Preprint, Institute of Mathematics, AS CR, Prague. 2007-12-18 (rev. 1, 2008-8-4)

Proof complexity of the cut-free calculus of structures

Emil Jeřábek

Institute of Mathematics of the Academy of Sciences

Žitná 25, 115 67 Praha 1, Czech Republic, email: `jerabek@math.cas.cz`

April 30, 2008

Abstract

We investigate the proof complexity of analytic subsystems of the deep inference proof system *SKSg* (the calculus of structures). Exploiting the fact that the cut rule ($i\uparrow$) of *SKSg* corresponds to the $\neg$-left rule in the sequent calculus, we establish that the "analytic" system $KSg + c\uparrow$ has essentially the same complexity as the monotone Gentzen calculus *MLK*. In particular, $KSg + c\uparrow$ quasipolynomially simulates *SKSg*, and admits polynomial-size proofs of some variants of the pigeonhole principle.

Keywords: proof complexity, calculus of structures, monotone sequent calculus, cut rule

1 Introduction

The *calculus of structures* (*CoS*) is a recent proof-theoretic formalism initially developed by Guglielmi [9, 10] as an alternative to the sequent calculus. It is based on the idea of *deep inference*: CoS rules can apply to any place deep inside a formula, in contrast to the usual sequent or Hilbert-style calculi, which only operate on the top part. The most popular CoS proof systems for the classical propositional logic—*SKSg* and its variants—were introduced by Brünnler [3]. The proof complexity of CoS proof systems was studied by Bruscoli and Guglielmi [5], who have shown that *SKSg* is polynomially equivalent to the usual sequent or Frege systems, but they leave open the question of the complexity of so-called *analytic* subsystems *KSg* and $KSg + c\uparrow$ of *SKSg*.

In sequent calculi, "analytic" is more or less a synonym for "cut-free": a proof system is analytic if formulas from the premises of any rule appear as building blocks (subformulas) in the conclusion of the rule, which means it has the *subformula property.* Cut-free proof systems and their subformula property have many applications in proof theory (e.g., ordinal analysis, conservativity results among fragments of arithmetic, decision procedures for nonclassical logics, interpolation and explicit definability, etc.), and indeed, the cut-elimination theorem was the main reason for Gerhard Gentzen to introduce the sequent calculus in the first place.

Due to the nature of the proof system, the subformula property does not make much sense in CoS. Nevertheless, substantially weaker notions of analyticity are used in the CoS literature (cf. [5]) under which some CoS systems are designated as analytic. In particular, there is a rule ($i\uparrow$, see Table 1) called the "cut rule", which is disallowed in analytic CoS. While analytic CoS bears some superficial resemblance to analytic sequent calculi (the cut rule is eliminable, and in fact, analytic CoS systems can simulate cut-free LK), there are also significant differences. As already mentioned, analytic CoS does not enjoy the subformula property. The most salient feature of the subformula property, which is sufficient for many of its applications, is that one can bound the complexity (e.g., the depth, or the number of quantifier alternations) of formulas appearing in the proof in terms of the complexity of formulas in its endsequent. (The same property is also responsible for the exponential gap between the proof complexity of cut-free and cut-full sequent calculi.) However, analytic CoS does not share this weaker property either: as we will see, there are depth-2 tautologies whose cut-free CoS proofs may contain arbitrary formulas. Another unorthodox feature is that the CoS cut rule is almost trivially reducible to its atomic special case.

The unusual behaviour of the analytic CoS systems and the CoS "cut" rule is explained by the equivalence of CoS to (two-sided) sequent calculus (Brünnler [4], McKinley [13]), where an LK-proof of a sequent $\Gamma \vdash \Delta$ corresponds to an $SKSg$-derivation of the formula $\bigvee \Delta$ from the premise $\bigwedge \Gamma$. We observe that in this translation, the CoS cut rule corresponds to the $\neg$-left sequent rule[1] (over a basic system, which may be chosen as $\neg$-free multiplicative linear logic on the sequent side, and a subsystem of KSg without any of the "structural rules" on the CoS side), whereas the sequent cut rule comes out essentially for free. The key point is that the correspondence is faithful in both directions, it provides for a translation of subsystems of $SKSg$ to subsystems of LK as well as vice versa[2].

In terms of proof complexity, the translation gives a polynomial equivalence of fragments of $SKSg$ to the corresponding fragments of the *tree-like* sequent calculus. We will use it to show that the "analytic" CoS system $KSg + c\uparrow$ has the same complexity as the tree-like monotone sequent calculus MLK. We present a simple elimination procedure for the coweakening ($w\uparrow$) rule; this allows us to work with the more convenient system $KSg+c\uparrow+w\uparrow$, which is equivalent to the $\neg$-left-free fragment of tree-like LK in the correspondence above. We establish that $KSg + c\uparrow + w\uparrow$ is polynomially equivalent to tree-like MLK with respect to derivations of monotone formulas from monotone formulas. We also show that this form of derivations is universal, in the sense that there is a natural translation of an arbitrary formula to a pair of a monotone assumption and a monotone conclusion which preserves (up to a polynomial) the size of $KSg + c\uparrow + w\uparrow$-derivations. For all purposes and intents, $KSg + c\uparrow$ and $KSg + c\uparrow + w\uparrow$ are thus polynomially equivalent to tree-like MLK; the only obstacle which prevents us from technically claiming the equivalence in this form is the mismatch in the languages of these systems (MLK cannot prove formulas, only monotone sequents).

[1] To avoid potential confusion: we work with formulas in negation normal form, i.e., $\neg$ is a primitive operation only on propositional variables, and its action is extended elsewhere by De Morgan's laws. In particular, this makes the $\neg$-left rule weak enough so that it is eliminable from proofs of a sequent with an empty antecedent.

[2] This is not true of the more usual embedding of the *one-sided* sequent calculus to CoS (as in [3, 5]), which is used to justify the label "cut" for the $i\uparrow$ rule. It allows to transform cut-free sequent proofs to KSg-proofs, but it lacks a matching translation of KSg back to cut-free sequent calculus. Indeed, KSg cannot be adequately expressed as a fragment of the one-sided sequent calculus: the distinction between the cut and $\neg$-left rules is lost in the one-sided calculus, where they are combined into a single rule (generally called the cut rule).

The monotone sequent calculus was studied by Atserias, Galesi, and Pudlák [1], who have shown that tree-like *MLK* *quasipolynomially* simulates full *LK* (wrt monotone sequents). Moreover, the available evidence seems to suggest that the calculi are in fact polynomially equivalent, although the problem remains open. As a corollary, we obtain that $KSg + c\uparrow$ quasipolynomially simulates (in the usual way, i.e., wrt proofs of formulas) *SKSg* (or equivalently, *LK*). Additionally, if the simulation of *LK* by tree-like *MLK* can be made polynomial, then *SKSg* and $KSg + c\uparrow$ are polynomially equivalent as well.

We also include another result on the complexity of *KSg*, which is not directly related to the correspondence with sequent calculus. We show that there exists a polynomial-time translation of *SKSg* to *KSg* based on a simple modification of the formula being proved. It can be also considered as a kind of a normal form for *SKSg*-proofs: all $i\uparrow$ (cut) inferences can be postponed until the end of the proof, and we can a priori bound their number, we need only one instance of $i\uparrow$ for each propositional variable appearing in the conclusion of the proof. We conclude that *KSg* cannot have feasible interpolation (under the same assumptions as *LK*), and we construct polynomial $KSg + c\uparrow$-proofs of some variants of the pigeonhole principle.

The paper is organized as follows. In Section 2 we present the relevant definitions and basic facts about the proof systems we are going to work with. In Section 3 we exhibit the "normal form" for $i\uparrow$ inferences, and its applications. In Section 4 we review the correspondence of CoS to the two-sided sequent calculus, and in Section 5 we discuss the connections to the monotone sequent calculus.

Acknowledgements

I wish to thank Marta Bílková for an interesting discussion on the topic, Richard McKinley and Alessio Guglielmi for comments on an earlier version of the paper (in particular, for drawing my attention to [4, 13]), and an anonymous referee for useful suggestions. The research was supported by grants IAA900090703 and IAA1019401 of GA AV ČR, and grant 1M0545 of MŠMT ČR.

