A Finitary System for First Order Logic

Abstract. We present an inference system for classical first order logic in which each inference rule, including the cut, only has a finite set of premises to chose from. The main conceptual contribution of this paper is the possibility of separating different sources of infinite choice, which happen to be entangled in the traditional cut rule.

Keywords. finitary system, cut elimination, consistency, first order predicate logic, deep inference.

1 Introduction

The cut rule in a Gentzen system [5] is infinitary, in the sense that, given its conclusion, there is an infinite choice of premises, corresponding to an infinite choice of eigenformulae. Much effort has been devoted to eliminating this source on infinity in various systems: theorems of cut elimination remove infinite choice together with the cut rule itself, and are at the core of proof theory. There is another source of infinite choice in the bottom-up construction of a first order proof, namely the choice in instantiating an existentially quantified variable. Research grounded in Herbrand's theorem [9] deals with this aspect and is at the core of automated deduction and logic programming.

This paper shows how one can eliminate all sources of infinite choice in a system of first order classical logic in a very simple way. The main idea we exploit is that there are actually two sources of infinite choice in the cut rule: an infinite choice of atoms and an infinite choice in how these atoms can be combined for making formulae. A third source of infinite choice, in the rule for existential quantification, is about choosing substitutions. We are able to separate the various kinds of infinite choice by making use of *deep inference*, which is the possibility of applying inference rules deep inside formulae.

In the sequent calculus, it is impossible to separate the two kinds of infinite choice in a cut rule without going through cut elimination. Instead, in the calculus of structures [7,2,1], whose main feature is the adoption of deep inference, one can straightforwardly reduce the cut rule to its atomic form, which has the advantage of not presenting infinite choice in combining atoms to shape a formula. Similar techniques reduce the instantiation rules into more elementary ones. Infinite choice in the elementary rules so produced can be attacked by simple considerations that essentially limit the range of possibilities to the atoms and terms that already appear in the conclusions of rules.

Systems in the calculus of structures offer the same proof theoretical properties as systems in the sequent calculus, in particular it is possible to prove cut elimination and many other normalisation results. The point we make in this paper is that it is possible to have finitary systems without having to use these more complex methods. As an example, we show how to prove consistency in our setting.

A way of looking at our results is to notice that, when proving in first order logic, the only infinity that remains is in the unboundedness of the proofs themselves, every other aspect in proof construction is finite: at any given step, there are finitely many inferences possible, and each inference rule can only be applied in a finite number of different ways. A good example is the contraction rule: it is always applicable, but only to a finite number of formulae and only in one way (one duplicates the chosen formula). The inference rules we get behave like contraction, for example, a cut can only produce two dual atoms that already appear in a formula.

In previous work, Brünnler and Tiu proved that classical logic can be presented in the calculus of structures in such a way that applying a rule only requires a bounded effort [1,2]. This paper improves on that result by bounding choice.

In Section 2 we introduce first order logic in the calculus of structures and in Section 3 we show how to reduce infinitary rules to finitary ones and we show the consistency argument.

2 First Order Logic in the Calculus of Structures

Variables are denoted by x and y. *Terms*, denoted by τ, are defined as usual in first-order predicate logic. *Atoms*, denoted by a, b, etc., are expressions of the form $p(\tau_1, \dots, \tau_n)$, where p is a *predicate symbol* of *arity* n and $\tau_1, \dots, \tau_n$ are terms. The negation of an atom is again an atom.

The *structures* of the language KSq are generated by

$$S ::= \mathsf{f} \mid \mathsf{t} \mid a \mid [\,\underbrace{S, \dots, S}_{>0}\,] \mid (\,\underbrace{S, \dots, S}_{>0}\,) \mid \exists x S \mid \forall x S \mid \bar{S} \quad .$$

