



Citation for published version:

Barrett, C & Guglielmi, A 2022, 'A Subatomic Proof System for Decision Trees', *ACM Transactions on Computational Logic*, vol. 23, no. 4, 26, pp. 1-25. <https://doi.org/10.1145/3545116>

DOI:

[10.1145/3545116](https://doi.org/10.1145/3545116)

Publication date:

2022

Document Version

Peer reviewed version

[Link to publication](#)

© ACM, 2022. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in *ACM Transactions on Computational Logic* <https://doi.org/10.1145/3545116>

University of Bath

Alternative formats

If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

A Subatomic Proof System for Decision Trees

CHRIS BARRETT, University of Bath, UK

ALESSIO GUGLIELMI, University of Bath, UK

We design a proof system for propositional classical logic that integrates two languages for Boolean functions: standard conjunction-disjunction-negation and binary decision trees. We give two reasons to do so. The first is proof-theoretical naturalness: the system consists of all and only the inference rules generated by the single, simple, linear scheme of the recently introduced subatomic logic. Thanks to this regularity, cuts are eliminated via a natural construction. The second reason is that the system generates efficient proofs. Indeed, we show that a certain class of tautologies due to Statman, which cannot have better than exponential cut-free proofs in the sequent calculus, have polynomial cut-free proofs in our system. We achieve this by using the same construction that we use for cut elimination. In summary, by expanding the language of propositional logic, we make its proof theory more regular and generate more proofs, some of which are very efficient.

That design is made possible by considering atoms as superpositions of their truth values, which are connected by self-dual, non-commutative connectives. A proof can then be projected via each atom into two proofs, one for each truth value, without a need for cuts. Those projections are semantically natural and are at the heart of the constructions in this paper. To accommodate self-dual non-commutativity, we compose proofs in deep inference.

CCS Concepts: • **Theory of computation** → **Proof theory**; *Linear logic*; **Oracles and decision trees**; *Proof complexity*; *Complexity theory and logic*.

Additional Key Words and Phrases: deep inference, open deduction, subatomic logic, cut elimination, Statman tautologies.

ACM Reference Format:

Chris Barrett and Alessio Guglielmi. 2022. A Subatomic Proof System for Decision Trees. *ACM Trans. Comput. Logic* 1, 1, Article 1 (January 2022), 25 pages. <https://doi.org/10.1145/3545116>

1 INTRODUCTION

In this work, we contribute to the proof theory of classical logic in two ways:

- we design a new proof system with a more natural and simpler normalisation theory;
- we show that, within this proof system, significantly smaller cut-free proofs become available.

We do so by augmenting the standard, Boolean algebraic language of propositional classical logic formulae with decision trees [21]. In other words, we build a new proof system around a language based on standard connectives (conjunction, disjunction, negation) and augmented by expressions in which atoms are connectives denoting choice: the formula $A b C$ has the truth value of formula A if atom b is negative and that of C if b is positive. For example, if c is negative, $A \wedge (B c ((D \vee E) f G))$ is interpreted as $A \wedge B$. If c is positive and f negative, the formula is interpreted as $A \wedge (D \vee E)$. We limit ourselves, in this paper, to propositional proof systems, but those can be extended with quantifiers and modalities in standard ways. At first sight, the integration of the two languages of Boolean operators and decision trees is not compelling because each language is complete for

Authors' addresses: Chris Barrett, University of Bath, Bath, UK; Alessio Guglielmi, University of Bath, Bath, UK.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.

1529-3785/2022/1-ART1 \$15.00
<https://doi.org/10.1145/3545116>

Boolean functions. However, an important reason for that integration comes from normalisation. Developing the basic theory requires some work, which we do in this paper. The highlight is a very simple and natural cut-elimination procedure that is, to the best of our knowledge, based on new principles that could have very wide applicability in other parts of proof theory.

The technical starting point is the observation that extending the standard language by decision trees is natural from the point of view of *subatomic logic* [2] in deep inference [5, 12, 15]. In subatomic logic, several inference rules that seem unrelated in the standard proof systems, such as identity, cut, contraction and weakening, are obtained as instances of a more general scheme that generates them all. Not only the design of the rules is unified this way, but also the study of normalisation can be done uniformly. In subatomic logic, all inference rules are linear in the usual, structural sense of dealing with the same variables in the premiss and the conclusion, with no repetitions. Non-linearity, which can manifest itself in inferences as duplication or negation of atoms, is revealed through an interpretation map that does not affect normalisation and that maps the subatomic proofs to the standard ones. Surprisingly, the subatomic logic scheme that generates all the rules of propositional logic also generates sound rules for decision trees.

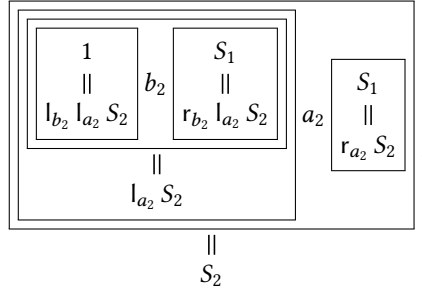
Despite the novelty, and perhaps the exoticism, of the subatomic perspective, the constructions in this paper are very natural and can be easily understood, at least at an intuitive level, without any previous exposure to background material. Indeed, they could be considered a good introduction to the whole area of deep inference. Our aim for the rest of the introduction is to show the main ideas with a simple example. Let us consider the series of tautologies defined by Statman in [19] (of which a more accessible, equivalent definition can be found in [9], Section 5.3.2). The series is defined in Section 6, but, for this introduction, the first three tautologies are sufficient:

$$\begin{aligned}
 S_1 &\equiv (\bar{a}_1 \wedge \bar{b}_1) \vee (a_1 \vee b_1) , \\
 S_2 &\equiv (\bar{a}_2 \wedge \bar{b}_2) \vee \\
 &\quad (((a_2 \vee b_2) \wedge \bar{a}_1) \wedge \\
 &\quad ((a_2 \vee b_2) \wedge \bar{b}_1)) \vee (a_1 \vee b_1) , \\
 S_3 &\equiv (\bar{a}_3 \wedge \bar{b}_3) \vee \\
 &\quad (((a_3 \vee b_3) \wedge \bar{a}_2) \wedge \\
 &\quad ((a_3 \vee b_3) \wedge \bar{b}_2)) \vee \\
 &\quad (((a_3 \vee b_3) \wedge (a_2 \vee b_2) \wedge \bar{a}_1) \wedge \\
 &\quad ((a_3 \vee b_3) \wedge (a_2 \vee b_2) \wedge \bar{b}_1)) \vee (a_1 \vee b_1) .
 \end{aligned}$$

S_1 is obviously a tautology. To see that S_2 is a tautology, one can do a case analysis on the truth values of a_2 and b_2 : if both atoms are false the tautology is trivial and, if any of them is true, S_2 reduces to S_1 because $a_2 \vee b_2$ is true. The same argument works for all the other tautologies in the sequence S_1, \dots, S_n, \dots . That constitutes a polynomial-time proof method, and it is immediate to translate it into a formal proof of some proof system containing a cut rule, which can be used for the case analysis. Indeed, Statman proved that there are polynomial-size tree-like sequent proofs with cut for the tautologies, but only exponential-size tree-like ones if cut is not allowed. It would seem that if we remove the cut from a proof system, we remove the means for generating small proofs by case analysis. However, it turns out that augmenting the language with decision trees circumvents this problem: we can indeed obtain polynomial-size cut-free proofs of Statman tautologies based on case analysis.

We can formalise the previous argument with decision trees in deep inference as follows. If we replace each occurrence of a_2 in S_2 with the unit 1 (for ‘true’), we obtain a formula $r_{a_2} S_2$, which we call the *right projection* of S_2 . Similarly, in the *left projection* $l_{a_2} S_2$, we substitute each occurrence

of a_2 by 0 ('false') and we obtain a formula $\lrcorner_{a_2} S_2$. From $(\lrcorner_{a_2} S_2) a_2 (r_{a_2} S_2)$ we can derive S_2 with a simple cut-free derivation (Lemma 5.8). Therefore, we can represent the above argument about deriving S_2 from S_1 as



A

where \parallel denotes a derivation from A to B. The construction is shown in detail in Figures 6 and 7.

B

In deep inference, particularly in subatomic logic, projections can be lifted from formulae to proofs with minimal effort. This way, we can transform a proof of A, which might contain cuts, into a proof of $(\lrcorner_b A) b (r_b A)$, where b is an atom that we are free to choose. From $(\lrcorner_b A) b (r_b A)$, in turn, we derive A. The proof of A so obtained has the remarkable property of not containing any cut on atom b . By repeating the construction on all the atoms appearing in cuts, we obtain a cut-free proof of A. It is interesting to note that, if we interpret the cuts in the original proof of A as case analyses, we have turned them into decision points in a decision tree. Semantically, they might constitute the same case analysis, but they have an entirely different computational and complexity-theoretic behaviour. We think that this idea is general, novel and natural, and indeed the main interest of this paper because it might have far-reaching consequences in proof theory and proof complexity.

As expected, augmenting a language with new connectives makes new proofs available, and, among them, some are smaller than those in the original language. In this sense, the results in this paper are not a surprise. However, one would not expect the proof system for the augmented language to be less complex than the one for the original language. In fact, new connectives – in our case, atoms – need new inference rules, and this normally means that the amount of information necessary to describe the proof system increases. Surprisingly, in our case, the opposite happens: we can describe the proof system of the augmented language with less information than is necessary for the proof system of the language without atoms-as-connectives. The reason is that *all* the instances of a general, but simple, proof scheme are sound for the extended language, and their collection is complete. This allows us to develop the proof theory of the new proof system working directly with one and only one subatomic scheme, and that is the main reason why we consider our proof system natural.

The findings in this paper corroborate those in [2], the foundational paper of subatomic logic, but the results in this paper have been obtained fortuitously, without any planning or expectation. What are the reasons for the unexpected success of the subatomic scheme? Unfortunately, we cannot offer any convincing explanation at this time. We hope that further research will allow us to understand these phenomena and formulate reasonable research questions.

2 DECISION TREES AND THE SUBATOMIC LANGUAGE

A decision tree is a data structure that forms the basis of many modern, efficient implementations of Boolean functions. For example, the decision tree represented by $A a B$ is read as 'if a then B,

else A' , where a is an atom, *i.e.*, a Boolean variable. Just as in subatomic logic, we consider atoms as binary self-dual, non-commutative connectives.

Suppose we wish to prove that two decision trees represent the same function; we could do so by building a proof system around decision trees. In what follows, we take an approach that enriches the language of decision trees with propositional connectives \wedge and \vee in order to express implication between formulae and gain access to the proof compression mechanisms of contraction and cut.

Definition 2.1. A *decision tree* is a binary tree, where each interior node is labelled by an atom and each leaf is labelled by a unit (0 or 1).

We use the word ‘atom’ both at the standard and the subatomic level. Typographically, we distinguish atoms in subatomic logic by using boldface.

Definition 2.2. Let \mathcal{A} be a countable set of *atoms* whose elements are denoted by $\mathbf{a}, \mathbf{b}, \mathbf{c}, \dots$. We define the *formulae* of the (subatomic) language DT as the language generated by the grammar

$$\text{DT} ::= 0 \mid 1 \mid (\text{DT} \vee \text{DT}) \mid (\text{DT} \wedge \text{DT}) \mid (\text{DT} \mathcal{A} \text{DT}),$$

where the distinct symbols 0, 1, \vee and \wedge do not appear in \mathcal{A} . We call 0 and 1 *units*. Atoms, \vee and \wedge are *connectives*. We denote formulae in DT as A, B, C, D, \dots . The expression $A \equiv B$ denotes syntactic equality of formulae. A *context* is a formula where some subformulae are substituted by holes, which are denoted $\{ \}$. Given a context with a single hole, denoted $K\{ \}$, and a formula A , let $K\{A\}$ denote the formula given by substituting A into the hole. In a formula $K\{A \mathbf{a} B\}$, for any $\mathbf{a} \in \mathcal{A}$, each atom occurrence in A or B is said to be *nested*. We may drop parentheses when there is no ambiguity. The *size* of a formula is the number of occurrences of units appearing in it.

