Ingredients of a Deep Inference Theorem Prover

Ozan Kahramanoğulları
ozank@doc.ic.ac.uk

Department of Computing, Imperial College

Abstract. Deep inference deductive systems for classical logic provide exponentially shorter proofs than the sequent calculus systems, however with the cost of higher nondeterminism and larger search space in proof search. We report on our ongoing work on proof search with deep inference deductive systems. We present systems for classical logic where nondeterminism in proof search is reduced by constraining the context management rule of these systems. We argue that a deep inference system for classical logic can outperform sequent calculus deductive systems in proof search when nondeterminism and the application of the contraction rule are controlled by means of invertible rules.

1 Introduction

Automated theorem proving is finding broader applications with the emergence of increasingly powerful SAT-solvers, which provide yes-no answers to unsatisfiability queries that are then often verified on an interactive theorem prover (see, e.g., [23, 1]). Along these lines, there is an increasing interest in analytical proofs such as those delivered by a sequent calculus system.

In the sequent calculus, proofs are constructed by applying the inference rules bottom-up at a top-level connective at each proof step. Allowing the application of the inference rules only at a top-level connective provides a rigid procedure for constructing proofs with little nondeterminism. This procedure also simplifies the standard techniques for studying the proof theoretical properties of deductive systems, e.g., cut-elimination. This is because the notion of main connective drives the proof construction as well as the case analysis in proof theoretical investigations (see, e.g., [22]).

For instance, when classical logic proofs are considered, the construction of a proof in the sequent calculus boils down to transforming the formula in the conclusion to a formula in conjunctive normal form. In the one-sided sequent calculus, this procedure is driven by the context management rule $\mathsf{R}\wedge$. [1]

$$\mathsf{R}\wedge \frac{\vdash R, \Phi \quad \vdash T, \Psi}{\vdash R \wedge T, \Phi, \Psi}$$

[1] This rule is often equipped with implicit contraction, which copies the context of the conjunction to both branches at every instance. This results in less nondeterminism in proof construction, however also in a greater size of the proof.

In an instance of this rule, the space between the two branches at the premise is a meta-level conjunction and the commas in the sequents are meta-level disjunctions. Thus, this rule implements the distributivity law that performs the transformation into conjunctive normal form. In a proof construction, once this rule is applied exhaustively, the axiom can be applied at the leaf of each branch to obtain a proof. However, this results in copying of the contexts at each proof construction step and, for some formulae, in an exponential growth of the size of the proof.

Example 1. Consider the formula

$$[(a \wedge b) \vee ([\bar{a} \vee \bar{b}] \wedge c \wedge [\bar{a} \vee \bar{b}] \wedge d) \vee \bar{c} \vee \bar{d}] \ .$$

This formula belongs to a class of tautologies, called *Statman* tautologies [15], proofs of which grow exponentially in the size of the formulae in the cut-free sequent calculus. This is because the sequent calculus can access the subformulae only by opening up the formula at the main connective and proofs are then constructed by spreading the context of the top-level conjunction to the branches of the proof tree.

It is possible to construct polynomial size proofs of Statman tautologies by applying the inference rules from inside to out, which is achieved in the sequent calculus by using the cut-rule. However, it is also possible to achieve this by allowing the application of the inference rules at arbitrary depths inside logical expressions: *deep inference* [8] is a recent proof theoretical methodology that generalises the notion of inference in the sequent calculus in such a way. Thus, in deep inference deductive systems, inference rules can be applied at arbitrary depths inside the logical expressions, similar to term rewriting rules [12, 14]. Deep inference provides a greater freedom in construction of the proofs.

Because previously available techniques on the sequent calculus do not generalise to the deep inference setting, proof theory with deep inference required the development of new tools and techniques (see, e.g., [8, 3, 18, 12]). The rich combinatoric analysis of proofs, provided by deep inference, reveals proof theoretical properties of different logics that are otherwise not observable by traditional proof theoretical means, e.g., permutability of inference rules in deductive systems for different logics (see, e.g., [18, 3, 4]). The duality between the cut rule and the axiom which remains hidden in the sequent calculus also becomes explicit in deep inference deductive systems (see, e.g., [8, 2, 19, 9]). Moreover, deep inference makes it possible to design deductive systems tailored for computer science applications [8, 10], and not designable with any bounded depth deductive system [21]. The computer science notion of locality, i.e., an operation having a bounded computational cost, also finds a meaningful proof theoretical interpretation in deep inference deductive systems [5, 17, 20].

When deep inference deductive systems are considered from the point of view of computation as proof search, we observe that the applicability of the inference rules at arbitrary depths inside logical expressions makes it possible to start the construction of the proofs from subformulae. This capability provides many more different proofs some of which are are exponentially shorter proofs than those provided by the sequent calculus, while some others correspond to the sequent calculus proofs; as Bruscoli and Guglielmi showed in [6], deep inference polynomially simulates the sequent calculus. However, in proof search with deep inference deductive systems the breadth of the search space increases rather quickly, because there are more number of applicable instances of the inference rules at each proof search step. This results in a trade-off between shorter proofs and larger breadth of the search space.