where t and f are the units *true* and *false*, $[S_1, \dots, S_h]$ is a *disjunction*, $(S_1, \dots, S_h)$ is a *conjunction*, $\exists$ is the *existential quantifier* and $\forall$ is the *universal quantifier*. $\bar{S}$ is the *negation* of the structure S. The units are not atoms. Structures are denoted by S, R, T, U and V. *Structure contexts*, denoted by $S\{\ \}$, are structures with one occurrence of $\{\ \}$, the *empty context* or *hole*, that does not appear in the scope of a negation. $S\{R\}$ denotes the structure obtained by filling the hole in $S\{\ \}$ with R. We drop the curly braces when they are redundant: for example, $S[R, T]$ stands for $S\{[R, T]\}$. Structures are *equivalent* modulo the smallest equivalence relation induced by the axioms shown in Fig. 1, where $\boldsymbol{R}$ and $\boldsymbol{T}$ are finite, non-empty sequences of structures. In general we do not distinguish between equivalent structures.

Associativity

$$[\boldsymbol{R},[\boldsymbol{T}],\boldsymbol{U}] = [\boldsymbol{R},\boldsymbol{T},\boldsymbol{U}]$$
$$(\boldsymbol{R},(\boldsymbol{T}),\boldsymbol{U}) = (\boldsymbol{R},\boldsymbol{T},\boldsymbol{U})$$

Units

$$(\mathsf{f},\mathsf{f}) = \mathsf{f} \qquad [\mathsf{f},R] = R$$
$$[\mathsf{t},\mathsf{t}] = \mathsf{t} \qquad (\mathsf{t},R) = R$$

Context Closure

$$\text{if} \quad R = T \quad \text{then} \quad \begin{array}{c} S\{R\} = S\{T\} \\ \bar{R} = \bar{T} \end{array}$$

Vacuous Quantifier

if y is not free in R then $\forall y R = \exists y R = R$

Variable Renaming

$$\text{if } y \text{ is not free in } R \quad \text{then} \quad \begin{array}{l} \forall x R = \forall y R[x/y] \\ \exists x R = \exists y R[x/y] \end{array}$$

Commutativity

$$[R,T] = [T,R]$$
$$(R,T) = (T,R)$$

Negation

$$\begin{array}{c} \bar{\mathsf{f}} = \mathsf{t} \\ \bar{\mathsf{t}} = \mathsf{f} \\ \overline{[R,T]} = (\bar{R},\bar{T}) \\ \overline{(R,T)} = [\bar{R},\bar{T}] \\ \overline{\exists x R} = \forall x \bar{R} \\ \overline{\forall x R} = \exists x \bar{R} \\ \bar{\bar{R}} = R \end{array}$$

Fig. 1. Syntactic equivalence of structures

An *inference rule* is a scheme of the kind $\rho \, \dfrac{S\{T\}}{S\{R\}}$, where ρ is the *name* of the rule, $S\{T\}$ is its *premise* and $S\{R\}$ is its *conclusion*. A (*formal*) *system* $\mathscr{S}$ is a set of inference rules. The *dual* of a rule is obtained by exchanging premise and conclusion and replacing each connective by its De Morgan dual.

A *derivation* Δ is a finite chain of instances of inference rules:

$$\begin{array}{rc} \pi' & \dfrac{T}{V} \\ \pi & \overline{} \\ & \vdots \\ \rho' & \underline{} \\ & U \\ \rho & \overline{R} \end{array} \quad .$$

A derivation can consist of just one structure. The topmost structure in a derivation is called the *premise* of the derivation, and the structure at the bottom is called its *conclusion*. A derivation Δ whose premise is T, whose conclusion is R, and whose inference rules are in $\mathscr{S}$ will be indicated with $\begin{array}{c} T \\ \Delta \Vert \mathscr{S} \\ R \end{array}$. A *proof* $\mathit{\Pi}$ in

$$
\mathsf{i}{\downarrow}\,\frac{S\{\mathsf{t}\}}{S[R,\bar{R}]} \qquad\qquad \mathsf{i}{\uparrow}\,\frac{S(R,\bar{R})}{S\{\mathsf{f}\}}
$$

$$
\mathsf{s}\,\frac{S([R,U],T)}{S[(R,T),U]}
$$

$$
\mathsf{u}{\downarrow}\,\frac{S\{\forall x[R,T]\}}{S[\forall xR,\exists xT]} \qquad\qquad\qquad\qquad \mathsf{u}{\uparrow}\,\frac{S(\exists xR,\forall xT)}{S\{\exists x(R,T)\}}
$$