Definition 2.3. We equip the language DT with an involution $\bar{\cdot} : \text{DT} \rightarrow \text{DT}$, which we call *negation*, defined inductively as follows, for all atoms \mathbf{a} :

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

We also use the same notation to denote the *dual* of connectives, *i.e.*, $\bar{\vee} = \wedge$, $\bar{\wedge} = \vee$ and $\bar{\mathbf{a}} = \mathbf{a}$.

We take as equalities on formulae the linear unit equations of classical logic, plus two new subatomic unit equations.

Definition 2.4. The equivalence $=$ on formulae is the minimal equivalence relation, closed under context (*i.e.*, $A = B$ implies $K\{A\} = K\{B\}$, for any context $K\{ \}$), defined by the following equations:

$$\begin{aligned} A \vee 0 &= A, & 0 \wedge 0 &= 0, & 0 \mathbf{a} 0 &= 0, \\ A \wedge 1 &= A, & 1 \vee 1 &= 1, & 1 \mathbf{a} 1 &= 1, \end{aligned}$$

for every atom \mathbf{a} .

For example, $1 \vee (0 \mathbf{a} 1)$ is equal under $=$ to $1 \vee (((0 \mathbf{a} 1) \wedge 1) \vee (0 \mathbf{b} 0))$, but neither formula is equal to 1.

The language DT expresses both the language of decision trees and the language of classical propositional formulae. The latter is obtained via an interpretation function that interprets as standard formulae all the subatomic formulae that do not contain irreducible nested atoms. We characterise these two sublanguages as follows.

Definition 2.5. Given $A \in \text{DT}$, we say that A is a *propositional formula* if there is a $B = A$ with no nested atom occurrences; we say that A is a *strict decision tree* if there is a $B = A$ with no conjunctions or disjunctions. We denote with Prop and SDT, respectively, the subsets of DT containing all the propositional formulae and the strict decision trees.

For example, a formula that contains no nested atom occurrences, and thus is in Prop, is $(1 \mathbf{a} 0) \vee (1 \mathbf{b} 0)$. Any formula equivalent to this is also in Prop, even if it contains nested atom occurrences. For example, $(1 \mathbf{a} (0 \mathbf{c} 0)) \vee (1 \mathbf{b} 0)$. Similarly, the formula $(1 \mathbf{b} 0) \mathbf{a} (1 \mathbf{c} 1)$ is in SDT, and thus so is the equivalent formula $(1 \mathbf{b} 0) \mathbf{a} (0 \vee 1)$, even though it contains a disjunction.

Remark 2.6. There is a natural map p from the formulae of classical propositional logic to the language Prop, given by substituting positive and negative atoms a and \bar{a} with the corresponding decision trees $0 \mathbf{a} 1$ and $1 \mathbf{a} 0$, respectively. For example, $p((1 \vee a) \wedge \bar{b}) = (1 \vee (0 \mathbf{a} 1)) \wedge (1 \mathbf{b} 0)$.

Throughout the paper, we exhibit constructions that allow us to translate between Prop and SDT within our proof system; that is, for any formula in Prop, we can find a semantically equivalent formula in SDT and vice versa, and proofs, within the proof system, supporting that equivalence. Prop and SDT are syntax for Boolean functions and can be interpreted as such via a standard evaluation function $\llbracket \cdot \rrbracket$. In the following, we use the symbols \wedge and \vee also as Boolean operators on the set of values $\{0, 1\}$, i.e., we use them in the semantics without typographically distinguishing them.

Definition 2.7. For any assignment of Boolean values to atoms $X: \mathcal{A} \rightarrow \{0, 1\}$, we define the *semantic interpretation* of formulae $\llbracket \cdot \rrbracket_X: \text{DT} \rightarrow \{0, 1\}$ as

$$\begin{aligned} \llbracket 0 \rrbracket_X &= 0, & \llbracket A \vee B \rrbracket_X &= \llbracket A \rrbracket_X \vee \llbracket B \rrbracket_X, & \llbracket A \mathbf{a} B \rrbracket_X &= \begin{cases} \llbracket A \rrbracket_X & \text{if } X(\mathbf{a}) = 0, \\ \llbracket B \rrbracket_X & \text{if } X(\mathbf{a}) = 1, \end{cases} \\ \llbracket 1 \rrbracket_X &= 1, & \llbracket A \wedge B \rrbracket_X &= \llbracket A \rrbracket_X \wedge \llbracket B \rrbracket_X, \end{aligned}$$

for every atom \mathbf{a} . We say that A *implies* B if, for every assignment X , $\llbracket A \rrbracket_X \leq \llbracket B \rrbracket_X$. We say that A is a *tautology* if for every assignment X , $\llbracket A \rrbracket_X = 1$. We say that two formulae A and B are *semantically equivalent* if for every assignment X , $\llbracket A \rrbracket_X = \llbracket B \rrbracket_X$.

For the sublanguages SDT and Prop, the semantic interpretation coincides with the standard one.

3 DEEP INFERENCE AND A SUBATOMIC PROOF SYSTEM FOR DECISION TREES

Deep inference is, essentially, the ability to apply inference rules at arbitrary depth within a formula [2, 12, 14]. Thus, derivations are composable horizontally by the same connectives as formulae, as well as in the usual vertical manner. Subatomic proof systems [1, 2] represent atoms as non-commutative self-dual connectives and, to accommodate this choice, we compose proofs by deep inference. Consequently, it is possible to adopt a unique, linear rule scheme that can generate all the standard inference rules for many logics, including classical, MALL [11] and BV [12].

For example, in system SKS for classical logic [5], non-linear rules such as identity and contraction appear in the subatomic system as the following instances of more general *linear* inferences. Working modulo some unit equations, we can interpret the left sides as the respective right sides, and vice versa:

$$\begin{aligned} \check{\vee} \bar{\mathbf{a}} \frac{(0 \vee 1) \mathbf{a} (1 \vee 0)}{(0 \mathbf{a} 1) \vee (1 \mathbf{a} 0)} &\leftrightarrow \frac{1}{a \vee \bar{a}}, \\ \check{\vee} \bar{\mathbf{a}} \frac{(0 \mathbf{a} 1) \vee (0 \mathbf{a} 1)}{(0 \vee 0) \mathbf{a} (1 \vee 1)} &\leftrightarrow \frac{a \vee a}{a}. \end{aligned}$$

Indeed, it turns out that *all* the instances of the general rule scheme are sound and complete for our extended language. That is not the case for the standard language of propositional classical logic, to which two instances of the scheme do not apply because they have nested atoms. Thus, by

extending the language, we simplify the proof system. This is so if we consider, as is customary in proof complexity, a proof system as an algorithm that checks proofs. A program applying one rule scheme is shorter than one that has to consider several, but not all, instances of the scheme. That comparison also holds in terms of Kolmogorov complexity.

The rule scheme is as follows.

Definition 3.1. The inference rules of *subatomic shape* are those of the form

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

for connectives $\alpha, \beta \in \{\vee, \wedge\} \cup \mathcal{A}$ and formulae A, B, C and D of DT. The *names* of inference rules of subatomic shape are given by $\alpha\hat{\beta}$ and $\beta\check{\alpha}$ respectively. We define $\check{\wedge} = \check{\vee} = \vee$, $\hat{\wedge} = \hat{\vee} = \wedge$ and $\check{a} = \hat{a} = a$. This notation relates two dual connectives, assigning a *weaker* $\check{\cdot}$ and a *stronger* $\hat{\cdot}$ of the pair.

Remark 3.2. Note that the name $a\check{\wedge}$ is not equal to $a\vee$ (which is not considered a name at all), even though $\check{\wedge} = \vee$. That is to say, although $\hat{\cdot}$ and $\check{\cdot}$ are considered as functions on connectives when appearing in a rule instance, they are considered as decorations when appearing in the name of a rule.

Remark 3.3. Some subatomic rules are assigned two names under this scheme, for example $\vee\hat{\wedge}$ and $\wedge\check{\vee}$, as in

$$\vee\hat{\wedge} \frac{(A \wedge B) \vee (C \hat{\wedge} D)}{(A \vee C) \wedge (B \vee D)} \quad \text{and} \quad \wedge\check{\vee} \frac{(A \wedge B) \vee (C \wedge D)}{(A \vee C) \wedge (B \check{\vee} D)}.$$

We consider this one rule with two distinct names. Another example is the pair $a\hat{b}$ and $a\check{b}$:

$$a\hat{b} \frac{(A \mathbf{b} B) \mathbf{a} (C \mathbf{b} D)}{(A \mathbf{a} C) \mathbf{b} (B \mathbf{a} D)} \quad \text{and} \quad a\check{b} \frac{(A \mathbf{a} B) \mathbf{b} (C \mathbf{a} D)}{(A \mathbf{b} C) \mathbf{a} (B \mathbf{b} D)}$$

(where one should keep in mind that \mathbf{a} and \mathbf{b} stand for any atom).

In the deep-inference literature, the cut rule is typically decomposed into several ‘up’ rules and the identity into several ‘down’ rules. It would be tempting to associate the saturation function with the up-down classification of rules. However, we do not have, at this point, any strong reason to do so, therefore we proceed in an agnostic way and we advise the reader not to make any assumption in this regard.

We define our subatomic proof system for decision trees, which we call DT^{sa} , as the set of all the possible rules of subatomic shape.

Definition 3.4. A *subatomic proof system* is a set of inference rules of subatomic shape, together with *equality rules*, for an equational theory $=$, given by $\frac{A}{B}$, for every formulae A and B such that $A = B$. System DT^{sa} is the subatomic proof system obtained by taking all the rules of subatomic shape $\alpha\hat{\beta}$ and $\beta\check{\alpha}$ for connectives $\alpha, \beta \in \{\vee, \wedge\} \cup \mathcal{A}$ together with the set of equality rules generated by $=$ from Definition 2.4.

See Figure 1 for the complete list of non-equality inference rules of DT^{sa} . The list of rules in the figure is only provided to facilitate the reading of the various derivations in this paper. We hope that the reader keeps in mind that all those rules are generated by the subatomic scheme and,

