Non-Elementary Complexities for Branching VASS, MELL, and Extensions *

Ranko Lazić
Dpt. of Computer Science, U. of Warwick, UK
lazic@dcs.warwick.ac.uk

Sylvain Schmitz
LSV, ENS Cachan & CNRS & INRIA, France
schmitz@lsv.ens-cachan.fr

Abstract

We study the complexity of reachability problems on branching extensions of vector addition systems, which allows us to derive new non-elementary complexity bounds for fragments and variants of propositional linear logic. We show that provability in the multiplicative exponential fragment is TOWER-hard already in the affine case—and hence non-elementary. We match this lower bound for the full propositional affine linear logic, proving its TOWER-completeness. We also show that provability in propositional contractive linear logic is ACKERMANN-complete.

Categories and Subject Descriptors F.2.2 [*Analysis of Algorithms and Problem Complexity*]: Nonnumerical Algorithms and Problems—Complexity of proof procedures; F.4.1 [*Mathematical Logic and Formal Languages*]: Mathematical Logic

General Terms Theory

Keywords Fast-growing complexity, linear logic, substructural logic, vector addition systems.

1. Introduction

The use of various classes of counter machines to provide computational counterparts to propositional substructural logics has been highly fruitful, allowing to prove for instance:

- the undecidability of provability in propositional linear logic (LL), thanks to a reduction from the halting problem in Minsky machines proved by Lincoln, Mitchell, Scedrov, and Shankar [14], who initiated much of this line of work,
- the decidability of the !-Horn fragment of multiplicative exponential linear logic, proved by Kanovich [9] by reduction to reachability in vector addition systems,
- the decidability of provability in affine linear logic, first shown by Kopylov using a notion of vector addition games [10],
- the ACKERMANN-completeness of provability in the conjunctive implicative fragment of relevance logic, proved by Urquhart [23], using reductions to and from expansive alternating vector addition systems, and
- the inter-reducibility between provability in multiplicative exponential linear logic and reachability in a model of branching vector addition systems, shown by de Groote, Guillaume, and Salvati [5].

1.1 Alternating Branching VASS

In this paper, we revisit the correspondences between propositional linear logic and counter systems with a focus on computational complexity. In Section 3, we define a model of *alternating branching vector addition systems* (ABVASS) with *full zero tests*. While this model can be seen as an extension and repackaging of Kopylov's vector games, its reachability problem enjoys very simple reductions to and from provability in LL, which are suitable for complexity statements (see Section 4). We prove that:

- coverability in the top-down, root-to-leaves direction is TOWER-complete, i.e. complete for the class of problems that can be solved with time or space resources bounded by a tower of exponentials whose height depends elementarily in the input size (see Section 5 for the upper bound and Section 6 for the lower bound), and
- coverability in the bottom-up, leaves-to-root direction is complete for ACKERMANN, i.e. complete for resources bounded by the Ackermann function of some primitive-recursive function of the input.

Due to space constraints, the technical details for this second point are omitted in this paper but can be found along with other material in the full paper at the address http://arxiv.org/abs/1401.6785.

1.2 Provability in Substructural Logics

Our complexity bounds for ABVASS translate into the exact same bounds for provability in fragments and variants of LL:

1.2.1 Affine Linear Logic

Affine Linear Logic (LLW) was proved decidable by Kopylov [10] in 1995 using vector addition games; a model-theoretic proof was later presented by Lafont [11].[1]

The best known complexity bounds for LLW are due to Urquhart [24]: by a reduction from coverability in vector addition systems [15], he derives an EXPSPACE lower bound, very far from

* Work partially supported by ANR grant ReacHard 11-BS02-001-01.

CSL-LICS '14, July 14–18, 2014, Vienna, Austria.

ACM 978-1-4503-2886-9/14/07…$15.00.
http://dx.doi.org/10.1145/2603088.2603129

[1] Variants of LLW are popular in the literature on implicit complexity; for instance, *light affine linear logic* [1] is known to type exactly the class FP of polynomial time computable functions. In this paper we are however interested in the complexity of the provability problem (for the propositional fragment), rather than the complexity of normalization. In terms of typed lambda calculi, our results pertain to the complexity of the type inhabitation problem.

the ACKERMANN upper bound he obtains from length function theorems for Dickson's Lemma [see e.g. 8].

1.2.2 Contractive Linear Logic

Contractive Linear Logic (LLC) was proved decidable by Okada and Terui [17] by model-theoretic methods.

Urquhart [23] showed the ACKERMANN-completeness of provability in a fragment of relevance logic, which is also a fragment of intuitionistic multiplicative additive LLC. To the best of our knowledge, there are no known complexity upper bounds for provability in LLC.

1.2.3 Multiplicative Exponential Linear Logic

The main open question in this area is whether the multiplicative exponential fragment (MELL) is decidable. It is related to many decision problems, for instance in computational linguistics [19, 20], cryptographic protocol verification [25], the verification of parallel programs [3], and data logics [2, 7].

Thanks to the reductions to and from the reachability problem in branching vector addition systems with states (BVASS) [5] and to the bounds of Lazić [13], we know that provability in MELL is 2-EXPSPACE-hard.

1.2.4 Summary of the Complexity Results

LLW We improve both the lower bound and the upper bound of Urquhart [24], and prove that LLW provability is complete for TOWER.

LLC We show that LLC provability is ACKERMANN-complete; the lower bound already holds for the multiplicative additive fragment MALLC.

MELL Our TOWER-hardness result for LLW already holds for affine MELL and thus for MELL, which improves over the 2-EXPSPACE lower bound of Lazić [13].

ILL All of our complexity bounds also hold for provability in the intuitionistic versions of our calculi. See the full paper for details.

2. Propositional Linear Logic

2.1 Classical Linear Logic

For convenience, we present here a sequent calculus for classical propositional linear logic that works with formulæ in negation normal form and considers one-sided sequents.

2.1.1 Syntax

Propositional linear logic formulæ are defined by the abstract syntax

$$\begin{aligned} A, B ::= {} & a \mid a^\perp && \text{(atomic)} \\ & \mid A \parr B \mid A \otimes B \mid \perp \mid \mathbf{1} && \text{(multiplicative)} \\ & \mid A \mathbin{\&} B \mid A \oplus B \mid \top \mid \mathbf{0} && \text{(additive)} \\ & \mid {!A} \mid {?A} && \text{(exponential)} \end{aligned}$$

where a ranges over atomic formulæ. We write "$A^\perp$" for the negation normal form of A, where negations are pushed to the atoms using the dualities $A^{\perp\perp} = A$, $(A \parr B)^\perp = A^\perp \otimes B^\perp$, $\perp^\perp = \mathbf{1}$, $(A \mathbin{\&} B)^\perp = A^\perp \oplus B^\perp$, $\top^\perp = \mathbf{0}$, and $(?A)^\perp = !A^\perp$. We write "$A \multimap B$" for the linear implication $A^\perp \parr B$.

2.1.2 Sequent Calculus

The rules of the sequent calculus manipulate multisets of formulæ, denoted by Γ, Δ, ..., so that the exchange rule is implicit; "$?\Gamma$" then denotes a multiset of formulæ all guarded by why-nots: $?\Gamma$ is of the form $?A_1, \ldots, ?A_n$.

$$\frac{}{\vdash A, A^\perp}\ \text{init} \qquad \frac{\vdash \Gamma, A \quad \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta}\ \text{cut}$$

$$\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\ \parr \quad \frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}\ \otimes \quad \frac{\vdash \Gamma}{\vdash \Gamma, \perp}\ \perp \quad \frac{}{\vdash \mathbf{1}}\ \mathbf{1}$$

$$\frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \mathbin{\&} B}\ \& \quad \frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B} \ \frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B}\ \oplus \quad \frac{}{\vdash \Gamma, \top}\ \top$$

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}\ ?\text{D} \quad \frac{\vdash \Gamma}{\vdash \Gamma, ?A}\ ?\text{W} \quad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}\ ?\text{C} \quad \frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A}\ ?\text{P}$$

The last four rules for exponential formulæ are called *dereliction* (?D), *logical weakening* (?W), *logical contraction* (?C), and *promotion* (?P).