2 Preliminaries

The fundamental notion of a general proof system was introduced by Cook and Reckhow [8].

Definition 2.1 Let L be a set of strings in a finite alphabet. A *proof system* for L is a polynomial-time function P such that L is the range of P. Any x such that $P(x) = y$ is called a *proof* of y. Let P and Q be proof systems for L. We say that P *polynomially simulates* (or *p-simulates*) Q, written as $Q \leq_p P$, if there exists a polynomial-time function f such that $Q = P \circ f$. If $P \leq_p Q$ and $Q \leq_p P$, the proof systems P and Q are *p-equivalent*.

We are interested in proof systems for the classical propositional logic, i.e., L is the set $TAUT$ of classical propositional tautologies. Typical proof systems, like sequent calculi, fit the Cook–Reckhow definition if we put

$$P(\pi) = \begin{cases} \varphi & \text{if } \pi \text{ is a proof with conclusion } \varphi, \\ \top & \text{if } \pi \text{ is not a valid proof.} \end{cases}$$

The definition of polynomial simulation amounts to the following when expanded: given a Q-proof π of a formula φ, we can construct in polynomial time a P-proof $f(\pi)$ of φ. The *size* $|\pi|$ of a proof π is strictly speaking the length of the string which represents π, but we will be content with a more liberal definition which only counts the number of occurrences of symbols (so that every propositional variables has size 1, independent of its index). For more background in proof complexity, the reader may consult e.g. Krajíček [12].

We proceed to define the CoS and sequent proof systems we will work with.

Definition 2.2 The *formulas* of the calculus of structures are built using the monotone connectives $\wedge$, $\vee$, $\top$, and $\bot$ from literals (atoms), which are propositional variables p_i and their negations $\neg p_i$. We extend $\neg$ to an involutive operation on all formulas using De Morgan's laws. A *context* $\xi\{\ \}$ is a formula in which exactly one *hole* $\{\ \}$ appears in place of a literal. If $\xi\{\ \}$ is a context, and φ a formula, we denote by $\xi\{\varphi\}$ the formula resulting by filling the hole with φ. A CoS *derivation* of a formula ψ from a formula φ is a sequence of formulas $\varphi = \varphi_0, \varphi_1, \dots, \varphi_m = \psi$, where φ_{i+1} is derived by a rule of the calculus from φ_i (all rules are unary). We will write

$$\frac{\varphi}{\overline{\overline{\psi}}} ,$$

possibly decorated with the name of the proof system or other information, if a derivation of ψ from φ exists. A *proof* of a formula φ is a derivation of φ from $\top$.

We consider the rules given in Table 1, where $\xi\{\ \}$ denotes an arbitrary context, and $\varphi, \psi, \chi, \omega$ are arbitrary formulas. We define *Sg* to be the calculus using the switch rule (s), and the eight $\wedge$ and $\vee$ rules. *KSg* is *Sg* together with $i{\downarrow}$ (identity), $w{\downarrow}$ (weakening), $c{\downarrow}$ (contraction), and the x_i rules. *SKSg* extends *KSg* by the rules $i{\uparrow}$ (cut), $w{\uparrow}$ (coweakening), and $c{\uparrow}$ (cocontraction).

If ϱ is one of the rules $i{\downarrow}$, $i{\uparrow}$, $c{\downarrow}$, $c{\uparrow}$, $w{\downarrow}$, $w{\uparrow}$, we denote by $a\varrho$ the restriction of ϱ which only allows a literal as the formula φ. The calculus *KS* consists of *Sg*, m, $ai{\downarrow}$, $aw{\downarrow}$, $ac{\downarrow}$, and the x_i rules. *SKS* is defined as $KS + \{ai{\uparrow}, aw{\uparrow}, ac{\uparrow}\}$.

Subsystems of $KSg + c{\uparrow}$ or $KS + ac{\uparrow}$ are called *analytic*[3].

Remark 2.3 It is obvious from the form of the CoS rules that given a derivation

$$\frac{\varphi}{\overline{\overline{\psi}}}$$

[3]There are issues with a general definition of analyticity in CoS, see e.g. [6]. We avoid the problem by giving a list; there seems to be a consensus in CoS sources that the systems we mentioned are analytic, whereas the $i{\uparrow}$ and $w{\uparrow}$ rules are not, which is all that matters for our purposes.

$\wedge u_1 \,\dfrac{\xi\{\varphi \wedge \top\}}{\xi\{\varphi\}}$	$\vee u_1 \,\dfrac{\xi\{\varphi \vee \bot\}}{\xi\{\varphi\}}$	$\wedge a \,\dfrac{\xi\{\varphi \wedge (\psi \wedge \chi)\}}{\xi\{(\varphi \wedge \psi) \wedge \chi\}}$
$\wedge u_2 \,\dfrac{\xi\{\varphi\}}{\xi\{\varphi \wedge \top\}}$	$\vee u_2 \,\dfrac{\xi\{\varphi\}}{\xi\{\varphi \vee \bot\}}$	$\vee a \,\dfrac{\xi\{\varphi \vee (\psi \vee \chi)\}}{\xi\{(\varphi \vee \psi) \vee \chi\}}$
$\wedge c \,\dfrac{\xi\{\varphi \wedge \psi\}}{\xi\{\psi \wedge \varphi\}}$	$\vee c \,\dfrac{\xi\{\varphi \vee \psi\}}{\xi\{\psi \vee \varphi\}}$	$s \,\dfrac{\xi\{\varphi \wedge (\psi \vee \chi)\}}{\xi\{(\varphi \wedge \psi) \vee \chi\}}$
$i{\downarrow} \,\dfrac{\xi\{\top\}}{\xi\{\varphi \vee \neg\varphi\}}$	$w{\downarrow} \,\dfrac{\xi\{\bot\}}{\xi\{\varphi\}}$	$c{\downarrow} \,\dfrac{\xi\{\varphi \vee \varphi\}}{\xi\{\varphi\}}$
$i{\uparrow} \,\dfrac{\xi\{\varphi \wedge \neg\varphi\}}{\xi\{\bot\}}$	$w{\uparrow} \,\dfrac{\xi\{\varphi\}}{\xi\{\top\}}$	$c{\uparrow} \,\dfrac{\xi\{\varphi\}}{\xi\{\varphi \wedge \varphi\}}$
$x_1 \,\dfrac{\xi\{\top\}}{\xi\{\top \vee \top\}}$	$x_2 \,\dfrac{\xi\{\bot \wedge \bot\}}{\xi\{\bot\}}$	$m \,\dfrac{\xi\{(\varphi \wedge \psi) \vee (\chi \wedge \omega)\}}{\xi\{(\varphi \vee \chi) \wedge (\psi \vee \omega)\}}$
$x_3 \,\dfrac{\xi\{\top \vee \top\}}{\xi\{\top\}}$	$x_4 \,\dfrac{\xi\{\bot\}}{\xi\{\bot \wedge \bot\}}$	

Table 1: Rules of the calculus of structures

with k lines and size s, we can construct a derivation of

$$\begin{array}{c}\xi\{\varphi\}\\ \hline\hline \xi\{\psi\}\end{array}$$

with k lines and size $s + k|\xi|$, for any context ξ. We will often tacitly use this observation.

In the original formulation, *KSg* and friends include an "equality rule" $=$, consisting of the transitive closure of the rules which we denote by $\wedge\cdots$, $\vee\cdots$, and x_i. While [5] show that $=$ is a polynomial-time recognizable rule (and thus acceptable as a rule in a Cook–Reckhow proof system), we find it too complicated to work with. We thus split it into several rules which are treated on the same footing as the other rules of the system, and give them individual names to ease reference. (This is only a cosmetic change in terms of the proof complexity of the systems, as any instance of the $=$ rule can be polynomially simulated by a sequence of the new rules.) We will nevertheless occasionally use the collective name $=$ for convenience.