$$
\mathsf{w}{\downarrow}\,\frac{S\{\mathsf{f}\}}{S\{R\}} \qquad\qquad \mathsf{w}{\uparrow}\,\frac{S\{R\}}{S\{\mathsf{t}\}}
$$

$$
\mathsf{c}{\downarrow}\,\frac{S[R,R]}{S\{R\}} \qquad\qquad \mathsf{c}{\uparrow}\,\frac{S\{R\}}{S(R,R)}
$$

$$
\mathsf{n}{\downarrow}\,\frac{S\{R[x/\tau]\}}{S\{\exists xR\}} \qquad\qquad\qquad\qquad \mathsf{n}{\uparrow}\,\frac{S\{\forall xR\}}{S\{R[x/\tau]\}}
$$

Fig. 2. System SKSgq

the calculus of structures is a derivation whose premise is the unit true. It will be denoted by $\begin{array}{c}\Pi\,\overline{\|}\,\mathscr{S}\\ R\end{array}$. A rule ρ is *derivable* for a system $\mathscr{S}$ if for every instance of $\rho\,\dfrac{T}{R}$ there is a derivation $\begin{array}{c}T\\ \|\,\mathscr{S}\\ R\end{array}$. A rule ρ is *admissible* for a system $\mathscr{S}$ if for every proof $\begin{array}{c}\overline{\|}\,\mathscr{S}\cup\{\rho\}\\ S\end{array}$ there is a proof $\begin{array}{c}\overline{\|}\,\mathscr{S}\\ S\end{array}$.

System SKSgq, shown in Fig. 2, has been introduced and shown to be sound and complete for classical predicate logic in [1]. The first S stands for "symmetric" or "self-dual", meaning that for each rule, its dual (or contrapositive) is also in the system. The K stands for "klassisch" as in Gentzen's LK and the second S says that it is a system in the calculus of structures. The g is for "general" (as opposed to atomic) contraction. The q denotes (first-order) quantifiers.

The first and last column show the rules that deal with quantifiers, in the middle there are the rules for the propositional fragment. The propositional rules $\mathsf{i}{\downarrow}$, s, $\mathsf{w}{\downarrow}$ and $\mathsf{c}{\downarrow}$ are called respectively *identity*, *switch*, *weakening* and *contraction*. The rule $\mathsf{u}{\downarrow}$ is called *universal*, because it roughly corresponds to the $\mathsf{R}\forall$ rule in sequent systems, while $\mathsf{n}{\downarrow}$ is called *instantiation*, because it corresponds to $\mathsf{R}\exists$.

In the sequent calculus, going up, the $\mathsf{R}\forall$ rule removes a universal quantifier from a formula to allow other rules to access this formula. In system SKSgq, inference rules apply deep inside formulae, so there is no need to remove the quantifier. Note that the premise of the $\mathsf{u}{\downarrow}$ rule implies its conclusion, which is not true for the $\mathsf{R}\forall$ rule of the sequent calculus. In all rules of SKSgq the premise implies the conclusion.

As usual, the substitution operation in the rules $\mathsf{n}{\downarrow}$ and $\mathsf{n}{\uparrow}$ requires τ to be free for x in R: quantifiers in R do not capture variables in τ. The term τ is not required to be free for x in $S\{R\}$: quantifiers in S may capture variables in τ.

The dual of rule carries the same name prefixed with a "co-", so e.g. $\mathsf{w}{\uparrow}$ is called *co-weakening*. The rule s is self-dual. The rule $\mathsf{i}{\uparrow}$ is special, it is called *cut*. Rules with a name that contains an arrow pointing downward are called *down-rules* and their duals are called *up-rules*. The system enjoys cut elimination: all up-rules are admissible, as has been shown in [1].