The *cut* rule can be eliminated in this calculus, which then enjoys the *subformula property*: in any rule except cut, the formulæ appearing in the premises are subformulæ of the formulæ appearing in the conclusion.

2.2 Fragments and Variants

Lincoln et al. [14] established most of the results on the decidability and complexity of provability in propositional linear logic. In particular, the full propositional linear logic (LL) is undecidable, while its multiplicative additive fragment (MALL, which excludes the exponential connectives and rules) is decidable in PSPACE. As mentioned in the introduction, the main open question in this area is whether the multiplicative exponential fragment (MELL, which excludes the additive connectives and rules) is decidable.

Regarding related logics, allowing respectively *structural weakening* (W) and *structural contraction* (C)

$$\frac{\vdash \Gamma}{\vdash \Gamma, A}\ \text{W} \qquad \frac{\vdash \Gamma, A, A}{\vdash \Gamma, A}\ \text{C}$$

instead of logical weakening and logical contraction gives rise to two decidable variants, called respectively *affine* linear logic (LLW) and *contractive* linear logic (LLC). The sequent calculi for LLW and LLC also enjoy cut elimination and the subformula property for cut-free proofs. We consider the intuitionistic variants of LL, LLW, and LLC in the full paper.

3. Alternating Branching VASS

We define a "tree" extension of vector addition systems with states (VASS) that combines two kinds of branching behaviors: those of alternating VASS (Section 3.3.1) and those of branching VASS (Section 3.3.2). With this combination, we obtain a reformulation of Kopylov's vector addition games [10], for which he showed that

1. the game is inter-reductible with LL provability
2. the "lossy" version of the game is inter-reducible with LLW provability.

We further add *full zero tests* to this model, as they make the reduction from LL provability straightforward (see Section 4) and can easily be removed (see Section 3.3.3).

3.1 Definitions

3.1.1 Syntax

An *alternating branching vector addition system with states and full zero tests* ($\text{ABVASS}_{\bar{0}}$) is a tuple $\mathcal{A} = \langle Q, d, T_u, T_f, T_s, T_z \rangle$ where Q is a finite set of *states*, d is a *dimension* in $\mathbb{N}$, and $T_u \subseteq Q \times \mathbb{Z}^d \times Q$, $T_f \subseteq Q^3$, $T_s \subseteq Q^3$ and $T_z \subseteq Q^2$ are respectively finite sets of *unary*, *fork*, *split* and *full zero test* rules.

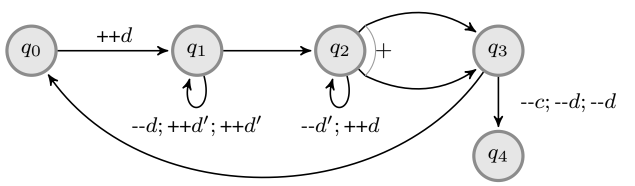

Figure 1. An example BVASS.

We denote unary rules $(q, \bar{u}, q_1)$ in T_u with $\bar{u}$ in $\mathbb{Z}^d$ by "$q \xrightarrow{\bar{u}} q_1$", fork rules (q, q_1, q_2) in T_f by "$q \to q_1 \wedge q_2$", split rules (q, q_1, q_2) in T_s by "$q \to q_1 + q_2$", and full zero test rules (q, q_1) in T_z by "$q \xrightarrow{\stackrel{?}{=}\bar{0}} q_1$".

3.1.2 Deduction Semantics

Given an ABVASS$_{\bar{0}}$, its semantics is defined by a deduction system over *configurations* $(q, \bar{v})$ in $Q \times \mathbb{N}^d$:

$$\frac{q, \bar{v}}{q_1, \bar{v} + \bar{u}} \text{ unary}$$

where "$+$" denotes component-wise addition in $\mathbb{N}^d$, if $q \xrightarrow{\bar{u}} q_1$ is a rule (and implicitly $\bar{v} + \bar{u}$ has no negative component, i.e. is in $\mathbb{N}^d$), and

$$\frac{q, \bar{v}}{q_1, \bar{v} \quad q_2, \bar{v}} \text{ fork} \qquad \frac{q, \bar{v}_1 + \bar{v}_2}{q_1, \bar{v}_1 \quad q_2, \bar{v}_2} \text{ split} \qquad \frac{q, \bar{0}}{q_1, \bar{0}} \text{ full-zero}$$

if $q \to q_1 \wedge q_2$, $q \to q_1 + q_2$, and $q \xrightarrow{\stackrel{?}{=}\bar{0}} q_1$ are rules of the system, respectively, and "$\bar{0}$" denotes the d-vector $\langle 0, \ldots, 0\rangle$ with zeroes on every coordinate. Such a deduction system can be employed either *top-down* or *bottom-up* depending on the decision problem at hand (as with tree automata); the top-down direction will correspond in a natural way to *proof search* in propositional linear logic, i.e. will correspond to the consequence to premises direction in the sequent calculus of Section 2.1.2.

3.1.3 Example

Let $\mathcal{A}$ be an ABVASS$_{\bar{0}}$ with five states $(q_0, q_1, q_2, q_3, q_4)$, of dimension 3, with six unary rules:

$$q_0 \xrightarrow{\langle 0,1,0\rangle} q_1 \qquad q_1 \xrightarrow{\langle 0,-1,2\rangle} q_1 \qquad q_1 \xrightarrow{\langle 0,0,0\rangle} q_2$$
$$q_2 \xrightarrow{\langle 0,1,-1\rangle} q_2 \qquad q_3 \xrightarrow{\langle 0,0,0\rangle} q_0 \qquad q_3 \xrightarrow{\langle -1,-2,0\rangle} q_4,$$

and with one split rule $q_2 \to q_3 + q_3$. There are no fork rules and no full zero test rules in $\mathcal{A}$, and so it is a BVASS (see Section 3.3.2). A depiction of $\mathcal{A}$ is in Figure 1, where we write c, d, d' for vector indices $1, 2, 3$ (respectively), and specify unary rules in terms of increments and decrements.

From state q_0 and with c, d, d' initialised to $4, 0, 0$ (i.e., from a root node labelled by $(q_0, \langle 4, 0, 0\rangle)$), $\mathcal{A}$ can reach q_2 with d, d' having values $2, 0$, perform the split rule by dividing c and d equally (i.e., branch to two nodes labelled by $(q_3, \langle 2, 1, 0\rangle)$), then in both threads reach q_2 again with d, d' having values $4, 0$, perform the split rule as before, and finally in all four threads reach q_4 with c, d, d' having values $0, 0, 0$ (i.e., have four leaf nodes, which are all labelled by $(q_4, \bar{0})$).

Further reasoning can show that $\mathcal{A}$ has a deduction tree whose root is labelled by $(q_0, \langle m, 0, 0\rangle)$ and with the state label at every leaf being q_4 if and only if $m \geq 4$. In fact, $\mathcal{A}$ is a slightly simplified version of the BVASS $\mathcal{B}_2$ in Section 6.