The $\wedge$ and $\vee$ rules enforce basic properties of the two connectives (associativity, commutativity, and neutrality of its unit), and we thus include them in the basic system *Sg*. On the other hand, the x_i rules are apparently only an auxiliary device used to reduce weakening and contraction to their atomic variants in *KS* and *SKS*, and in particular, the x_i rules do not have a nice interpretation in the sequent calculus (as we will see shortly). Notice that $x_{1,2,3,4}$ are redundant in *KSg* (they are derivable from $\vee u_2 + w{\downarrow}$, $w{\downarrow} + \wedge u_1$, $c{\downarrow}$, and $w{\downarrow}$, respectively).

The next theorem shows that the up and down i, w, c rules can be reduced to the atomic cases (using the x_i rules, and—in the case of contraction—the m rule), which means that the choice between $(S)KS$ and $(S)KSg$ is only a matter of convenience. We will generally find it more natural to work with the *KSg* and *SKSg* variants, but we will sometimes appeal to the restricted versions of the rules as well.

Theorem 2.4 (Bruscoli, Guglielmi [5]) *The calculus KSg is polynomially equivalent to KS, and SKSg is polynomially equivalent to SKS.*

In more detail, any instance of $i{\downarrow}$ *(*$i{\uparrow}$*,* $w{\downarrow}$*,* $w{\uparrow}$*) has a polynomial-time constructible derivation using* s*,* $=$*, and* $ai{\downarrow}$ *(*$ai{\uparrow}$*,* $aw{\downarrow}$*,* $aw{\uparrow}$*, respectively). Any instance of* $c{\downarrow}$ *(*$c{\uparrow}$*) has a polynomial-time constructible derivation using* m*,* $=$*, and* $ac{\downarrow}$ *(*$ac{\uparrow}$*).*

Definition 2.5 A *sequent* is an expression of the form $\Gamma \vdash \Delta$, where Γ and Δ are finite sequences of formulas. For consistency with CoS we work with formulas in negation normal form, with $\neg$ defined as an operator as in Definition 2.2. A (dag-like) *proof* in a sequent calculus is a sequence of sequents, each of which is derived from some of the previous sequents by a rule of the calculus. A proof is *tree-like* if every sequent is used at most once as a hypothesis. A proof of the sequent $\vdash \varphi$ is considered a proof of the formula φ.

We consider the rules introduced in Table 2. We define *MLL* to be the calculus consisting of the identity (i), cut, and exchange (e-l, e-r) rules together with the left and right rules for $\wedge$, $\vee$, $\bot$, $\top$. The *monotone Gentzen calculus MLK* consists of *MLL* and the weakening (w-l, w-r) and contraction (c-l, c-r) rules. The *Gentzen calculus LK* extends *MLK* by the left and right rules for $\neg$.

Remark 2.6 We adhere to proof complexity conventions with regard to the shape of sequent proofs. For proof theorists: a "tree-like proof" is just a proof, and a "dag-like proof" is a proof where repeated occurrences of a subproof may be replaced with a simple reference to the first occurrence (and thus do not contribute to the overall size of the proof). Tree-like *LK* is well-known to be polynomially equivalent to *LK* (Krajíček [11, 12]), but this is not necessarily true for its subsystems.

MLL is a notational variant of the $\neg$-free fragment of the multiplicative linear logic, hence the name. The monotone calculus *MLK* is more properly defined as the subsystem of *LK* which only allows monotone (i.e., $\neg$-free) formulas to appear in the proof. This makes no significant difference, as long as we use *MLK* only to prove monotone sequents (we can replace negative literals which sneak in an *MLK*-proof by new variables).

The Gentzen calculus *LK* is polynomially equivalent to other standard proof systems, such as Frege (or Hilbert-style)systems, and natural deduction [8].

The x_i rules are special instances of weakening and contraction, which will be used only to translate the corresponding CoS rules.

Theorem 2.7 (Bruscoli, Guglielmi [5]) *The calculus SKSg is polynomially equivalent to LK.*

Theorem 2.8 (Atserias et al. [1]) *Tree-like MLK quasipolynomially simulates LK.*

In more detail, if a monotone sequent in n *variables has an LK-proof of size* s*, then we can construct in quasipolynomial time its tree-like MLK-proof of size* $s^{O(1)}n^{O(\log n)}$ *with* $s^{O(1)}$ *lines.*

$$
i\,\frac{}{\varphi \vdash \varphi} \qquad \top\text{-r}\,\frac{}{\vdash \top} \qquad \bot\text{-r}\,\frac{\Gamma \vdash \Delta}{\Gamma \vdash \bot, \Delta}
$$

$$
cut\,\frac{\Gamma \vdash \varphi, \Delta \qquad \Pi, \varphi \vdash \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda} \qquad \top\text{-l}\,\frac{\Gamma \vdash \Delta}{\Gamma, \top \vdash \Delta} \qquad \bot\text{-l}\,\frac{}{\bot \vdash}
$$

$$
e\text{-r}\,\frac{\Gamma \vdash \Delta, \varphi, \psi, \Lambda}{\Gamma \vdash \Delta, \psi, \varphi, \Lambda} \qquad \wedge\text{-r}\,\frac{\Gamma \vdash \varphi, \Delta \qquad \Pi \vdash \psi, \Lambda}{\Gamma, \Pi \vdash \varphi \wedge \psi, \Delta, \Lambda} \qquad \vee\text{-r}\,\frac{\Gamma \vdash \varphi, \psi, \Delta}{\Gamma \vdash \varphi \vee \psi, \Delta}
$$

$$
e\text{-l}\,\frac{\Gamma, \varphi, \psi, \Pi \vdash \Delta}{\Gamma, \psi, \varphi, \Pi \vdash \Delta} \qquad \wedge\text{-l}\,\frac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \varphi \wedge \psi \vdash \Delta} \qquad \vee\text{-l}\,\frac{\Gamma, \varphi \vdash \Delta \qquad \Pi, \psi \vdash \Lambda}{\Gamma, \Pi, \varphi \vee \psi \vdash \Delta, \Lambda}
$$

$$
w\text{-r}\,\frac{\Gamma \vdash \Delta}{\Gamma \vdash \varphi, \Delta} \qquad c\text{-r}\,\frac{\Gamma \vdash \varphi, \varphi, \Delta}{\Gamma \vdash \varphi, \Delta} \qquad \neg\text{-r}\,\frac{\Gamma, \varphi \vdash \Delta}{\Gamma \vdash \neg\varphi, \Delta}
$$

$$
w\text{-l}\,\frac{\Gamma \vdash \Delta}{\Gamma, \varphi \vdash \Delta} \qquad c\text{-l}\,\frac{\Gamma, \varphi, \varphi \vdash \Delta}{\Gamma, \varphi \vdash \Delta} \qquad \neg\text{-l}\,\frac{\Gamma \vdash \varphi, \Delta}{\Gamma, \neg\varphi \vdash \Delta}
$$

$$
x_1\,\frac{\Gamma \vdash \top, \Delta}{\Gamma \vdash \top, \top, \Delta} \qquad x_2\,\frac{\Gamma, \bot \vdash \Delta}{\Gamma, \bot, \bot \vdash \Delta}
$$

$$
x_3\,\frac{\Gamma \vdash \top, \top, \Delta}{\Gamma \vdash \top, \Delta} \qquad x_4\,\frac{\Gamma, \bot, \bot \vdash \Delta}{\Gamma, \bot \vdash \Delta}
$$

Table 2: Rules of the sequent calculus

3 An almost simulation of $i\uparrow$

Lemma 3.1 *Given a context $\xi\{\ \}$, and a formula φ, there are polynomial-time constructible Sg-proofs of*

$$
\frac{\xi\{\varphi\}}{\overline{\varphi \vee \xi\{\bot\}}}\ .
$$

Proof: By induction on the complexity of ξ. The base case $\xi\{\ \} = \{\ \}$ is an instance of $\vee u_2$. The induction step for conjunction can be derived as

$$
\begin{array}{r c}
\text{I.H.} & \dfrac{\psi \wedge \xi\{\varphi\}}{\overline{\psi \wedge (\varphi \vee \xi\{\bot\})}} \\
s, \vee c & \dfrac{}{\overline{\varphi \vee (\psi \wedge \xi\{\bot\})}}
\end{array},
$$

and the induction step for disjunction follows easily from $\vee a$ and $\vee c$. $\square$

