

CONSISTENCY WITHOUT CUT ELIMINATION

Kai Brunnler and Alessio Guglielmi (TU Dresden)

22.4.2002 - updated on 22.9.2007

In this note we show how to get consistency for first order classical logic, in a purely syntactic way, without going through cut elimination. The procedure is very simple and it uses the calculus of structures [WS] in an interesting way. It also shows how finitariness (in the sense of finite choice of premises for each rule) is actually a *triviality* (contrarily to what one would guess from textbooks).

We all know that Gödel's incompleteness forbids easy solutions for very expressive cases, so this method can't be a magic bullet for the foundations of mathematics. Nonetheless, there are two points that we believe are worthy:

- 1) The method is general and could be helpful for easy checking of consistency in other cases.
- 2) It makes clear that *there is no deep connection* between cut elimination and finitariness, as entailed by the subformula property (which in turn entails consistency). This is folklore for the sequent calculus, but in the case of the calculus of structures the construction is more perspicuous.

First we give a general summary of the steps involved, for a generic logic; then we get into the details for classical logic.

Consistency in Four Easy Steps

Step 1

Write your logic in the calculus of structures, making sure that the rules are defined according to our recipe [AG1]. This ensures that the cut rule can trivially be replaced by a cut rule in atomic form. Then, you have the atomic cut rule, which is

$$\begin{array}{c}
 S(a, \neg a) \\
 a \uparrow \text{-----} \cdot \\
 S\{\perp\}
 \end{array}$$

where \perp is the unit for disjunction.

(In the sequent calculus, the cut rule cannot trivially be restricted to atomic form. You'd have to go through full-blown cut elimination.)

Step 2

Replace $ai\uparrow$ by $fai\uparrow$ (finitary cut, see [AG2]), which is:

$$\frac{S(a, \neg a)}{fai\uparrow \text{-----} S\{\perp\}} \quad \text{where } a \text{ or } \neg a \text{ appears in } S\{ \}.$$

Provability is not affected by this substitution: proving this claim is almost trivial, and it probably always is for every system (look at the proof below for classical logic).

This means that you get a system that proves the same logic, but all its rules can only be applied in a finite number of different ways: the equivalent of a subformula property.

This is a weak form of cut elimination: you only eliminate *certain* cuts, those that introduce atoms that have nothing to do with the conclusion of the proof. Doing this *does not require* going through each and every mutual relation of rules in the system.

Step 3

Prove that falsehood, or empty, or any given structure, can not be proved. This is consistency in a weak form, if you like. This is very easy for classical logic and BV [AG3]. We guess it's the same for other systems.

Step 4

Prove a stronger form of consistency, like: you can't have a proof of R and another of $\neg R$ at the same time. There is a nice easy trick for doing this in the calculus of structures, given 'weak' consistency.

* * *

At this point you're done. Of course, you still can perform the equivalent of cut elimination in the sequent calculus.

Consistency for Propositional Classical Logic

We apply the procedure above to classical propositional logic. We refer here to systems in [BT], reproduced below for convenience.

Step 1

Classical logic in the calculus of structures: here is a presentation with atomic interaction rules and non-atomic contraction and weakening. Things work the same for the entirely atomic presentation.

1) Language:

- 1) R, T and U stand for generic structures;
- 2) a is an atom, which is a structure;
- 3) f and t are structures, but not atoms; f stands for the unit 'false', t stands for the unit 'true';
- 4) $[R_1, \dots, R_n]$ and (R_1, \dots, R_n) are structures, for $n > 0$; $[R, T]$ stands for $R \vee T$, (R, T) stands for $R \wedge T$;
- 5) the structure $\neg R$ is the negation of R;
- 6) $S\{ \}$ is a structure with a hole that does not occur inside a negation;
- 7) $S[R, T]$ and $S(R, T)$ are shortcuts of $S\{[R, T]\}$ and $S\{(R, T)\}$, respectively;

2) Equations:

Structures are considered equivalent modulo =, so defined:

- 1) associativity and commutativity of $[...]$ and $(...)$: e.g., $[((a, b), c), d] = [d, (a, c, b)]$;
- 2) De Morgan: e.g., $\neg[a, (b, c)] = (\neg a, [\neg b, \neg c])$.
- 3) units: $[t, t] = t$; $[f, R] = R$;
 $(f, f) = f$; $(t, R) = R$.

3) Rules:

1) Interaction:

$$\begin{array}{ccc}
 S\{t\} & & S(a, \neg a) \\
 ai \downarrow \text{-----} & & ai \uparrow \text{-----} \\
 S[a, \neg a] & & S\{f\}
 \end{array}$$

2) Core:

$$\begin{array}{c}
 S([R, U], T) \\
 s \text{-----} \\
 S[(R, T), U]
 \end{array}$$

3) Non-core:

$$\begin{array}{ccc}
 S[R, R] & & S\{R\} \\
 c \downarrow \text{-----} & & c \uparrow \text{-----} \\
 S\{R\} & & S(R, R) \\
 \\
 S\{f\} & & S\{R\} \\
 w \downarrow \text{-----} & & w \uparrow \text{-----} \\
 S\{R\} & & S\{t\}
 \end{array}$$