3.2 Decision Problems

3.2.1 Reachability

Given an ABVASS$_{\bar{0}}$ $\mathcal{A}$ and a finite set of states Q_ℓ, we denote by a *root judgement* "$\mathcal{A}, Q_\ell \rhd q, \bar{v}$" the fact that there exists a deduction tree $\mathcal{D}$ in $\mathcal{A}$ with root label $(q, \bar{v})$ and leaf labels in $Q_\ell \times \{\bar{0}\}$. We call $\mathcal{D}$ a reachability witness for $(q, \bar{v})$. Given furthermore a state q_r, the *reachability* problem asks whether $\mathcal{A}, Q_\ell \rhd q_r, \bar{0}$; we call a reachability witness for $(q_r, \bar{0})$ a *reachability witness*.

We will see in Section 4 that this reachability problem is equivalent to provability in LL; the problem is also related to games played over vectors of natural numbers, see the full paper. It is however undecidable:

Fact 1. *Reachability in ABVASS$_{\bar{0}}$ is undecidable.*

Proof. Reachability is already undecidable in the more restricted model of AVASS, see Fact 2 below. □

3.2.2 Lossy Reachability

In order to obtain decidability, we must weaken the ABVASS$_{\bar{0}}$ model or the decision problem. For the former, let us denote by $\bar{e}_i$ the unit vector in $\mathbb{N}^d$ with one on coordinate i and zero everywhere else. Then a *lossy* ABVASS$_{\bar{0}}$ can be understood as featuring a rule $q \xrightarrow{-\bar{e}_i} q$ for every q in Q and $0 < i \leq d$. We rather define it by extending its deduction system with

$$\frac{q, \bar{v}}{q, \bar{v} - \bar{e}_i} \text{ loss}$$

for every q in Q and $0 < i \leq d$. We write '$\rhd_\ell$' for root judgements where losses can occur. In terms of proof search in linear logic, losses will correspond to structural weakening, which is the distinguishing feature of affine linear logic.

Top-Down Coverability An alternative way to see the reachability problem in lossy ABVASS$_{\bar{0}}$ is to weaken the problem. Let us define a variant of ABVASS$_{\bar{0}}$ that feature *full resets* instead of full zero tests: we denote in this case rules (q, q_1) in T_z by $q \xrightarrow{:=\bar{0}} q_1$ and associate a different semantics:

$$\frac{q, \bar{v}}{q_1, \bar{0}} \text{ full-reset}$$

We call the resulting model ABVASSr. Given an ABVASSr $\mathcal{A}$, a state q_r, and a finite set of states Q_ℓ, the *top-down coverability* or *leaf coverability* problem asks whether there exists a deduction tree $\mathcal{D}$ with root label $(q_r, \bar{0})$ and such that, for each leaf, there exists some q_ℓ in Q_ℓ and some $\bar{v}$ in $\mathbb{N}^d$ such that the leaf label is $(q_\ell, \bar{v})$; we then call $\mathcal{D}$ a *coverability witness*.

The reachability problem for lossy ABVASS$_{\bar{0}}$ is then equivalent to top-down coverability for ABVASSr. Observe indeed that the unary, fork, and split rules are *monotone*: if $\bar{v} \leq \bar{w}$ for the product ordering, i.e. if $\bar{v}(i) \leq \bar{w}(i)$ for all $0 < i \leq d$, and a configuration $(q, \bar{v})$ allows to apply a rule and result in some configurations $(q_1, \bar{v}_1)$ and (possibly) $(q_2, \bar{v}_2)$, then $(q, \bar{w})$ allows to apply the same rule and to obtain some $(q_1, \bar{w}_1)$ and $(q_2, \bar{w}_2)$ with $\bar{v}_1 \leq \bar{w}_1$ and $\bar{v}_2 \leq \bar{w}_2$. This means that losses in an ABVASS$_{\bar{0}}$ can be applied as late as possible, either right before a full zero test or at the leaves—which corresponds exactly to top-down coverability for ABVASSr.

3.2.3 Expansive Reachability

In order to model structural contractions during proof search, it is natural to consider another variant of ABVASS$_{\bar{0}}$ called *expansive* ABVASS$_{\bar{0}}$ and equipped with the deduction rules

$$\frac{q, \bar{v} + \bar{e}_i}{q, \bar{v} + 2\bar{e}_i} \text{ expansion}$$

for every q in Q and $0 < i \leq d$. We write '$\triangleright_e$' for root judgements where expansions can occur. This is a restriction over ABVASS$_{\bar{0}}$ since expansions can be emulated through two unary rules $q \xrightarrow{-\bar{e}_i} q' \xrightarrow{2\bar{e}_i} q$. Expansive reachability is not quite dual to lossy rechability—we deal with *increasing reachability* in the full paper.

3.3 Restrictions

Note that ABVASS$_{\bar{0}}$ generalize *vector addition systems with states* (VASS), which are ABVASS$_{\bar{0}}$ with only unary rules. They also generalize two "branching" extensions of VASS, which have been defined in relation with propositional linear logic. Since these restrictions do not feature full zero tests, their lossy reachability problem is equivalent to their top-down coverability problem.

3.3.1 Alternating VASS

Alternating VASS were originally called "and-branching" counter machines by Lincoln et al. [14], and were introduced to prove the undecidability of propositional linear logic. Formally, an AVASS is an ABVASS$_{\bar{0}}$ which only features unary and fork rules, i.e. with $T_s = T_z = \emptyset$.

Fact 2 (Lincoln et al. [14])**.** *Reachability in AVASS is undecidable.*

Proof Idea. By a reduction from the halting problem in Minsky machines: note that a zero test $q \xrightarrow{c \stackrel{?}{=} 0} q'$ on a counter c can be emulated through a fork $q \to q' \wedge q_c$, where unary rules $q_c \xrightarrow{-\bar{e}_{c'}} q_c$ for all $c' \neq c$ allow to empty the counters different from c, and a last unary rule $q_c \xrightarrow{\bar{0}} q_\ell$ to the single target state allows to check that c was indeed equal to zero. ☐

Alternating VASS do not allow to model LL proof search in full; Kanovich [9] identified the matching LL fragment, called the $(!, \oplus)$-Horn fragment.

The complexity of the other basic reachability problems on AVASS is known:

- motivated by the complexity of fragments of relevance logic, Urquhart [23] proved that expansive reachability is complete for Ackermannian time, and
- motivated by the complexity of vector addition games (see the full paper), Courtois and Schmitz [4] showed that lossy reachability is 2-ExpTime-complete.

3.3.2 Branching VASS

Inspired by the correspondences between the !-Horn fragment of linear logic and VASS unearthed by Kanovich [9], de Groote et al. [5] defined BVASS—which they originally dubbed "vector addition tree automata"—as a model of counter machines that matches MELL. Formally, a BVASS is an ABVASS$_{\bar{0}}$ with only unary and split rules, i.e. with $T_f = T_z = \emptyset$. This model turned out to be equivalent to independently defined models in linguistics [19] and protocol verification [25]; see [20] for a survey.

Whether BVASS reachability is decidable is an open problem, and is interreducible with MELL provability. Lazić [13] proved the best known lower bound to this day, which is 2-ExpSpace-hardness. Two related problems were shown to be 2-ExpTime-complete by Demri et al. [6], namely increasing rechability (see the full paper) and boundedness.

3.3.3 Alternating Branching VASS

Kopylov [10] defined a one-player vector game, which matches essentially the reachability problem in ABVASS, i.e. in ABVASS$_{\bar{0}}$ with $T_z = \emptyset$. The *elementary* fragment of ILL defined by Larchey-Wendling and Galmiche [12] is another counterpart to ABVASS.

While allowing full zero tests is helpful in the reduction from LL provability, they can be dispensed with at little expense. Let us first introduce some notation. If node n is an ancestor of a node n' in a deduction tree $\mathcal{D}$, and the labels of n and n' are the same, we write $\mathcal{D}[n \leftarrow n']$ for the *shortening* of $\mathcal{D}$ obtained by replacing the subtree of rule applications rooted at n by the one rooted at n'. Observe that, if $\mathcal{D}$ is a reachability witness (resp. a coverability witness), then $\mathcal{D}[n \leftarrow n']$ is also a reachability witness (resp. a coverability witness).