$$\begin{array}{ccc}
\frac{\wedge\hat{\wedge} \frac{(A \wedge B) \wedge (C \wedge D)}{(A \wedge C) \wedge (B \wedge D)}}{\wedge\hat{\wedge} \frac{(A \wedge B) \wedge (C \wedge D)}{(A \wedge C) \wedge (B \wedge D)}} & \frac{\vee\hat{\wedge} \frac{(A \wedge B) \vee (C \wedge D)}{(A \vee C) \wedge (B \vee D)}}{\vee\hat{\wedge} \frac{(A \wedge B) \vee (C \wedge D)}{(A \vee C) \wedge (B \vee D)}} & \frac{\mathbf{a}\hat{\wedge} \frac{(A \wedge B) \mathbf{a} (C \wedge D)}{(A \mathbf{a} C) \wedge (B \mathbf{a} D)}}{\mathbf{a}\hat{\wedge} \frac{(A \wedge B) \mathbf{a} (C \wedge D)}{(A \mathbf{a} C) \wedge (B \mathbf{a} D)}} \\
\frac{\wedge\hat{\vee} \frac{(A \vee B) \wedge (C \wedge D)}{(A \wedge C) \vee (B \wedge D)}}{\wedge\hat{\vee} \frac{(A \vee B) \wedge (C \wedge D)}{(A \wedge C) \vee (B \wedge D)}} & \frac{\vee\hat{\vee} \frac{(A \vee B) \vee (C \wedge D)}{(A \vee C) \vee (B \vee D)}}{\vee\hat{\vee} \frac{(A \vee B) \vee (C \wedge D)}{(A \vee C) \vee (B \vee D)}} & \frac{\mathbf{a}\hat{\vee} \frac{(A \vee B) \mathbf{a} (C \wedge D)}{(A \mathbf{a} C) \vee (B \mathbf{a} D)}}{\mathbf{a}\hat{\vee} \frac{(A \vee B) \mathbf{a} (C \wedge D)}{(A \mathbf{a} C) \vee (B \mathbf{a} D)}} \\
\frac{\wedge\hat{\mathbf{a}} \frac{(A \mathbf{a} B) \wedge (C \mathbf{a} D)}{(A \wedge C) \mathbf{a} (B \wedge D)}}{\wedge\hat{\mathbf{a}} \frac{(A \mathbf{a} B) \wedge (C \mathbf{a} D)}{(A \wedge C) \mathbf{a} (B \wedge D)}} & \frac{\vee\hat{\mathbf{a}} \frac{(A \mathbf{a} B) \vee (C \mathbf{a} D)}{(A \vee C) \mathbf{a} (B \vee D)}}{\vee\hat{\mathbf{a}} \frac{(A \mathbf{a} B) \vee (C \mathbf{a} D)}{(A \vee C) \mathbf{a} (B \vee D)}} & \frac{\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)}}{\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)}} \\
\\
\frac{\vee\hat{\vee} \frac{(A \vee B) \vee (C \vee D)}{(A \vee C) \vee (B \vee D)}}{\vee\hat{\vee} \frac{(A \vee B) \vee (C \vee D)}{(A \vee C) \vee (B \vee D)}} & \frac{\wedge\hat{\vee} \frac{(A \wedge B) \vee (C \wedge D)}{(A \vee C) \wedge (B \vee D)}}{\wedge\hat{\vee} \frac{(A \wedge B) \vee (C \wedge D)}{(A \vee C) \wedge (B \vee D)}} & \frac{\mathbf{a}\hat{\vee} \frac{(A \mathbf{a} B) \vee (C \mathbf{a} D)}{(A \vee C) \mathbf{a} (B \vee D)}}{\mathbf{a}\hat{\vee} \frac{(A \mathbf{a} B) \vee (C \mathbf{a} D)}{(A \vee C) \mathbf{a} (B \vee D)}} \\
\frac{\vee\hat{\wedge} \frac{(A \vee B) \wedge (C \vee D)}{(A \wedge C) \vee (B \vee D)}}{\vee\hat{\wedge} \frac{(A \vee B) \wedge (C \vee D)}{(A \wedge C) \vee (B \vee D)}} & \frac{\wedge\hat{\wedge} \frac{(A \wedge B) \wedge (C \wedge D)}{(A \wedge C) \wedge (B \vee D)}}{\wedge\hat{\wedge} \frac{(A \wedge B) \wedge (C \wedge D)}{(A \wedge C) \wedge (B \vee D)}} & \frac{\mathbf{a}\hat{\wedge} \frac{(A \mathbf{a} B) \wedge (C \mathbf{a} D)}{(A \wedge C) \mathbf{a} (B \vee D)}}{\mathbf{a}\hat{\wedge} \frac{(A \mathbf{a} B) \wedge (C \mathbf{a} D)}{(A \wedge C) \mathbf{a} (B \vee D)}} \\
\frac{\vee\hat{\mathbf{a}} \frac{(A \vee B) \mathbf{a} (C \vee D)}{(A \mathbf{a} C) \vee (B \mathbf{a} D)}}{\vee\hat{\mathbf{a}} \frac{(A \vee B) \mathbf{a} (C \vee D)}{(A \mathbf{a} C) \vee (B \mathbf{a} D)}} & \frac{\wedge\hat{\mathbf{a}} \frac{(A \wedge B) \mathbf{a} (C \wedge D)}{(A \mathbf{a} C) \wedge (B \mathbf{a} D)}}{\wedge\hat{\mathbf{a}} \frac{(A \wedge B) \mathbf{a} (C \wedge D)}{(A \mathbf{a} C) \wedge (B \mathbf{a} D)}} & \frac{\mathbf{a}\hat{\mathbf{b}} \frac{(A \mathbf{a} B) \mathbf{b} (C \mathbf{a} D)}{(A \mathbf{b} C) \mathbf{a} (B \mathbf{b} D)}}{\mathbf{a}\hat{\mathbf{b}} \frac{(A \mathbf{a} B) \mathbf{b} (C \mathbf{a} D)}{(A \mathbf{b} C) \mathbf{a} (B \mathbf{b} D)}}
\end{array}$$

Fig. 1. System DT^{sa} (excluding equations). Every inference rule is generated by the unique subatomic rule scheme, and no instances of the scheme are excepted from inclusion. Atoms \mathbf{a} and \mathbf{b} range over all of \mathcal{A} . Some rules such as $\mathbf{a}\hat{\mathbf{b}}$ and $\wedge\hat{\vee}$ appear twice (denoted by distinct names).

$$\begin{array}{c}
= \\
\frac{1}{\vee\hat{\mathbf{a}} \left(1 \vee \left(\frac{\wedge\hat{\vee} \frac{(0 \wedge 1) \vee (1 \wedge 0)}{(0 \vee 1) \wedge (1 \vee 0)} \mathbf{b} 0}{1} \right) \mathbf{a} (0 \vee (1 \mathbf{b} 1)) \right)} \\
(1 \mathbf{a} 0) \vee \frac{\mathbf{b}\hat{\mathbf{a}} \frac{(1 \mathbf{b} 0) \mathbf{a} (1 \mathbf{b} 1)}{(1 \mathbf{a} 1) \mathbf{b} (0 \mathbf{a} 1)}}{1}
\end{array}$$

Fig. 2. An example proof in system DT^{sa} .

indeed, the best way to understand our work is to familiarise oneself with the scheme rather than using Figure 1 as a dictionary.

The first subatomic system for classical logic was introduced in [2] as SAKS. In this paper, we refer to that system as SKS^{sa} . Note that we deal with commutativity and associativity of \wedge and \vee using inference rules of subatomic shape, rather than equality rules.

Definition 3.5. The subatomic proof system for classical logic, SKS^{sa} , is the set of inference rules $\{\vee\hat{\wedge}, \wedge\hat{\vee}, \vee\hat{\vee}, \wedge\hat{\wedge}, \vee\hat{\wedge}, \wedge\hat{\vee}\} \cup (\bigcup_{\mathbf{a} \in \mathcal{A}} \{\vee\hat{\mathbf{a}}, \wedge\hat{\mathbf{a}}, \wedge\hat{\mathbf{a}}, \vee\hat{\mathbf{a}}\})$, together with the set of equality rules generated by $=$ from Definition 2.4.

The system DT^{sa} can be seen as the subatomic system SKS^{sa} for classical logic extended with the rule $\mathbf{a}\hat{\mathbf{b}}$ (or, equivalently, $\mathbf{a}\hat{\mathbf{b}}$), which allows the manipulation of decision trees. For example, the

proof in Figure 2 would be in SKS^{sa} except for this rule. In other words, we can define $\text{SKS}^{\text{sa}} = \text{DT}^{\text{sa}} \setminus \bigcup_{a,b \in \mathcal{A}} \{\widehat{\text{ab}}, \widetilde{\text{ab}}\}$.

Note that this definition includes some rules not originally in SKS^{sa} as it appears in [2], namely $\{\wedge\check{\lambda}, \wedge\check{\lambda}, \vee\widehat{\nu}, \vee\widehat{\nu}\}$. However, each of these rules can be replaced with a derivation in the original system for only a linear cost in complexity. An example of this can be found after the following definition, which defines deep-inference proofs according to the open deduction formalism [14].

Definition 3.6. Given a subatomic proof system \mathcal{S} , a derivation ϕ in \mathcal{S} with premiss $A \equiv \text{pr } \phi$ and conclusion $B \equiv \text{cn } \phi$, also said *from A to B*, is denoted by

$$\boxed{\begin{array}{c} A \\ \phi \parallel \mathcal{S} \\ B \end{array}};$$

its *width* is denoted by $w \phi$ and its *height* by $h \phi$. We define derivations and the functions pr , cn , w and h inductively; given ψ and χ derivations in \mathcal{S} , a derivation ϕ can be:

- (1) $\phi \equiv A$, where A is a formula; in this case, $\text{pr } \phi \equiv \text{cn } \phi \equiv A$, $w \phi$ is the size of A and $h \phi = 0$;
- (2) a *composition by inference*

$$\phi \equiv \boxed{\begin{array}{c} \psi \\ \rho \frac{\psi}{\chi} \\ \chi \end{array}},$$

where $\rho \frac{\psi}{\chi}$ is an instance of an inference rule in \mathcal{S} , which could be an equality rule; in this case, $\text{pr } \phi \equiv \text{pr } \psi$, $\text{cn } \phi \equiv \text{cn } \chi$, $w \phi = \max(w \psi, w \chi)$ and $h \phi = h \psi + h \chi + 1$;

- (3) a *composition by connective*

$$\phi \equiv (\psi \alpha \chi),$$

where $\alpha \in \{\vee, \wedge\} \cup \mathcal{A}$; in this case, $\text{pr } \phi \equiv (\text{pr } \psi) \alpha (\text{pr } \chi)$, $\text{cn } \phi \equiv (\text{cn } \psi) \alpha (\text{cn } \chi)$, $w \phi = w \psi + w \chi$ and $h \phi = \max(h \psi, h \chi)$.

We consider composition by inference associative. We call a derivation with premiss 1 a *proof*. The *size* of a derivation is the number of occurrences of units appearing within it. We omit the name of a derivation or of a proof system if there is no ambiguity. The boxes around derivations are essentially brackets, and so we sometimes also omit them.

Remark 3.7. Later in the text, we refer to the *mirror image* of a derivation. For example, the following two rules are one the mirror image of the other:

$$\vee\check{\lambda} \frac{(A \vee B) \wedge (C \vee D)}{(A \wedge C) \vee (B \vee D)} \quad \text{and} \quad \frac{(D \vee C) \wedge (B \vee A)}{(D \vee B) \vee (C \wedge A)}.$$

Technically, the rule at the right is not included in the proof system. We can either choose to include the mirror image rules in the proof system, or for each mirror image construction we can make use of commutativity. In both cases, all important proof-theoretic properties are preserved.

PROPOSITION 3.8. *System DT^{sa} is sound with respect to the semantics defined by $\llbracket \cdot \rrbracket$, i.e., if there exists a derivation in DT^{sa} with premiss A and conclusion B , then A implies B .*

PROOF. By inspection of the rules. □

Remark 3.9. For every inference rule $\alpha\tilde{\beta}$ of subatomic shape from A to B , there exists a *dual inference rule* $\gamma\hat{\delta}$ of subatomic shape from \bar{B} to \bar{A} , with $\gamma \equiv \bar{\alpha}$ and $\delta \equiv \bar{\beta}$, and vice versa. Thus, given a derivation ϕ from A to B in DT^{sa} , we can obtain a *dual derivation* from \bar{B} to \bar{A} by inverting ϕ , dualising every formula occurring in it and renaming each inference rule as its dual. For example, the dual of

$$\frac{\frac{\frac{\frac{(A \vee B) \wedge (C \vee D)}{\vee\tilde{\lambda}}}{(A \wedge C) \vee (B \vee D)}}{\vee\tilde{\lambda}} \quad \wedge \quad \frac{\frac{\frac{(E \vee F) \wedge (G \vee H)}{\vee\tilde{\lambda}}}{(E \wedge G) \vee (F \vee H)}}{\vee\tilde{\lambda}}}{((A \wedge C) \wedge (E \wedge G)) \vee ((B \vee D) \vee (F \vee H))}$$

is

$$\frac{\wedge\hat{\nu} \quad \frac{\frac{\frac{((\bar{A} \vee \bar{C}) \vee (\bar{E} \vee \bar{G})) \wedge ((\bar{B} \wedge \bar{D}) \wedge (\bar{F} \wedge \bar{H}))}{\wedge\hat{\nu}}}{(\bar{A} \wedge \bar{B}) \vee (\bar{C} \wedge \bar{D})}}{\wedge\hat{\nu}} \quad \vee \quad \frac{\frac{\frac{(\bar{E} \vee \bar{G}) \wedge (\bar{F} \wedge \bar{H})}{\wedge\hat{\nu}}}{(\bar{E} \wedge \bar{F}) \vee (\bar{G} \wedge \bar{H})}}{\wedge\hat{\nu}}}{\wedge\hat{\nu}} .$$

The dual of a sound derivation is sound.

