A Subatomic Proof System for Decision Trees

C. Barrett, A. Guglielmi
University of Bath, Dept. of Computer Science

Subatomic proof systems [4] represent atoms as non-commutative self-dual connectives. As a consequence of this choice, it is possible to adopt a unique shape that is able to generate all the standard inference rules. In fact, a wide class of logics including classical, MALL and BV admit subatomic proof systems. In [4], simple sufficient conditions are determined for subatomic systems to enjoy generalized cut elimination theorems. Indeed, subatomic systems seem to provide a good level of abstraction for the study of cut elimination in a *general* rather than ad-hoc manner.

Given a language with connectives α, β, we define the shape of the rules as follows:

$$\alpha\hat{\beta}\,\frac{(A\ \beta\ B)\ \alpha\ (C\ \hat{\beta}\ D)}{(A\ \alpha\ C)\ \beta\ (B\ \alpha\ D)} \qquad \beta\check{\alpha}\,\frac{(A\ \beta\ B)\ \alpha\ (C\ \beta\ D)}{(A\ \alpha\ C)\ \beta\ (B\ \check{\alpha}\ D)}$$

Consider the two rules above: we call them the *up* ($\alpha\hat{\beta}$) and the *down* ($\beta\check{\alpha}$) rules. In the case of classical subatomic formulae, we fix $\hat{\wedge} = \hat{\vee} = \wedge$ and $\check{\wedge} = \check{\vee} = \vee$. This notation relates two dual connectives, assigning a stronger $\hat{\ }$ and weaker $\check{\ }$ of the pair. Thus, we can express, for example, the switch rule of system SKS [1] for classical logic in subatomic form as $\vee\check{\wedge}$:

$$\vee\check{\wedge}\,\frac{(A \vee B) \wedge (C \vee D)}{(A \wedge C) \vee (B\ \check{\wedge}\ D)} \quad \equiv \quad \vee\check{\wedge}\,\frac{(A \vee B) \wedge (C \vee D)}{(A \wedge C) \vee (B \vee D)}$$

Subatomic languages consider propositional atoms a as terms $(0\ \mathbf{a}\ 1)$, and dually $\bar{a}$ as $(1\ \mathbf{a}\ 0)$. The new connective $\mathbf{a}$ is self-dual, and we fix $\hat{\mathbf{a}} = \check{\mathbf{a}} = \mathbf{a}$. The set of rules $\{\vee\check{\wedge}, \vee\check{\mathbf{a}}, \wedge\check{\mathbf{a}}, \wedge\check{\vee}, \vee\check{\vee}\}$ and their duals $\{\wedge\hat{\vee}, \wedge\hat{\mathbf{a}}, \vee\hat{\mathbf{a}}, \vee\hat{\wedge}, \wedge\hat{\wedge}\}$ define the subatomic system $\mathrm{SKS^{sa}}$ for classical logic. This system was introduced in [4], where it is called SAKS. Note how the non-linear rules of system SKS for classical logic, contraction and cut, can be embedded into the subatomic system as instances of certain rules. Working modulo some unit equations, the correspondence is clear. For example, $(0 \wedge 1)\ \mathbf{a}\ (1 \wedge 0)\ =\ 0\ \mathbf{a}\ 0 = 0$.

$$\vee\hat{\mathbf{a}}\,\frac{(0\ \mathbf{a}\ 1) \vee (0\ \mathbf{a}\ 1)}{(0 \vee 0)\ \mathbf{a}\ (1 \vee 1)} \quad \frac{a \vee a}{a} \qquad\qquad \wedge\hat{\mathbf{a}}\,\frac{(0\ \mathbf{a}\ 1) \wedge (1\ \mathbf{a}\ 0)}{(0 \wedge 1)\ \mathbf{a}\ (1 \wedge 0)} \quad \frac{a \wedge \bar{a}}{0}$$

$$\vee\check{\mathbf{a}}\,\frac{(0 \vee 1)\ \mathbf{a}\ (1 \vee 0)}{(0\ \mathbf{a}\ 1) \vee (1\ \mathbf{a}\ 0)} \quad \frac{1}{a \vee \bar{a}} \qquad\qquad \wedge\check{\mathbf{a}}\,\frac{(0 \wedge 0)\ \mathbf{a}\ (1 \wedge 1)}{(0\ \mathbf{a}\ 1) \wedge (0\ \mathbf{a}\ 1)} \quad \frac{a}{a \wedge a}$$

'Nesting' these new connectives – where one atom is contained within the scope of another, e.g., $((0\ \mathbf{a}\ 1)\ \mathbf{b}\ 0)$ – has not been fully explored. We start to do that here, and in the case of classical logic we propose a natural semantics for such formulae given by *decision trees (DTs)*. Further, we define and investigate the proof system $\mathrm{DT^{sa}}$, given by taking all possible rules of the given shape for the connectives of (subatomic) classical logic. It is a 'completion' of the subatomic system for classical logic, $\mathrm{SKS^{sa}}$, and is indeed a conservative extension of classical logic. In particular, $\mathrm{DT^{sa}}$ has no restriction on nesting and thus allows us to make inferences about DTs.

