Atomic Cut Elimination for Classical Logic

Kai Brünnler
kai.bruennler@inf.tu-dresden.de

Technische Universität Dresden, Fakultät Informatik, D - 01062 Dresden, Germany

Abstract. System SKS is a set of rules for classical propositional logic presented in the calculus of structures. Like sequent systems and unlike natural deduction systems, it has an explicit cut rule, which is admissible. In contrast to sequent systems, the cut rule can easily be reduced to atomic form. This allows for a very simple cut elimination procedure based on plugging in parts of a proof, like normalisation in natural deduction and unlike cut elimination in the sequent calculus. It should thus be a good common starting point for investigations into both proof search as computation and proof normalisation as computation.

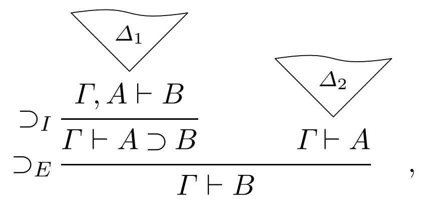
Keywords. sequent calculus, natural deduction, cut elimination, classical logic, atomic cut.

1 Introduction

The two well-known connections between proof theory and language design, *proof search as computation* and *proof normalisation as computation*, have mainly used different proof-theoretic formalisms. While designers of functional programming languages prefer natural deduction, because of the close correspondence between proof normalisation and reduction in related term calculi [4,8], designers of logic programming languages prefer the sequent calculus [7], because infinite choice and much of the unwanted non-determinism is limited to the cut rule, which can be eliminated.

System SKS [2] is a set of inference rules for classical propositional logic presented in a new formalism, the *calculus of structures* [5]. This system admits the good properties usually found in sequent systems: in particular, all rules that induce infinite choice in proof search are admissible. Thus, in principle, it is as suitable for proof search as systems in the sequent calculus. In this paper I will present a cut elimination procedure for SKS that is very similar to normalisation in natural deduction. It thus allows us to develop, at least for the case of classical logic, both the proof search and the proof normalisation paradigm of computation in the same formalism and starting from the same system of rules.

Cut elimination in the sequent calculus and normalisation in natural deduction, widely perceived as 'morally the same', differ quite a bit, technically. Compared to cut elimination, (weak) normalisation is simpler, involving neither permutation of a multicut rule, nor induction on the cut-rank. The equivalent of a cut in natural deduction, for example,



is eliminated as follows: first, assumption A and all its copies are removed from Δ_1. Second, the derivation Δ_2, with the context strengthened accordingly, is plugged into all the leaves of Δ_1 where assumption A was used.

This method relies on the fact that no rule inside Δ_1 can change the premise A, which is why it does not work for the sequent calculus. To eliminate a cut in the sequent calculus, one has to cope with the fact that logical rules may be applied to both eigenformulas of the cut. This is usually done by permuting up the cut rule step-by-step. However, given a cut with an *atomic* cut formula a inside a sequent calculus proof, we can trace the occurrence of a and its copies produced by contraction, identify all the leaves where they are used in identity axioms, and plug in subproofs in very much the same way as in natural deduction. The problem for the sequent calculus is that cuts are not atomic, in general.

The calculus of structures generalises the one-sided sequent calculus. It has led not only to inference systems with interesting new properties for classical and linear logic [2,9,10], but also to inference systems for new logics that are problematic for the sequent calculus [5,6,3].

Derivations in the calculus of structures enjoy a top-down symmetry that is not available in the sequent calculus: they are chains of one-premise inference rules. 'Meta-level conjunction' (the branching of the proof tree) and 'object-level conjunction' (the connective in a formula) are identified. The two notions of formula and sequent are also identified, they merge into the notion of *structure*, which is a formula subject to equivalences that are usually imposed on sequents. This simplification makes explicit the duality between the identity axiom and the cut rule [5]:

$$\textit{identity}\,\frac{S\{\textit{true}\}}{S\{R \vee \bar{R}\}} \qquad\qquad \textit{cut}\,\frac{S\{R \wedge \bar{R}\}}{S\{\textit{false}\}}$$

The identity rule is read bottom-up as: if inside a structure there occurs a disjunction of a structure R and its negation, then it can be replaced by the constant *true*. The notion of duality between cut and identity is precisely the one that is known as *contrapositive*.

