Proof Compression via Subatomic Logic and Guarded Substitutions

Victoria Barrett[1], Alessio Guglielmi[2], Benjamin Ralph[3], and Lutz Straßburger[4]

[1] Inria Saclay
victoria.barrett@inria.fr
[2] University of Bath
[3] University of Bath
[4] Inria Saclay

The affinity between structural proof theory and the mathematical foundations of computation establishes mechanisms of proof compression as a natural object of study. The most prominent proof compression mechanism is the *cut rule* [11, 12], which allows lemmas to be reused in a proof. And indeed, eliminating the cuts from a proof can lead to a non-elementary blow-up [6]. In propositional logic, this blow-up is still exponential [19].

In the area of proof complexity, a distinct subfield of proof theory, other mechanisms of proof compression have been studied, the most notable ones being substitution [9] and Tseitin extension [20]. In both cases, proof compression is achieved by permitting propositional variables to replace arbitrary subformulae in a proof. The cost of eliminating either rule from a proof is an exponential blow-up (and it is not known whether this can be done more efficiently).

Given their conceptual similarity, it is perhaps not so surprising that, in the presence of cut, extension and substitution are *p-equivalent*, i.e., a system with cut and substitution can polynomially simulate one with cut and Tseitin extension [9], and vice versa [16]. This has been shown in the setting of Frege systems, which always contain the cut because of the presence of the *modus ponens* rule. Moreover, even in the absence of cut, it has also been demonstrated that the two proof compression mechanisms of substitution and Tseitin extension are p-equivalent [17,18]. However, it is not known if cut-free systems with extension or substitution can p-simulate systems with cut and extension or substitution.

This leaves us with two powerful proof compression mechanisms: (i) the cut and (ii) extension/substitution. It is an open question whether one of them subsumes the other, or whether they are truly independent.

In this work we give a surprising answer to this question. We observe that both proof compression mechanisms are subsumed by a more general one, namely, *guarded substitution*, which is a variant of explicit substitutions [1].

To see how this is possible, let us first observe that the proof compression of cut and substitution comes from the ability of reusing information. And the most basic inference rules that deal with duplication of information are the rules of *contraction* and *cocontraction* shown below:[1]

$$\mathsf{c}{\downarrow}\,\frac{A \vee A}{A} \qquad \text{and} \qquad \mathsf{c}{\uparrow}\,\frac{A}{A \wedge A} \tag{1}$$

A move to a deep inference proof system [7, 14], which allow for finer granularity in the design of the inference rules, enables the restriction of these rules to their atomic form, shown

[1] In fact, the combination of contraction and cocontraction form another mechanism of proof compression, when cut is absent. This has been investigated in [8,10,15]. However, we will not go into further details of this, as we develop in this work a cut-free system that can p-simulate cut.

$$\mathsf{cut}\,\frac{\vdash \Gamma, A \qquad \vdash \bar{A}, \Delta}{\vdash \Gamma, \Delta} \quad\leadsto\quad \mathsf{i}{\uparrow}\,\frac{A \wedge \bar{A}}{\mathsf{0}} \quad\leadsto\quad \mathsf{ai}{\uparrow}\,\frac{a \wedge \bar{a}}{\mathsf{0}} \quad\leadsto\quad \frac{(\mathsf{0}\;\boldsymbol{a}\;\mathsf{1}) \wedge (\mathsf{1}\;\boldsymbol{a}\;\mathsf{0})}{(\mathsf{0} \wedge \mathsf{1})\;\boldsymbol{a}\;(\mathsf{1} \wedge \mathsf{0})} \quad\leadsto\quad \hat{\boldsymbol{a}}{\wedge}\,\frac{(A\;\boldsymbol{a}\;B) \wedge (C\;\boldsymbol{a}\;D)}{(A \wedge C)\;\boldsymbol{a}\;(B \wedge D)}$$

Figure 1: Evolution of cut from the sequent calculus via deep inference to subatomic proof theory

below, provided there is an additional purely linear inference rule in the system [7]. This rule, called *medial*, is shown in the middle below:

$$\mathsf{ac}{\downarrow}\,\frac{a \vee a}{a} \qquad \mathsf{m}\,\frac{(A \wedge B) \vee (C \wedge D)}{(A \vee C) \wedge (B \vee D)} \qquad \mathsf{ac}{\uparrow}\,\frac{a}{a \wedge a} \tag{2}$$

This leads to the proof system SKS [7], consisting only of atomic rules (like $\mathsf{ac}{\downarrow}$ and $\mathsf{ac}{\uparrow}$ above) that change the size of a formula, and purely linear rules (like m above) that only rearrange subformulae without changing the size.