We show this system to be complete with respect to the expected DT semantics. We also have a novel and exceedingly simple proof of cut elimination, an example of which is detailed in this abstract. It is striking that the subatomic methodology leads directly to the discovery of such a proof system.

Decision Trees and Semantics of Subatomic Formulae

A decision tree is a data structure that forms the basis of many modern, efficient implementations of boolean functions [5]. It is a binary tree of conditionals, where each node is labelled by a boolean variable, and each leaf a unit 0 or 1. An example of a DT is given by the abstract syntax tree of a formula such as $((0\ \mathbf{a}\ 1)\ \mathbf{b}\ (1\ \mathbf{c}\ (0\ \mathbf{b}\ 1)))$. Intuitively, we can read the DT represented by $A\ \mathbf{a}\ B$ as 'if a then B, else A'.

Suppose we wish to prove that two DTs represent the same function. In what follows, we detail an approach which enriches the language of DTs with propositional connectives $\wedge$ and $\vee$, in order to be able to express implication between formulae, as well as gaining access to the proof compression mechanisms of contraction and cut.

We fix our set of *subatomic classical formulae*, equipped with an involution $\bar{\cdot}$ and given by:

$$A, B, C, D \quad ::= \quad 0 \mid 1 \mid (A \;\mathbf{a}\; B) \mid (A \wedge B) \mid (A \vee B)$$

$$\overline{A \wedge B} = \bar{A} \vee \bar{B} \quad \overline{A \vee B} = \bar{A} \wedge \bar{B} \quad \overline{A \;\mathbf{a}\; B} = \bar{A} \;\mathbf{a}\; \bar{B} \quad \bar{0} = 1 \quad \bar{1} = 0$$

where $\mathbf{a} \in \mathcal{A}$ is drawn from a countable set of variables. Commutativity and associativity of $\wedge$ and $\vee$ are handled by inference rules of the subatomic shape. We take as equalities on these formulae the linear[1] unit equations of classical logic, plus $0 = 0 \;\mathbf{a}\; 0$ and $1 = 1 \;\mathbf{a}\; 1$ for every variable $\mathbf{a} \in \mathcal{A}$.

For any assignment of boolean values to variables given by a function $X{:}\,\mathcal{A} \to \{0,1\}$, we can define the semantics of subatomic classical formulae via a function $[\![-]\!]_X$ as follows:

$$[\![0]\!]_X = 0 \quad [\![1]\!]_X = 1 \quad [\![A \wedge B]\!]_X = [\![A]\!]_X \wedge [\![B]\!]_X \quad [\![A \vee B]\!]_X = [\![A]\!]_X \vee [\![B]\!]_X$$

$$[\![A \;\mathbf{a}\; B]\!]_X = \begin{cases} [\![A]\!]_X & \text{if } X(a) = 0 \\ [\![B]\!]_X & \text{if } X(a) = 1 \end{cases}$$