Theorem 3.2 *Given an SKSg-proof of a formula φ in variables p_i, $i < n$, we can construct in polynomial time a KSg-proof of the formula*

$$
\varphi \vee \bigvee_{i<n} (p_i \wedge \neg p_i).
$$

Proof: We substitute truth constants for variables other than p_i, $i < n$, which may appear in the proof. We eliminate instances of $w\uparrow$ and $c\uparrow$ in favor of $i\uparrow$, using subproofs of the form

$$
\begin{array}{r}
\wedge u_2, \vee u_2 \\ w\downarrow \\ s \\ i\uparrow \\ \vee u_1
\end{array}
\begin{array}{c}
\psi \\ \hline\hline \psi \wedge (\bot \vee \top) \\ \hline \psi \wedge (\neg\psi \vee \top) \\ \hline (\psi \wedge \neg\psi) \vee \top \\ \hline \bot \vee \top \\ \hline \top
\end{array}
\qquad
\begin{array}{r}
\wedge u_2, i\downarrow \\ c\downarrow \\ s \\ i\uparrow, \vee u_1
\end{array}
\begin{array}{c}
\psi \\ \hline\hline \psi \wedge ((\neg\psi \vee \neg\psi) \vee (\psi \wedge \psi)) \\ \hline \psi \wedge (\neg\psi \vee (\psi \wedge \psi)) \\ \hline (\psi \wedge \neg\psi) \vee (\psi \wedge \psi) \\ \hline\hline \psi \wedge \psi
\end{array} .
$$

By Theorem 2.4, we may assume that all instances of $i\uparrow$ in the proof are atomic. We put all formulas in the proof into the context $\{\,\} \vee \bigvee_{i<n}(p_i \wedge \neg p_i)$. We prefix the derivation with the subproof

$$
\begin{array}{r}
\vee u_2 \\ w\downarrow
\end{array}
\begin{array}{c}
\top \\ \hline \top \vee \bot \\ \hline \top \vee \bigvee_{i<n}(p_i \wedge \neg p_i)
\end{array} .
$$

Finally, we replace instances

$$
\frac{\xi\{p_j \wedge \neg p_j\} \vee \bigvee_{i<n}(p_i \wedge \neg p_i)}{\xi\{\bot\} \vee \bigvee_{i<n}(p_i \wedge \neg p_i)}
$$

of $i\uparrow$ by derivations of the form

$$
\begin{array}{r}
(*) \\ \vee a, \vee c, c\downarrow
\end{array}
\begin{array}{c}
\xi\{p_j \wedge \neg p_j\} \vee \bigvee_{i<n}(p_i \wedge \neg p_i) \\ \hline\hline \xi\{\bot\} \vee (p_j \wedge \neg p_j) \vee \bigvee_{i<n}(p_i \wedge \neg p_i) \\ \hline\hline \xi\{\bot\} \vee \bigvee_{i<n}(p_i \wedge \neg p_i)
\end{array} ,
$$

where $(*)$ follows by Lemma 3.1. □

Remark 3.3 As easy as it is, Theorem 3.2 has a profound impact on the proof complexity of the analytic systems *KSg* and *KSg* $+ c\uparrow$. The mapping

$$
\nu \colon \varphi(\vec{p}) \mapsto \varphi(\vec{p}) \vee \bigvee_i (p_i \wedge \neg p_i)
$$

is a simple poly-time function such that $\nu\varphi$ is equivalent to φ, and ν provides an interpretation of *SKSg* in *KSg*. Indeed, it is much simpler than the usual translations of propositional formulas to the language of resolution or algebraic proof systems. For most practical purposes, *KSg* thus has the same complexity as *SKSg* (i.e., as *LK*). Moreover, for many formulas φ we can actually eliminate the extra disjunct altogether (see Example 3.6). The ν interpretation preserves some important properties of proof systems too, see Corollary 3.5.

Theorem 3.2 also shows that "analytic" CoS proofs may contain formulas of arbitrary complexity, independent of the complexity of the formula being proved. Indeed, we can sneak any formula ψ in an *SKSg*-proof using a subproof of the form

$$
\begin{array}{r}
w\downarrow \\ i\uparrow
\end{array}
\begin{array}{c}
\xi\{\bot\} \\ \hline \xi\{\psi \wedge \neg\psi\} \\ \hline \xi\{\bot\}
\end{array} ,
$$

and ψ will stay in the KSg-proof constructed in Theorem 3.2, provided we are proving a formula of the form $\nu\varphi$, and all variables of ψ appear among $\vec{p}$. For a more natural example, see Example 3.6.

Definition 3.4 Let $\varphi_0(\vec{p},\vec{q}) \vee \varphi_1(\vec{p},\vec{r})$ be a classical tautology using the indicated variables, where $\vec{p}$, $\vec{q}$ and $\vec{r}$ are disjoint. Its *interpolant* is a Boolean circuit $C(\vec{p})$ such that

$$e\big(\varphi_{e(C)}\big) = 1$$

for any assignment e. A classical propositional proof system P has *feasible interpolation*, if every tautology $\varphi = \varphi_0 \vee \varphi_1$ as above has an interpolant of size polynomial in the size of the shortest P-proof of φ.

Feasible interpolation is a measure of the strength of proof systems. Weak proof systems, such as resolution or cut-free sequent calculus, admit feasible interpolation, whereas strong proof systems typically lack it (under reasonable assumptions). In particular, Bonet et al. [2] proved that LK does not have feasible interpolation if integer factoring is hard for $P/poly$.

Corollary 3.5 *If LK does not have feasible interpolation, then neither does KSg.*

Proof: Given an LK proof of $\varphi(\vec{p},\vec{q}) \vee \psi(\vec{p},\vec{r})$, we can construct a KSg-proof of

$$(*) \qquad \Big(\varphi(\vec{p},\vec{q}) \vee \bigvee_i (p_i \wedge \neg p_i) \vee \bigvee_i (q_i \wedge \neg q_i)\Big) \vee \Big(\psi(\vec{p},\vec{r}) \vee \bigvee_i (r_i \wedge \neg r_i)\Big)$$

by Theorems 2.7 and 3.2. The formula $(*)$ preserves the separation of variables in $\varphi \vee \psi$, and as the extra disjuncts are false, any circuit interpolating $(*)$ also interpolates $\varphi \vee \psi$. $\square$

Example 3.6 There are polynomial-time constructible $KSg + c{\uparrow}$-proofs of the functional pigeonhole principle

$$PHP_n^{n+1} = \bigvee_{i<n+1} \bigwedge_{j<n} \neg p_{i,j} \vee \bigvee_{\substack{i<n+1 \\ j<j'<n}} (p_{i,j} \wedge p_{i,j'}) \vee \bigvee_{\substack{j<n \\ i<i'<n+1}} (p_{i,j} \wedge p_{i',j}).$$

Proof: Buss [7] has constructed polynomial proofs of PHP in LK, hence also in $SKSg$ by Theorem 2.7. Using Theorem 3.2, we produce KSg-proofs of

$$PHP_n^{n+1} \vee \bigvee_{\substack{i<n+1 \\ j<n}} (p_{i,j} \wedge \neg p_{i,j}).$$

It thus suffices to construct $KSg + c{\uparrow}$-derivations of

$$\frac{p_{i,j} \wedge \neg p_{i,j}}{PHP_n^{n+1}}$$

for every i, j. We assume $i = j = 0$ to simplify the notation, and derive

sequent rule	w-r	w-l	c-r	c-l	$\neg$-r	$\neg$-l	x_i
CoS rule	$w{\downarrow}$	$w{\uparrow}$	$c{\downarrow}$	$c{\uparrow}$	$i{\downarrow}$	$i{\uparrow}$	x_i