Just like in the sequent calculus, the identity axiom can easily be reduced to atomic form. The symmetry of the calculus of structures allows to reduce the cut to atomic form in the same way as the identity axiom, i.e. without having to go through cut elimination. Atomicity of the cut then admits a very simple cut elimination procedure that is similar to normalisation in natural deduction.

After introducing basic notions of the calculus of structures, I show system SKS with atomic contraction, weakening, identity and, most significantly, atomic cut. Then, after establishing some lemmas, I present the cut elimination procedure.

2 The Calculus of Structures

Definition 1. Propositional variables p and their negations $\bar{p}$ are *atoms*, with the negation of $\bar{p}$ defined to be p. Atoms are denoted by a, b, ... The *structures* of the language KS are generated by

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

where t and f are the units *true* and *false*, $[S_1,\dots,S_h]$ is a *disjunction* and $(S_1,\dots,S_h)$ is a *conjunction.* $\bar{S}$ 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 congruence relation induced by the equations 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

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

Commutativity

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

Singleton

$$[R] = R = (R)$$

Units

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

Negation

$$\bar{\mathsf{t}} = \mathsf{f}$$
$$\bar{\mathsf{f}} = \mathsf{t}$$
$$\overline{[R_1, \dots, R_h]} = (\bar{R}_1, \dots, \bar{R}_h)$$
$$\overline{(R_1, \dots, R_h)} = [\bar{R}_1, \dots, \bar{R}_h]$$
$$\bar{\bar{R}} = R$$

Fig. 1. Equations on structures

Definition 2. 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.* In an instance of ρ, the structure taking the place of R is called *redex* and the structure taking the place of T is called *contractum.* A (*formal*) *system* $\mathscr{S}$ is a set of inference rules. To clarify the use of the equational theory where it is not obvious, I will use the rule $= \dfrac{T}{R}$ where R and T are equivalent structures.

Definition 3. The *dual* of a rule is obtained by exchanging premise and conclusion and replacing each connective by its De Morgan dual.

Definition 4. A *derivation* Δ in a certain formal system is a finite chain of instances of inference rules in the system:

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

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

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

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

$$\mathsf{aw}{\downarrow}\,\frac{S\{\mathsf{f}\}}{S\{a\}} \qquad \mathsf{aw}{\uparrow}\,\frac{S\{a\}}{S\{\mathsf{t}\}}$$

$$\mathsf{ac}{\downarrow}\,\frac{S[a,a]}{S\{a\}} \qquad \mathsf{ac}{\uparrow}\,\frac{S\{a\}}{S(a,a)}$$

Fig. 2. System SKS

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

3 System SKS

System SKS, shown in Fig. 2, has been introduced and shown to be sound and complete for classical propositional logic in [2]. 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 last S says that it is a system on structures.

The rules $\mathsf{ai}\!\downarrow, \mathsf{s}, \mathsf{m}, \mathsf{aw}\!\downarrow, \mathsf{ac}\!\downarrow$ are called respectively *atomic identity*, *switch*, *medial*, *atomic weakening* and *atomic contraction*. Their dual rules carry the same name prefixed with a "co-", so e.g. $\mathsf{aw}\!\uparrow$ is called *atomic co-weakening*. The rules s and m are their own duals. The rule $\mathsf{ai}\!\uparrow$ is special, it is called *atomic cut*. Rules $\mathsf{ai}\!\downarrow, \mathsf{aw}\!\downarrow, \mathsf{ac}\!\downarrow$ are called *down-rules* and their duals are called *up-rules*. In [2], by a semantic argument, all up-rules were shown to be admissible. By removing them we obtain system KS, shown in Fig. 3, which is complete.

Cut-free sequent systems fulfill the subformula property. Our case is different, because the notions of formula and sequent are merged. System KS does not fulfill a "substructure property" just as sequent systems do not fulfill a "subsequent property". However, when seen bottom-up, no rule in system KS introduces new atoms. It thus satisfies the main aspect of the subformula property: when given a conclusion of a rule there is only a finite number of premises to choose from. In proof search, for example, the branching of the search tree is finite.

$$\mathsf{ai}\!\downarrow \frac{S\{\mathsf{t}\}}{S[a,\bar{a}]} \qquad \mathsf{aw}\!\downarrow \frac{S\{\mathsf{f}\}}{S\{a\}} \qquad \mathsf{ac}\!\downarrow \frac{S[a,a]}{S\{a\}}$$

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

Fig. 3. System KS

Identity, cut, weakening and contraction are restricted to atoms in system SKS. The general versions of those rules are shown in Fig. 4.