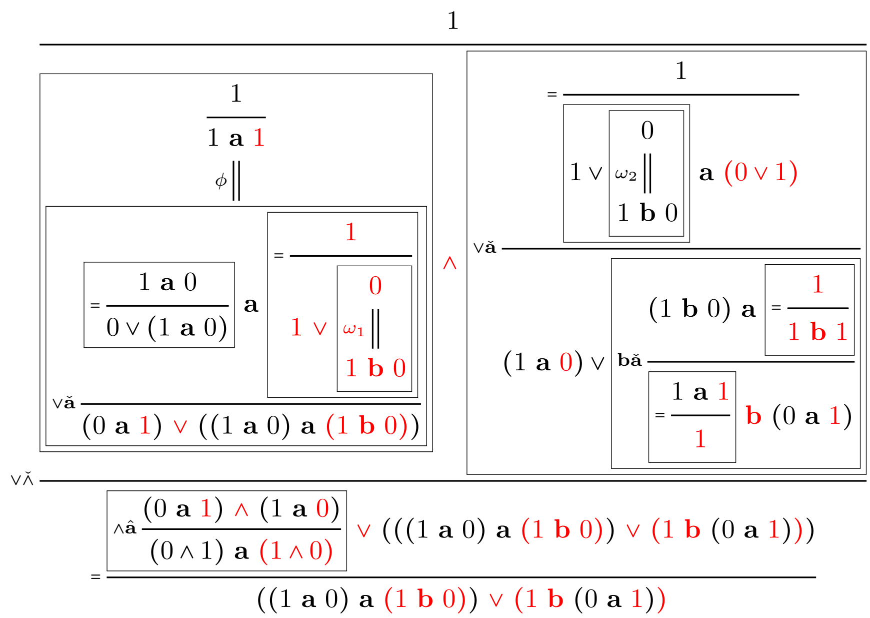
There is a natural embedding of the formulae of propositional classical logic into this language, given by substituting atoms a and $\bar{a}$ for their corresponding DTs $(0 \;\mathbf{a}\; 1)$ and $(1 \;\mathbf{a}\; 0)$ respectively. Formulae that involve 'nesting' of atoms are not in the image of this embedding. System $\mathsf{SKS}^{\mathsf{sa}}$ has a restricted language, in which no nested atoms may appear. Thus, the system maintains a correspondence with the more traditional system SKS for classical logic.

We consider as a *cut on* $\mathbf{a}$ any instance of the rule $\wedge\hat{\mathbf{a}}$ such that the premise is equal to $(0 \;\mathbf{a}\; 1) \wedge (1 \;\mathbf{a}\; 0)$ or $(1 \;\mathbf{a}\; 0) \wedge (0 \;\mathbf{a}\; 1)$, and thus with conclusion equal to 0. These are exactly the inferences which correspond to an atomic cut when interpreted in system SKS.

Deep Inference

Deep inference is the ability to apply inference rules arbitrarily deep within a formula [2]. Thus, derivations are composable horizontally by the same connectives as formulae, as well as in the usual vertical manner.

In the following example, we prove $((0 \;\mathbf{a}\; 1) \;\mathbf{a}\; (0 \;\mathbf{b}\; 1)) \to (1 \;\mathbf{b}\; (0 \;\mathbf{a}\; 1))$, *i.e.*, $\overline{(0 \;\mathbf{a}\; 1) \;\mathbf{a}\; (0 \;\mathbf{b}\; 1)} \vee (1 \;\mathbf{b}\; (0 \;\mathbf{a}\; 1))$. In particular, we take note of the *cut* $(\wedge\hat{\mathbf{a}})$ of $(0 \;\mathbf{a}\; 1)$ and $(1 \;\mathbf{a}\; 0)$. Rules labelled with $=$ are equations on formulae.



(1)

[1] In particular, this means that we take $A = A \vee 0$ and $A = A \wedge 1$, but not the equations $A \wedge 0 = 0$ or $A \vee 1 = 1$ in general.

The derivation ϕ used in this proof, from $1 \mathbf{a} 1$ to $(1 \mathbf{a} 0) \mathbf{a} 1$, is an instance of the *DT-weakening* construction, which we omit details of in this abstract. Briefly, in DTs such as $(A \mathbf{a} B) \mathbf{a} C$, there is semantically irrelevant information – the truth of the formula does not depend on B. A DT-weakening can be used to derive $(A \mathbf{a} B) \mathbf{a} C$ from $A \mathbf{a} C$. The 'traditional' weakenings such as ω_1 or ω_2 can be derived using certain applications of inference rules to units.

The *red* elements are those that are in the *right projection* of $\mathbf{a}$. This is defined on formulae by replacing every subformula of the form $A \mathbf{a} B$ with B. The left projection is defined analogously. The obvious extension of this definition to derivations is well-defined, with any inference rules that are broken easily fixed. For example, the highlighting above shows the right projection of the derivation indeed gives a valid subproof, and this is the case in general. We will use projections of derivations later, in the proof of cut elimination.