Table 3: Correspondence of sequent and CoS rules

$$
\begin{array}{rc}
=, i{\downarrow} & \dfrac{p_{0,0} \wedge \neg p_{0,0}}{p_{0,0} \wedge \neg p_{0,0} \wedge \left(\bigwedge_{j\neq 0} \neg p_{0,j} \vee \bigvee_{j\neq 0} p_{0,j}\right)} \\
=, s & \dfrac{}{\bigwedge_j \neg p_{0,j} \vee \left(p_{0,0} \wedge \bigvee_{j\neq 0} p_{0,j}\right)} \\
c{\uparrow} & \dfrac{}{\bigwedge_j \neg p_{0,j} \vee \left(\bigwedge_{j\neq 0} p_{0,0} \wedge \bigvee_{j\neq 0} p_{0,j}\right)} \\
=, s & \dfrac{}{\bigwedge_j \neg p_{0,j} \vee \bigvee_{j\neq 0}(p_{0,0} \wedge p_{0,j})} \\
=, w{\downarrow} & \dfrac{}{PHP_n^{n+1}}\,.
\end{array}
$$

□

Remark 3.7 A similar argument shows that there are polynomial-time constructible $KSg + c{\uparrow}$-proofs of the onto pigeonhole principle

$$\bigvee_{i<n+1} \bigwedge_{j<n} \neg p_{i,j} \vee \bigvee_{j<n} \bigwedge_{i<n+1} \neg p_{i,j} \vee \bigvee_{\substack{j<n \\ i<i'<n+1}} (p_{i,j} \wedge p_{i',j}).$$

However, it does not seem to work for the multi-function pigeonhole principle

$$\bigvee_{i<n+1} \bigwedge_{j<n} \neg p_{i,j} \vee \bigvee_{\substack{j<n \\ i<i'<n+1}} (p_{i,j} \wedge p_{i',j}).$$

Notice that this situation matches the known upper bounds for the monotone sequent calculus *MLK*: Atserias et al. [1] have constructed polynomial tree-like *MLK*-proofs of the functional and onto pigeonhole principles (expressed as sequents of monotone formulas), but for the most general version of *PHP* only the quasipolynomial proof given by Theorem 2.8 is known.

4 The correspondence of CoS to the sequent calculus

In this section we present the simulation of CoS in tree-like sequent calculus and back, as described in [4, 13]. We include a detailed proof so that it is clear that the translation is polynomial-time, and to highlight the key information on which fragments of *SKSg* and *LK* correspond to each other.

Definition 4.1 If Γ is a sequence of formulas, we let $\bigwedge \Gamma$ be the conjunction of its elements bracketed to the right, where the empty conjunction is $\top$. Notice that $\bigwedge \Gamma$ is, provably in *Sg*, independent of the choice of bracketing or order, and $\bigwedge \Gamma_1 \wedge \bigwedge \Gamma_2$ is equivalent to $\bigwedge(\Gamma_1 \cup \Gamma_2)$. Big disjunctions $\bigvee \Delta$ are handled similarly.

Theorem 4.2 *Let R be a set of non-MLL sequent rules from Table 2, and let R' be the matching set of CoS rules according to Table 3. Given a tree-like $MLL + R$-proof of size s of a sequent $\Gamma \vdash \Delta$, we can construct in polynomial time an $Sg + R'$-derivation of* $\dfrac{\bigwedge \Gamma}{\bigvee \Delta}$ *with $O(s)$ lines, and size $O(s^2)$.*

Proof: By induction on the length of the derivation. The identity rule translates to the trivial derivation $\dfrac{\varphi}{\varphi}$. An instance of the cut rule

$$\frac{\Gamma \vdash \varphi, \Delta \qquad \Pi, \varphi \vdash \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda}$$

is simulated by the derivation

$$\begin{array}{r c}
 & \bigwedge \Gamma \wedge \bigwedge \Pi \\
\text{I.H.} & \overline{\overline{}} \\
 & (\varphi \vee \bigvee \Delta) \wedge \bigwedge \Pi \\
s, \wedge c & \overline{\overline{}} \\
 & (\bigwedge \Pi \wedge \varphi) \vee \bigvee \Delta \\
\text{I.H.} & \overline{\overline{}} \\
 & \bigvee \Lambda \vee \bigvee \Delta
\end{array} .$$

(Here and below we do not indicate instances of the $\wedge \cdots$ and $\vee \cdots$ rules needed to manipulate big conjunctions and disjunctions, as remarked in Definition 4.1.) The rules $\wedge$-l, $\vee$-r, $\bot$-l, $\top$-r, e-l, and e-r are obvious, and the rules $\top$-l and $\bot$-r are handled by an easy application of $\wedge u_1$ and $\vee u_2$, respectively. The steps for $\wedge$-r and $\vee$-l are as follows:

$$\begin{array}{r c}
 & \bigwedge \Gamma \wedge \bigwedge \Pi \\
\text{I.H.} & \overline{\overline{}} \\
 & (\varphi \vee \bigvee \Delta) \wedge \bigwedge \Pi \\
\text{I.H.} & \overline{\overline{}} \\
 & (\varphi \vee \bigvee \Delta) \wedge (\psi \vee \bigvee \Lambda) \\
s & \overline{} \\
 & ((\varphi \vee \bigvee \Delta) \wedge \psi) \vee \bigvee \Lambda \\
s, \wedge c & \overline{\overline{}} \\
 & (\varphi \wedge \psi) \vee \bigvee \Delta \vee \bigvee \Lambda
\end{array}
\qquad
\begin{array}{r c}
 & \bigwedge \Gamma \wedge \bigwedge \Pi \wedge (\varphi \vee \psi) \\
s, \vee c & \overline{\overline{}} \\
 & \bigwedge \Gamma \wedge (\varphi \vee (\bigwedge \Pi \wedge \psi)) \\
s & \overline{} \\
 & (\bigwedge \Gamma \wedge \varphi) \vee (\bigwedge \Pi \wedge \psi) \\
\text{I.H.} & \overline{\overline{}} \\
 & \bigvee \Delta \vee (\bigwedge \Pi \wedge \psi) \\
\text{I.H.} & \overline{\overline{}} \\
 & \bigvee \Delta \vee \bigvee \Lambda
\end{array} .$$

This completes the proof for *MLL*. The $\neg$-rules can be simulated by $i\uparrow$ and $i\downarrow$ using

$$\begin{array}{r c}
 & \bigwedge \Gamma \wedge \neg\varphi \\
\text{I.H.} & \overline{\overline{}} \\
 & (\varphi \vee \bigvee \Delta) \wedge \neg\varphi \\
s, \wedge c, \vee c & \overline{\overline{}} \\
 & \bigvee \Delta \vee (\neg\varphi \wedge \varphi) \\
i\uparrow, \vee u_1 & \overline{\overline{}} \\
 & \bigvee \Delta
\end{array}
\qquad
\begin{array}{r c}
 & \bigwedge \Gamma \\
i\downarrow, \wedge u_2 & \overline{\overline{}} \\
 & \bigwedge \Gamma \wedge (\varphi \vee \neg\varphi) \\
s & \overline{} \\
 & (\bigwedge \Gamma \wedge \varphi) \vee \neg\varphi \\
\text{I.H.} & \overline{\overline{}} \\
 & \bigvee \Delta \vee \neg\varphi
\end{array} .$$

The other rules from Table 3 are completely straightforward. □

Remark 4.3 The translation of the cut rule employed an instance of switch, but notice that it was only needed to shuffle the side-formulas around. It is not necessary in the important special case with $\Delta = \varnothing$; the cut rule then basically follows from mere transitivity of derivation, hence it cannot be prevented by throwing away any CoS rule or a set of rules.

We formulated the proof of Theorem 4.2 as an inductive argument, but it can also be easily visualized globally. The transformation basically consists of a left-to-right depth-first traversal of the sequent proof tree; when visiting a particular node, we can make some CoS inferences (depending on the rule) before we visit its ancestors (e.g., the s inferences in the simulation of $\vee$-l above), after we visit them (cf. $\wedge$-r above), or between visiting the first and the second ancestor of a binary rule (cf. the simulation of cut). The context describing the path from the current node to the root is carried along, the relevant CoS inferences are actually performed inside this context. (The depth of the CoS inference thus corresponds to the depth of the sequent proof tree.)