Lemma 3. *There is a logarithmic-space reduction from (lossy, resp. expansive) ABVASS$_{\bar{0}}$ reachability to (lossy, resp. expansive) ABVASS reachability, and a polynomial time Turing reduction that preserves the system dimension.*

Proof. Suppose $\mathcal{A}$ is an ABVASS$_{\bar{0}}$ with set of states Q and dimension d.

For a logarithmic-space reduction, the key observation is that, if there exists a witness for an instance of (lossy, resp. expansive) reachability for $\mathcal{A}$, then by repeated shortenings, there must be one in which, along every vertical path, the number of occurences of full zero tests is at most $|Q| - 1$.

It therefore suffices to decide the problem for an ABVASS $\mathcal{A}^\dagger$ whose set of states is $\{1, \ldots, |Q|\} \times Q$, whose dimension is $|Q| \cdot d$, and which simulates $\mathcal{A}$ up to $|Q| - 1$ full zero tests along any vertical path. In any state (i, q), $\mathcal{A}^\dagger$ behaves like $\mathcal{A}$ in state q, but using the ith d-tuple of its vector components. To simulate a full zero test $q \xrightarrow{\stackrel{?}{=} \bar{0}} q'$ in $\mathcal{A}$, $\mathcal{A}^\dagger$ changes state from (i, q) to $(i + 1, q)$, postponing the check that the ith d-tuple of vector components are zero until the leaves.

For a reduction that preserves d, we define the set of *root states* relative to a subset X of Q by

$$\mathrm{Root}_{\mathcal{A}}(X) \stackrel{\text{def}}{=} \{q \in Q \mid \mathcal{A}, X \triangleright q, \bar{0}\} \tag{1}$$

as the set of states q such that there exists a deduction in $\mathcal{A}$ with root label $(q, \bar{0})$ and leaf labels in $X \times \{\bar{0}\}$. The (lossy, resp. expansive) reachability problem for $\langle \mathcal{A}, q_r, Q_\ell \rangle$ then reduces to checking whether q_r belongs to $\mathrm{Root}_{\mathcal{A}}(Q_\ell)$.

Writing $\mathcal{A}'$ for the corresponding ABVASS, we can compute $\mathrm{Root}_{\mathcal{A}'}(X)$ using $|Q|$ calls to an oracle for (lossy, resp. expansive) ABVASS reachability. Moreover, since $\mathrm{Root}_{\mathcal{A}'}(X) \supseteq X$ is monotone, we can use a least fixed point computation that discovers root states according to the number of full zero tests along the branches of their reachability witnesses:

$$\mathrm{Root}_{\mathcal{A}}(Q_\ell) = \mu X . \mathrm{Root}_{\mathcal{A}'}(Q_\ell) \cup \mathrm{Root}_{\mathcal{A}'}(X \cup T_z^{-1}(X)) \,. \tag{2}$$

This computation converges after at most $|Q|$ steps, and therefore works in polynomial time relative to the same oracle. ☐

3.4 Computational Complexity

3.4.1 Non-Elementary Complexity Classes

We will use in this paper two complexity classes [see 21]:

$$\textsc{Tower} \stackrel{\text{def}}{=} \bigcup_{e \in \textsc{FElem}} \textsc{DTime}\big(\mathrm{tower}(e(n))\big) \tag{3}$$

is the class of problems that can be solved with a deterministic Turing machine in time tower of some elementary function e of the input, where $\mathrm{tower}(0) \stackrel{\text{def}}{=} 1$ and $\mathrm{tower}(n+1) \stackrel{\text{def}}{=} 2^{\mathrm{tower}(n)}$ defines towers of exponentials. Similarly,

$$\textsc{Ackermann} \stackrel{\text{def}}{=} \bigcup_{p \in \textsc{FPR}} \textsc{DTime}\big(\mathrm{Ack}(p(n))\big) \tag{4}$$

is the class of problems solvable in time Ack of some primitive recursive function p of the input size, where "Ack" denotes the Ackermann function—any standard definition of Ack yields the same complexity class [21].

Completeness for TOWER is understood relative to many-one elementary reductions, and completeness for ACKERMANN relative to many-one primitive-recursive reductions.

3.4.2 ABVASS$_{\bar{0}}$ Complexity

For a set T_u of unary rules, we write $\max^-(T_u)$ (resp. $\max^+(T_u)$) for the largest absolute value of any negative (resp. positive) integer in a vector in T_u, and $\max(T_u)$ for their overall maximum. We assume a binary encoding of the vectors in unary rules, thus $\max(T_u)$ might be exponential in the size of the ABVASS$_{\bar{0}}$. We can however reduce to *ordinary* ABVASS$_{\bar{0}}$, i.e. ABVASS$_{\bar{0}}$ with $\bar{u} = \bar{e}_i$ or $\bar{u} = -\bar{e}_i$ for some $0 < i \leq d$ whenever $q \xrightarrow{\bar{u}} q_1$ is a unary rule:

Lemma 4. *There is a logarithmic space reduction from reachability in (lossy, resp. expansive) ABVASS$_{\bar{0}}$ to reachability in (lossy, resp. expansive) ordinary ABVASS$_{\bar{0}}$.*

Proof Idea. The idea is to encode each of the d coordinates of the original ABVASS$_{\bar{0}}$ into $\lfloor\log(\max(T_u)+1)\rfloor$ coordinates, and each unary rule to apply a binary encoding of $\bar{u}$ to those new coordinates; see for instance [20] where this construction is detailed for BVASS. The expansive case requires to first explicitly encode expansions as unary rules. □

Lossy Case One of the main results of this paper is the following:

Theorem 5. *Reachability in lossy BVASS and lossy ABVASS$_{\bar{0}}$ is TOWER-complete.*

Proof. The upper bound is proved in Section 5. We present the hardness proof in detail in Section 6. □

Note that Theorem 5 entails an improvement for BVASS reachability over the 2-EXPSPACE lower bound of Lazić [13].

Expansive Case Regarding expansive ABVASS$_{\bar{0}}$, we can adapt the proofs of Urquhart [23] for expansive AVASS and the relevance calculus LR+ to show:

Theorem 6. *Reachability in expansive AVASS and expansive ABVASS$_{\bar{0}}$ is ACKERMANN-complete.*

Proof. The lower bound is due to Urquhart [23], who proved hardness of expansive AVASS reachability by a direct reduction from the halting problem of Minsky machines with counter values bounded by the Ackermann function. The upper bound can be proved following essentially the same arguments as Urquhart's for LR+, using length function theorems for Dickson's Lemma [see e.g. 8]. See the full paper for a proof. □

Theorem 6 allows to derive the same ACKERMANN bounds for provability in MALLC and LLC, see the full paper.