Theorem 5. *The rules $\mathsf{i}\!\downarrow$, $\mathsf{w}\!\downarrow$ and $\mathsf{c}\!\downarrow$ are derivable in $\{\mathsf{ai}\!\downarrow, \mathsf{s}\}$, $\{\mathsf{aw}\!\downarrow\}$ and $\{\mathsf{ac}\!\downarrow, \mathsf{m}\}$, respectively. Dually, the rules $\mathsf{i}\!\uparrow$, $\mathsf{w}\!\uparrow$ and $\mathsf{c}\!\uparrow$ are derivable in $\{\mathsf{ai}\!\uparrow, \mathsf{s}\}$, $\{\mathsf{aw}\!\uparrow\}$ and $\{\mathsf{ac}\!\uparrow, \mathsf{m}\}$, respectively.*

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

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

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

Fig. 4. General identity, weakening, contraction and their duals

Proof. By an easy structural induction on the structure that is cut, weakened or contracted. Details are in [2]. The case for the cut is shown here. A cut introducing the structure (R,T) together with its dual structure $[\bar{R},\bar{T}]$ is replaced by two cuts on smaller structures:

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

□

So, while general identity, weakening, contraction and their duals do not belong to SKS, they will be freely used in derivations in SKS to denote multiple instances of the corresponding rules in SKS according to Theorem 5.

Remark 6. Sequent calculus derivations easily correspond to derivations in system SKS. For instance, the cut of sequent systems in Gentzen-Schütte form [11]:

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

4 Cut Elimination

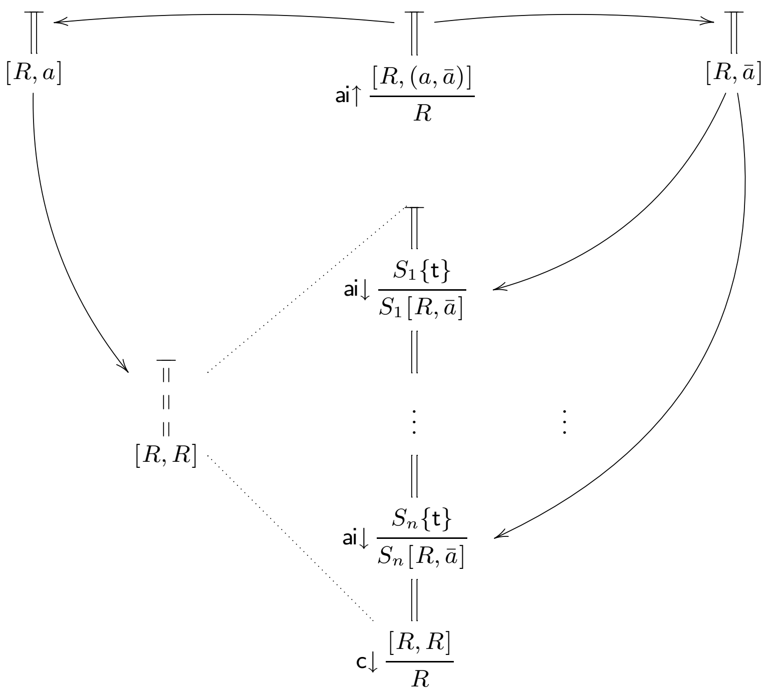
In the calculus of structures, there is more freedom in applying inference rules than in the sequent calculus. While this allows for a richer combinatorial analysis of proofs, it is a significant challenge for cut elimination. During cut elimination, the sequent calculus allows to get into the crucial situation where on one branch a logical rule applies to the main connective of the eigenformula and on the other branch the corresponding rule applies to the dual connective of the dual eigenformula. In the calculus of structures, rules apply deep inside a context, they are not restricted to main connectives. The methodology of the sequent calculus thus does not apply. For example, one cannot permute the cut over the switch rule. One can generalise the cut in order to permute it over switch, but this requires a case analysis that is far more complicated than in the sequent calculus. Contraction is an even bigger problem. Despite many efforts, no cut elimination procedure along these lines has been found for system SKS.