A proof of R is a chain of rule instances whose premise is t and conclusion is R, denoted by

$$\begin{array}{c}
 t \\
 | \\
 R
 \end{array}
 .$$

Step 2

The rules $ai \uparrow$ and $w \uparrow$ are non-finitary, since, given a conclusion, they yield infinitely many premises. Being non-core, $w \uparrow$ can immediately be eliminated by using a generic cut and $w \downarrow$:

$$\begin{array}{c}
S(R, [t, f]) = S\{R\} \\
\hline
S[t, (R, f)] \\
w\downarrow \hline
S[t, (R, \neg R)] \\
i\uparrow \hline
S\{t\}
\end{array}
.$$

Generic cuts can be reduced to atomic cuts by recursively applying

$$\begin{array}{c}
S(R, T, [\neg R, \neg T]) \\
\hline
S(R, [\neg R, (T, \neg T)]) \\
i\uparrow \hline
S(R, \neg R) \\
i\uparrow \hline
S\{f\}
\end{array}
.$$

So, the only infinitary rule we are left with is $ai\uparrow$. How do we get rid of it in case we have a proof?

Consider the rule

$$\begin{array}{c}
S(a, \neg a) \\
fai\uparrow \hline
S\{\perp\}
\end{array}
\quad \text{where } a \text{ or } \neg a \text{ appears in } S\{ \}.$$

This rule is finitary, and we're going to show that, given a proof of R , we can always get a proof of R where no $ai\uparrow$ appears, but the only cuts that appear are $fai\uparrow$ instances.

Take a proof of R and individuate the bottommost instance of $ai\uparrow$ that violates the proviso, as in:

$$\begin{array}{c}
t \\
| \\
S(a, \neg a) \\
ai\uparrow \hline
S\{f\} \\
| \\
R
\end{array}
,$$

where neither a nor $\neg a$ appear in $S\{ \}$. We can then replace all occurrences of a and $\neg a$ in the subproof above $S\{f\}$ with t and f , respectively, and we still have a proof of R . All rule instances stay valid or become trivial, for example

$$\begin{array}{c}
S\{t\} \\
ai\downarrow \hline
S[t, f]
\end{array}
\quad \text{and} \quad
\begin{array}{c}
S(t, f) \\
ai\uparrow \hline
S\{f\}
\end{array}
,$$

can just be removed, since $t = [t, f]$ and $(t, f) = f$.

Please notice that if a or $\neg a$ appeared in $S\{ \}$, this would not work, because it could destroy the derivation from $S\{f\}$ to R .

Proceeding inductively upwards, we remove all infinitary atomic cuts.

Step 3

We have to show that there is no proof

$$\begin{array}{c} \mathbf{t} \\ | \\ \mathbf{f} \end{array} .$$

After having done Step 2, we know that we can restrict ourselves to the finitary case, which means that we just have to show that there is no such proof when no atom appears in the proof. So, only f and t can appear in the proof, and we essentially have to look at the units' equations.

Let us call 'boolean structure' any structure freely made from f and t by $[_,_]$ and $(_,_)$. It's easy to show that f is not equal to t . Then we just have to show that no rule, when applied to a boolean structure, has premise t and conclusion f . This is simply done by inspection of all rules, which means inspection of s , $c\downarrow$ and $w\downarrow$. Easy case analysis.

Step 4

We want to prove that if R is provable, then $\neg R$ is not provable. The following flipping construction is not possible in the sequent calculus.

Take

$$\begin{array}{c} \mathbf{t} \\ | \\ R \end{array}$$

and flip it, by negating everything and using the corules for every rule instance (for this we need the entire system above), to get

$$\begin{array}{c} \neg R \\ | \\ \mathbf{f} \end{array} .$$

If $\neg R$ were provable we could then prove f , which is absurd.

Consistency for Predicate Classical Logic

We only sketch the argument, by noting the differences with respect to the propositional case; see [BT] for details of the predicate system.

We eliminate all those instances of atomic cut,

$$\text{ai} \uparrow \frac{S(p(t_1, \dots, t_n), \neg p(t_1, \dots, t_n))}{S\{f\}},$$

in which predicate symbol p does not occur in S .

This is done in the same way as in the propositional case. We replace all occurrences of instances of $p(t_1, \dots, t_n)$ by t and of $\neg p(t_1, \dots, t_n)$ by f . This works because the only extra rules in the predicate case that care about atoms are:

1) the equation

$$\exists x.R = \forall x.R = R \quad \text{if } x \text{ is not free in } R;$$

2) the rule

$$n \downarrow \frac{S\{R[x \leftarrow t]\}}{S\{\exists x.R\}}.$$

Instances of both of them are still valid if t or f replaces arbitrary atoms in R .

References

[AG1] Alessio Guglielmi. Recipe. Manuscript, 2002, URL: <http://cs.bath.ac.uk/ag/p/AG2.pdf>.

[AG2] Alessio Guglielmi. Finitary cut. Manuscript, 2002, URL: <http://cs.bath.ac.uk/ag/p/AG3.pdf>.

[AG3] Alessio Guglielmi. A system of interaction and structure. ACM Transactions on Computational Logic, Vol. 8 (1:1) 2007, pp. 1-64, URL: <http://cs.bath.ac.uk/ag/p/SystIntStr.pdf>.

[BT] Kai Brännler 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.iam.unibe.ch/~kai/Papers/lcl-lpar.pdf>.

Web Site

[WS] <http://alessio.guglielmi.name/res/cos>.