4. Relationships Between LL and ABVASS$_{\bar{0}}$

4.1 From LL to ABVASS$_{\bar{0}}$

We present here a direct reduction from LL provability to ABVASS$_{\bar{0}}$ reachability, which relies on the subformula property of the sequent calculus. Consider for this a formula F of linear logic; we know that, if $\vdash F$ is provable, then it has a cut-free proof tree, where all the nodes are labeled by multisets of subformulæ of F. More precisely, a sequent $\vdash \Gamma$ appearing in such a proof can be written as $\vdash ?\Psi, \Delta$ where the formulæ in Δ are *not* guarded by why-nots;

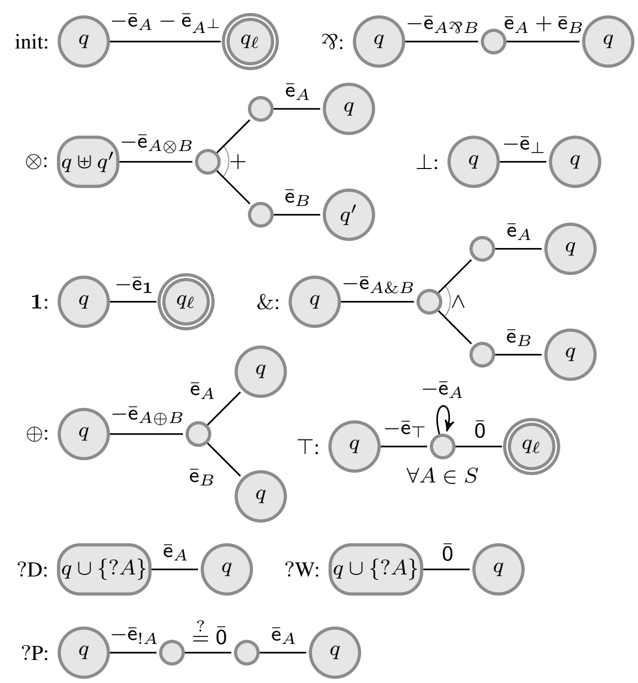

Figure 2. The rules of $\mathcal{A}_F$; q and q' are subsets of $S_?$.

writing $S_?$ for the ?-guarded subformulæ of F and S for its remaining subformulæ, it means that $?\Psi$ is a multiset over $S_?$ and Δ a multiset over S. Let us denote by "$\vdash_F$" the provability relation restricted to subformulæ of F; then $\vdash F$ if and only if $\vdash_F F$.

We define an ABVASS$_{\bar{0}}$ $\mathcal{A}_F$ that maintains an encoding of a sequent $\vdash_F ?\Psi, \Delta$ as the configuration $(\sigma(?\Psi), \Delta)$ over $2^{S_?} \times \mathbb{N}^S$ where, for any multiset m over some set E,

$$\sigma(m) \stackrel{\text{def}}{=} \{e \in E \mid m(e) > 0\} \tag{5}$$

denotes the *support* of the multiset. The ABVASS$_{\bar{0}}$ $\mathcal{A}_F$ includes $2^{S_?}$, a distinguished root state q_r, and a distinguished leaf state q_ℓ as part of its state space; it works in dimension $|S|$ and its rules and intermediate states are depicted in Figure 2. It encodes the sequent calculus of Section 2.1.2 in a straightforward way; the rules maintain the following invariant, which can be checked by induction on the height of deduction trees and proof trees:

$$\mathcal{A}_F, \{q_\ell\} \triangleright \sigma(?\Psi), \Delta \quad \text{iff} \quad \vdash_F ?\Psi, \Delta \ . \tag{6}$$

Two cases arise at the root of deductions in $\mathcal{A}_F$: either $F = ?F'$, and we add a rule $q_r \xrightarrow{\bar{0}} \{?F'\}$ to the rules depicted in Figure 2, or F is not guarded by a why-not, and we add a rule $q_r \xrightarrow{\bar{e}_F} \emptyset$. Then, by (6), $\mathcal{A}_F, \{q_\ell\} \triangleright q_r, \bar{0}$ if and only if $\vdash_F F$.

It is worth noting that the logical contraction rule (?C) is handled implicitly by the use of supports, and that, for each $q \uplus q' \subseteq S_?$, $\mathcal{A}_F$ features a single split rule, for (⊗), a single fork rule, for (&), and a single full zero test rule, for (?P). This means in particular that MELL provability can be reduced to BVASS$_{\bar{0}}$ reachability—and thus by the proof of Lemma 3, to BVASS reachability. Also observe that structural weakening (W) and structural contraction (C) can be handled respectively by losses and expansions in $\mathcal{A}_F$. We conclude:

Proposition 7. *There are polynomial space reductions:*

1. *from (affine, resp. contractive) LL provability to (lossy, resp. expansive) ABVASS$_{\bar{0}}$ reachability,*

2. *from (affine, resp. contractive) MELL provability to (lossy, resp. expansive) BVASS$_{\bar{0}}$ reachability.*

Our reductions incur an exponential blow-up in the number of states—however, as we will see with our complexity upper bounds, this is not an issue, because the main source of complexity in ABVASS$_{\bar{0}}$ is, by far, the dimension of the system, which is here linear in $|F|$. We provide similar reductions for the intuitionistic cases in the full paper, where the proof for an intuitionistic version of (6) is also provided in greater detail.

4.2 From ABVASS$_{\bar{0}}$ to LL

In order to exhibit a reduction from ABVASS$_{\bar{0}}$ reachability to LL provability, we extend a similar reduction proved by Lincoln, Mitchell, Scedrov, and Shankar [14] in the case of AVASS (also employed by Urquhart [23]). The general idea is to encode ABVASS$_{\bar{0}}$ configurations as sequents and ABVASS$_{\bar{0}}$ deductions as proofs in LL extended with a *theory*, where encoded ABVASS$_{\bar{0}}$ rules are provided as an additional set of non-logical axioms.

4.2.1 Linear Logic with a Theory

In the framework of Lincoln et al., a theory T is a finite set of *axioms* $C, p_1^\perp, \ldots, p_m^\perp$ where C is a MALL formula and each p_i is an atomic proposition. Proofs in LL+T can employ two new rules

$$\frac{}{\vdash C, p_1^\perp, \ldots, p_m^\perp}\, T \qquad \frac{\vdash C, p_1^\perp, \ldots, p_m^\perp \quad \vdash C^\perp, \Delta}{\vdash p_1^\perp, \ldots, p_m^\perp, \Delta}\text{ directed cut}$$

where $C, p_1^\perp, \ldots, p_m^\perp$ belongs to T.

A proof in LL+T is directed if all its cuts are *directed cuts*. By adapting the LL cut-elimination proof, Lincoln et al. show:

Fact 8 ([14])**.** *If there is a proof of $\vdash \Gamma$ in LL+T, then there is a directed proof of $\vdash \Gamma$ in LL+T.*

The axioms of a theory T can be translated in pure LL by

$$\ulcorner C, p_1^\perp, \ldots, p_m^\perp \urcorner \stackrel{\text{def}}{=} C^\perp \otimes p_1 \otimes \cdots \otimes p_m\ . \tag{7}$$

Fact 9 ([14])**.** *For any finite set of axioms T, $\vdash \Gamma$ is provable in LL+T if and only if $\vdash ?\ulcorner T \urcorner, \Gamma$ is provable in LL.*

4.2.2 Encoding ABVASS$_{\bar{0}}$

Given an ABVASS $\mathcal{A} = \langle Q, d, T_u, T_f, T_s, \emptyset\rangle$, a configuration $(q, \bar{v})$ in $Q \times \mathbb{N}^d$ is encoded as the sequent

$$\theta(q, \bar{v}) \stackrel{\text{def}}{=} \vdash q^\perp, (e_1^\perp)^{\bar{v}(1)}, \ldots, (e_d^\perp)^{\bar{v}(d)} \tag{8}$$

where $Q \uplus \{e_i \mid i = 1, \ldots, d\}$ is included in the set of atomic propositions and A^n stands for the formula A repeated n times.