The next insight comes from the concept of *subatomic proof theory* [2,3,5] which splits the atoms into binary connectives. A formula "$A\;\boldsymbol{a}\;B$" is then interpreted as "if a is false then A, and if a is true then B". In this setting, we can write a as $\mathsf{0}\;\boldsymbol{a}\;\mathsf{1}$ and its dual $\bar{a}$ as $\mathsf{1}\;\boldsymbol{a}\;\mathsf{0}$. The two rules of $\mathsf{ac}{\downarrow}$ and $\mathsf{ac}{\uparrow}$ from above now become:

$$\frac{(\mathsf{0}\;\boldsymbol{a}\;\mathsf{1}) \vee (\mathsf{0}\;\boldsymbol{a}\;\mathsf{1})}{(\mathsf{0} \vee \mathsf{0})\;\boldsymbol{a}\;(\mathsf{1} \vee \mathsf{1})} \qquad \text{and} \qquad \frac{(\mathsf{0} \wedge \mathsf{0})\;\boldsymbol{a}\;(\mathsf{1} \wedge \mathsf{1})}{(\mathsf{0}\;\boldsymbol{a}\;\mathsf{1}) \wedge (\mathsf{0}\;\boldsymbol{a}\;\mathsf{1})} \tag{3}$$

which are just instances of the general rules

$$\check{\vee}\boldsymbol{a}\,\frac{(A\;\boldsymbol{a}\;B) \vee (C\;\boldsymbol{a}\;D)}{(A \vee C)\;\boldsymbol{a}\;(B \vee D)} \qquad \text{and} \qquad \hat{\wedge}\boldsymbol{a}\,\frac{(A \wedge B)\;\boldsymbol{a}\;(C \wedge D)}{(A\;\boldsymbol{a}\;C) \wedge (B\;\boldsymbol{a}\;D)} \tag{4}$$

which have the same shape as the medial rule in (2) above. The same principle also applies to the cut rule: in Figure 1 we see the evolution of the cut, starting from the sequent calculus, first becoming an atomic rule, and finally a subatomic rule. In this way we can obtain a proof system for propositional logic in which *all* inference rules are linear rewriting steps [2], except for the rules dealing with the units, for example

$$\frac{A}{A \wedge \mathsf{1}} \qquad \frac{A}{A \vee \mathsf{0}} \qquad \frac{A \wedge \mathsf{1}}{A} \qquad \frac{A \vee \mathsf{0}}{A} \tag{5}$$

Even though these are rather trivial inference steps in a standard proof system, they are the only ones that break a rigidly defined notion of linearity in a subatomic proof system. To achieve what is called a strictly linear system, these can be eliminated, but the naive way of doing so leads to an exponential blow-up of the size of the proof [4, 5]. However, by allowing explicit substitutions as constructors in formulae and derivations, the size of the proof expansion can be reduced to a polynomial [5]. The resulting system (called KDTS in [5]) is p-equivalent to SKS (and therefore also to standard Frege systems without extension or substitution) and still contains the cut (in its linear form, as shown on the right in Figure 1). And, unsurprisingly, eliminating the cut from this system leads to an exponential blow-up [5]. Furthermore, it is unknown whether these explicit substitutions can in any way polynomially simulate Tseitin extension or substitution in Frege systems.

In other words, in terms of proof complexity, nothing has been gained so far with respect to what we said at the beginning of this introduction; even in the unfamiliar climes of a subatomic proof system with explicit substitutions, it appears that both cut and extension/substitution operate as independent means of compressing proofs.

This motivates our paper, in which we introduce *guarded substitutions* that offer us a distinct new means of proof compression. Guarded substitutions are a variant of explicit substitution that, instead of representing the replacement of every occurrence of a free variable, only select a certain subset of the free occurrences of the given variable. To make this formal, we assign to every variable occurrence a *range*, and to every guarded substitution a *guard*, and the substitution can apply to the variable occurrence, if the guard is in the range. For example, the formula $\langle A|x\rangle((x \wedge x) \vee (y \wedge x))$ with an ordinary explicit substitution becomes $(A \wedge A) \vee (y \wedge A)$ when the substitution is carried out, whereas the formula $\langle\!\langle A|x\rangle\!\rangle^p((x^{q,p} \wedge x^r) \vee (y^p \wedge x^{p,r}))$ becomes $(A \wedge x^r) \vee (y^p \wedge A)$ when the substitution is carried out, because only the first and the last x have the guard p in its range.

With this additional construct, the new system, that we call KSubG, can polynomially simulate Frege systems with substitution. Furthermore, we can do this with the cut-free fragment KSubG^-. In other words, guarded substitution can polynomially simulate the cut, as well as Tseitin extension and substitution in Frege systems. And surprisingly, this can even be done when we only allow the units $\mathsf{0}$ and $\mathsf{1}$ in the place of the formula to be substituted (the A in the example above).

Since Frege systems with substitution are known to be the most potent propositional proof systems, in the sense that they can polynomially simulate every other proof system for classical propositional logic, our system KSubG has now the same property, with the additional feature that every inference step is linear. There is never an inference that adds or deletes information.

This becomes possible because a subatomic derivation can be interpreted as a *superposition* of standard derivations. Even though this idea has been around since the beginning of subatomic proof theory [13], only our guarded substitutions allow to make use of this. We can see such a superposition as executing several similar shaped derivations in parallel, and the guarded substitutions can be used to read out the correct results.