The following definition introduces derivation composition.

Definition 3.10. Let ϕ and ψ be two derivations such that $\text{cn } \phi \equiv \text{pr } \psi$. We define

$$\boxed{\begin{array}{c} \phi \\ \dots \\ \psi \end{array}},$$

which we call the (*synchronal*) *composition* of ϕ and ψ , as follows:

(1) if $\phi \equiv A \in \text{DT}$, then $\boxed{\begin{array}{c} A \\ \dots \\ \psi \end{array}} \equiv \psi$; similarly, if $\psi \equiv A \in \text{DT}$, then $\boxed{\begin{array}{c} \phi \\ \dots \\ A \end{array}} \equiv \phi$;

(2) if $\phi \equiv \rho \frac{\chi}{\omega}$, then $\boxed{\begin{array}{c} \rho \frac{\chi}{\omega} \\ \dots \\ \psi \end{array}} \equiv \rho \frac{\chi}{\boxed{\begin{array}{c} \omega \\ \dots \\ \psi \end{array}}}$; similarly, if $\psi \equiv \rho \frac{\chi}{\omega}$, then $\boxed{\begin{array}{c} \phi \\ \dots \\ \rho \frac{\chi}{\omega} \end{array}} \equiv \boxed{\begin{array}{c} \phi \\ \dots \\ \chi \\ \dots \\ \omega \end{array}}$;

(3) if $\phi \equiv \chi \alpha \omega$ and $\psi \equiv \chi' \alpha \omega'$, then $\boxed{\begin{array}{c} \chi \alpha \omega \\ \dots \\ \chi' \alpha \omega' \end{array}} \equiv \boxed{\begin{array}{c} \chi \\ \dots \\ \chi' \end{array}} \alpha \boxed{\begin{array}{c} \omega \\ \dots \\ \omega' \end{array}}$.

For example, we have

$$\boxed{\begin{array}{c} \frac{\wedge\hat{\nu} \quad \frac{(A \wedge B) \vee (C \wedge D)}{\wedge E}}{(A \vee C) \wedge (B \vee D)} \\ \dots \\ \frac{\vee\tilde{\lambda} \quad \frac{(A \vee C) \wedge (B \vee D)}{\wedge E}}{(A \wedge B) \vee (C \vee D)} \end{array}} \equiv \frac{\wedge\hat{\nu} \quad \frac{(A \wedge B) \vee (C \wedge D)}{\vee\tilde{\lambda}}}{(A \vee C) \wedge (B \vee D)} \wedge E .$$

We call cuts those inferences that correspond to an atomic cut in system SKS via the embedding in Remark 2.6 and the equational theory =.

Definition 3.11. A cut on a is any instance of the rule

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

such that $A = D = 0$ and $B = C = 1$, or $A = D = 1$ and $B = C = 0$ in the equational theory $=$. We define an *identity* as the dual of a cut. A *cut-free* (resp., *identity-free*) derivation is a derivation that contains no cuts (resp., identities) on any atom.

For example, a cut on a , corresponding to the standard atomic cut rule

$$\frac{a \wedge \bar{a}}{0}$$

is

$$\wedge \hat{a} \frac{((0 \wedge 0) \mathbf{a} (0 \vee 1)) \wedge ((1 \vee 1) \mathbf{a} (0 \wedge 1))}{((0 \wedge 0) \wedge (1 \vee 1)) \mathbf{a} ((0 \vee 1) \wedge (0 \wedge 1))}.$$

At the subatomic level, what we call a cut does not look at all like a cut in the traditional sense. However, a subatomic cut is the only way to produce a standard cut via the embedding. Therefore, if we eliminate the subatomic cuts, we also eliminate the standard ones.

Indeed, the whole normalisation theory of SKS^{sa} , when restricted to Prop , can be lifted from that of SKS , which is the standard system for the standard language of propositional logic. In particular, we have the following proposition.

PROPOSITION 3.12. *SKS^{sa} is implicationally complete for Prop and cut-free complete for tautologies of Prop .*

PROOF. The proof follows from the following three facts (see [2] for more details).

- If there is a derivation from A to B in SKS , then there is a derivation from $p A$ to $p B$ in SKS^{sa} , where p is the map of Remark 2.6; the latter derivation is cut-free if the former is. Such a derivation is obtained by translating via p all its formulae and applying the inference rules of SKS^{sa} in an obvious way.
- SKS is implicationally complete and cut-free complete.
- For every formula C in Prop there is a formula D in the standard language of propositional logic such that $C = p D$. \square

The implicational completeness and cut-free completeness of DT^{sa} also rely on that of SKS , in a very similar way to the above. We prove them in Propositions 4.6 and 4.7, but, for those, we need to develop a bit of theory.

4 COMPLETENESS

In this section, we prove the completeness of our proof system with respect to the semantics of Definition 2.7. We first describe a few constructions that are useful in this proof, and later in the paper.

PROPOSITION 4.1. *In system DT^{sa} , for every formula A , there exist cut-free and identity-free derivations of the form*

$$\begin{array}{ccc} 0 & & A \\ \parallel & \text{and} & \parallel \\ A & & 1 \end{array}.$$

Their width is $O(n)$ and their height is constant, where n is the size of A .

PROOF. We construct the first derivation by replacing each unit 1 appearing in A with

$$\begin{array}{c} 0 \\ \hline (0 \wedge 1) \vee (1 \wedge 0) \\ \wedge \check{\vee} \\ \hline (0 \vee 1) \wedge (1 \vee 0) \\ \hline 1 \end{array}$$

This makes the premiss equal to 0 and the conclusion equal to A . The second derivation can be obtained as the dual of the first. No cuts or identities are used. \square

Definition 4.2. We call *weakenings* and *coweakenings* respectively the derivations constructed in Proposition 4.1.

LEMMA 4.3. *In system DT^{sa} , for every formula A , there exist cut-free and identity-free derivations of the form*

$$\begin{array}{ccc} A & A & A \text{ a } 0 \quad 0 \text{ a } A \\ \parallel & \parallel & \parallel \quad \parallel \\ A \text{ a } 1 & 1 \text{ a } A & A \quad A \end{array} \text{ and } \begin{array}{ccc} A \text{ a } 0 & 0 \text{ a } A & \\ \parallel & \parallel & \\ A & A & \end{array}$$

Their height and width are $O(n)$, where n is the size of A .

PROOF. We construct the first derivation by structural induction on A . The second derivation can be constructed as the mirror image. The final two can be recovered as duals of the first two.

If $A \equiv 0$, the necessary derivation is given by a weakening. If $A \equiv 1$, take the derivation

$$\begin{array}{c} 1 \\ \hline 1 \text{ a } 1 \end{array}$$

If $A \equiv B \alpha C$ for some connective α , construct

$$\begin{array}{c} \boxed{\begin{array}{c} B \\ \parallel \\ B \text{ a } 1 \end{array}} \quad \alpha \quad \boxed{\begin{array}{c} C \\ \parallel \\ C \text{ a } 1 \end{array}} \\ \hline \alpha \hat{\wedge} \\ (B \alpha C) \text{ a } \quad \boxed{\begin{array}{c} 1 \alpha 1 \\ \hline 1 \end{array}} \end{array}$$

Note that the occurrence of $\alpha \hat{\wedge}$ here is not an instance of the cut rule, and, when dualised, not an instance of the identity rule. \square

The semantic equivalence between $B \text{ a } C$ and $(B \wedge (1 \text{ a } 0)) \vee ((0 \text{ a } 1) \wedge C)$ is cut-free provable within the system, *i.e.*, there is a cut-free derivation from one to the other and vice versa. These constructions allow us to nest and un-nest atom occurrences, which in turn allows us to show that for any formula of DT, there is a semantically equivalent formula in Prop (see Definition 2.5), and vice versa, and that this equivalence is provable within the system. This reduces the completeness of system DT^{sa} to that of system SKS^{sa} , which is known.

LEMMA 4.4. *In system DT^{sa} , for every formulae B and C , we can construct cut-free derivations of the form*

$$\begin{array}{ccc} B \text{ a } C & & (B \wedge (1 \text{ a } 0)) \vee ((0 \text{ a } 1) \wedge C) \\ \parallel & \text{and} & \parallel \\ (B \wedge (1 \text{ a } 0)) \vee ((0 \text{ a } 1) \wedge C) & & B \text{ a } C \end{array}$$

Their width and height are $O(n)$, where n is the size of $(B \text{ a } C)$.

PROOF. If $B \neq 0 \neq C$, we construct the two derivations as

$$\begin{array}{c} \frac{\frac{B}{(B \wedge 1) \vee (0 \wedge 0)}}{=} \mathbf{a} \frac{C}{(0 \wedge 0) \vee (1 \wedge C)} \\ \text{va} \\ \frac{\frac{\frac{(B \wedge 1) \mathbf{a} (0 \wedge 0)}{\wedge \hat{\mathbf{a}}} \left(\frac{B \mathbf{a} 0}{\phi \parallel B} \wedge (1 \mathbf{a} 0) \right)}{\wedge \hat{\mathbf{a}}} \vee \frac{\frac{(0 \wedge 0) \mathbf{a} (1 \wedge C)}{\wedge \hat{\mathbf{a}}} \left((0 \mathbf{a} 1) \wedge \frac{0 \mathbf{a} C}{\psi \parallel C} \right)}{\wedge \hat{\mathbf{a}}}}{\vee} \end{array}$$

and

$$\begin{array}{c} \frac{\frac{\frac{B}{\chi \parallel B \mathbf{a} 1} \wedge (1 \mathbf{a} 0)}{\wedge \hat{\mathbf{a}}} \vee \frac{(0 \mathbf{a} 1) \wedge \frac{C}{\omega \parallel 1 \mathbf{a} C}}{\wedge \hat{\mathbf{a}}}}{\wedge \hat{\mathbf{a}}} \\ \text{va} \\ \frac{\frac{(B \wedge 1) \vee (0 \wedge 1)}{=} \mathbf{a} \frac{(1 \wedge 0) \vee (1 \wedge C)}{=} \mathbf{a} \frac{B}{=} \mathbf{a} \frac{C}{=} \end{array}$$

respectively, where ϕ, ψ, χ and ω are instances of the constructions of Lemma 4.3. No occurrence of rule $\wedge \hat{\mathbf{a}}$ can be a cut. If $B = 0$ or $C = 0$, the latter derivation contains a cut in at least one of the subderivations

$$\frac{\frac{\frac{B}{\chi \parallel B \mathbf{a} 1} \wedge (1 \mathbf{a} 0)}{\wedge \hat{\mathbf{a}}}}{\wedge \hat{\mathbf{a}}} \quad \text{and} \quad \frac{(0 \mathbf{a} 1) \wedge \frac{C}{\omega \parallel 1 \mathbf{a} C}}{\wedge \hat{\mathbf{a}}}}{\wedge \hat{\mathbf{a}}}}{\wedge \hat{\mathbf{a}}}$$

To fix this, we replace the problematic subderivation(s) with

$$\frac{\frac{\frac{B}{=} \mathbf{a} 0 \wedge (1 \mathbf{a} 0)}{\wedge \hat{\mathbf{a}}}}{\wedge \hat{\mathbf{a}}} \quad \text{and} \quad \frac{(0 \mathbf{a} 1) \wedge \frac{C}{=} \mathbf{a} 0}{\wedge \hat{\mathbf{a}}}}{\wedge \hat{\mathbf{a}}}}{\left(\left(\frac{0}{=} \mathbf{a} 0 \wedge 1 \right) \mathbf{a} \left(\frac{0}{\pi \parallel 1} \wedge 0 \right) \right)} \quad \text{and} \quad \left(0 \wedge \frac{0}{\theta \parallel 1} \right) \mathbf{a} \left(1 \wedge \frac{0}{=} \mathbf{a} C \right)$$

respectively, where in each case π and θ are instances of the weakening construction.