By Lemma 4 we assume $\mathcal{A}$ to be in ordinary form. We construct from the rules of $\mathcal{A}$ a theory T consisting of sequents of form $\vdash q^\perp, c_1^\perp, \ldots, c_m^\perp, C$ with q in Q the originating state, c_j in $\{e_i \mid 0 < i \leq d\}$, and C a MALL formula containing the destination state(s) positively. Here are the axioms corresponding to each type of rule:

$q \xrightarrow{\bar{e}_i} q_1$	$q^\perp, q_1 \otimes e_i$
$q \xrightarrow{-\bar{e}_i} q_1$	$q^\perp, e_i^\perp, q_1$
$q \to q_1 \land q_2$	$q^\perp, q_1 \oplus q_2$
$q \to q_1 + q_2$	$q^\perp, q_1 \parr q_2$

By Lemma 3, we do not need to consider the case of full zero tests. Here is nevertheless how they could be encoded, provided we slightly extended the reduction of LL+T to LL in Fact 9 to allow exponentials in T:

$q \xrightarrow{\stackrel{?}{=}\bar{0}} q_1$	$q^\perp, !q_1$

Claim 10.1. For all $(q, \bar{v})$ in $Q \times \mathbb{N}^d$, $\mathcal{A}, Q_\ell \rhd q, \bar{v}$ if and only if $\vdash \theta(q, \bar{v}), ?Q_\ell$ in LL+T.

Proof. The AVASS case is proved by Lincoln et al. [14, Lemmata 3.5 and 3.6] by induction on the height of deduction trees in $\mathcal{A}$ and the number of directed cuts in a directed proof in LL+T (with minor adaptations for $?Q_\ell$). Thus, we only need to prove that split rules preserve this statement.[2]

Assume for the direct implication that $\mathcal{A}, Q_\ell \rhd q, \bar{v}$ as the result of a split rule $q \to q_1 + q_2$, thus $\bar{v} = \bar{v}_1 + \bar{v}_2$ and $\mathcal{A}, Q_\ell \rhd q_1, \bar{v}_1$ and $\mathcal{A}, Q_\ell \rhd q_2, \bar{v}_2$. By induction hypothesis, $\vdash \theta(q_1, \bar{v}_1), ?Q_\ell$ and $\vdash \theta(q_2, \bar{v}_2), ?Q_\ell$, and we can prove

$$\vdash q_1^\perp \otimes q_2^\perp, (c_1^\perp)^{\bar{v}_1(1)+\bar{v}_2(1)}, \ldots, (c_d^\perp)^{\bar{v}_1(d)+\bar{v}_2(d)}, ?Q_\ell, ?Q_\ell \tag{9}$$

using ($\otimes$), and after $|Q_\ell|$ logical contractions and a directed cut with $\vdash q^\perp, q_1 \parr q_2$, we obtain $\vdash \theta(q, \bar{v}), ?Q_\ell$ as desired.

Conversely, assume that the last applied directed cut has

$$\vdash q_1^\perp \otimes q_2^\perp, (c_1^\perp)^{\bar{v}(1)}, \ldots, (c_d^\perp)^{\bar{v}(d)}, ?Q_\ell \tag{10}$$

and $\vdash q^\perp, q_1 \parr q_2$ as premises. The only rules that allow to prove (10) are (?D), (?C) or (?W) applied to some q_ℓ in Q_ℓ, and ($\otimes$). Logical contractions are irrelevant to the claim, and wlog. we can apply derelictions above ($\otimes$), thus we know that (10) is the result of ($\otimes$) followed by a series of (?W). Hence $\vdash \theta(q_1, \bar{v}_1), ?Q_1$ and $\vdash \theta(q_2, \bar{v}_2), ?Q_2$ with $\bar{v} = \bar{v}_1 + \bar{v}_2$ and $Q_\ell \supseteq Q_1 \cup Q_2$. By induction hypothesis, $\mathcal{A}, Q_1 \rhd q_1, \bar{v}_1$ and $\mathcal{A}, Q_2 \rhd q_2, \bar{v}_2$. Because $Q_1 \subseteq Q_\ell$ and $Q_2 \subseteq Q_\ell$ this entails $\mathcal{A}, Q_\ell \rhd q_1, \bar{v}_1$ and $\mathcal{A}, Q_\ell \rhd q_2, \bar{v}_2$, from which a split allows to derive $\mathcal{A}, Q_\ell \rhd q, \bar{v}$ as desired. □

Proposition 10. *There are logarithmic space reductions*

1. *from ABVASS$_{\bar{0}}$ reachability to LL provability and*
2. *from BVASS$_{\bar{0}}$ reachability to MELL provability.*

Proof. By Lemma 3 we can eliminate full zero tests. For 1, by Claim 10.1 and Fact 9, $\mathcal{A}, Q_\ell \rhd q_r, \bar{0}$ if and only if $\vdash q_r^\perp, ?Q_\ell, ?\ulcorner T \urcorner$. Regarding 2, simply observe that additive connectives are only used for the encoding of fork rules. □

4.2.3 Affine Case

Adapting the proof of Proposition 10 to the affine case is relatively straightforward. For starters, Fact 8 also holds for LLW+T using the cut elimination procedure for LLW, and allowing structural weakenings does not influence the proof of Fact 9 in [14, Lemmata 3.2 and 3.3]. We show:

Proposition 11. *There are logarithmic space reductions*

1. *from ABVASS$_{\bar{0}}$ lossy reachability to LLW provability and*
2. *from BVASS$_{\bar{0}}$ lossy reachability to MELLW provability.*

This relies on an extension of Claim 10.1:

Claim 11.1. For all $(q, \bar{v})$ in $Q \times \mathbb{N}^d$, $\mathcal{A}, Q_\ell \rhd_\ell q, \bar{v}$ with lossy semantics if and only if $\vdash \theta(q, \bar{v}), ?Q_\ell$ in LLW+T.

4.2.4 Contractive Case

Again, Fact 8 is straightforward to adapt to LLC+T using cut elimination. Fact 9 can be strengthened to avoid exponentials in the contractive case; see the full paper for a proof:

Lemma 12. *For a finite set of axioms T, $\vdash \Gamma$ is provable in LLC+T if and only if $\vdash \top \oplus \bigoplus_{t \in T} \ulcorner t \urcorner, \Gamma$ is provable in LLC.*

[2] de Groote et al. [5] show how to handle split rules in IMELL, but they do not rely on the LL+T framework, which motivates considering this case.

Proof. For the direct implication, we consider a directed proof of $\vdash_{\mathrm{LLC}+T} \Gamma$. By induction on the number of directed cuts, we build an LLC proof of $\vdash_{\mathrm{LLC}} \top \oplus \bigoplus_{t\in T} \ulcorner t \urcorner, \Gamma$. For the base case, an LLC+T proof without directed cuts is also an LLC proof, thus $\vdash_{\mathrm{LLC}+T} \top, \Gamma$ using the ($\top$) rule, and $\vdash \top \oplus \bigoplus_{t\in T} \ulcorner t \urcorner, \Gamma$ by $|T|$ applications of ($\oplus$). For the induction step, consider a directed cut of an axiom $t = C, p_1^\perp, \ldots, p_m^\perp$ in T with $\vdash_{\mathrm{LLC}+T} C^\perp, \Delta$. We have $\vdash_{\mathrm{LLC}} C, C^\perp$ and $\vdash_{\mathrm{LLC}} p_i, p_i^\perp$ for all $0 < i \leq m$ by the (init) rule, and $m + 1$ applications of ($\otimes$) yield $\vdash_{\mathrm{LLC}} t, \ulcorner t \urcorner$. By induction hypothesis $\vdash_{\mathrm{LLC}} \top \oplus \bigoplus_{t\in T} \ulcorner t \urcorner, C^\perp, \Delta$, thus a (normal) cut yields $\vdash_{\mathrm{LLC}} \top \oplus \bigoplus_{t\in T} \ulcorner t \urcorner, \ulcorner t \urcorner, p_1^\perp, \ldots, p_m^\perp, \Delta$. Using $|T|$ applications of ($\oplus$) allows to prove $\vdash_{\mathrm{LLC}} \top \oplus \bigoplus_{t\in T} \ulcorner t \urcorner, \top \oplus \bigoplus_{t\in T} \ulcorner t \urcorner, p_1^\perp, \ldots, p_m^\perp, \Delta$ and a structural contraction yields the desired LLC proof.