Two new techniques were developed to eliminate cuts in the calculus of structures. The first is called *decomposition*, and has been used in [6,9] for some systems related to linear logic. Proving termination of decomposition is rather involved [9]. It makes essential use of the exponentials of linear logic which restrict the use of contraction. So far, this technique could not be used for classical logic with its unrestricted contraction. The second technique is called *splitting* [5], and essentially makes available a situation corresponding to the one described above for the sequent calculus. Splitting covers the broadest range of systems in the calculus of structures, it not only applies to the systems mentioned above, but has recently also been applied to system SKS (but the proof is not published yet). Compared to splitting, the procedure given here is much simpler. In fact, I do not know of any other system with such a simple cut elimination procedure.

In the sequent calculus as well as in sequent-style natural deduction, a derivation is a tree. Seen bottom-up, a cut splits the tree into two branches. To apply a cut, one is forced to split the context among the two branches (in the case of multiplicative context treatment) or to duplicate the context (in the case of additive context treatment). In the calculus of structures, the cut rule does not split the proof.

The crucial idea, illustrated in Fig. 5, is that we can do that during cut elimination. This allows us to plug-in proofs just like in natural deduction:

we duplicate the proof above a cut and remove atom a from the copy shown on the left and the atom $\bar{a}$ from the copy shown on the right. We choose one copy, the one on the left in this case, and replace a by R throughout the proof, breaking some instances of identity. They are fixed by substituting the proof on the right. A contraction is applied to obtain a cut-free proof of R.



$[R, a]$
ai↑ $\frac{[R,(a,\bar{a})]}{R}$
$[R, \bar{a}]$
ai↓ $\frac{S_1\{\mathsf{t}\}}{S_1[R,\bar{a}]}$
$[R, R]$
ai↓ $\frac{S_n\{\mathsf{t}\}}{S_n[R,\bar{a}]}$
c↓ $\frac{[R,R]}{R}$

Fig. 5. Elimination of one atomic cut

In contrast to the sequent calculus, the cut is not the only problematic rule in system SKS. The rule aw↑ also induces infinite choice in proof-search. Fortunately, we can not only eliminate the cut rule, but also the other up-rules. Each up-rule individually can be shown to be admissible for system KS. However, since we are going to eliminate the cut anyway, to eliminate rules aw↑ and ac↑ the following lemma is sufficient.

Lemma 7. Each rule in SKS is derivable for identity, cut, switch and its dual rule.

Proof. An instance of $\rho\!\uparrow \dfrac{S\{T\}}{S\{R\}}$ can be replaced by $\begin{array}{l} \mathsf{i}\!\downarrow \dfrac{S\{T\}}{S(T,[R,\bar{R}])} \\ \mathsf{s} \dfrac{\phantom{S(T,[R,\bar{R}])}}{S[R,(T,\bar{R})]} \\ \rho\!\downarrow \dfrac{\phantom{S[R,(T,\bar{R})]}}{S[R,(T,\bar{T})]} \\ \mathsf{i}\!\uparrow \dfrac{\phantom{S[R,(T,\bar{T})]}}{S\{R\}} \end{array}$.

The same holds for down-rules. □

When plugging in a derivation in natural deduction, its context has to be strengthened, to fit into the leaf into which it is plugged. Adding to a context in natural deduction is easy, since it is a flat object, a set or a multiset. In the calculus of structures, contexts are more general, nested objects. The following definition is used to strengthen contexts.

Definition 8. Given a derivation Δ, the derivation $S\{\Delta\}$ is obtained as follows:

$$\Delta = \begin{array}{l} \pi' \dfrac{T}{V} \\ \pi \dfrac{}{\vdots} \\ \rho' \dfrac{}{U} \\ \rho \dfrac{}{R} \end{array} \qquad S\{\Delta\} = \begin{array}{l} \pi' \dfrac{S\{T\}}{S\{V\}} \\ \pi \dfrac{\phantom{S\{V\}}}{\vdots} \\ \rho' \dfrac{\phantom{S\{U\}}}{S\{U\}} \\ \rho \dfrac{\phantom{S\{U\}}}{S\{R\}} \end{array} \quad .$$

Definition 9. An instance of atomic cut is called *shallow* if it is of the following form:

$$\mathsf{ai}\!\uparrow \frac{[S,(a,\bar{a})]}{S} \quad .$$

Lemma 10. The atomic cut is derivable for shallow atomic cut and switch.