For example, the derivation

$$\Psi = (x \vee x) \wedge \mathsf{mix}\,\frac{y \wedge \mathsf{0}}{y \vee \mathsf{0}}$$

is a superposition of the derivation

$$\Phi = \mathsf{ai}{\uparrow}\,\frac{\boxed{\mathsf{ac}{\downarrow}\,\frac{a \vee a}{a}} \wedge \boxed{\mathsf{aw}{\downarrow}\,\frac{\mathsf{0}}{\bar{a}}}}{\mathsf{0}}$$

because by substituting $\mathsf{0}$ for x and $\mathsf{1}$ for y in Ψ we can recover the result of assuming that a is false in Φ, and by substituting $\mathsf{1}$ for x and $\mathsf{0}$ for y we can recover the result of assuming that a is true. Then Φ can be encoded by the subatomic derivation with guarded substitutions

$$\langle\!\langle \mathsf{0}|x, \mathsf{1}|y\rangle\!\rangle^l \,\langle\!\langle \mathsf{1}|x, \mathsf{0}|y\rangle\!\rangle^r \,\langle \Psi|z\rangle(z^l \,\boldsymbol{a}\, z^r)$$

We can recover from this the conclusion of Φ, and the resulting derivation contains no cuts.

This allows for the factorisation of lemmas with different inputs, essentially performing the work of both modus ponens and the Frege substitution rule. In this way, we are able to p-simulate substitution Frege in a cut-free subatomic system with guarded substitutions.

References

[1] Martín Abadi, Luca Cardelli, Pierre-Louis Curien, and Jean-Jacques Lévy. Explicit substitutions. *J. of Functional Programming*, 1(4):375–416, 1991.

[2] Andrea Aler Tubella. *A Study of Normalisation Through Subatomic Logic*. PhD thesis, University of Bath, 2017.

[3] Chris Barrett and Alessio Guglielmi. A subatomic proof system for decision trees. *ACM Transactions on Computational Logic*, 23(4):26:1–25, 2022.

[4] Victoria Barrett. *A Strictly Linear Proof System for Propositional Classical Logic*. PhD thesis, University of Bath, 2024.

[5] Victoria Barrett, Alessio Guglielmi, and Benjamin Ralph. A strictly linear subatomic proof system. In *CSL 2025 - 33rd EACSL Annual Conference on Computer Science Logic*, Amsterdam, Netherlands, February 2025.

[6] George Boolos. Don't eliminate cut. *Journal of Philosophical Logic*, 13:373–378, 1984.

[7] Kai Brünnler and Alwen Fernanto Tiu. A local system for classical logic. In R. Nieuwenhuis and A. Voronkov, editors, *LPAR 2001*, volume 2250 of *LNAI*, pages 347–361. Springer, 2001.

[8] Paola Bruscoli, Alessio Guglielmi, Tom Gundersen, and Michel Parigot. A quasipolynomial normalisation in deep inference via atomic flows and threshold formulae. *Logical Methods in Computer Science*, 12((1:5)):1–30, 2016.

[9] Stephen A Cook and Robert A Reckhow. The relative efficiency of propositional proof systems. *The journal of symbolic logic*, 44(1):36–50, 1979.

[10] Anupam Das. On the pigeonhole and related principles in deep inference and monotone systems. In Thomas Henzinger and Dale Miller, editors, *Joint Meeting of the 23rd EACSL Annual Conference on Computer Science Logic (CSL) and the 29th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 36:1–10. ACM, 2014.

[11] Gerhard Gentzen. Untersuchungen über das logische Schließen. I. *Mathematische Zeitschrift*, 39:176–210, 1935.

[12] Gerhard Gentzen. Untersuchungen über das logische Schließen. II. *Mathematische Zeitschrift*, 39:405–431, 1935.

[13] Alessio Guglielmi. Subatomic logic. note, November 2002.

[14] Alessio Guglielmi and Lutz Straßburger. Non-commutativity and MELL in the calculus of structures. In Laurent Fribourg, editor, *Computer Science Logic, CSL 2001*, volume 2142 of *LNCS*, pages 54–68. Springer-Verlag, 2001.

[15] Emil Jeřábek. Proof complexity of the cut-free calculus of structures. *Journal of Logic and Computation*, 19(2):323–339, 2009.

[16] Jan Krajíček and Pavel Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *The Journal of Symbolic Logic*, 54(3):1063–1079, 1989.

[17] Novak Novakovic and Lutz Straßburger. On the power of substitution in the calculus of structures. *ACM Trans. Comput. Log.*, 16(3):19, 2015.

[18] Lutz Straßburger. Extension without cut. *Annals of Pure and Applied Logic*, 163(12):1995–2007, 2012.

[19] Anne Sjerp Troelstra and Helmut Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, second edition, 2000.

[20] G. S. Tseitin. On the complexity of derivation in propositional calculus. *Zapiski Nauchnykh Seminarou LOMI*, 8:234–259, 1968.