In deep inference deductive systems, context management is performed by the switch rule. In [13, 11], we have introduced a rule called *deep lazy interaction switch* that reduces nondeterminism in proof search. We have shown that multiplicative linear logic remains complete when this rule is replaced with the switch rule in these systems.

The rule deep lazy interaction switch exploits an interaction scheme, which is determined by the instances of the axiom rule, between the atoms of the formula being proved. Exchanging this rule with the switch rule in deductive systems results in a reduction in nondeterminism without loosing the shorter proofs available due to deep inference. However, non-deterministic proof search still remains computationally expensive, in general, because the search space expands rather quickly. In contrast, in the sequent calculus, it is possible to overcome nondeterminism in classical logic by resorting to invertible rules, which essentially transform the formula being proved into a formula in conjunctive normal form. However, this transformation results in an exponential growth of the size of the proof in the size of the formula being proved due to the implicit applications of the contraction rule that copy contexts.

In this paper, we argue that our approach for reducing nondeterminism in deep inference proof search can be exploited together with the idea of using invertible rules to design a competitive deep inference theorem prover that delivers shorter proofs than those obtained by the sequent calculus. We first show that the techniques presented in [13, 11] can be carried to classical logic to replace the switch rule in this system with the deep lazy interaction switch rule. With the aim of moving from non-deterministic proof search to deterministic proof construction, we then propose a set of rules that includes rules obtained from deep lazy interaction switch rule. We then introduce a strategy that works on these rules. This strategy aims at reducing the size of the formula being proved by exploiting deep inference and applying invertible rules while going up in the proof. For this purpose, we delay the application of the rules for context management and contraction, thus minimise the growth of the proofs in contrast to the proofs in the sequent calculus systems. We conjecture that the strategy preserves the short proofs that are available due to deep inference.

2 Classical Logic with Deep Inference

In deep inference, the laws such as associativity and commutativity, which are usually implicitly imposed on formulae, become explicit by means of an underlying equational system: we work with congruence classes of formulae that we call structures (see, e.g., [8, 2]). Let us see classical logic structures.

Definition 1. *There are countably many* atoms*, denoted by $a, b, c, \ldots$. The* structures *P, Q, R, $S \ldots$ of classical logic are generated by*

$$R ::= a \mid \bar{a} \mid \mathsf{tt} \mid \mathsf{ff} \mid [R \vee R] \mid (R \wedge R) \quad ,$$

where a stands for any atom; negation is defined on the atoms as a (non-identical) involution $\bar{\cdot}$, thus dual atom occurences, as a and $\bar{a}$, can appear in the structures. tt and ff are the units true *and* false*. A structure $[R \vee R]$ is a* disjunction, *$(R \wedge R)$ is a* conjunction*. Structures are considered to be equivalent modulo relation $\approx$, which is the smallest congruence relation induced by the equational system consisting of the equations for* associativity *and* commutativity *for disjunction and conjunction, and the* equations *$[\mathsf{ff} \vee R] \approx R$, $(\mathsf{tt} \wedge R) \approx R$, $[\mathsf{tt} \vee \mathsf{tt}] \approx \mathsf{tt}$ and $(\mathsf{ff} \wedge \mathsf{ff}) \approx \mathsf{ff}$* for unit*. We denote the structures in the same equivalence class by picking a structure from the equivalence class. If there is no ambiguity, when writing the structures, we drop the superfluous brackets by resorting to the equations for associativity.*

Remark 1. Negation is defined only on atoms. This is not a limitation because of De Morgan laws.

Example 2. With respect to the congruence relation $\approx$, for the following structures we have $[(\bar{b} \wedge (\bar{a} \wedge \bar{c})) \vee [b \vee [a \vee c]]] \approx [[b \vee (((\bar{a} \wedge \bar{c}) \wedge \mathsf{tt}) \wedge \bar{b})] \vee [c \vee a]]$ and we can denote both structures with $[(\bar{a} \wedge \bar{b} \wedge \bar{c}) \vee a \vee b \vee c]$.

Definition 2. *A* structure context*, denoted as in $S\{\ \}$, is a structure with a hole. The structure R is a* substructure *of $S\{R\}$ and $S\{\ \}$ is its* context*.*

Example 3. Let $S\{\ \} = [\{\ \} \vee b \vee c]$, $R = \bar{a}$, $T = (\bar{b} \wedge \bar{c})$ and $U = a$. Then $S[(R \wedge T) \vee U] = [(\bar{a} \wedge \bar{b} \wedge \bar{c}) \vee a \vee b \vee c]$.