The width of these derivations is $O(n)$. The derivations ϕ, ψ, χ and ω each have height $O(n)$ and thus each derivation has height $O(n)$, too. \square

LEMMA 4.5. *In system DT^{sa} , for every formula A in DT, there exist some semantically equivalent formula B in Prop and cut-free derivations*

$$\begin{array}{ccc} A & & B \\ \parallel & \text{and} & \parallel \\ B & & A \end{array}$$

PROOF. We construct the first derivation. If there exist nested atoms occurring in A , then we have $A \equiv K\{C \text{ a } D\}$ for some atom a , where either C or D contain an atom occurrence. Applying the first construction from Lemma 4.4 to $C \text{ a } D$ strictly reduces the number of nested atoms in A . Repeatedly applying this construction until there are no more nested atoms yields a derivation with conclusion B that is semantically equivalent to A , and such that B is in Prop.

Repeatedly applying the second construction from Lemma 4.4 to B allows us to nest atom occurrences until we have recovered A . \square

The following proposition derives from the same idea as Proposition 3.12: the implicational completeness for the extended language is derived from the implicational completeness of SKS.

PROPOSITION 4.6. *System DT^{sa} is implicationally complete with respect to the semantics $\llbracket \cdot \rrbracket$, i.e., if A implies B then there exists a derivation with premiss A and conclusion B .*

PROOF. Using Lemma 4.5, we can find formulae C and D in Prop that are semantically equivalent to A and B respectively, together with derivations ϕ from A to C and χ from D to B in DT^{sa} . Because C and D are both in Prop, and because C implies D , there exists a derivation with premiss C and conclusion D in system SKS^{sa} by implicational completeness of SKS^{sa} , as per Proposition 3.12, which we call ψ . We can thus construct within DT^{sa} the derivation

$$\begin{array}{c} A \\ \phi \parallel_{DT^{sa}} \\ C \\ \dots \\ C \\ \psi \parallel_{SKS^{sa}} . \\ D \\ \dots \\ D \\ \chi \parallel_{DT^{sa}} \\ B \end{array}$$

\square

The following proposition also derives from the same idea as Proposition 3.12. We do have here a cut-elimination procedure, but an indirect one, relying on a translation to Prop.

PROPOSITION 4.7. *System DT^{sa} is cut-free complete with respect to the semantics $\llbracket \cdot \rrbracket$, i.e., if A is a tautology, there exists a cut-free proof of A .*

PROOF. Using Lemma 4.5, we can find a formula B in Prop that is semantically equivalent to A , together with a cut-free derivation ψ from B to A . Since B is a tautology in Prop, there exists a cut-free proof ϕ of B in SKS^{sa} by the cut-free completeness of SKS^{sa} for Prop, as per Proposition 3.12. In fact, this is also a proof in DT^{sa} . Thus construct

$$\begin{array}{c} 1 \\ \phi \parallel_{SKS^{sa}} \\ B \\ \dots \\ B \\ \psi \parallel_{DT^{sa}} \\ A \end{array}$$

\square

Can we obtain a direct, and better, proof of cut elimination? The next section is devoted to that.

5 CUT ELIMINATION

In this section, we describe a construction whereby a derivation can be projected, via a given atom, into two derivations, one for each truth value of that atom.

More in detail, we know that if we are given a formula, there is an obvious notion of left (resp., right) projection on an atom a given by taking the value of that formula when a is false (resp., true). This can be done both on standard and subatomic formulae. As we see in this section, in subatomic logic, we can lift the definition of projection from formulae to derivations with minimal effort, because of the linear nature of the inference scheme. Thus, we can transform a proof of A into a proof of $(l_a A) a (r_a A)$, where a is an atom that we are free to choose and $l_a A$ and $r_a A$ are left and right projections of A on a , respectively (the precise definition follows in this section). In that proof, there are no cuts on a because there are no occurrences of a in the projections. Remarkably, we can also build a proof from $(l_a A) a (r_a A)$ to A that does not contain cuts on a . Therefore, by repeating the construction on all the atoms appearing in cuts, we obtain a cut-free proof of A .

LEMMA 5.1. *In system DT^{sa} , for every formula A and every connective $\alpha \in \{\vee\} \cup \mathcal{A}$, $\beta \in \{\wedge\} \cup \mathcal{A}$, we can construct cut-free and identity-free derivations of the form*

$$\frac{A \alpha A}{A} \quad \text{and} \quad \frac{A}{A \beta A}.$$

Their size is $O(n^2)$, where n is the size of A .

PROOF. We construct the first derivation by structural induction on A . The second derivation can be recovered as the dual of the first.

If $A \equiv 0$, take $= \frac{0 \alpha 0}{0}$. If $A \equiv 1$, take $= \frac{1 \alpha 1}{1}$. Otherwise, we have $A \equiv B \gamma C$, for some connective γ , and we construct the derivation

$$\frac{\gamma \tilde{\alpha} \left(\frac{B \gamma C}{B} \alpha \frac{B \gamma C}{C} \right)}{\gamma \tilde{\alpha} \left(\frac{B \alpha B}{B} \gamma \frac{C \alpha C}{C} \right)}.$$

The width of the derivation is $O(n)$, and its height is $O(n)$. □

Definition 5.2. We call *contractions* and *cocontractions* respectively the derivations constructed in Lemma 5.1.

LEMMA 5.3. *In system DT^{sa} , for every formulae A and B and every context $K\{ \}$, there exist cut-free and identity-free derivations of the form*

$$\frac{K\{A\} \wedge B}{K\{A \wedge B\}} \quad \text{and} \quad \frac{K\{A \vee B\}}{K\{A\} \vee B}.$$

Their width is $O(m + n + l)$ and their height is $O(ml)$, where m is the size of $K\{ \}$, n is the size of A and l is the size of B .

PROOF. We construct the first derivation; the second can be obtained as the dual of the first. We proceed by structural induction on $K\{ \}$. In the base case, $K\{ \} \equiv \{ \}$ and we take the trivial

derivation. If $K\{ \} \equiv H\{ \} \wedge C$, $K\{ \} \equiv H\{ \} \vee C$ or $K\{ \} \equiv H\{ \} \mathbf{a} C$, where \mathbf{a} is any atom, take

$$\begin{array}{c}
 \boxed{ \begin{array}{c} (H\{A\} \wedge C) \wedge \\ \hline \frac{B}{B \wedge 1} \end{array} } \\
 \wedge \hat{\wedge} \overline{ \begin{array}{c} \boxed{ \begin{array}{c} H\{A\} \wedge B \\ \parallel \\ H\{A \wedge B\} \end{array} } \wedge \frac{C \wedge 1}{C} } \\
 \end{array} , \quad
 \begin{array}{c}
 \boxed{ \begin{array}{c} (H\{A\} \vee C) \wedge \\ \hline \frac{B}{B \vee 0} \end{array} } \\
 \wedge \check{\vee} \overline{ \begin{array}{c} \boxed{ \begin{array}{c} H\{A\} \wedge B \\ \parallel \\ H\{A \wedge B\} \end{array} } \vee \frac{C \vee 0}{C} } \\
 \end{array} \quad \text{and} \quad
 \begin{array}{c}
 \boxed{ \begin{array}{c} (H\{A\} \mathbf{a} C) \wedge \\ \hline \frac{B}{\phi \parallel B \mathbf{a} 1} \end{array} } \\
 \wedge \hat{\mathbf{a}} \overline{ \begin{array}{c} \boxed{ \begin{array}{c} H\{A\} \wedge B \\ \parallel \\ H\{A \wedge B\} \end{array} } \mathbf{a} \frac{C \wedge 1}{C} } \\
 \end{array} ,
 \end{array}$$

respectively, where ϕ is as in Lemma 4.3. The remaining cases of $K\{ \} \equiv C \wedge H\{ \}$, $K\{ \} \equiv C \vee H\{ \}$ and $K\{ \} \equiv C \mathbf{a} H\{ \}$ can all be dealt with similarly. These derivations contain no instances of the cut or identity rule.

The width of each derivation is $O(n + m + l)$. Note that the height of the third case dominates the other cases; ϕ is of height $O(l)$, whereas the corresponding part of the other two derivations is an equality rule of constant height. There are $O(m)$ inductive steps in the derivation, each contributing height $O(l)$, thus the height of the derivation in total is $O(ml)$. \square

In decision trees such as $(A\mathbf{a}B)\mathbf{a}C$, the truth of the formula does not depend on B . The equivalence of that formula to $A \mathbf{a} C$ is provable within the system. We state and prove the general case below.

LEMMA 5.4. *In system DT^{sa} , for every formulae A, B and C and every atom \mathbf{a} , there exist cut-free derivations of the form*

$$\begin{array}{c}
 K\{A\} \mathbf{a} \\
 \parallel \\
 K\{A \mathbf{a} B\} \mathbf{a} C
 \end{array}
 \quad \text{and} \quad
 \begin{array}{c}
 C \mathbf{a} K\{A\} \\
 \parallel \\
 C \mathbf{a} K\{B \mathbf{a} A\}
 \end{array} ,$$

for any context $K\{ \}$. Their width is $O(m + n)$ and their height is $O(m + n)$, where m is the size of $K\{ \}$ and n is the size of $(A \mathbf{a} B) \mathbf{a} C$.

PROOF. We construct the first derivation; the second can be obtained as the mirror image of the first. The derivation is shown in Figure 3.

We take derivations ϕ , ψ and χ to be instances of the constructions that nest and un-nest atom occurrences from Lemma 4.4, while ω is an instance of weakening. The derivation π is the construction from Lemma 5.3. The derivation contains no cuts.

The width of the derivation is $O(m + n)$. The subderivations ϕ and χ have height $O(m + n)$, while ψ has height $O(n)$ and π has height $O(m)$ so overall the height of the derivation is $O(m + n)$. \square

Definition 5.5. We call *DT-weakenings* the derivations constructed in Lemma 5.4.

The following definition of the ‘projection’ of an atom on a derivation is key to our proof of cut elimination. The right projection on \mathbf{a} is defined on formulae by replacing every subformula of the form $A \mathbf{a} B$ with B , with the left projection defined analogously. For example, the right projection on \mathbf{a} of the formula $(0 \vee (0 \mathbf{a} 1)) \wedge (1 \mathbf{a} (0 \mathbf{a} 1))$ is $(0 \vee 1) \wedge 1$. The obvious extension of this definition to derivations is well defined, with any inference rules that are broken easily fixed.

Definition 5.6. For any atom \mathbf{a} , we denote the *right projection on \mathbf{a}* of a derivation ϕ in system DT^{sa} as $r_{\mathbf{a}} \phi$, which we define inductively as follows.

- (1) The base cases are $r_{\mathbf{a}} 0 \equiv 0$ and $r_{\mathbf{a}} 1 \equiv 1$.

$$\begin{array}{c}
K\{A\} \mathbf{a} C \\
\phi \parallel \\
\boxed{\boxed{K\{A\} \wedge \frac{\frac{\frac{1}{1 \wedge 1} \quad \mathbf{a} \quad \frac{0}{0 \wedge 0}}{\wedge \tilde{\mathbf{a}}}}{(1 \mathbf{a} 0) \wedge (1 \mathbf{a} 0)}}} \\
= \\
\boxed{\boxed{\boxed{K\{A\} \wedge (1 \mathbf{a} 0)} \\
\pi \parallel \\
\left. \begin{array}{c}
A \wedge (1 \mathbf{a} 0) \\
= \\
\left. \begin{array}{c}
(A \wedge (1 \mathbf{a} 0)) \vee \frac{\frac{0}{\omega \parallel}}{(0 \mathbf{a} 1) \wedge B} \\
\psi \parallel \\
A \mathbf{a} B
\end{array} \right\} \\
\wedge (1 \mathbf{a} 0)
\end{array} \right\} \\
\chi \parallel \\
K\{A \mathbf{a} B\} \mathbf{a} C \\
\vee ((0 \mathbf{a} 1) \wedge C)
\end{array}$$

Fig. 3. Derivation of DT-weakening.