Sequent calculus derivations easily correspond to derivations in system SKSgq. For instance, the cut of sequent systems in Gentzen-Schütte form [14]:

$$\mathsf{Cut}\,\frac{\vdash \Phi, A \quad \vdash \Psi, \bar{A}}{\vdash \Phi, \Psi} \qquad \text{corresponds to} \qquad \mathsf{i}{\uparrow}\,\dfrac{\mathsf{s}\,\dfrac{\mathsf{s}\,\dfrac{([\Phi, A], [\Psi, \bar{A}])}{[\Phi, (A, [\Psi, \bar{A}])]}}{[\Phi, \Psi, (A, \bar{A})]}}{[\Phi, \Psi]} \quad .$$

Besides deep inference, the calculus of structures employs a notion of top-down symmetry for derivations. Symmetry makes possible to reduce the cut rule to its atomic form without performing cut elimination: this would be impossible by solely adopting deep inference. Here is an example that makes use of symmetry by flipping derivations: assuming that we can not prove f in the system, having a proof of R implies that there is no proof of $\bar{R}$. We assume that we have both proofs:

$$\begin{array}{c} \overline{\Vert} \\ R \end{array} \qquad \text{and} \qquad \begin{array}{c} \overline{\Vert} \\ \bar{R} \end{array} \quad ,$$

dualise the proof of R, to get

$$\begin{array}{c} \bar{R} \\ \Vert \\ \mathsf{f} \end{array} \quad ,$$

and compose this derivation with the proof of $\bar{R}$ to get a proof of f, which is a contradiction:

$$\begin{array}{c} \overline{\Vert} \\ \bar{R} \\ \Vert \\ \mathsf{f} \end{array} \quad .$$

3 Reducing and Eliminating Infinitary Rules

There are three infinitary rules in system SKSgq: the co-weakening, the cut, and the instantiation rule. In the following we will see for each of these rules how to to replace them by finitary ones without affecting provability.

3.1 The Co-weakening Rule

The rule $\mathsf{w}{\uparrow}$ is clearly infinitary, since there is an infinite choice of atoms, but it can immediately be eliminated by using a cut and an instance of $\mathsf{w}{\downarrow}$ as follows:

$$\mathsf{w}{\uparrow}\,\frac{S\{a\}}{S\{\mathsf{t}\}} \quad\rightsquigarrow\quad =\,\dfrac{\mathsf{i}{\uparrow}\,\dfrac{\mathsf{w}{\downarrow}\,\dfrac{\mathsf{s}\,\dfrac{\mathsf{s}\,\dfrac{S\{R\}}{S(R,[\mathsf{t},\mathsf{f}])}}{S[\mathsf{t},(R,\mathsf{f})]}}{S[\mathsf{t},(R,\bar{R})]}}{S[\mathsf{t},\mathsf{f}]}}{S\{\mathsf{t}\}} \quad .$$

3.2 The Cut Rule

The cut is the most prominent infinitary rule. The first source of infinite choice we will remove is the arbitrary size of the cut formula. To that end, consider the *atomic cut* rule:

$$\mathsf{ai}{\uparrow}\,\frac{S(a,\bar{a})}{S\{\mathsf{f}\}}$$

The following theorem, also proved in [1], allows us to restrict ourselves to atomic cuts.

Theorem 1. The rule $\mathsf{i}{\uparrow}$ is derivable for $\{\mathsf{ai}{\uparrow},\mathsf{s},\mathsf{u}{\uparrow}\}$.

Proof. By an easy structural induction on the structure that is cut. A cut introducing the structure (R,T) together with its dual structure $[\bar{R},\bar{T}]$ is replaced by two cuts on smaller structures:

$$\mathsf{i}{\uparrow}\,\frac{S(R,T,[\bar{R},\bar{T}])}{S\{\mathsf{f}\}} \quad\rightsquigarrow\quad \mathsf{i}{\uparrow}\,\dfrac{\mathsf{i}{\uparrow}\,\dfrac{\mathsf{s}\,\dfrac{\mathsf{s}\,\dfrac{S(R,T,[\bar{R},\bar{T}])}{S(R,[\bar{R},(T,\bar{T})])}}{S[(R,\bar{R}),(T,\bar{T})]}}{S(R,\bar{R})}}{S\{\mathsf{f}\}} \quad .$$