Definition 3. *An* inference rule *is a scheme of the kind $\rho \dfrac{T}{R}$, where ρ is the* name *of the rule, T is its* premise *and R is its* conclusion*. A typical deep inference rule has the shape $\rho \dfrac{S\{T\}}{S\{R\}}$ and specifies a step of rewriting* [2] *determined by the implication $T \Rightarrow R$ inside a generic context $S\{\ \}$. In an instance of ρ, we say that R is the* redex *and T is the* contractum*. A system $\mathscr{S}$ is a set of inference rules.*

Definition 4. *[19] The following are the rules of the system KSg, which are called* atomic interaction *($\mathsf{ai}{\downarrow}$), switch (s),* contraction *($\mathsf{c}{\downarrow}$) and* weakening *($\mathsf{w}{\downarrow}$), respectively.*

$$\mathsf{ai}{\downarrow}\,\frac{S\{\mathsf{tt}\}}{S[a \vee \bar{a}]} \qquad \mathsf{s}\,\frac{S([R \vee U] \wedge T)}{S[(R \wedge T) \vee U]} \qquad \mathsf{c}{\downarrow}\,\frac{S[R \vee R]}{S\{R\}} \qquad \mathsf{w}{\downarrow}\,\frac{S\{\mathsf{ff}\}}{S\{R\}}$$

[2] Because we consider the inference rules for proof-search, we consider their bottom-up applications which result in proofs that grow bottom-up from the conclusion. Thus, in this paper, these rewritings are those that rewrite the conclusion to the premise.

Definition 5. *A* derivation Δ *is a finite chain of instances of inference rules. A derivation can consist of just one structure. The top-most structure in a derivation is called the* premise*, and the bottom-most structure is called the* conclusion. *A derivation Δ whose premise is T, conclusion is R, and inference rules are in $\mathscr{S}$ will be written as* $\begin{array}{c} T \\ \Delta \Vert \mathscr{S} \\ R \end{array}$ *. A* proof Π *is a finite derivation whose premise is the unit* tt*. A rule ρ is* derivable *for a system $\mathscr{S}$ if for every instance of the* $\rho \dfrac{T}{R}$ *there is a derivation in system $\mathscr{S}$ with the premise T and the conclusion R. Two systems $\mathscr{S}$ and $\mathscr{S}'$ are* equivalent *if they prove the same structures.*

3 Proof Search

Applicability of the inference rules to substructures provides shorter proofs that are not available in the sequent calculus. However, this also results in a greater non-determinism in proof search: with deep inference, the inference rules become applicable at many more positions than in the sequent calculus. Because of this, the breadth of the search space grows rather quickly. In [13, 11], we have introduced a technique for reducing this non-determinism in proof search without loosing the shorter proofs that are available due to deep inference. In the following, we first show that this technique can be applied to system KSg in order to reduce the size of the breadth of the search space, and integrated into a proof construction strategy that exploits invertible rules.

Definition 6. *Given a structure R,* $\mathsf{at}\,R$ *is the set of all the atoms in R. We define* $\mathsf{at}\,\overline{R}$ *as the set obtained by negating all the atoms in the set* $\mathsf{at}\,R$*.*

Example 4. For $R = [\bar{a} \vee b \vee (a \wedge \bar{b})]$, we have $\mathsf{at}\,R = \mathsf{at}\,\overline{R} = \{a, \bar{a}, b, \bar{b}\}$.

Definition 7. *A structure R is a* proper disjunction*, if there are two structures R' and R'' with $R = [R' \vee R'']$ where $R' \neq \mathsf{tt} \neq R''$ and $R' \neq \mathsf{ff} \neq R''$. A structure R is a* proper conjunction*, if there are two structures R' and R'' with $R = (R' \wedge R'')$ where $R' \neq \mathsf{tt} \neq R''$ and $R' \neq \mathsf{ff} \neq R''$.*

Example 5. The structure $[a \vee b]$ is a proper disjunction, whereas a and $(a \wedge [b \vee c])$ are not. $(a \wedge [b \vee c])$ is a proper conjunction.

Definition 8. *[11] Consider the switch rule*

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

We say that it is deep lazy interaction switch (dlis) *if U is not a proper disjunction, R is not a proper conjunction and* $\mathsf{at}\,\overline{R} \cap \mathsf{at}\,U \neq \emptyset$*.*

The idea behind the condition of the rule dlis can be explained as follows: the atomic interaction rule (rule $\mathsf{ai}\!\downarrow$) annihilates dual atoms in a disjunction while going up in a proof. This indicates an interaction scheme in proof construction that is determined by dual atoms sharing a disjunction context, i.e., $S[S_1\{a\} \vee S_2\{\bar{a}\}]$. Because the switch rule manages the disjunctive context of a conjunction in order to bring the structures closer in a disjunction, we can constraint this rule to manage the context in a way that will result in interactions, that is, instances of the rule $\mathsf{ai}\!\downarrow$. If structures R and U do not contain dual atoms, that is, if $\mathsf{at}\,\overline{R} \cap \mathsf{at}\,U = \emptyset$, then they cannot interact. Thus, the rule dlis allows to bring only interacting structures closer in a disjunction.