We illustrate it in Figure 1, which shows an LK-proof together with its $SKSg$ translation.

$$
\dfrac{\dfrac{}{p \vdash p}\,i \qquad \mathbf{1}\ \dfrac{\dfrac{}{q \vdash q}\,i}{\vdash \neg q, q}\,\neg\text{-r}}{p \vdash p \wedge \neg q, q}\,\wedge\text{-r}\ \mathbf{2}
\qquad \mathbf{3} \qquad
\dfrac{\dfrac{\dfrac{\mathbf{4}\ \dfrac{\neg\text{-l}\,\dfrac{\dfrac{}{p \vdash p}\,i}{p, \neg p \vdash}\,\mathbf{5} \qquad \neg\text{-l}\,\dfrac{\dfrac{}{q \vdash q}\,i}{\neg q, q \vdash}\,\mathbf{6}}{p, \neg q, \neg p \vee q \vdash}\,\vee\text{-l}}{\neg p \vee q, p, \neg q \vdash}\,e\text{-l}}{\neg p \vee q, p \wedge \neg q \vdash}\,\wedge\text{-l}}{}
$$

$$
\dfrac{p \vdash p \wedge \neg q, q \qquad \neg p \vee q, p \wedge \neg q \vdash}{p, \neg p \vee q \vdash q}\,cut
$$

$$
\begin{array}{rc l}
i{\downarrow} & \dfrac{p \wedge (\neg p \vee q)}{(p \wedge (\neg q \vee q)) \wedge (\neg p \vee q)} & \mathbf{1} \\
s & \dfrac{}{((p \wedge \neg q) \vee q) \wedge (\neg p \vee q)} & \mathbf{2} \\
s & \dfrac{}{(p \wedge \neg q \wedge (\neg p \vee q)) \vee q} & \mathbf{3} \\
2 \times s & \dfrac{}{((p \wedge \neg p) \vee (\neg q \wedge q)) \vee q} & \mathbf{4} \\
i{\uparrow} & \dfrac{}{(\neg q \wedge q) \vee q} & \mathbf{5} \\
i{\uparrow} & \dfrac{}{q} & \mathbf{6}
\end{array}
$$

Figure 1: An example of a sequent proof, and its CoS representation

We omit instances of the $=$ rule from the diagram; for the remaining inference steps we indicate by numbers the place in the traversal sequence of the sequent proof-tree where they come from. Notice that most points of the traversal sequence do not actually generate any CoS inference.

Lemma 4.4 *Given a context $\xi\{\ \}$, and formulas φ, ψ, we can construct in polynomial time a tree-like cut-free MLL-derivation of $\xi\{\varphi\} \vdash \xi\{\psi\}$ from $\varphi \vdash \psi$ with $O(|\xi|)$ lines, and size $O(|\xi| \cdot (|\xi| + |\varphi| + |\psi|))$.*

Proof: By induction on the complexity of ξ. The base case is trivial, and the induction steps for $\wedge$ and $\vee$ follow by

$$
\wedge\text{-l}\,\dfrac{\wedge\text{-r}\,\dfrac{\text{I.H.}\,\dfrac{\varphi \vdash \psi}{\xi\{\varphi\} \vdash \xi\{\psi\}} \qquad \dfrac{}{\chi \vdash \chi}\,i}{\xi\{\varphi\}, \chi \vdash \xi\{\psi\} \wedge \chi}}{\xi\{\varphi\} \wedge \chi \vdash \xi\{\psi\} \wedge \chi}
\qquad
\vee\text{-r}\,\dfrac{\vee\text{-l}\,\dfrac{\text{I.H.}\,\dfrac{\varphi \vdash \psi}{\xi\{\varphi\} \vdash \xi\{\psi\}} \qquad \dfrac{}{\chi \vdash \chi}\,i}{\xi\{\varphi\} \vee \chi \vdash \xi\{\psi\}, \chi}}{\xi\{\varphi\} \vee \chi \vdash \xi\{\psi\} \vee \chi}\,.
$$

$\square$

Lemma 4.5 *Given a multiset* Γ *of* k *formulas of total size* s*, we can construct in polynomial time a cut-free tree-like MLL-derivation of the sequents*

$$\Gamma \vdash \bigwedge \Gamma \qquad\qquad \bigvee \Gamma \vdash \Gamma$$

with $O(k)$ *lines, and size* $O(ks)$*.*

Proof: Exercise. □

Theorem 4.6 *Let* R *be a set of non-MLL sequent rules from Table 2, and let* R' *be the matching set of CoS rules according to Table 3. Given an* $Sg + R'$*-derivation of*

$$\frac{\bigwedge \Gamma}{\bigvee \Delta}$$

of size s*, we can construct in polynomial time a tree-like* $MLL + R$*-proof of the sequent* $\Gamma \vdash \Delta$ *with* $O(s)$ *lines, and size* $O(s^2)$*.*

Proof: By Lemma 4.5, we may assume that $\Gamma = \{\varphi\}$ and $\Delta = \{\psi\}$ are singletons. Given a CoS derivation $\varphi = \varphi_0, \ldots, \varphi_k = \psi$, we can construct proofs of the sequents $\varphi_i \vdash \varphi_{i+1}$, and derive $\varphi \vdash \psi$ by k cuts. It thus suffices to handle a single CoS inference step. By Lemma 4.4, we may assume that the inference is shallow. The rest is just a matter of perseverance; we indicate derivation of some of the rules below, and leave the rest to the reader.

$$\wedge\text{-l}\,\dfrac{\top\text{-l}\,\dfrac{i\,\dfrac{}{\varphi \vdash \varphi}}{\varphi, \top \vdash \varphi}}{\varphi \wedge \top \vdash \varphi} \qquad \wedge\text{-r}\,\dfrac{i\,\dfrac{}{\varphi \vdash \varphi} \qquad \top\text{-r}\,\dfrac{}{\vdash \top}}{\varphi \vdash \varphi \wedge \top} \qquad \wedge\text{-l}\,\dfrac{e\text{-l}\,\dfrac{\wedge\text{-r}\,\dfrac{i\,\dfrac{}{\psi \vdash \psi} \qquad i\,\dfrac{}{\varphi \vdash \varphi}}{\psi, \varphi \vdash \psi \wedge \varphi}}{\varphi, \psi \vdash \psi \wedge \varphi}}{\varphi \wedge \psi \vdash \psi \wedge \varphi}$$

$$\wedge\text{-l}\,\dfrac{\wedge\text{-r}\,\dfrac{\wedge\text{-r}\,\dfrac{i\,\dfrac{}{\varphi \vdash \varphi} \qquad i\,\dfrac{}{\psi \vdash \psi}}{\varphi, \psi \vdash \varphi \wedge \psi} \qquad i\,\dfrac{}{\chi \vdash \chi}}{\varphi, \psi, \chi \vdash (\varphi \wedge \psi) \wedge \chi}}{\varphi \wedge (\psi \wedge \chi) \vdash (\varphi \wedge \psi) \wedge \chi} \qquad \wedge\text{-l}, \vee\text{-r}\,\dfrac{\vee\text{-l}\,\dfrac{\wedge\text{-r}\,\dfrac{i\,\dfrac{}{\varphi \vdash \varphi} \qquad i\,\dfrac{}{\psi \vdash \psi}}{\varphi, \psi \vdash \varphi \wedge \psi} \qquad i\,\dfrac{}{\chi \vdash \chi}}{\varphi, \psi \vee \chi \vdash \varphi \wedge \psi, \chi}}{\varphi \wedge (\psi \vee \chi) \vdash (\varphi \wedge \psi) \vee \chi}$$

$$\top\text{-l}, \vee\text{-r}\,\dfrac{\neg\text{-r}\,\dfrac{i\,\dfrac{}{\varphi \vdash \varphi}}{\vdash \varphi, \neg\varphi}}{\top \vdash \varphi \vee \neg\varphi} \qquad \wedge\text{-l}, \bot\text{-r}\,\dfrac{\neg\text{-l}\,\dfrac{i\,\dfrac{}{\varphi \vdash \varphi}}{\varphi, \neg\varphi \vdash}}{\varphi \wedge \neg\varphi \vdash \bot} \qquad w\text{-r}\,\dfrac{\bot\text{-l}\,\dfrac{}{\bot \vdash}}{\bot \vdash \varphi}$$