A Proof System for Decision Trees

We define our subatomic proof system for DTs, which we call $\mathsf{DT}^{\mathsf{sa}}$, as the set of all possible up ($\alpha\hat{\beta}$) and down ($\beta\check{\alpha}$) rules for $\alpha, \beta \in \{\wedge, \vee\} \cup \mathcal{A}$. The full set of *down* rules is:

$$\vee\check{\vee}\frac{(A \vee B) \vee (C \vee D)}{(A \vee C) \vee (B \vee D)} \qquad \wedge\check{\vee}\frac{(A \wedge B) \vee (C \wedge D)}{(A \vee C) \wedge (B \vee D)} \qquad \mathbf{a}\check{\vee}\frac{(A \mathbf{a} B) \vee (C \mathbf{a} D)}{(A \vee C) \mathbf{a} (B \vee D)}$$

$$\vee\check{\wedge}\frac{(A \vee B) \wedge (C \vee D)}{(A \wedge C) \vee (B \vee D)} \qquad \wedge\check{\wedge}\frac{(A \wedge B) \wedge (C \wedge D)}{(A \wedge C) \wedge (B \vee D)} \qquad \mathbf{a}\check{\wedge}\frac{(A \mathbf{a} B) \wedge (C \mathbf{a} D)}{(A \wedge C) \mathbf{a} (B \vee D)}$$

$$\vee\check{\mathbf{a}}\frac{(A \vee B) \mathbf{a} (C \vee D)}{(A \mathbf{a} C) \vee (B \mathbf{a} D)} \qquad \wedge\check{\mathbf{a}}\frac{(A \wedge B) \mathbf{a} (C \wedge D)}{(A \mathbf{a} C) \wedge (B \mathbf{a} D)} \qquad \mathbf{a}\check{\mathbf{b}}\frac{(A \mathbf{a} B) \mathbf{b} (C \mathbf{a} D)}{(A \mathbf{b} C) \mathbf{a} (B \mathbf{b} D)}$$

Some of these are admissible, or even derivable, for system $\mathsf{SKS}^{\mathsf{sa}}$, for example $\mathbf{a}\check{\wedge}$ and $\wedge\check{\wedge}$. The rules given below are the dual set to those above – that is, they are all the *up* rules of the system. Some self-dual rules such as $\mathbf{a}\check{\mathbf{b}}$ and $\wedge\check{\vee}$ are included in both the up and down sets.

$$\wedge\hat{\wedge}\frac{(A \wedge B) \wedge (C \wedge D)}{(A \wedge C) \wedge (B \wedge D)} \qquad \vee\hat{\wedge}\frac{(A \wedge B) \vee (C \wedge D)}{(A \vee C) \wedge (B \vee D)} \qquad \mathbf{a}\hat{\wedge}\frac{(A \wedge B) \mathbf{a} (C \wedge D)}{(A \mathbf{a} C) \wedge (B \mathbf{a} D)}$$

$$\wedge\hat{\vee}\frac{(A \vee B) \wedge (C \wedge D)}{(A \wedge C) \vee (B \wedge D)} \qquad \vee\hat{\vee}\frac{(A \vee B) \vee (C \wedge D)}{(A \vee C) \vee (B \vee D)} \qquad \mathbf{a}\hat{\vee}\frac{(A \vee B) \mathbf{a} (C \wedge D)}{(A \mathbf{a} C) \vee (B \mathbf{a} D)}$$

$$\wedge\hat{\mathbf{a}}\frac{(A \mathbf{a} B) \wedge (C \mathbf{a} D)}{(A \wedge C) \mathbf{a} (B \wedge D)} \qquad \vee\hat{\mathbf{a}}\frac{(A \mathbf{a} B) \vee (C \mathbf{a} D)}{(A \vee C) \mathbf{a} (B \vee D)} \qquad \mathbf{a}\hat{\mathbf{b}}\frac{(A \mathbf{b} B) \mathbf{a} (C \mathbf{b} D)}{(A \mathbf{a} C) \mathbf{b} (B \mathbf{a} D)}$$

We can prove completeness of system $\mathsf{DT}^{\mathsf{sa}}$ relative to the semantics $[\![-]\!]$ given above. There exist cut-free derivations within the system between $A \mathbf{a} B$ and $(A \wedge (1 \mathbf{a} 0) \vee ((0 \mathbf{a} 1) \wedge B)$, in both directions. Thus, we can reduce the proof of completeness to that of subatomic system $\mathsf{SKS}^{\mathsf{sa}}$ for classical logic, which is known.

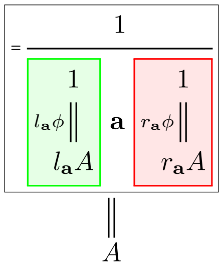
Cut Elimination

There are several notable cut-free derivations available in the system. We will mention three constructions, the derivations of which are left as easy exercises; here we use $l_{\mathbf{a}}$ and $r_{\mathbf{a}}$ to denote the left and right projections on $\mathbf{a}$, as discussed earlier.