Example 6. Consider the structure $[(\bar{a} \wedge \bar{b} \wedge \bar{c}) \vee a \vee b \vee c]$ with the following instances of the switch rule where the R and U structures of the switch rule are shaded:

$$(i.)\ \frac{[([a \vee \bar{a}] \wedge \bar{b} \wedge \bar{c}) \vee b \vee c]}{[(\bar{a} \wedge \bar{b} \wedge \bar{c}) \vee a \vee b \vee c]} \qquad (ii.)\ \frac{[([(\bar{b} \wedge \bar{c}) \vee a] \wedge \bar{a}) \vee b \vee c]}{[(\bar{a} \wedge \bar{b} \wedge \bar{c}) \vee a \vee b \vee c]}$$

$(i.)$ is an instance of dlis, because $R = \bar{a}$ and $U = a$, and thus $\mathsf{at}\,\bar{R} \cap \mathsf{at}\,U = \{a\} \neq \emptyset$. $(ii.)$ is not instances of dlis.

In [11], we have show that the rule dlis can replace the rule s in a deep inference system for multiplicative linear logic that consists of the rules $\mathsf{ai}\!\downarrow$ and s. The system obtained remains complete for multiplicative linear logic and admits a much smaller proof search space than the original system, while preserving the shorter proofs that are available due to deep inference. This is because the rule dlis exploits the interaction scheme, determined by the rule $\mathsf{ai}\!\downarrow$ that annihilates dual atoms inside disjunction while going up in the proofs.

The switch rule simulates the context management rule $\mathsf{R}\wedge$ of the sequent calculus as follows:

$$\mathsf{R}\wedge\,\frac{\vdash R, \Phi \quad \vdash T, \Psi}{\vdash R \wedge T, \Phi, \Psi} \qquad \leadsto \qquad \mathsf{s}\,\frac{\mathsf{s}\,\frac{([R \parr \Phi] \otimes [T \parr \Psi])}{[([R \parr \Phi] \otimes T) \parr \Psi]}}{[(R \otimes T) \parr \Phi \parr \Psi]}$$

Once the rule $\mathsf{R}\wedge$ is applied, the sequents in the two branches cannot exchange any formula, thus the communication between the formula at the two branches becomes impossible while going up in the proofs. Proofs are thus constructed by obtaining dual atoms at each branch, so that the axiom can be applied, and weakening rule is applied to annihilate the excessive formulae at each branch that do not contribute to the instance of the axiom.

The rule interaction switch (dlis) realises this idea, while exploiting the capability of the switch rule to manage the contexts gradually and constraining the formula that is considered at each proof search step, however with the possibility to be applied at arbitrary depths inside logical expressions. In its shallow instances, this rule is analogous to constraining the rule $\mathsf{R}\wedge$ such that the sequent Φ consists of a single formula, and R is not a conjunction formula. Because such a restriction in the sequent calculus would not affect the size of the proofs, this allows the rule dlis to polynomially simulate the sequent calculus proofs while providing shorter proofs due to deep inference. Because the rule dlis does not impose any restriction on the deep applicability of the inference rules, it does not cause a loss in shorter proofs.

Thanks to deep inference, we can construct proofs of classical logic structures with system KSg where the contraction and weakening rules are pushed to the bottom of the proofs as in the theorem below. This capability provides a means to carry the ideas in [11] to prove that the rule s can be replaced with the rule dlis in system KSg without loosing completeness.

Definition 9. System KSg with deep lazy interaction switch*, or system* KSgdli *is the system* $\{ \mathsf{ai}\!\downarrow, \mathsf{dlis}, \mathsf{c}\!\downarrow, \mathsf{w}\!\downarrow \}$.

Theorem 1. *[14] If a structure R has a proof in system* KSg*, then there exist structures R_1, R_2, R_3, R_1', R_2', and R_3' and proofs of the following forms:*