- (2) If ϕ is a horizontal composition of two derivations ψ and χ by either the connective \mathbf{a} or a connective $\beta \neq \mathbf{a}$ (which is possibly another atom), define

$$r_{\mathbf{a}} \left(\begin{array}{|c|} \hline A \\ \hline \psi \parallel \\ \hline B \\ \hline \end{array} \mathbf{a} \begin{array}{|c|} \hline C \\ \hline \chi \parallel \\ \hline D \\ \hline \end{array} \right) \equiv \begin{array}{|c|} \hline r_{\mathbf{a}} C \\ \hline r_{\mathbf{a}} \chi \parallel \\ \hline r_{\mathbf{a}} D \\ \hline \end{array} \quad \text{and} \quad r_{\mathbf{a}} \left(\begin{array}{|c|} \hline A \\ \hline \psi \parallel \\ \hline B \\ \hline \end{array} \beta \begin{array}{|c|} \hline C \\ \hline \chi \parallel \\ \hline D \\ \hline \end{array} \right) \equiv \begin{array}{|c|} \hline r_{\mathbf{a}} A \\ \hline r_{\mathbf{a}} \psi \parallel \\ \hline r_{\mathbf{a}} B \\ \hline \end{array} \beta \begin{array}{|c|} \hline r_{\mathbf{a}} C \\ \hline r_{\mathbf{a}} \chi \parallel \\ \hline r_{\mathbf{a}} D \\ \hline \end{array} .$$

- (3) If ϕ is the composition of derivations ψ and χ by the inference rule $\mathbf{a}\hat{\vee}$, define

$$r_{\mathbf{a}} \mathbf{a}\hat{\vee} \frac{\frac{\frac{E}{\psi \parallel}}{(A \vee B) \mathbf{a} (C \wedge D)}}{(A \mathbf{a} C) \vee (B \mathbf{a} D)} \frac{\chi \parallel}{F} \equiv \frac{\frac{\frac{\frac{r_{\mathbf{a}} E}{r_{\mathbf{a}} \psi \parallel}}{r_{\mathbf{a}} C \wedge r_{\mathbf{a}} D}}{(0 \vee r_{\mathbf{a}} C) \wedge (0 \vee r_{\mathbf{a}} D)}}{(0 \wedge 0) \vee (r_{\mathbf{a}} C \vee r_{\mathbf{a}} D)}}{r_{\mathbf{a}} C \vee r_{\mathbf{a}} D} \frac{r_{\mathbf{a}} \chi \parallel}{r_{\mathbf{a}} F} ;$$

we do analogously when the composition is by inference $\mathbf{a}\hat{\wedge}$.

- (4) If ϕ is the composition of derivations ψ and χ by an inference rule $\alpha\tilde{\wedge}$ or $\alpha\hat{\wedge}$, for any connective α , define

$$r_a \frac{\begin{array}{c} E \\ \psi \parallel \\ (A \alpha B) \mathbf{a} (C \alpha D) \\ \hline (A \mathbf{a} C) \alpha (B \mathbf{a} D) \\ \chi \parallel \\ F \end{array}}{\alpha\tilde{\wedge}} \equiv \frac{\begin{array}{c} r_a E \\ r_a \psi \parallel \\ r_a (C \alpha D) \\ \hline r_a (C \alpha D) \\ r_a \chi \parallel \\ r_a F \end{array}}{\alpha\tilde{\wedge}}$$

and

$$r_a \frac{\begin{array}{c} E \\ \psi \parallel \\ (A \mathbf{a} B) \alpha (C \mathbf{a} D) \\ \hline (A \alpha C) \mathbf{a} (B \alpha D) \\ \chi \parallel \\ F \end{array}}{\alpha\hat{\wedge}} \equiv \frac{\begin{array}{c} r_a E \\ r_a \psi \parallel \\ r_a (B \alpha D) \\ \hline r_a (B \alpha D) \\ r_a \chi \parallel \\ r_a F \end{array}}{\alpha\hat{\wedge}}.$$

- (5) Otherwise, if ϕ is the composition of derivations ψ and χ by an inference or equality ρ , then ρ survives the projection unchanged (although it may act on different formulae) and we define

$$r_a \frac{\begin{array}{c} E \\ \psi \parallel \\ A \\ \rho \frac{\quad}{B} \\ \chi \parallel \\ F \end{array}}{\rho} \equiv \frac{\begin{array}{c} r_a E \\ r_a \psi \parallel \\ r_a A \\ \rho \frac{\quad}{r_a B} \\ r_a \chi \parallel \\ r_a F \end{array}}{\rho};$$

if ρ is an equality, note that if $A = B$, then $r_a A = r_a B$ for our equivalence relation \equiv .

We also analogously define the *left projection on a* of a derivation ϕ , denoted $l_a \phi$.

We could see the above definition as a term-rewriting procedure. In that case, note that there are no critical pairs, therefore the procedure is confluent. In other words, given a derivation, a projection is unique.

For an example of the left projection of a proof, see Figure 4.

Remark 5.7. There are no occurrences of atom \mathbf{a} in either $l_a \phi$ or $r_a \phi$, and thus no cuts on \mathbf{a} in either – even if there are cuts on \mathbf{a} in ϕ .

LEMMA 5.8. *In system DT^{sa} , for every formula A and every atom \mathbf{a} , there exist derivations*

$$\begin{array}{c} (l_a A) \mathbf{a} (r_a A) \\ \parallel \\ A \end{array} \quad \text{and} \quad \begin{array}{c} A \\ \parallel \\ (l_a A) \mathbf{a} (r_a A) \end{array},$$

the first of which is cut-free. The size of each derivation is $O(n^3)$, where n is the size of A . If A contains no nested atom occurrences \mathbf{a} , the size of each derivation is $O(n^2)$.

PROOF. We construct the first derivation; the second can be obtained dually. We proceed by induction on the size of A . In the base case, where $A \equiv 0$ or $A \equiv 1$, we take the derivation

$$= \frac{0 \mathbf{a} 0}{0} \quad \text{or} \quad = \frac{1 \mathbf{a} 1}{1},$$

$$\begin{array}{c}
= \\
\hline
\begin{array}{c}
\text{1} \\
\text{a} \ (1 \vee 0) \\
\phi \parallel \\
\frac{\frac{\frac{1 \text{ a } 0}{0 \vee (1 \text{ a } 0)}}{\text{a}} \left(\frac{0}{1 \vee \omega_1 \parallel} \right)}{\text{a}} \left(\frac{0}{1 \vee \omega_2 \parallel} \right) \\
\text{v}\check{\alpha} \\
(0 \text{ a } 1) \vee ((1 \text{ a } 0) \text{ a } (1 \text{ b } 0))
\end{array}
\wedge
\begin{array}{c}
\left(\frac{0}{1 \vee \omega_2 \parallel} \right) \text{a} (0 \vee (1 \text{ b } 1)) \\
\text{v}\check{\alpha} \\
(1 \text{ b } 0) \text{a} (1 \text{ b } 1) \\
\text{b}\check{\alpha} \\
\frac{1 \text{ a } 1}{1} \text{b} (0 \text{ a } 1)
\end{array} \\
\hline
\text{v}\check{\chi} \\
\frac{\frac{(0 \text{ a } 1) \wedge (1 \text{ a } 0)}{(0 \wedge 1) \text{ a } (1 \wedge 0)}}{\text{cut}} \vee (((1 \text{ a } 0) \text{ a } (1 \text{ b } 0)) \vee (1 \text{ b } (0 \text{ a } 1))) \\
= \\
\hline
((1 \text{ a } 0) \text{ a } (1 \text{ b } 0)) \vee (1 \text{ b } (0 \text{ a } 1))
\end{array}$$

Fig. 4. An example proof of the implication $((0 \text{ a } 1) \text{ a } (0 \text{ b } 1)) \rightarrow (1 \text{ b } (0 \text{ a } 1))$, i.e., $\overline{(0 \text{ a } 1) \text{ a } (0 \text{ b } 1)} \vee (1 \text{ b } (0 \text{ a } 1))$ in system DT^{sa} . The black elements are those in the *left projection on a* of the proof, while the pale elements are those discarded in taking the left projection. The derivation ϕ is an instance of DT-weakening, while ω_1 and ω_2 are weakenings. Figure 2 displays a fragment of this proof, with the definition of ω_2 unfolded.

respectively.

If $A \equiv B \beta C$, for $\beta \neq \text{a}$ (but β is possibly another atom), we construct

$$\beta\check{\alpha} \frac{(l_a B \beta l_a C) \text{ a } (r_a B \beta r_a C)}{\frac{\frac{l_a B \text{ a } r_a B}{\psi \parallel} B}{\beta} \frac{l_a C \text{ a } r_a C}{\chi \parallel} C},$$

where ψ and χ are obtained via the inductive assumption.

If $A \equiv B \text{ a } C$, we construct

$$\omega \parallel \frac{l_a B \text{ a } r_a C}{\frac{\frac{l_a B \text{ a } r_a B}{\psi \parallel} B}{\text{a}} \frac{l_a C \text{ a } r_a C}{\chi \parallel} C},$$

where ψ and χ are obtained via the inductive assumption and ω is made up of two instances of DT-weakening. Note that if A contains no nested atom occurrences a , then $A \equiv B \text{ a } C \equiv (l_a B) \text{ a } (l_a C) \equiv (l_a A) \text{ a } (r_a A)$, i.e., we can simply adopt the trivial derivation.

The width of the derivation is $O(n)$; the DT-weakenings used in the second case are of height $O(n)$, while the $\beta\check{\alpha}$ of the first case is of constant size. Thus, in the worst case, we have DT-weakenings of height $O(n)$ stacked $O(n)$ times by the inductive step, for a total height of $O(n^2)$. Thus, the total size is $O(n^3)$. However, if A contains no nested atom occurrences a , we never need the DT-weakening case and thus we have constant size rules stacked $O(n)$ times, for a total size of $O(n^2)$. \square

$$\begin{array}{c}
= \\
\hline
\begin{array}{ccc}
& & 1 \\
& & \hline
& & \begin{array}{|c|} \hline 0 \\ \hline \end{array} \\
& & \left(1 \vee \begin{array}{|c|} \hline \omega_2 \| \\ \hline \end{array} \right) \\
& & \left. \vphantom{\left(1 \vee \begin{array}{|c|} \hline \omega_2 \| \\ \hline \end{array} \right)} \right\} a \\
& & \left. \vphantom{\left(1 \vee \begin{array}{|c|} \hline \omega_2 \| \\ \hline \end{array} \right)} \right\} \begin{array}{|c|} \hline 1 \mathbf{b} 0 \\ \hline \end{array} \\
& & \hline
& \frac{\vee \tilde{\lambda}}{\left(0 \vee 1 \right) \wedge \left(1 \vee \begin{array}{|c|} \hline \omega_2 \| \\ \hline \end{array} \right)} \\
& & \left(0 \wedge 1 \right) \vee \left(1 \vee \left(1 \mathbf{b} 0 \right) \right) \\
& & \hline
& = \\
& & \frac{1 \vee \left(1 \mathbf{b} 0 \right)}{\vee \tilde{a}}
\end{array}
\quad \begin{array}{ccc}
& & 1 \\
& & \hline
& & \left(1 \vee \begin{array}{|c|} \hline \omega_1 \| \\ \hline \end{array} \right) \\
& & \left. \vphantom{\left(1 \vee \begin{array}{|c|} \hline \omega_1 \| \\ \hline \end{array} \right)} \right\} a \\
& & \left. \vphantom{\left(1 \vee \begin{array}{|c|} \hline \omega_1 \| \\ \hline \end{array} \right)} \right\} \begin{array}{|c|} \hline 1 \mathbf{b} 0 \\ \hline \end{array} \\
& & \hline
& \frac{\vee \tilde{\lambda}}{\left(1 \wedge 0 \right) \vee \left(\left(1 \mathbf{b} 0 \right) \vee \left(1 \mathbf{b} 1 \right) \right)} \\
& & \left(1 \mathbf{b} 0 \right) \wedge \left(0 \vee \left(1 \mathbf{b} 1 \right) \right) \\
& & \hline
& = \\
& & \frac{\left(1 \mathbf{b} 0 \right) \vee \left(1 \mathbf{b} 1 \right)}{\vee \tilde{a}}
\end{array}
\end{array}$$

$$\begin{array}{ccc}
\begin{array}{|c|} \hline 1 \mathbf{a} \left(1 \mathbf{b} 0 \right) \\ \hline \end{array} & & \begin{array}{|c|} \hline \left(1 \mathbf{b} 0 \right) a \left(1 \mathbf{b} 1 \right) \\ \hline \end{array} \\
\psi \| & & \text{b}\tilde{a} \\
\begin{array}{|c|} \hline \left(1 \mathbf{a} 0 \right) a \left(1 \mathbf{b} 0 \right) \\ \hline \end{array} & \vee & \begin{array}{|c|} \hline \begin{array}{|c|} \hline 1 \mathbf{a} 1 \\ \hline \end{array} \\ \hline \end{array} \\
& & = \\
& & \frac{1}{\mathbf{b} \left(0 \mathbf{a} 1 \right)}
\end{array}$$

Fig. 5. An example of the cut-elimination construction applied once, for atom a , to the proof we presented in Figure 4. The two shaded boxes at the left and the right of a denote the left and right projections of the proof of Figure 4, respectively. The derivation ψ is an instance of DT-weakening. Some equality rule instances have been grouped.