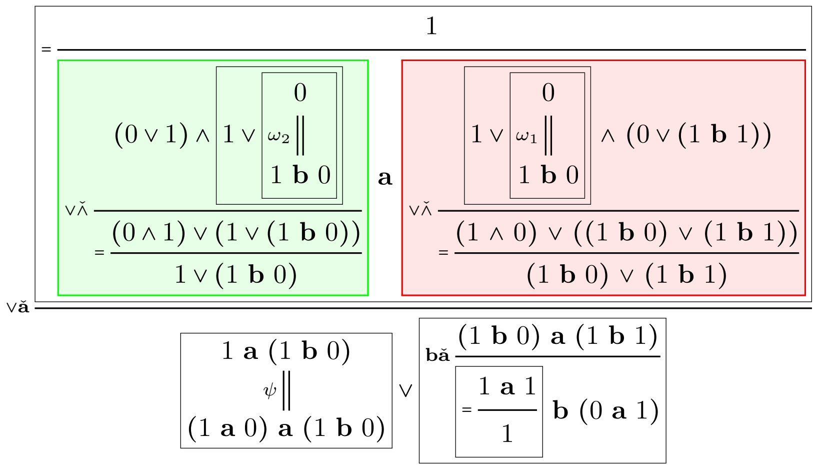
$$\begin{array}{c} A \mathbf{a} A \\ \| \\ A \end{array} , \qquad \begin{array}{c} A \mathbf{a} C \\ \| \\ (A \mathbf{a} B) \mathbf{a} C \end{array} \qquad \begin{array}{c} l_{\mathbf{a}}A \mathbf{a} r_{\mathbf{a}}A \\ \| \\ A \end{array}$$

The first is a novel type of *generalized contraction*, a construction in [4] which forms the basis of the generalized *decomposition* procedure (*i.e.*, the first half of the cut elimination theorem). The second construction is a DT-weakening, an instance of which was used in our earlier example (1). The third construction can be easily obtained with the help of the first two and is useful for cut elimination. This derivation reorders formulae so that a particular connective, in this case **a**, is at the top.

Again, we use $l_{\mathbf{a}}$ and $r_{\mathbf{a}}$ to denote left and right projections on **a**. In particular, a projection on **a** can be applied to a proof, resulting in a new proof which does not contain any instances of **a**, and thus contains no cuts on **a**. Using these projections and the third construction discussed above, assuming we have some proof ϕ of a formula A, we can construct the following:



Iterating this process for each **a** thus yields a proof with no cuts. The next figure shows an example, where we take ϕ to be the proof we presented earlier in (1). In the example, the derivation ψ is a DT-weakening. The green and red boxes denote the left and right projections respectively, and in particular the red box in the example corresponds to the highlighted subproof of (1).



This is strikingly simple. Further, this construction could be the basis of canonical analytic forms for proofs – under certain suitable equivalences. The proof is similar in this way to the 'experiments method' for system SKS [3].

It is natural to ask if the generalization of the subatomic language which makes the representation of DTs possible can be applied in the case of logics other than classical. We claim that system $\mathrm{SKS}^{\mathrm{sa}} \cup \{\hat{\mathbf{ab}}\}$ is a sensible extension of classical logic. A subset of the rules of $\mathrm{SKS}^{\mathrm{sa}}$ gives a proof system for MLL, and it is natural to ask if $\mathrm{MLL} \cup \{\hat{\mathbf{ab}}\}$ might itself be a sensible proof system – and, more generally, if we can apply the same extension to proof systems for MALL and BV. These questions are left for future research.

Acknowledgements

The authors would like to thank Andrea Aler Tubella, Anupam Das and Willem Heijltjes for helpful exchanges and insights, as well as the reviewers for their comments.

References

[1] Kai Brünnler and Alwen Fernanto Tiu. A local system for classical logic. In R. Nieuwenhuis and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, volume 2250 of *Lecture Notes in Computer Science*, pages 347–361. Springer-Verlag, 2001.

[2] Alessio Guglielmi, Tom Gundersen, and Michel Parigot. A proof calculus which reduces syntactic bureaucracy. In *Proceedings of the 21st International Conference on Rewriting Techniques and Applications*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2010.

[3] Benjamin Ralph. *Modular Normalisation of Classical Proofs*. PhD thesis, University of Bath, 2019.

[4] Andrea Aler Tubella and Alessio Guglielmi. Subatomic proof systems: Splittable systems. *ACM Transactions on Computational Logic (TOCL)*, 19, 2018.

[5] Ingo Wegener. *Branching Programs and Binary Decision Diagrams*. Society for Industrial and Applied Mathematics, 2000.