A cut introducing the structure $\forall xR$ together with its dual structure $\exists x\bar{R}$ is replaced by a cut inside an existential quantifier followed by an instance of $\mathsf{u}\!\uparrow$:

$$\mathsf{i}\!\uparrow\frac{S(\forall xR,\exists x\bar{R})}{S\{\mathsf{f}\}} \qquad\rightsquigarrow\qquad =\frac{\mathsf{i}\!\uparrow\dfrac{\mathsf{u}\!\uparrow\dfrac{S(\forall xR,\exists x\bar{R})}{S\{\exists x(R,\bar{R})\}}}{S\{\exists x\mathsf{f}\}}}{S\{\mathsf{f}\}} \quad .$$

These reductions can be repeated until all cuts are atomic. □

The rule $\mathsf{ai}\!\uparrow$ still is infinitary, since there is an infinite choice of atoms. Let us take a closer look at the atoms:

$$\mathsf{ai}\!\uparrow\frac{S(p(\tau_1,\dots,\tau_n),\overline{p(\tau_1,\dots,\tau_n)})}{S\{\mathsf{f}\}} \quad .$$

There are both an infinite choice of predicate symbols p and an infinite choice of terms for each argument of p. Let $\vec{\tau}$ denote $\tau_1,\dots,\tau_n$ and $\vec{x}$ denote $x_1,\dots,x_n$. Since cuts can be applied inside existential quantifiers, we can delegate the choice of terms to a sequence of $\mathsf{n}\!\downarrow$ instances:

$$\mathsf{ai}\!\uparrow\frac{S(p(\vec{\tau}),\overline{p(\vec{\tau})})}{S\{\mathsf{f}\}} \qquad\rightsquigarrow\qquad =\frac{\mathsf{ai}\!\uparrow\dfrac{\mathsf{n}\!\downarrow^{n}\dfrac{S(p(\vec{\tau}),\overline{p(\vec{\tau})})}{S\{\exists\vec{x}(p(\vec{x}),\overline{p(\vec{x})})\}}}{S\{\exists\vec{x}\mathsf{f}\}}}{S\{\mathsf{f}\}} \quad .$$

The remaining cuts are restricted in that they do not introduce arbitray terms but just existential variables. Let us call this restricted form $\mathsf{vai}\!\uparrow$:

$$\mathsf{vai}\!\uparrow\frac{S(p(\vec{x}),\overline{p(\vec{x})})}{S\{\mathsf{f}\}} \quad .$$

The only infinite choice that remains is the one of the predicate symbol p. To remove it, consider the rule *finitary atomic cut*

$$\mathsf{fai}\!\uparrow\frac{S(p(\vec{x}),\overline{p(\vec{x})})}{S\{\mathsf{f}\}} \qquad \text{where } p \text{ appears in the conclusion.}$$

This rule is finitary, and we will show that we can easily transform a proof into one where the only cuts that appear are $\mathsf{fai}\!\uparrow$ instances.

Take a proof in the system we obtained so far, that is SKSgq without $\mathsf{w}\!\uparrow$, and with $\mathsf{vai}\!\uparrow$ instead of $\mathsf{i}\!\uparrow$. Individuate the bottommost instance of $\mathsf{vai}\!\downarrow$ that violates the proviso of $\mathsf{fai}\!\uparrow$:

$$\begin{array}{c}\overline{\Vert}\\ \mathsf{vai}\!\uparrow\dfrac{S(p(\vec{x}),\overline{p(\vec{x})})}{S\{\mathsf{f}\}}\end{array}\quad,$$

where p does not appear in $S\{\mathsf{f}\}$. We can then replace all instances of $p(\vec{x})$ and $\overline{p(\vec{x})}$ in the proof above the cut with t and f, respectively, to obtain a proof of $S\{\mathsf{f}\}$. It is easy to check that all rule instances stay valid or become trivial; the cut

$$\mathsf{vai}\!\uparrow\frac{S(\mathsf{t},\mathsf{f})}{S\{\mathsf{f}\}}\quad,$$

can just be removed, since $(\mathsf{t},\mathsf{f})=\mathsf{f}$.