The previous lemma shows the semantic equivalence between $(l_a A) a (r_a A)$ and A because it establishes the existence of derivations that go from one formula to the other and back. (The same could be established semantically, of course.)

THEOREM 5.9. *In system DT^{sa} , the cut rule is admissible.*

PROOF. Take a proof ϕ of a formula A . If there exists a cut on a , construct

$$\begin{array}{c}
= \\
\hline
\begin{array}{ccc}
& & 1 \\
& & \hline
& & \begin{array}{|c|} \hline 1 \\ \hline \end{array} \\
& & \left. \vphantom{\begin{array}{|c|} \hline 1 \\ \hline \end{array}} \right\} a \\
& & \left. \vphantom{\begin{array}{|c|} \hline 1 \\ \hline \end{array}} \right\} \begin{array}{|c|} \hline l_a \phi \| \\ \hline \end{array} \\
& & \left. \vphantom{\begin{array}{|c|} \hline 1 \\ \hline \end{array}} \right\} \begin{array}{|c|} \hline l_a A \\ \hline \end{array} \\
& & \hline
& \frac{\chi \|}{\mathbf{a}} \\
& & \begin{array}{|c|} \hline r_a \phi \| \\ \hline \end{array} \\
& & \left. \vphantom{\begin{array}{|c|} \hline r_a \phi \| \\ \hline \end{array}} \right\} a \\
& & \left. \vphantom{\begin{array}{|c|} \hline r_a \phi \| \\ \hline \end{array}} \right\} \begin{array}{|c|} \hline r_a A \\ \hline \end{array} \\
& & \hline
& & \mathbf{b}
\end{array}$$

where χ is the cut-free derivation of Lemma 5.8. This proof has no cuts on a . Iterating this construction yields a cut-free proof. \square

For an example of the cut-elimination procedure, see Figure 5. Note that the procedure blows up the size of the proof exponentially. In particular, it grows exponentially in the number of *distinct* atoms in the original proof upon which there are cuts (rather than exponentially in the number of cuts). The procedure is not confluent, due to an arbitrary ordering of the atoms upon which there are cuts. The proof is, somewhat, conceptually similar to the ‘experiments method’ for system SKS [17], which is, however, confluent.

The semantical nature of the cut-elimination procedure, in addition to its simplicity and the simplicity of the proof system itself, are the reasons why we consider DT^{sa} natural.

$$\begin{array}{c}
1 \\
= \\
\boxed{\begin{array}{c} \check{\vee} b_1 \\ \boxed{\begin{array}{c} (1 \vee 0) \mathbf{b}_1 (0 \vee 1) \\ \boxed{\begin{array}{c} (1 \mathbf{b}_1 0) \\ = \\ 1 \wedge \bar{b}_1 \end{array}} \vee \boxed{\begin{array}{c} (0 \mathbf{b}_1 1) \\ = \\ 0 \vee b_1 \end{array}} \end{array}} \\
\end{array}} \mathbf{a}_1 \left(\left(0 \wedge \begin{array}{c} 0 \\ \omega_1 \parallel \\ \bar{b}_1 \end{array} \right) \vee \left(1 \vee \begin{array}{c} 0 \\ \omega_2 \parallel \\ b_1 \end{array} \right) \right) \\
\phi \parallel \\
(\bar{a}_1 \wedge \bar{b}_1) \vee (a_1 \vee b_1)
\end{array}$$

Fig. 6. Base case for the construction of polynomial-size proofs of Statman tautologies.

6 STATMAN TAUTOLOGIES

We mentioned in the introduction that our proof system generates efficient proofs. Indeed, we show that a certain class of tautologies due to Statman, which cannot have better than exponential cut-free proofs in the sequent calculus, have polynomial cut-free proofs in our system. We achieve this by using projections – the same construction that we use for cut elimination. Further, the proofs are semantically natural, being a formalisation of the argument described in the introduction.

The emphasis of this section is on the method rather than the result. We already knew that Statman tautologies can be proved with no cuts and in polynomial time in deep inference [6]. Indeed, a very small amount of deep inference is necessary to achieve that, meaning that allowing inference just below the root connective of formulae is all that is needed to obtain polynomial-size proofs. We also know that that small amount of deep inference is enough to polynomially simulate proofs with inferences at any depth [10] and still retain cut-freeness. Intuitively, we achieve those speed-ups because deep inference captures the distributivity laws more efficiently than the sequent calculus, without involving cuts.

Cuts provide an essentially different compression mechanism from distributivity. Especially when they are reduced to their atomic form, cuts can be interpreted as a case-analysis mechanism. In fact, in an atomic cut, the two dual hypotheses on the truth value of an atom are formulated and then employed in the proof above the cut. The point made in this section is that polynomial-size proofs of the Statman tautologies can be obtained via a natural case analysis, as explained in the introduction, but without cuts.

Definition 6.1. We call *Statman tautologies* the formulae S_1, S_2, \dots :

$$\begin{array}{l}
S_1 \equiv (\bar{a}_1 \wedge \bar{b}_1) \vee (a_1 \vee b_1), \\
\cdots, \\
S_n \equiv (\bar{a}_n \wedge \bar{b}_n) \vee ((A_{n-1}^n \wedge B_{n-1}^n) \vee \cdots \vee (A_1^n \wedge B_1^n)) \vee (a_1 \vee b_1), \\
\cdots,
\end{array}$$

where a_i and b_i stand for $0 \mathbf{a}_i 1$ and $0 \mathbf{b}_i 1$, and \bar{a}_i and \bar{b}_i stand for $1 \mathbf{a}_i 0$ and $1 \mathbf{b}_i 0$, we work modulo associativity and, for $n > k \geq 1$:

$$\begin{array}{l}
A_k^n \equiv (a_n \vee b_n) \wedge \cdots \wedge (a_{k+1} \vee b_{k+1}) \wedge \bar{a}_k, \\
B_k^n \equiv (a_n \vee b_n) \wedge \cdots \wedge (a_{k+1} \vee b_{k+1}) \wedge \bar{b}_k.
\end{array}$$

$$\begin{array}{c}
S_{n-1} \\
\pi \parallel \\
\boxed{\begin{array}{c} S_{n-1} \\ \rho \parallel \\ \boxed{\begin{array}{c} S_{n-1} \\ = \frac{1}{\phi} b_n \\ \frac{0 \vee (1 \wedge \bar{a}_{n-1} \wedge 1 \wedge \bar{b}_{n-1})}{\vee \bigvee_{k=1}^{n-2} (1 \wedge A_k^{n-1} \wedge 1 \wedge B_k^{n-1}) \vee (a_1 \vee b_1)} \\ \chi \parallel \end{array}} \\ \begin{array}{c} \frac{\bar{b}_n}{1 \wedge \bar{b}_n} \vee \left(\frac{b_n}{0 \vee b_n} \wedge \bar{a}_{n-1} \wedge \frac{b_n}{0 \vee b_n} \wedge \bar{b}_{n-1} \right) \\ \vee \bigvee_{k=1}^{n-2} \left(\frac{b_n}{0 \vee b_n} \wedge A_k^{n-1} \wedge \frac{b_n}{0 \vee b_n} \wedge B_k^{n-1} \right) \vee (a_1 \vee b_1) \\ \hline \text{I}_{a_n} S_n \end{array} \\ \end{array}} \\
\theta \parallel \\
S_n
\end{array}$$

where $\phi \equiv 1 \vee$

$$\boxed{\begin{array}{c} 0 \\ \omega \parallel \\ (0 \wedge \bar{a}_{n-1} \wedge 0 \wedge \bar{b}_{n-1}) \\ \vee \bigvee_{k=1}^{n-2} (0 \wedge A_k^{n-1} \wedge 0 \wedge B_k^{n-1}) \vee (a_1 \vee b_1) \end{array}}$$

and $\psi \equiv$

$$\boxed{\begin{array}{c} (\bar{a}_{n-1} \wedge \bar{b}_{n-1}) \vee \bigvee_{k=1}^{n-2} (A_k^{n-1} \wedge B_k^{n-1}) \vee (a_1 \vee b_1) \\ = \\ \left(0 \wedge \begin{array}{c} 0 \\ \parallel \\ \bar{b}_n \end{array} \right) \vee \left(\left(1 \vee \begin{array}{c} 0 \\ \parallel \\ b_n \end{array} \right) \wedge \bar{a}_{n-1} \wedge \left(1 \vee \begin{array}{c} 0 \\ \parallel \\ b_n \end{array} \right) \wedge \bar{b}_{n-1} \right) \\ \vee \bigvee_{k=1}^{n-2} \left(\left(1 \vee \begin{array}{c} 0 \\ \parallel \\ b_n \end{array} \right) \wedge A_k^{n-1} \wedge \left(1 \vee \begin{array}{c} 0 \\ \parallel \\ b_n \end{array} \right) \wedge B_k^{n-1} \right) \vee (a_1 \vee b_1) \end{array}}$$

Fig. 7. Inductive case for the construction of polynomial-size proofs of Statman tautologies.

Note that, in the previous definition, $A_k^n \equiv (a_n \vee b_n) \wedge A_k^{n-1}$ and $B_k^n \equiv (a_n \vee b_n) \wedge B_k^{n-1}$, whenever $n - 1 > k$.

THEOREM 6.2. *There exist cut-free proofs of the Statman tautologies of size $O(m^{2.5})$, where m is the size of the tautologies.*

PROOF. Given S_1, S_2, \dots, S_n , where S_n is of size $O(m) = O(n^2)$, the proof is by induction on n ; the base case is in Figure 6 and the inductive case in Figure 7.

In Figure 6, ϕ is the construction from Lemma 5.8, ω_1 and ω_2 are weakenings, and the size of the whole derivation is (obviously) a constant.

In Figure 7, each derivation from S_i to S_{i+1} is of size $O(m^2)$ and again we use projections: θ and χ are the construction from Lemma 5.8, with size $O(m^2)$ (since there are no nested \mathbf{a}_n 's in S_n); ω is a weakening with size $O(m)$; π is a cocontraction of size $O(m^2)$; ρ is the construction from Lemma 4.3, also with size $O(m^2)$. There are $O(n) = O(m^{0.5})$ inductive steps in the proof, giving a total size of $O(m^{2.5})$. \square

This bound on the size of Statman tautologies matches the bound for dag-like, cut-free sequent proofs presented in [9] (Theorem 5.3.3).

The example on Statman tautologies that we have presented here shows that we can have proofs that are short, based on natural case analysis and cut-free. No other proof system that we know of exhibits those three properties. At this point, we do not know how far our case-analysis technique can go, but we do not expect it to have the same power as case analysis by cuts. This could be the subject of future research.

7 STRICT DECISION TREES

We have seen how to translate between DT and Prop within system DT^{sa} . Here, we see how to translate between DT and SDT, and thus also between Prop and SDT.

LEMMA 7.1. *In system DT^{sa} , for every formula A in DT, there exists some semantically equivalent formula B in SDT and derivations*

$$\begin{array}{ccc} A & & B \\ \parallel & \text{and} & \parallel \\ B & & A \end{array} .$$

PROOF. Here, we construct the second derivation. The first derivation can be obtained dually.