$$
\begin{array}{c} \mathsf{tt} \\ \Delta_3 \Big\Vert \{ \mathsf{w}\!\downarrow \} \\ R_3 \\ \Delta_2 \Big\Vert \{ \mathsf{ai}\!\downarrow \} \\ R_2 \\ \Delta_1 \Big\Vert \{ \mathsf{s}, \mathsf{c}\!\downarrow \} \\ R \end{array}
\quad \overset{i.}{\rightsquigarrow} \quad
\begin{array}{c} \mathsf{tt} \\ \Delta_3 \Big\Vert \{ \mathsf{w}\!\downarrow \} \\ R_3 \\ \Delta_2 \Big\Vert \{ \mathsf{ai}\!\downarrow \} \\ R_2 \\ \Delta_{1,b} \Big\Vert \{ \mathsf{s} \} \\ R_1 \\ \Delta_{1,a} \Big\Vert \{ \mathsf{c}\!\downarrow \} \\ R \end{array}
\quad \overset{ii.}{\rightsquigarrow} \quad
\begin{array}{c} \mathsf{tt} \\ \Delta_2 \Big\Vert \{ \mathsf{ai}\!\downarrow \} \\ R_3' \\ \Delta_3 \Big\Vert \{ \mathsf{w}\!\downarrow \} \\ R_2 \\ \Delta_{1,b} \Big\Vert \{ \mathsf{s} \} \\ R_1 \\ \Delta_{1,a} \Big\Vert \{ \mathsf{c}\!\downarrow \} \\ R \end{array}
\quad \overset{iii.}{\rightsquigarrow} \quad
\begin{array}{c} \mathsf{tt} \\ \Delta_2 \Big\Vert \{ \mathsf{ai}\!\downarrow \} \\ R_3' \\ \Delta_{1,b}' \Big\Vert \{ \mathsf{s} \} \\ R_2' \\ \Delta_3' \Big\Vert \{ \mathsf{w}\!\downarrow \} \\ R_1' \\ \Delta_{1,a} \Big\Vert \{ \mathsf{c}\!\downarrow \} \\ R \end{array}
$$

Theorem 2. *Systems* KSg *and* KSgdli *are equivalent.*

Proof (sketch). Every proof in KSgdli is a proof in KSg. For the other direction, apply Theorem 1 to a tautology R to obtain a proof of the form

$$
\begin{array}{c} \mathsf{tt} \\ \Pi \Big\Vert \{ \mathsf{ai}\!\downarrow, \mathsf{s} \} \\ R' \\ \Big\Vert \{ \mathsf{w}\!\downarrow, \mathsf{c}\!\downarrow \} \\ R \end{array}
$$

The proof Π above is in a system which is similar to multiplicative linear logic with the only difference that the equalities $[\mathsf{tt} \vee \mathsf{tt}] \approx \mathsf{tt}$ and $(\mathsf{ff} \wedge \mathsf{ff}) \approx \mathsf{ff}$ hold for this system. We can thus replace the proof Π with a proof in $\{ \mathsf{dlis}, \mathsf{ai}\!\downarrow \}$ by applying the procedure in [11] where we replace proofs of multiplicative linear logic structures in system $\{ \mathsf{s}, \mathsf{ai}\!\downarrow \}$ with proofs in $\{ \mathsf{dlis}, \mathsf{ai}\!\downarrow \}$.

The completeness argument that we use in [11] is based on the strong relationship between cut-elimination and completeness: we use a technique, called *splitting*, which was originally introduced as a cut-elimination technique for deep inference systems [8]. In Theorem 2, we use the splitting argument on system $\{\, \mathsf{ai}\!\downarrow ,\ \mathsf{dlis} \,\}$ together with a semantic cut elimination argument on system KSg. However, Theorem 2 indicates that a splitting theorem on system KSgdli can be used to show the completeness of this system.

Remark 2 (Splitting for system KSgdli*).* We can state the splitting theorem for system KSgdli as follows: for all structures R, T, and P, if $[(R \wedge T) \vee P]$ is provable in KSgdli then there exist P_1, P_2, and a derivation $\begin{array}{c}[P_1 \vee P_2]\\ \Delta \,\|\, \mathsf{KSgdli}\\ P\end{array}$ such that $[R \vee P_1]$ and $[T \vee P_2]$ are provable in KSgdli. Given that system KSg and KSgdli are equivalent, we can easily state this result by resorting to the contraction rule such that derivation Δ is given by the derivation $\mathsf{c}\!\downarrow \dfrac{[P \vee P]}{P}$. This is because there exist proofs of $[R \vee P]$ and $[T \vee P]$ in KSg, thus also in KSgdli by Theorem 2. A constructive proof of this result is a topic of ongoing work.

Remark 3. In [6], Bruscoli and Guglielmi show that Statman tautologies have quadratic size proofs in the size of the proved tautologies in system KSg, in contrast to their exponential size proofs in the sequent calculus. It is straight-forward to see that system KSgdli preserves these quadratic size proofs of Statman tautologies.

When deep inference systems are considered from a proof theoretical point of view, the equalities for unit contribute to the simplicity of this system. However, from the point of view of proof construction, these equalities result in redundant rule instances. Because of this, it is desirable to control their applications in proof construction by means of inference rules that replace these equalities.