□

5 The relationship to the monotone sequent calculus

The goal of this section is to show the equivalence of the analytic CoS system $KSg + c{\uparrow}$ to the monotone sequent calculus *MLK*, building on the correspondence given by Theorems 4.2 and 4.6. A minor problem arises because of the *MLK* w-l rule, which translates to the

$w{\uparrow}$ rule, absent in $KSg + c{\uparrow}$. (The $w{\uparrow}$ CoS rule is not considered analytic.) We resolve it by showing that $w{\uparrow}$ can be easily eliminated from $KSg + c{\uparrow} + w{\uparrow}$-proofs of formulas. Although the elimination cannot work literally for general derivations with assumptions other than $\top$ (indeed, an instance of $w{\uparrow}$ is itself an example of a derivation which cannot be simulated in $KSg+c{\uparrow}$, polynomially or otherwise), something to similar effect can be achieved as well. The upshot is that $KSg + c{\uparrow}$ and $KSg + c{\uparrow} + w{\uparrow}$ can be treated as essentially identical systems.

The simulation naturally works for atomic variants of the structural rules, we obtain the general version as a corollary using Theorem 2.4.

Theorem 5.1 *The calculus $KS + ac{\uparrow}$ polynomially simulates $KS + ac{\uparrow} + aw{\uparrow}$, and KS polynomially simulates $KS + aw{\uparrow}$.*

More generally, given a $KS \pm ac{\uparrow} + aw{\uparrow}$ derivation of $\dfrac{\varphi}{\psi}$ *, we can construct in polynomial time a $KS \pm ac{\uparrow}$ derivation of* $\dfrac{\varphi'}{\psi}$ *, where φ' differs from φ by substitution of $\top$ for some occurrences of literals.*

Proof: Let π be a derivation of $\dfrac{\varphi}{\psi}$ in $KS + aw{\uparrow} \pm ac{\uparrow}$. Let $\dfrac{\xi\{a\}}{\xi\{\top\}}$ be the topmost instance of $aw{\uparrow}$ occurring in π. We mark some occurrences of the literal a in π as follows:

- The indicated occurrence of a in the line $\xi\{a\}$ above is marked.
- If $\dfrac{\zeta\{\omega\}}{\zeta\{\chi\}}$ is an inference in π, then any occurrence of a in ζ which is marked in the conclusion of the rule is also marked in the premise.
- If $\dfrac{\zeta\{\omega\}}{\zeta\{\chi\}}$ is an instance of s, m, or $=$ in π, then for any marked occurrence of a in χ, the corresponding occurrence of a in ω (defined in an obvious way) is also marked.
- If $\dfrac{\zeta\{a \vee a\}}{\zeta\{a\}}$ is an instance of $ac{\downarrow}$ in π, and the indicated occurrence of a in the conclusion is marked, then both occurrences of a in the premise are also marked.
- If $\dfrac{\zeta\{a\}}{\zeta\{a \wedge a\}}$ is an instance of $ac{\uparrow}$ in π, and both indicated occurrences of a in the conclusion are marked, then the occurrence in the premise is also marked.

We replace all marked occurrences of a by $\top$, and move on to the next instance of $aw{\uparrow}$, repeating the process until we reach the end of the proof.

The transformation does not change the size of the proof, and it turns all instances of $aw{\uparrow}$ into trivial inferences $\dfrac{\xi\{\top\}}{\xi\{\top\}}$, which may be simplified by removing one of the lines. The definition of marking ensures that inference steps in π remain valid instances of the same rule, except for the following cases:

- If only one occurrence of a is marked in the conclusion of an inference $\dfrac{\zeta\{a\}}{\zeta\{a \wedge a\}}$, it is transformed into $\dfrac{\zeta\{a\}}{\zeta\{\top \wedge a\}}$, which is an instance of $\wedge u_1$.

- An instance $\dfrac{\zeta\{\top\}}{\zeta\{a \vee \neg a\}}$ of $i\downarrow$ may turn into $\dfrac{\zeta\{\top\}}{\zeta\{\top \vee \neg a\}}$ or $\dfrac{\zeta\{\top\}}{\zeta\{\top \vee \top\}}$. The latter is an instance of x_1, and the former can be fixed by a subderivation

$$\vee u_2 \frac{\zeta\{\top\}}{aw\downarrow \dfrac{\zeta\{\top \vee \bot\}}{\zeta\{\top \vee \neg a\}}} .$$

- An instance $\dfrac{\zeta\{\bot\}}{\zeta\{a\}}$ of $aw\downarrow$ may turn into $\dfrac{\zeta\{\bot\}}{\zeta\{\top\}}$. This is an instance of $w\downarrow$; if we insist on using only $aw\downarrow$, it can be eliminated as in Theorem 2.4.

$\square$

Corollary 5.2 *KSg* $+ c\uparrow$ *polynomially simulates KSg* $+ c\uparrow + w\uparrow$*, and KSg polynomially simulates KSg* $+ w\uparrow$*.*

Given a KSg$\pm c\uparrow + w\uparrow$*-derivation of* $\dfrac{\varphi}{\psi}$ *, we can construct in polynomial time a KSg*$\pm c\uparrow$ *derivation of* $\dfrac{\varphi'}{\psi}$ *, where* φ' *is as in Theorem 5.1.*

Proof: Use Theorems 2.4 and 5.1. $\square$

We proceed with the basic equivalence of *KSg*$+c\uparrow+w\uparrow$ and *MLK* wrt monotone sequents.

Theorem 5.3 *Let* Γ *and* Δ *be multisets of monotone formulas.*

(*i*) *Given a tree-like MLK-proof of* $\Gamma \vdash \Delta$*, we can construct in polynomial time a KSg* $+ c\uparrow + w\uparrow$*-derivation of* $\dfrac{\bigwedge \Gamma}{\bigvee \Delta}$ *.*

(*ii*) *Given a KSg* $+ c\uparrow + w\uparrow$*-derivation of* $\dfrac{\bigwedge \Gamma}{\bigvee \Delta}$ *, we can construct in polynomial time a tree-like MLK-proof of* $\Gamma \vdash \Delta$*.*

Proof: (*i*) is a special case of Theorem 4.2.

(*ii*): By Theorem 2.4, we may assume that all instances if $i\downarrow$ in the proof are atomic. We replace all negative literals occurring in the proof with $\top$. This preserves the validity of all inference steps in the proof, except for instances of $ai\downarrow$, which turn into $\dfrac{\xi\{\top\}}{\xi\{p \vee \top\}}$. We can simulate the latter by $w\downarrow$, hence we obtain a derivation in *Sg* $+ \{w\downarrow, w\uparrow, c\downarrow, c\uparrow\}$. We can transform it in an *MLK*-proof by Theorem 4.6. $\square$

What remains to show is that arbitrary formulas can be satisfactorily approximated by monotone sequents, so that we can make Theorem 5.3 into a simulation of general *KSg* $+ c\uparrow$-proofs.

Definition 5.4 If $\varphi(p_0, \ldots, p_{n-1})$ is a formula using only the indicated variables, let $\varphi^{\mathbf{m}}(\vec{p}, \vec{q})$ be the monotone formula such that $\varphi = \varphi^{\mathbf{m}}(\vec{p}, \neg\vec{p})$, where $q_0, \ldots, q_{n-1}$ is a sequence of fresh variables.

Theorem 5.5 *Let φ and ψ be formulas in the variables p_i, $i < n$.*

(i) LK-proofs of ψ and $\{p_i \vee q_i;\ i < n\} \vdash \psi^{\mathbf{m}}(\vec{p}, \vec{q})$ are constructible from each other in polynomial time.