We proceed by induction on the number of distinct atoms (not atom occurrences) appearing in A . If there are no atoms in A , the formula is made up of only conjunctions and disjunctions of units, which is equal under $=$ to a unit 0 or 1 and thus is in SDT.

If A contains at least one atom \mathbf{a} , we construct the following derivation:

$$\boxed{\begin{array}{|c|c|} \hline C & D \\ \hline \phi \parallel & \mathbf{a} \quad \psi \parallel \\ \hline \mathbf{l}_a A & \mathbf{r}_a A \\ \hline \end{array}},$$

$$\chi \parallel$$

$$A$$

where χ is the construction from Lemma 5.8, and ϕ and ψ can be obtained from the inductive hypothesis since both $\mathbf{l}_a A$ and $\mathbf{r}_a A$ must contain strictly fewer distinct atoms than A (they no longer contain any occurrences of \mathbf{a}). Take $B \equiv C \mathbf{a} D$. Because C and D are both in SDT by the inductive hypothesis, B is also in SDT. That B is semantically equivalent to A follows from Lemma 5.8. \square

Remark 7.2. Lemma 7.1 can be used to obtain a B that is an *ordered* decision tree, *i.e.*, given a total ordering \leq on atoms, in every path from the root to a leaf of (the syntax tree of) B , each atom occurs at most once and the sequence of atom occurrences given by each such path respects \leq .

Remark 7.3. Lemma 7.1 yields an easy alternate proof of completeness. Given a tautology A , the construction of Lemma 7.1 yields an equivalent decision tree B and a derivation ϕ from B to A . In

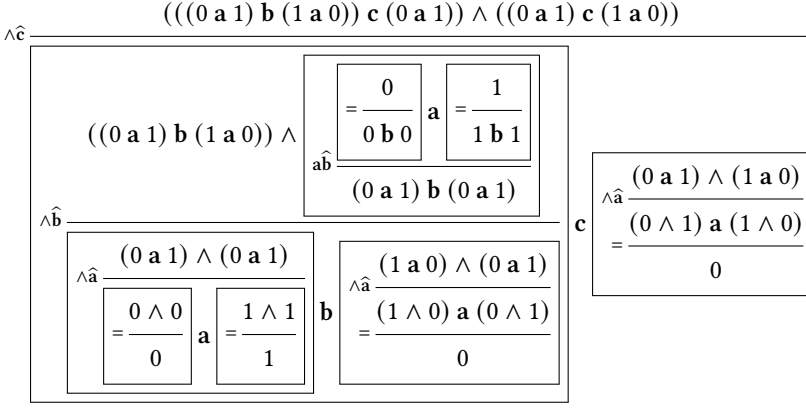


Fig. 8. A derivation corresponding to a run of the common ‘apply’ algorithm on two reduced ordered DTs.

fact, by Remark 7.2, we can take B to be an ordered decision tree. Thus, since B is also a tautology, B must be equal under $=$ to the unit 1, and so ϕ is a proof – in fact, a cut-free one.

We can now translate between the two languages Prop and SDT within our system.

THEOREM 7.4. *In system DT^{sa} , given a formula A in Prop, there exists a semantically equivalent formula B in SDT, a derivation from A to B and a derivation from B to A . Given a formula C in SDT, there exists a semantically equivalent formula D in Prop, a derivation from C to D and a derivation from D to C .*

PROOF. It follow immediately from Lemmas 4.5 and 7.1. □

A *reduced ordered decision tree (RODT)* is an ordered decision tree where every subformula of the form $(A a A)$ is replaced with A [21]. For a given Boolean function and a total ordering on atoms, there is a unique RODT that represents that function and respects the ordering in the sense of Remark 7.2. Instead of decision trees, RODTs are often used in practical applications.

Remark 7.5. Our proof system contains derivations that correspond to runs of some common algorithms used for decision trees. An especially common algorithm, known as ‘apply’, takes two RODTs A and B with the same ordering and a binary Boolean connective \wedge or \vee , and produces a new RODT with that ordering, which represents the same function as $A \wedge B$ or $A \vee B$ respectively.

For example, in the case of applying \wedge to two RODTs $((0 a 1) b (1 a 0)) c (0 a 1)$ and $(0 a 1) c (1 a 0)$, which both have ordering $c < b < a$, the derivation corresponding to a run of the algorithm is given in Figure 8. The conclusion of the derivation is the RODT representation of the premiss, respecting the ordering $c < b < a$.

8 CONCLUSIONS AND FURTHER RESEARCH

Decision trees and closely related structures have been extended in the past with Boolean operators, typically to improve their expressiveness and efficiency, as in [3]. More recently, proof theory has entered the picture in [8], where the authors provide sequent proof systems for decision trees and study their complexity. Against that background, the present work could be seen as a way to design a proof system for decision trees whose proof theory is guaranteed to be natural by the previous research on deep inference. Indeed, nested atoms and the corresponding rules could

simply be added to the current standard system for first-order classical logic, which is presented in [4], and that would effortlessly extend the expressivity of our proof system. We believe that our cut-elimination technique based on projections would extend to that case but we have not explored this yet.

The simplicity of our proof system and cut-elimination procedure is striking, and is a direct consequence of deep inference’s ability to deal with non-commutative connectives [20]. Also, the work [18] is very close to ours because it uses subatomic logic in deep inference to study the normalisation of a proof system with projections, which behave very similarly to decision trees. We would expect that, eventually, all these lines of research will converge towards a standard proof theory of Boolean functions, including decision trees, that supports the applications and whose complexity is well understood.

The expressiveness of our proof system can naturally be extended by quantifiers and modalities. There is now a rich literature in deep inference about standard methods to do so [13]. As we mentioned, the cut-elimination procedure in this paper is exponential in the size of the proof to be normalised. However, in deep inference, quasi-polynomial cut-elimination procedures are available for propositional classical logic [7, 16]. It is natural to wonder whether those results could be extended to our proof system, and possibly improved.

ACKNOWLEDGMENTS

We would like to thank Andrea Aler Tubella, Victoria Barrett, Anupam Das and Willem Heijltjes for helpful exchanges and insights. We also thank the referees for criticism and suggestions that have greatly benefited the paper. This research has been supported by EPSRC project EP/K018868/1, ‘Efficient and Natural Proof Systems’.

REFERENCES

- [1] Andrea Aler Tubella. 2017. *A Study of Normalisation Through Subatomic Logic*. Ph.D. Dissertation. University of Bath. <https://people.bath.ac.uk/ag248/aat/phd.pdf>
- [2] Andrea Aler Tubella and Alessio Guglielmi. 2017. Subatomic Proof Systems: Splittable Systems. *ACM Transactions on Computational Logic* 19, 1 (2017), 5:1–33. <https://doi.org/10.1145/3173544>
- [3] Henrik Reif Andersen and Henrik Hulgaard. 1997. Boolean Expression Diagrams. In *12th Annual IEEE Symposium on Logic in Computer Science (LICS)*, Glynn Winskel (Ed.). IEEE, 88–98. <https://doi.org/10.1109/LICS.1997.614938>
- [4] Kai Brünnler. 2006. Cut Elimination Inside a Deep Inference System for Classical Predicate Logic. *Studia Logica* 82, 1 (2006), 51–71. <https://doi.org/10.1007/s11225-006-6605-4>
- [5] Kai Brünnler and Alwen Fernanto Tiu. 2001. A Local System for Classical Logic. In *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR) (Lecture Notes in Computer Science, Vol. 2250)*, R. Nieuwenhuis and Andrei Voronkov (Eds.). Springer, 347–361. https://doi.org/10.1007/3-540-45653-8_24
- [6] Paola Bruscoli and Alessio Guglielmi. 2009. On the Proof Complexity of Deep Inference. *ACM Transactions on Computational Logic* 10, 2 (2009), 14:1–34. <https://doi.org/10.1145/1462179.1462186>
- [7] Paola Bruscoli, Alessio Guglielmi, Tom Gundersen, and Michel Parigot. 2016. Quasipolynomial Normalisation in Deep Inference Via Atomic Flows and Threshold Formulae. *Logical Methods in Computer Science* 12, 1 (2016), 5:1–30. [https://doi.org/10.2168/LMCS-12\(2:5\)2016](https://doi.org/10.2168/LMCS-12(2:5)2016)
- [8] Sam Buss, Anupam Das, and Alexander Knop. 2020. Proof Complexity of Systems of (Non-Deterministic) Decision Trees and Branching Programs. In *28th EACSL Annual Conference on Computer Science Logic (CSL) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 152)*, Maribel Fernández and Anca Muscholl (Eds.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 12:1–17. <https://doi.org/10.4230/LIPIcs.CSL.2020.12>
- [9] Peter Clote and Evangelos Kranakis. 2002. *Boolean Functions and Computation Models*. Springer. <https://doi.org/10.1007/978-3-662-04943-3>
- [10] Anupam Das. 2011. On the Proof Complexity of Cut-Free Bounded Deep Inference. In *Automated Reasoning with Analytic Tableaux and Related Methods (Tableaux) (Lecture Notes in Artificial Intelligence, Vol. 6793)*, Kai Brünnler and George Metcalfe (Eds.). Springer, 134–148. https://doi.org/10.1007/978-3-642-22119-4_12
- [11] Jean-Yves Girard. 1987. Linear Logic. *Theoretical Computer Science* 50 (1987), 1–102. [https://doi.org/10.1016/0304-3975\(87\)90045-4](https://doi.org/10.1016/0304-3975(87)90045-4)

- [12] Alessio Guglielmi. 2007. A System of Interaction and Structure. *ACM Transactions on Computational Logic* 8, 1 (2007), 1:1–64. <https://doi.org/10.1145/1182613.1182614>
- [13] Alessio Guglielmi. 2022. Deep Inference. Web site at <http://alessio.guglielmi.name/res/cos>.
- [14] Alessio Guglielmi, Tom Gundersen, and Michel Parigot. 2010. A Proof Calculus Which Reduces Syntactic Bureaucracy. In *21st International Conference on Rewriting Techniques and Applications (RTA) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 6)*, Christopher Lynch (Ed.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 135–150. <https://doi.org/10.4230/LIPIcs.RTA.2010.135>
- [15] Alessio Guglielmi and Lutz Straßburger. 2001. Non-commutativity and MELL in the Calculus of Structures. In *Computer Science Logic (CSL) (Lecture Notes in Computer Science, Vol. 2142)*, L. Fribourg (Ed.). Springer, 54–68. https://doi.org/10.1007/3-540-44802-0_5
- [16] Emil Jeřábek. 2009. Proof Complexity of the Cut-Free Calculus of Structures. *Journal of Logic and Computation* 19, 2 (2009), 323–339. <https://doi.org/10.1093/logcom/exn054>
- [17] Benjamin Ralph. 2019. *Modular Normalisation of Classical Proofs*. Ph.D. Dissertation. University of Bath. <https://people.bath.ac.uk/ag248/br/phd.pdf>
- [18] Luca Roversi. 2018. *Subatomic Systems Need Not Be Subatomic*. Technical Report. Università di Torino. <https://arxiv.org/pdf/1804.08105.pdf>
- [19] Richard Statman. 1978. Bounds for Proof-Search and Speed-Up in the Predicate Calculus. *Annals of Mathematical Logic* 15 (1978), 225–287. [https://doi.org/10.1016/0003-4843\(78\)90011-6](https://doi.org/10.1016/0003-4843(78)90011-6)
- [20] Alwen Tiu. 2006. A System of Interaction and Structure II: The Need for Deep Inference. *Logical Methods in Computer Science* 2, 2 (2006), 4:1–24. [https://doi.org/10.2168/LMCS-2\(2:4\)2006](https://doi.org/10.2168/LMCS-2(2:4)2006)
- [21] Ingo Wegener. 2000. *Branching Programs and Binary Decision Diagrams: Theory and Applications*. Society for Industrial and Applied Mathematics. <https://doi.org/10.1137/1.9780898719789>