University of Technology, 2002, URL: http://www.ki.inf.tu-dresden.de/~guglielm/Research/Notes/AG4.pdf.

[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.

[KB] Kai Brünnler. Two restrictions on contraction. Technical Report WV-02-04, Dresden University of Technology, 2002, URL: http://www.ki.inf.tu-dresden.de/~kai/RestrictionsOnContraction.pdf, submitted.

Web Site

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