(ii) $KSg + c\uparrow + w\uparrow$-derivations of $\dfrac{\varphi}{\psi}$ and

$$\frac{\bigwedge_{i<n}(p_i \vee q_i) \wedge \varphi^{\mathbf{m}}(\vec{p}, \vec{q})}{\psi^{\mathbf{m}}(\vec{p}, \vec{q})}$$

are constructible from each other in polynomial time.

Proof: (i): Given a proof of

$$\{p_i \vee q_i;\ i < n\} \vdash \psi^{\mathbf{m}}(\vec{p}, \vec{q}),$$

we substitute $\neg p_i$ for q_i in the whole proof, and cut away the unwanted formulas from the antecedent using separate subproofs of $\vdash p_i \vee \neg p_i$.

On the other hand, we can construct a polynomial-size derivation of

$$\varphi^{\mathbf{m}}(\vec{p}, \neg\vec{p}) \vdash \varphi^{\mathbf{m}}(\vec{p}, \vec{q})$$

from assumptions

$$\neg p_i \vdash q_i$$

as in Lemma 4.4. We weaken all sequents in the derivation by $\{p_i \vee q_i;\ i < n\}$, and get rid of the extra assumptions using subproofs of

$$p_i \vee q_i, \neg p_i \vdash q_i.$$

We obtain a polynomial-size proof of

$$\{p_i \vee q_i;\ i < n\}, \varphi \vdash \varphi^{\mathbf{m}}(\vec{p}, \vec{q}),$$

we can thus derive

$$\{p_i \vee q_i;\ i < n\} \vdash \varphi^{\mathbf{m}}(\vec{p}, \vec{q})$$

from $\vdash \varphi$ by a cut.

(ii): Given a derivation of

$$\frac{\bigwedge_{i<n}(p_i \vee q_i) \wedge \varphi^{\mathbf{m}}(\vec{p}, \vec{q})}{\psi^{\mathbf{m}}(\vec{p}, \vec{q})},$$

we substitute $\neg p_i$ for q_i in the whole proof, and prefix it with a polynomial-size subproof

$$i\downarrow, = \frac{\varphi}{\bigwedge_{i<n}(p_i \vee \neg p_i) \wedge \varphi}.$$

Assume we are given a derivation of $\dfrac{\varphi}{\psi}$. We may assume that no variables except $\vec{p}$ appear in the proof, and that all instances of $i\downarrow$ are atomic. We put all formulas in the proof into the context $\bigwedge_{i<n}(p_i \vee q_i) \wedge \{\ \}$, and replace $\neg p_i$ with q_i. This transformation preserves the inference steps, except for instances of $ai\downarrow$, which we simulate by polynomial-size derivations

$$\frac{\bigwedge_{i<n}(p_i \vee q_i) \wedge \xi\{\top\}}{\bigwedge_{i<n}(p_i \vee q_i) \wedge \xi\{p_i \vee q_i\}}$$

constructed dually to Lemma 3.1. We get rid of the extra $\bigwedge_{i<n}(p_i \vee q_i) \wedge \cdots$ at the end of the proof by $w\uparrow$. □

Corollary 5.6 *$KSg + c\uparrow$-proofs of $\psi(\vec{p})$, and tree-like MLK-proofs of*

$$\{p_i \vee q_i;\ i < n\} \vdash \psi^{\mathbf{m}}(\vec{p}, \vec{q}),$$

are constructible from each other in polynomial time.

Proof: Use Corollary 5.2, and Theorems 5.3 and 5.5. □

Corollary 5.7 *$KSg + c\uparrow$ quasipolynomially simulates SKSg. If a formula φ in n variables has an SKSg-proof of size s, it has a $KSg + c\uparrow$-proof of size $s^{O(1)} n^{O(\log n)}$.*

Proof: By Theorems 2.7, 5.5 (i), 2.8, and Corollary 5.6. □

Corollary 5.8 *The following are equivalent.*

(i) *$KSg + c\uparrow$ polynomially simulates SKSg.*

(ii) *Tree-like MLK polynomially simulates LK with respect to monotone sequents whose antecedent is of the form $\{p_i \vee q_i;\ i < n\}$.*

Proof: (ii) $\to$ (i) follows immediately from Theorem 5.5 and Corollary 5.6.

(i) $\to$ (ii): Given an LK-proof of $\{p_i \vee q_i;\ i < n\} \vdash \Delta(\vec{p}, \vec{q}, \vec{r})$, we can construct an LK-proof of the formula $\varphi = (\bigvee \Delta)(\vec{p}, \neg\vec{p}, \vec{r})$. We can make it tree-like, and by Theorem 4.2 we can construct an *SKSg*-proof of φ. By assumption, we can construct a $KSg + c\uparrow$-proof of φ as well, which we can transform in a tree-like MLK-proof of

$$\{p_i \vee q_i;\ i < n\}, \{r_j \vee s_j;\ j < m\} \vdash \bigvee \Delta$$

by Corollary 5.6. As Δ does not involve $\vec{s}$, we may eliminate the extra assumptions by substitution of $\top$ for s_j, and we finish the proof using Lemma 4.5. □

Remark 5.9 We do not know what is the complexity of *KSg* without the $c\uparrow$ rule. On the one hand, Theorem 3.2 suggests that it is close to *SKSg*. On the other hand, attempts to eliminate $c\uparrow$ in an obvious way in the spirit of Theorem 5.1 seem to incur an exponential blow-up of the proof. We thus cannot exclude the possibility that the $c\uparrow$ rule provides a significant speed-up over *KSg*.

References

[1] Albert Atserias, Nicola Galesi, and Pavel Pudlák, *Monotone simulations of non-monotone proofs*, Journal of Computer and System Sciences 65 (2002), no. 4, pp. 626–638.

[2] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz, *No feasible interpolation for TC^0-Frege proofs*, in: Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science, 1997, pp. 254–263.

[3] Kai Brünnler, *Deep inference and symmetry in classical proofs*, Logos Verlag, Berlin, 2004.

[4] ———, *Deep inference and its normal form of derivations*, in: Logical Approaches to Computational Barriers, 2nd Conference on Computability in Europe, CiE 2006, Proceedings (A. Beckmann, U. Berger, B. Löwe, and J. V. Tucker, eds.), Lecture Notes on Computer Science vol. 3988, Springer, 2006, pp. 65–74.

[5] Paola Bruscoli and Alessio Guglielmi, *On the proof complexity of deep inference*, ACM Transactions on Computational Logic (2008), accepted.

[6] ———, *On analytic inference rules in the calculus of structures*, note, `http://cs.bath.ac.uk/ag/p/Onan.pdf`, 2007.

[7] Samuel R. Buss, *Polynomial size proofs of the propositional pigeonhole principle*, Journal of Symbolic Logic 52 (1987), pp. 916–927.

[8] Stephen A. Cook and Robert A. Reckhow, *The relative efficiency of propositional proof systems*, Journal of Symbolic Logic 44 (1979), no. 1, pp. 36–50.

[9] Alessio Guglielmi, *A system of interaction and structure*, ACM Transactions on Computational Logic 8 (2007), no. 1, pp. 1–64.

[10] Alessio Guglielmi and Lutz Straßburger, *Non-commutativity and MELL in the calculus of structures*, in: Computer Science Logic: 15th International Workshop, CSL 2001, Proceedings (L. Fribourg, ed.), Lecture Notes in Computer Science vol. 2142, Springer, 2001, pp. 54–68.

[11] Jan Krajíček, *Lower bounds to the size of constant-depth propositional proofs*, Journal of Symbolic Logic 59 (1994), no. 1, pp. 73–86.

[12] ———, *Bounded arithmetic, propositional logic, and complexity theory*, Encyclopedia of Mathematics and Its Applications vol. 60, Cambridge University Press, 1995.

[13] Richard McKinley, *Classical categories and deep inference*, in: Structures and Deduction – the Quest for the Essence of Proofs (satellite workshop of ICALP 2005) (P. Bruscoli, F. Lamarche, and C. Stewart, eds.), Technische Universität Dresden, 2005, pp. 19–33.