For the converse implication, if $\vdash_{\mathrm{LLC}} \top \oplus \bigoplus_{t\in T} \ulcorner t \urcorner, \Gamma$, then $\vdash_{\mathrm{LLC}+T} \top \oplus \bigoplus_{t\in T} \ulcorner t \urcorner, \Gamma$. Then $\vdash_{\mathrm{LLC}+T} \mathbf{1}$, and for each axiom $t = C, p_1^\perp, \ldots, p_m^\perp$ in T, we can prove $\vdash_{\mathrm{LLC}+T} C \parr p_1^\perp \parr \cdots \parr p_m^\perp$ by m applications of ($\parr$) from $\vdash_{\mathrm{LLC}+T} t$, i.e. $\vdash_{\mathrm{LLC}+T} \ulcorner t \urcorner^\perp$. Thus $|T|$ applications of ($\&$) yield $\vdash_{\mathrm{LLC}+T} \mathbf{1} \,\&\, \bigwith_{t\in T} \ulcorner t \urcorner^\perp$, and a (normal) cut shows $\vdash_{\mathrm{LLC}+T} \Gamma$. ☐

Without loss of generality, we can assume that $Q_\ell = \{q_\ell\}$ for a state q_ℓ with no applicable rule in $\mathcal{A}$. We extend Claim 10.1 and Proposition 10 to the contractive case; see the full paper:

Claim 13.1. For all $(q, \bar{\mathrm{v}})$ in $Q \times \mathbb{N}^d$, $\mathcal{A}, \{q_\ell\} \rhd_e q, \bar{\mathrm{v}}$ using expansive semantics if and only if $\vdash \theta(q, \bar{\mathrm{v}}), q_\ell^s$ in LLC+T for some $s > 0$.

Proposition 13. *There is a logarithmic space reduction from ABVASS$_{\bar{0}}$ expansive reachability to MALLC provability.*

5. TOWER Upper Bounds

To show that the reachability problem for lossy ABVASS$_{\bar{0}}$ is in TOWER, we establish by induction over the dimension d a bound on the height of minimal reachability witnesses, following in this the reasoning used by Rackoff [18] to show that the coverability problem for VASS is in EXPSPACE. The main new idea here is that, where there is freedom to choose how values of vector components are distributed when performing split rules top-down (see Section 3.1), splitting them equally (or with the difference of 1) allows sufficient lower bounds to be established along vertical paths in deduction trees for the inductive argument to go through. Since the bounds we obtain on the heights of smallest witnessing deduction trees are exponentiated at every inductive step (rather than multiplied as in Rackoff's proof), the resulting complexity upper bound involves a tower of exponentials, but will be shown broadly optimal in Section 6.

The following lemma in fact addresses the equivalent top-down coverability problem (see Section 3.2.2), and considers systems without full resets thanks to Lemma 3. We first define some terminology. We say that a deduction tree is:

- $(q_r, \bar{\mathrm{v}}_0)$-*rooted* iff that is the label of its root;
- Q_ℓ-*leaf-covering* iff, for every leaf label $(q, \bar{\mathrm{v}})$, we have $q \in Q_\ell$;
- of *height* h iff that is the maximum number of edges, i.e. the maximum number of rule applications, along any path from the root to a leaf.

For integers $d, m \geq 0$ and $s \geq 1$, we define a natural number $H(d, s, m)$ recursively:

$$H(0, s, m) \stackrel{\text{def}}{=} s \,, \tag{11}$$

$$H(d + 1, s, m) \stackrel{\text{def}}{=} s(m \cdot 2^{H(d,s,m)})^{d+1} + H(d, s, m) \,. \tag{12}$$

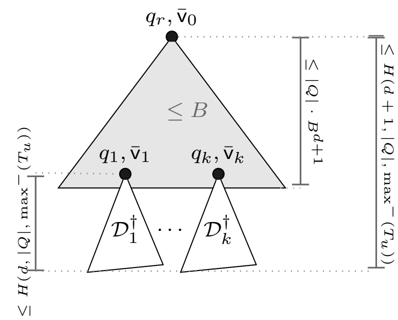

Figure 3. Induction step in the proof of Lemma 14.

Lemma 14. *If an ABVASS $\mathcal{A} = \langle Q, d, T_u, T_f, T_s, \emptyset\rangle$ has a $(q_r, \bar{\mathrm{v}}_0)$-rooted Q_ℓ-leaf-covering deduction tree, then it has such a deduction tree of height at most $H(d, |Q|, \max^-(T_u))$.*

Proof. We use induction on the dimension d.

If $\mathcal{A}$ is 0-dimensional, then the labels in its deduction trees are states only. Starting with a deduction tree whose root label is q_r and whose every leaf label is in Q_ℓ, we obtain by repeated shortenings a deduction tree in which labels along every branch are mutually distinct, with height at most $|Q| - 1$.

Suppose that $\mathcal{A} = \langle Q, d{+}1, T_u, T_f, T_s, \emptyset\rangle$, and $\mathcal{D}$ is a $(q_r, \bar{\mathrm{v}}_0)$-rooted Q_ℓ-leaf-covering deduction tree. Let

$$B \stackrel{\text{def}}{=} 2^{H(d,|Q|,\max^-(T_u))} \cdot \max^-(T_u) \,, \tag{13}$$

and let $\{n_1, \ldots, n_k\}$ be the set of all nodes of $\mathcal{D}$ such that, for all i, we have:

- all vector components in labels of ancestors of n_i are smaller than B;
- for some $0 < j_i \leq d + 1$, we have $\bar{\mathrm{v}}_i(j_i) \geq B$, where the label of n_i is $(q_i, \bar{\mathrm{v}}_i)$.

By repeated shortenings, we can assume that the length (i.e., the number of edges) of every path in $\mathcal{D}$, which is from the root either to some n_i or to a leaf with no n_i ancestor, is at most $|Q| \cdot B^{d+1}$, the number of possible labels with all vector components smaller than B.

In the remainder of the argument, we apply the induction hypothesis below each of the nodes n_i. More precisely, let $\mathcal{A}_i$ denote the d-dimensional ABVASS obtained from $\mathcal{A}$ by projecting onto vector indices $\{1, \ldots, d + 1\} \setminus \{j_i\}$. (The only change is in the set of unary rules.) From the subtree of $\mathcal{D}$ rooted at n_i, we know that $\mathcal{A}_i$ has a $(q_i, \bar{\mathrm{v}}_i(-j_i))$-rooted Q_ℓ-leaf-covering deduction tree. (Here $\bar{\mathrm{w}}(-j)$ is the projection of $\bar{\mathrm{w}}$ to all indices except j.) Let $\mathcal{D}_i$ be such a deduction tree, which we can choose of height at most $H(d, |Q|, \max^-(T_u))$ by induction hypothesis.

Now, to turn $\mathcal{D}_i$ into a $(q_i, \bar{\mathrm{v}}_i)$-rooted deduction tree $\mathcal{D}_i^\dagger$ of $\mathcal{A}$, we have to do two things:

1. For every application of a unary rule $q \xrightarrow{\bar{\mathrm{u}}} q'$ in $\mathcal{D}_i$, decide which unary rule $q \xrightarrow{\bar{\mathrm{u}}'} q'$ of $\mathcal{A}$ such that $\bar{\mathrm{u}} = \bar{\mathrm{u}}'(-j_i)$ to apply: we do that arbitrarily.
2. For every application of a split rule $q \to q' + q''$ in $\mathcal{D}_i$, decide how to split the vector component x with index j_i: we do that by balancing, i.e. picking the corresponding components x_1 and x_2 of the two child vectors so that $|x_1 - x_2| \leq 1$.