Definition 10. *We define the following rules that are called* unit 1*,* unit 2*,* unit 3 *and* unit 4*.*

$$\mathsf{u}_1\!\downarrow \frac{S\{R\}}{S[R \vee \mathsf{ff}]} \qquad \mathsf{u}_2\!\downarrow \frac{S\{R\}}{S(R \wedge \mathsf{tt})} \qquad \mathsf{u}_3\!\downarrow \frac{S\{\mathsf{tt}\}}{S[\mathsf{tt} \vee \mathsf{tt}]} \qquad \mathsf{u}_4\!\downarrow \frac{S\{\mathsf{ff}\}}{S(\mathsf{ff} \wedge \mathsf{ff})}$$

Remark 4. In the rest of the paper, we assume that the relation $\approx$ is defined by only associativity and commutativity of the logical operators. This is because the equalities for unit, in Definition 1, become redundant in proofs when we introduce the rules in Definition 10.

Definition 11. *The following rules are called* co-contraction *(*$\mathsf{c}\!\uparrow$*) [3],* conjunction weakening *(*$\mathsf{w}_1\!\downarrow$*) and* disjunction weakening *(*$\mathsf{w}_2\!\downarrow$*).*

$$\mathsf{c}\!\uparrow \frac{S\{R\}}{S(R \wedge R)} \qquad \mathsf{w}_1\!\downarrow \frac{S\{\mathsf{ff}\}}{S(R \wedge \mathsf{ff})} \qquad \mathsf{w}_2\!\downarrow \frac{S\{\mathsf{tt}\}}{S[R \vee \mathsf{tt}]}$$

Proposition 1. *The rules* $\mathsf{w}_1\!\downarrow$ *and* $\mathsf{w}_2\!\downarrow$ *are derivable for system* KSgdli*.*

Remark 5. The rule $\mathsf{c}\!\uparrow$ is the contrapositive of the rule $\mathsf{c}\!\downarrow$.

Definition 12. *The following rule is called* contractive deep lazy interaction switch (cs)

$$\mathsf{cs}\,\frac{S[([R \vee U] \wedge T) \vee U]}{S[(R \wedge T) \vee U]}$$

where U is not a proper disjunction, R is not a proper conjunction and $\mathsf{at}\,\overline{R} \cap \mathsf{at}\,U \neq \emptyset$*.*

Proposition 2. *The rule* cs *is derivable for* $\{\mathsf{dlis}\,,\, \mathsf{c}\!\downarrow\}$*.*

Proof. Take the following derivation.

$$\mathsf{c}\!\downarrow\,\dfrac{\mathsf{dlis}\,\dfrac{S[([R \vee U] \wedge T) \vee U]}{S[(R \wedge T) \vee U \vee U]}}{S[(R \wedge T) \vee U]} \quad \rightsquigarrow \quad \mathsf{cs}\,\frac{S[([R \vee U] \wedge T) \vee U]}{S[(R \wedge T) \vee U]}$$

Definition 13. *The system* $\{\, \mathsf{u}_1\!\downarrow\,,\, \mathsf{u}_2\!\downarrow\,,\, \mathsf{u}_3\!\downarrow\,,\, \mathsf{u}_4\!\downarrow\,,\, \mathsf{w}_1\!\downarrow\,,\, \mathsf{w}_2\!\downarrow\,,\, \mathsf{c}\!\uparrow\,,\, \mathsf{ai}\!\downarrow\,,\, \mathsf{cs} \,\}$ *is called* system KG.

Theorem 3. *Systems* KG *and* KSgdli *are equivalent.*

Proof (sketch). All the rules of system KG are derivable for system KSgdli. For the proof of the other direction, take a proof of a tautology R obtained by applying Theorem 2 where the instances of the rules $\mathsf{c}\!\downarrow$ and $\mathsf{w}\!\downarrow$ are at the bottom of the proof. We can permute all the instances of the rule $\mathsf{w}\!\downarrow$ to the top of the proof where they can be replaced with the instances of the rules $\mathsf{w}_1\!\downarrow$ and $\mathsf{w}_2\!\downarrow$. We replace each instance of the rule $\mathsf{c}\!\downarrow$, that has a proper disjunction as its redex, with two instances of this rule applied to two structures in the disjunction. We then permute up all the instances of the contractions, where we can either annihilate the substructures of the contractum in an instance of one of the rules $\mathsf{u}_1\!\downarrow\,,\, \mathsf{u}_2\!\downarrow\,,\, \mathsf{w}_1\!\downarrow\,,\, \mathsf{w}_2\!\downarrow\,,\, \mathsf{c}\!\uparrow$ and $\mathsf{ai}\!\downarrow$, or we can replace the instances of the rules $\mathsf{c}\!\downarrow$ and dlis with the instances of cs as shown in the proof of Proposition 2.

Remark 6. All the rules of system KG are invertible rules.