Please notice that if p appeared in $S\{\mathsf{f}\}$, then this would not work, because it could destroy the rule instance below $S\{\mathsf{f}\}$.

Proceeding inductively upwards, we remove all infinitary atomic cuts.

3.3 The Instantiation Rule

The same techniques also work for instantiation. Consider these two restricted versions of $\mathsf{n}\!\downarrow$:

$$\mathsf{n}\!\downarrow_1\frac{S\{R[x/f(\vec{x})]\}}{S\{\exists xR\}}\qquad\text{and}\qquad\mathsf{n}\!\downarrow_2\frac{S\{R[x/y]\}}{S\{\exists xR\}}\quad.$$

An instance of $\mathsf{n}\!\downarrow$ that is not an instance of $\mathsf{n}\!\downarrow_2$ can inductively be replaced by instances of $\mathsf{n}\!\downarrow_1$ (chose variables for $\vec{x}$ that are not free in R):

$$\mathsf{n}\!\downarrow\frac{S\{R[x/f(\vec{\tau})]\}}{S\{\exists xR\}}\qquad\rightsquigarrow\qquad =\dfrac{\mathsf{n}\!\downarrow_1\dfrac{\mathsf{n}\!\downarrow^n\dfrac{S\{R[x/f(\vec{\tau})]\}}{S\{\exists\vec{x}R[x/f(\vec{x})]\}}}{S\{\exists\vec{x}\exists xR\}}}{S\{\exists xR\}}\quad.$$

This process can be repeated until all instances of $\mathsf{n}\!\downarrow$ are either instances of $\mathsf{n}\!\downarrow_1$ or $\mathsf{n}\!\downarrow_2$.

Now consider the following finitary rules,

$$\mathsf{fn}\!\downarrow_1\frac{S\{R[x/f(\vec{x})]\}}{S\{\exists xR\}}\qquad\text{and}\qquad\mathsf{fn}\!\downarrow_2\frac{S\{R[x/y]\}}{S\{\exists xR\}}\quad.$$

where $\mathsf{fn}{\downarrow}_1$ carries the proviso that the function symbol f either occurs in the conclusion or is a fixed constant c, and $\mathsf{fn}{\downarrow}_2$ carries the proviso that the variable y appears in the conclusion (no matter whether free or bound or in a vacuous quantifier).

Infinitary instances of $\mathsf{n}{\downarrow}_1$ and $\mathsf{n}{\downarrow}_2$, i.e. those that are not instances of $\mathsf{fn}{\downarrow}_1$ and $\mathsf{fn}{\downarrow}_2$, respectively, are easily replaced by finitary rules similarly to how the infinitary cuts were eliminated. Take the constant symbol c that is fixed in the proviso of $\mathsf{fn}{\downarrow}_1$, and throughout the proof above an infinitary instance of $\mathsf{n}{\downarrow}_1$, replace all terms that are instances of $f(\vec{x})$ by c. For $\mathsf{n}{\downarrow}_2$ we do the same to all occurences of y, turning it into an instance of $\mathsf{fn}{\downarrow}_1$.

3.4 Consistency

We now define the finitary system FKSgq to be

$$(\mathsf{SKSgq} \setminus \{\mathsf{i}{\uparrow}, \mathsf{w}{\uparrow}, \mathsf{n}{\downarrow}\}) \cup \{\mathsf{fai}{\uparrow}, \mathsf{fn}{\downarrow}_1, \mathsf{fn}{\downarrow}_2\} \quad ,$$

and, for what we said above, state

Theorem 2. Each structure is provable in system SKSgq if and only it is provable in system FKSgq.

To put finitariness at work, we show consistency of system FKSgq. Of course, for this purpose it suffices to have finitary cut. Having infinite choice in instantiation would not affect the following argument.

Theorem 3. The unit f is not provable in system FKSgq.