Proof. An easy induction locally replaces an instance of atomic cut by a shallow atomic cut followed by instances of switch. Details are in [1]. □

Lemma 11. Each proof $\begin{array}{c} \Vert \mathsf{KS} \\ T\{a\} \end{array}$ can be transformed into a proof $\begin{array}{c} \Vert \mathsf{KS} \\ T\{\mathsf{t}\} \end{array}$.

Proof. Starting with the conclusion, going up in the proof, in each structure we replace the occurrence of a and its copies, that are produced by contractions, by the unit t. Replacements inside the context of any rule instance do not affect the validity of this rule instance. Instances of the rules m and s remain valid, also in the case that atom occurrences are replaced inside redex and contractum. Instances of the other rules are replaced by the following derivations:

$$\mathsf{ac}{\downarrow}\,\frac{S[a,a]}{S\{a\}} \quad \leadsto \quad =\,\frac{S[\mathsf{t},\mathsf{t}]}{S\{\mathsf{t}\}}$$

$$\mathsf{aw}{\downarrow}\,\frac{S\{\mathsf{f}\}}{S\{a\}} \quad \leadsto \quad \begin{array}{r}
= \dfrac{S\{\mathsf{f}\}}{S([\mathsf{t},\mathsf{t}],\mathsf{f})} \\
\mathsf{s}\,\dfrac{}{S[\mathsf{t},(\mathsf{t},\mathsf{f})]} \\
= \dfrac{}{S\{\mathsf{t}\}}
\end{array}$$

$$\mathsf{ai}{\downarrow}\,\frac{S\{\mathsf{t}\}}{S[a,\bar{a}]} \quad \leadsto \quad \begin{array}{r}
= \dfrac{S\{\mathsf{t}\}}{S[\mathsf{t},\mathsf{f}]} \\
\mathsf{aw}{\downarrow}\,\dfrac{}{S[\mathsf{t},\bar{a}]}
\end{array} \quad .$$

□

Properly equipped, we now turn to cut elimination.

Theorem 12. Each proof $\begin{array}{c}\big\| \,\mathsf{SKS} \\ T\end{array}$ can be transformed into a proof $\begin{array}{c}\big\| \,\mathsf{KS} \\ T\end{array}$.

Proof. By Lemma 7, the only rule left to eliminate is the cut. By Lemma 10, we replace all cuts by shallow cuts. The topmost instance of cut, together with the proof above it, is singled out:

$$\begin{array}{c}\Big\| \,\mathsf{KS}\cup\{\mathsf{ai}{\uparrow}\} \\ T\end{array} \quad = \quad \begin{array}{c}
\Pi \big\| \,\mathsf{KS} \\
\mathsf{ai}{\uparrow}\,\dfrac{[R,(a,\bar{a})]}{R} \\
\Delta \big\| \,\mathsf{KS}\cup\{\mathsf{ai}{\uparrow}\} \\
T
\end{array} \quad .$$

Lemma 11 is applied twice on Π to obtain

$$\begin{array}{c}\Pi_1 \big\| \,\mathsf{KS} \\ [R,a]\end{array} \quad \text{and} \quad \begin{array}{c}\Pi_2 \big\| \,\mathsf{KS} \\ [R,\bar{a}]\end{array} \quad .$$

Starting with the conclusion, going up in proof Π_1, in each structure we replace the occurrence of a and its copies, that are produced by contractions, by the structure R.