Example 7. A proof of the structure in Example 1 in system KG is as follows. (We omit the instances of the rules $\mathsf{u}_1\!\downarrow$, $\mathsf{u}_2\!\downarrow$, $\mathsf{u}_3\!\downarrow$, $\mathsf{u}_4\!\downarrow$, $\mathsf{w}_1\!\downarrow$ and $\mathsf{w}_2\!\downarrow$.)

$$\mathsf{cs}\,\dfrac{\mathsf{ai}\!\downarrow\,\dfrac{\mathsf{cs}\,\dfrac{\mathsf{ai}\!\downarrow\,\dfrac{\mathsf{c}\!\uparrow\,\dfrac{\mathsf{cs}\,\dfrac{\mathsf{ai}\!\downarrow\,\dfrac{\mathsf{ai}\!\downarrow\,\dfrac{\mathsf{tt}}{[b \vee \bar{b} \vee \bar{a} \vee \bar{c} \vee \bar{d}]}}{[([a \vee \bar{a}] \wedge b) \vee \bar{a} \vee \bar{b} \vee \bar{c} \vee \bar{d}]}}{[(a \wedge b) \vee \bar{a} \vee \bar{b} \vee \bar{c} \vee \bar{d}]}}{[(a \wedge b) \vee ([\bar{a} \vee \bar{b}] \wedge [\bar{a} \vee \bar{b}]) \vee \bar{c} \vee \bar{d}]}}{[(a \wedge b) \vee ([\bar{a} \vee \bar{b}] \wedge [\bar{a} \vee \bar{b}] \wedge [d \vee \bar{d}]) \vee \bar{c} \vee \bar{d}]}}{[(a \wedge b) \vee ([\bar{a} \vee \bar{b}] \wedge [\bar{a} \vee \bar{b}] \wedge d) \vee \bar{c} \vee \bar{d}]}}{[(a \wedge b) \vee ([\bar{a} \vee \bar{b}] \wedge [c \vee \bar{c}] \wedge [\bar{a} \vee \bar{b}] \wedge d) \vee \bar{c} \vee \bar{d}]}}{[(a \wedge b) \vee ([\bar{a} \vee \bar{b}] \wedge c \wedge [\bar{a} \vee \bar{b}] \wedge d) \vee \bar{c} \vee \bar{d}]}$$

Definition 14. *An instance of the rule* $\mathsf{cs}*$ *is an instance of the rule* cs*, given in Definition 12, if the rule* cs *cannot be applied bottom-up to the structures* R *and* U*.*

Definition 15. *A proof system* $\mathscr{S}$ *p-simulates a proof system* $\mathscr{S}'$ *if there is a polynomial time computable algorithm that transforms every proof in* $\mathscr{S}'$ *into a proof in* $\mathscr{S}$*.*

Definition 16. *For any two rules* ρ_1 *and* ρ_2*, the* $\rho_1 \prec \rho_2$ *denotes a strategy for the bottom-up application of the rules* ρ_1 *and* ρ_2 *such that when possible always* ρ_1 *is applied exhaustively before* ρ_2*.*

Conjecture 1. For any sequent calculus system $\mathscr{S}$ for classical logic where the inference rules can be applied only at the top-level (main) connective, the following holds: (*i.*) System KG p-simulates $\mathscr{S}$. (*ii.*) There are classes of classical logic structures, e.g., Statman tautologies, for which the strategy given below provides polynomial size proofs in contrast to their exponential size proofs in $\mathscr{S}$.

$$\mathsf{u}_1\!\downarrow\ \prec \mathsf{u}_2\!\downarrow\ \prec \mathsf{u}_3\!\downarrow\ \prec \mathsf{u}_4\!\downarrow\ \prec \mathsf{w}_1\!\downarrow\ \prec\ \mathsf{w}_2\!\downarrow\ \prec\ \mathsf{c}\!\uparrow\ \prec\ \mathsf{ai}\!\downarrow\ \prec\ \mathsf{cs}*$$

All the rule instances, except those of the rule cs, shrink the size of the structure being proved. Then applying the rule $\mathsf{cs}*$ should allow to construct proofs by annihilating substructures. Because we prioritise the instances of this rule in deeper contexts the condition of this rule is applied to smaller structures and also the structures that are copied by this rule remain smaller substructures.

4 Discussion

We have defined a set of rules within the methodology of deep inference for classical propositional logic, with the aim of providing the ingredients of a deep inference theorem prover that performs better than shallow inference theorem provers. In order to overcome the nondeterminism that is present due to deep applicability of the inference rules, we have integrated the techniques that were presented in [13, 11]. The design principle of system KG and the strategy that we define on this system is keeping all the rules invertible, while delaying the application of the contraction rule when possible and preserving deep inference. This way, we can apply the inference rules with the aim of shrinking the size of the formula being proved and minimise the exponential growth in the size of the formula due to the instances of the contraction rule when it is possible.