Proof. No atoms, but only f, t and vacuous quantifiers can appear in such a proof. It is easy to show that f is not equivalent to t. Then we show that no rule can have a premise equivalent to t and a conclusion equivalent to f. This is simply done by inspection of all the rules in FKSgq.

From the two theorems above we immediately get

Corollary 4. The unit f is not provable in system SKSgq.

3.5 The Dust Under the Carpet

The seemingly innocuous equations for variable renaming and vacuous quantifier, technically speaking, are infinitary in the choice of variables. There is no reason to believe that vacuous quantifier and variable renaming can get in the way of finitariness; in fact, from automated deduction we know that this can be treated in a finitary way. At this time we could show a finitary system where inference rules keep track of the global structure of derivations in much the same way as Miller does in [10]. We are not entirely happy with this solution, because it loses some of the purity of rules. We are confident that a perfectly satisfying solution can be found, and we reserve it for the journal version of this paper.

4 Conclusion

In this paper we showed simple proof theoretical techniques for making a system of first order classical logic finitary. We believe that these considerations help make clear that finitariness and cut elimination, or other normalisation techniques, are conceptually independent.

Some of the techniques we used, for example the replacement of an atom and its dual by t and f, are folklore. However, in order to produce a finitary system they have to be combined with the reduction of the cut rule to its atomic form. This crucial ingredient is provided by deep inference and top-down symmetry, which are not available in the sequent calculus.

In the calculus of structures, there are presentations of various modal logics [11], linear logic [13,12] and various extensions of it [7,8,3] and noncommutative logics [4]. All these systems are similar to system SKSgq in that they include rules which follow a scheme [6], which ensures atomicity of cut and identity. So it is certainly possible to use these methods for these logics.

References

1. Kai Brünnler. Locality for classical logic. Technical Report WV-02-15, Dresden University of Technology, 2002. Available at http://www.wv.inf.tu-dresden.de/~kai/LocalityClassical.pdf.
2. Kai Brünnler and Alwen Fernanto 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.
3. Paola Bruscoli. A purely logical account of sequentiality in proof search. In Peter J. Stuckey, editor, *Logic Programming, 18th International Conference*, volume 2401 of *Lecture Notes in Artificial Intelligence*, pages 302–316. Springer-Verlag, 2002.
4. Pietro Di Gianantonio. Structures in cyclic linear logic. Technical report, Università di Udine, 2003.
5. Gerhard Gentzen. Investigations into logical deduction. In M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland, Amsterdam, 1969.
6. Alessio Guglielmi. Recipe. Manuscript. Available on the web at http://www.wv.inf.tu-dresden.de/~guglielm/Research/Notes/AG2.pdf, 2002.
7. Alessio Guglielmi. A system of interaction and structure. Technical Report WV-02-10, 2002. Available at http://www.wv.inf.tu-dresden.de/~guglielm/Research/Gug/Gug.pdf.
8. Alessio Guglielmi and Lutz Straßburger. A non-commutative extension of MELL. In Matthias Baaz and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning, LPAR 2002*, volume 2514 of *LNAI*, pages 231–246. Springer-Verlag, 2002.
9. J. Herbrand. *Recherches sur la théorie de la démonstration*. PhD thesis, Université de Paris, 1930.
10. Dale Miller. Abstract syntax and logic programming. In Andrei Voronkov, editor, *Proceedings of the Second Russian Conference on Logic Programming*, LNAI. Springer-Verlag, 1991.

11. Charles Stewart and Phiniki Stouppa. A systematic proof theory for several modal logics. Technical Report WV-03-08, Technische Universität Dresden, 2003. Submitted.
12. Lutz Straßburger. MELL in the calculus of structures. Technical Report WV-2001-03, Dresden University of Technology, 2001. Accepted by TCS. Available at http://www.ki.inf.tu-dresden.de/~lutz/els.pdf.
13. Lutz Straßburger. A local system for linear logic. In Matthias Baaz and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning, LPAR 2002*, volume 2514 of *LNAI*, pages 388–402. Springer-Verlag, 2002.
14. Anne Sjerp Troelstra and Helmut Schwichtenberg. *Basic Proof Theory.* Cambridge University Press, 1996.