Replacements inside the context of any rule instance do not affect the validity of this rule instance. Instances of the rules m and s remain valid, also in the case that atom occurrences are replaced inside redex and contractum. Instances of $\mathsf{ac}{\downarrow}$ and $\mathsf{aw}{\downarrow}$ are replaced by their general versions:

$$\mathsf{ac}{\downarrow}\,\frac{S[a,a]}{S\{a\}} \quad \leadsto \quad \mathsf{c}{\downarrow}\,\frac{S[R,R]}{S\{R\}}$$

$$\mathsf{aw}{\downarrow}\,\frac{S\{\mathsf{f}\}}{S\{a\}} \quad \leadsto \quad \mathsf{w}{\downarrow}\,\frac{S\{\mathsf{f}\}}{S\{R\}} \quad .$$

Instances of $\mathsf{ai}{\downarrow}$ are replaced by $S\{\Pi_2\}$:

$$\mathsf{ai}{\downarrow}\,\frac{S\{\mathsf{t}\}}{S[a,\bar{a}]} \quad \leadsto \quad \begin{array}{c} S\{\mathsf{t}\} \\ S\{\Pi_2\}\Big\Vert\,\mathsf{KS} \\ S[R,\bar{a}] \end{array} \quad .$$

The result of this process of substituting Π_2 into Π_1 is a proof Π_3, from which we build

$$\begin{array}{c} \Pi_3\Big\Vert\,\mathsf{KS} \\ \mathsf{c}{\downarrow}\,\dfrac{[R,R]}{R} \\ \Delta\Big\Vert\,\mathsf{KS}\cup\{\mathsf{ai}{\downarrow}\} \\ T \end{array}$$

Proceed inductively downward with the remaining instances of cut. □

5 Conclusion

System SKS seems a good starting point for developing both the proof search as well as the proof normalisation paradigm in one system. Since all up-rules are admissible, it is suitable for proof search as computation.

The cut elimination procedure given is simpler than those for sequent calculi. The way in which proofs are substituted resembles normalisation in natural deduction. This hopefully allows for a computational interpretation in the proof normalisation as computation paradigm.

Of course, a lot of work remains to done. In the proof search as computation realm, given the admissibility of cut, a suitable notion of *uniform proof* as in [7] should be obtainable. For proof normalisation as computation, natural questions to be considered are strong normalisation and confluence of the cut elimination procedure when imposing as little strategy as possible. Similarly to [8], a term calculus should be developed and its computational meaning be made precise. Intuitionistic logic is a more familiar setting for this, so the possibility of treating intuitionistic logic should be explored.

A natural question is whether this procedure scales to more expressive cases, for example to predicate logic. System SKSq extends system SKS by first-order quantifiers [1]. There, cut elimination is proved via a translation to the sequent calculus. The procedure presented here does not appear to easily scale to system SKSq. The problem, which does not occur in shallow inference systems like sequent calculus or natural deduction, are existential quantifiers in the context of a cut which bind variables both in a and $\bar{a}$. The procedure easily extends to *closed* atomic cuts, that is, cuts where the eigenformula is an atom prefixed by quantifiers that bind all its variables. The question then is how to reduce general cuts to closed atomic cuts. If this problem were solved, then the procedure would scale to predicate logic. Hopefully this will lead to a cut elimination procedure for predicate logic, which is simpler than other cut elimination procedures, as happened for propositional logic.

Acknowledgements
This work has been supported by the DFG Graduiertenkolleg 334. I would like to thank Alessio Guglielmi, for inspiration, and Alwen Fernanto Tiu and Michel Parigot, for comments which helped simplifying an earlier version of the cut elimination procedure. Alessio Guglielmi, Charles Stewart and Lutz Straßburger carefully read previous drafts of this work and prompted improvements.

Web Site
Information about the calculus of structures is available from:

http://www.wv.inf.tu-dresden.de/˜guglielm/Research/ .

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. Jean Gallier. Constructive logics. Part I: A tutorial on proof systems and typed λ-calculi. *Theoretical Computer Science*, 110:249–339, 1993.
5. 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.
6. 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.
7. Dale Miller, Gopalan Nadathur, Frank Pfenning, and Andre Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, 51:125–157, 1991.
8. M. Parigot. $\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction. In *LPAR 1992*, volume 624 of *Lecture Notes in Computer Science*, pages 190–201. Springer-Verlag, 1992.
9. 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.
10. 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.
11. Anne Sjerp Troelstra and Helmut Schwichtenberg. *Basic Proof Theory.* Cambridge University Press, 1996.