In [6], Bruscoli and Guglielmi show that Statman tautologies have quadratic size proofs in the size of the proved tautologies in system KSg, in contrast to their exponential size proofs in the sequent calculus. It is straight-forward to see that system KSgdli preserves these quadratic size proofs of Statman tautologies. We believe that system KG and the search strategy that we have defined on this system do not result in a loss of also other short proofs that are available due to deep inference, because our restrictions promote the interaction between dual atoms, while preserving deep applicability of the inference rules. Our long term goal is developing analytic deep inference theorem provers for modal logics [7, 16] and fragments of linear logics [19, 18]. In this regard, splitting theorems can provide the proof theoretical counter-parts of strategies, similar to selection rules in resolution theorem provers.

Acknowledgements: The author would like to thank Alessio Guglielmi, Paola Bruscoli and anonymous referees for comments and improvements.

References

1. Hasan Amjad. Compressing propositional refutations. In S. Merz and T. Nipkow, editors, *Proceedings of the 6th International Workshop on Automated Verification of Critical Systems*, volume 185 of *ENTCS*, pages 3–15. Elsevier, 2007.
2. Kai Brünnler. Atomic cut elimination for classical logic. In M. Baaz and J. A. Makowsky, editors, *CSL 2003*, volume 2803 of *LNCS*, pages 86–97. Springer, 2003.
3. Kai Brünnler. *Deep Inference and Symmetry in Classical Proofs*. PhD thesis, Technische Universität Dresden, 2003.
4. Kai Brünnler. Deep inference and its normal form of derivations. In Arnold Beckmann, Ulrich Berger, Benedikt Löwe, and John V. Tucker, editors, *Computability in Europe 2006*, volume 3988 of *LNCS*, pages 65–74. Springer, July 2006.
5. Kai Brünnler. Locality for classical logic. *Notre Dame Journal of Formal Logic*, 47(4):557–580, 2006.
6. Paola Bruscoli and Alessio Guglielmi. On the proof complexity of deep inference. *ACM Transactions on Computational Logic*, 2008. In press.
7. Rajeev Goré and Alwen Tiu. Classical modal display logic in the calculus of structures and minimal cut-free deep inference calculi for S5. *Journal of Logic and Computation*, 17(4):767–794, 2007.
8. Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1):1–64, 2007.
9. Alessio Guglielmi and Tom Gundersen. Normalisation control in deep inference via atomic flows. *Logical Methods in Computer Science*, 2008. In press.
10. Alessio Guglielmi and Lutz Straßburger. A system of interaction and structure IV: The exponentials. In the second round of revision for Mathematical Structures in Computer Science., 2007.
11. Ozan Kahramanoğulları. Interaction and depth against nondeterminism in deep inference proof search. submitted, 2008.
12. Ozan Kahramanoğulları. *Nondeterminism and Language Design in Deep Inference*. PhD thesis, TU Dresden, 2006.
13. Ozan Kahramanoğulları. Reducing nondeterminism in the calculus of structures. In Miki Hermann and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning, Proceedings of the 13th International Conference, LPAR 2006, Phnom Penh, Cambodia*, volume 4246 of *LNCS*, pages 272–286. Springer, 2006.
14. Ozan Kahramanoğulları. Maude as a platform for designing and implementing deep inference systems. In *Proceedings of the Eighth International Workshop on Rule-Based Programming, RULE'07*, ENTCS. Elsevier, 2008. In press.
15. Richard Statman. Bounds for proof-search and speed-up in the predicate calculus. *Annals of Mathematical Logic*, 15:225–287, 1978.

16. Phiniki Stouppa. A deep inference system for the modal logic S5. *Studia Logica*, 85(2):199–214, 2007.
17. Lutz Straßburger. A local system for linear logic. In M. Baaz and A. Voronkov, editors, *LPAR 2002*, volume 2514 of *LNAI*, pages 388–402. Springer, 2002.
18. Lutz Straßburger. *Linear Logic and Noncommutativity in the Calculus of Structures*. PhD thesis, TU Dresden, 2003.
19. Lutz Straßburger. MELL in the calculus of structures. *Theoretical Computer Science*, 309:213–285, 2003.
20. Alwen Fernanto Tiu. A local system for intuitionistic logic. In Miki Hermann and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning, Proceedings of the 13th International Conference, LPAR 2006, Phnom Penh, Cambodia*, volume 4246 of *LNCS*, pages 242–256. Springer, 2006.
21. Alwen Fernanto Tiu. A system of interaction and structure II: the need for deep inference. *Logical Methods in Computer Science*, 2 (2:4):1–24, April 2006.
22. Anne Sjerp Troelstra and Helmut Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, 1996.
23. Tjark Weber. Integrating a sat solver with an lcf-style theorem prover. In A. Armando and A. Cimatti, editors, *Proceedings of the Third Workshop on Pragmatics of Decision Procedures in Automated Reasoning*, volume 144 of *ENTCS*, pages 67–78. Elsevier, 2006.