We claim that $\mathcal{D}_i^\dagger$ thus obtained is indeed a $(q_i, \bar{\mathrm{v}}_i)$-rooted Q_ℓ-leaf-covering deduction tree of $\mathcal{A}$. Since the node labels in $\mathcal{D}_i^\dagger$ differ from those in $\mathcal{D}_i$ only by the extra j_ith components, it suffices to show that all the latter are non-negative. In fact, at the root of $\mathcal{D}_i^\dagger$, we have $\bar{\mathsf{v}}_i(j_i) \geq B$, and it follows by a straightforward induction that, for every node n in $\mathcal{D}_i^\dagger$ whose distance from the root is h (which is at most $H(d, |Q|, \max^-(T_u))$), its vector label $\bar{\mathsf{w}}$ satisfies

$$\bar{\mathsf{w}}(j_i) \geq 2^{H(d,|Q|,\max^-(T_u))-h} \cdot \max^-(T_u) \; . \qquad (14)$$

It remains to observe that, by replacing for each $0 < i \leq k$, the subtree of $\mathcal{D}$ rooted at n_i by $\mathcal{D}_i^\dagger$, the height of the resulting deduction tree (see Figure 3 for a depiction) is at most

$$|Q| \cdot B^{d+1} + H(d, |Q|, \max^-(T_u)) = H(d+1, |Q|, \max^-(T_u)),$$

thereby establishing the lemma. ☐

The following auxiliary function and proposition will be useful for deriving the complexity upper bounds. Let

$$H'(d, s, m) \stackrel{\text{def}}{=} 4(d+1)(s+m+1)H(d, s, m) \; . \qquad (15)$$

We show in the full paper that:

Proposition 15. *For all $d, m \geq 0$ and $s \geq 1$, we have:*

$$H'(d+1, s, m) \leq 2^{H'(d,s,m)} \; .$$

We are now in a position to establish the membership in TOWER. More precisely, since the height of the tower of exponentials in the bounds we obtained is equal to the system dimension, the problem with the dimension d fixed will be in d-EXPTIME.

Theorem 16. *Reachability for lossy ABVASS$_{\bar{0}}$ is in* TOWER. *For every fixed dimension d, it is in* PTIME *if $d = 0$, and in d-*EXPTIME *if $d \geq 1$.*

Proof. By Lemma 3, it suffices to consider an ABVASS. We argue in terms of the top-down coverability problem (see Section 3.2.2): given an ABVASS $\mathcal{A} = \langle Q, d, T_u, T_f, T_s, \emptyset \rangle$, a state q_r and a set of states Q_ℓ, to decide whether $\mathcal{A}$ has a $(q_r, \bar{0})$-rooted Q_ℓ-leaf-covering deduction tree.

By Lemma 14, if $\mathcal{A}$ has such a deduction tree, then it has one of height at most $H(d, |Q|, \max^-(T_u))$. Observing that, in such a deduction tree, all vector components are bounded by

$$(\max^+(T_u) + 1) \cdot H(d, |Q|, \max^-(T_u)) \; ,$$

we conclude that it can be guessed and checked in

$$O((d+1) \cdot \log((\max^+(T_u) + 1) \cdot H'(d, |Q|, \max^-(T_u))))$$

space by an alternating algorithm which manipulates at most three configurations of $\mathcal{A}$ at a time.

The memberships in the statement (for ABVASS) follow from the fact that $H'(0, |Q|, \max^-(T_u))$ is polynomial, by Proposition 15, and since ALOGSPACE = PTIME, APSPACE = EXPTIME, and $(d-1)$-AEXPSPACE = d-EXPTIME. ☐

By Proposition 7, this shows:

Corollary 17. *LLW provability is in* TOWER.

6. TOWER Lower Bounds

The rough pattern of our hardness proof resembles those by e.g. Urquhart [23], where a fast-growing function is computed weakly, then its result is used to allocate space for simulating a universal machine, and finally the inverse of the function is computed weakly for checking purposes. Indeed, we simulate Minsky machines whose counters are tower-bounded, but the novelty here is in the inverse computations. Specifically, for each Minsky counter c, we maintain its dual $\hat{c}$ and simulate each zero test on c by a split rule that launches a thread to check that $\hat{c}$ has the maximum value.

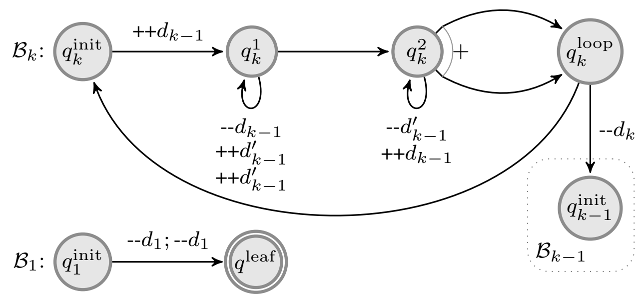

Figure 4. Defining $\mathcal{B}_k$ for $k > 1$ (above), and $\mathcal{B}_1$ (below).

Recalling that such rules split all values non-deterministically, we must construct the simulating system carefully so that such non-determinism cannot result in erroneous behaviours.

The auxiliary threads check that a counter is at least $\text{tower}(k)$ by seeking to apply split rules at least $\text{tower}(k-1)$ times along every branch. The difficulty here is, similarly, how to count up to $\text{tower}(k-1)$ or more in a manner which is robust with respect to the non-determinism of the split rules.

A hierarchy of BVASS for the latter purpose is given in Figure 4— recall the depicting conventions in Section 3.1.3. Thus, after the unary rule from q_k^{loop} that decrements d_k, we have that $\mathcal{B}_k$ behaves like $\mathcal{B}_{k-1}$ from state q_{k-1}^{init}.

Lemma 18. *For every $k \geq 1$ and vector of naturals $\bar{\mathsf{v}}_0$ such that $\bar{\mathsf{v}}_0(d_i) = \bar{\mathsf{v}}_0(d'_i) = 0$ for all $i < k$, we have that $\mathcal{B}_k$ has a $(q_k^{\text{init}}, \bar{\mathsf{v}}_0)$-rooted $\{q^{\text{leaf}}\}$-leaf-covering deduction tree if and only if $\bar{\mathsf{v}}_0(d_k) \geq \text{tower}(k)$.*

Proof. We proceed by induction on k, where the base case $k = 1$ is immediate, so let us consider $k > 1$ and $\bar{\mathsf{v}}_0$ such that $\bar{\mathsf{v}}_0(d_i) = \bar{\mathsf{v}}_0(d'_i) = 0$ for all $i < k$.

If $\bar{\mathsf{v}}_0(d_k) \geq \text{tower}(k)$, we observe that $\mathcal{B}_k$ can proceed from $(q_k^{\text{init}}, \bar{\mathsf{v}}_0)$ as follows:

- each loop at q_k^1 empties d_{k-1}, i.e. doubles d_{k-1} and transfers it to d'_{k-1};
- each loop at q_k^2 empties d'_{k-1}, i.e. transfers d'_{k-1} to d_{k-1};
- each split from q_k^2 divides d_{k-1} into two equal values, and divides d_k into two values that differ by at most 1.

In any deduction tree thus obtained, at every node which is the hth node with state label q_k^{loop} from the root, and whose vector label is $\bar{\mathsf{w}}$, we have:

$$\bar{\mathsf{w}}(d_{k-1}) = h \; , \quad \bar{\mathsf{w}}(d'_{k-1}) = 0 \; , \quad \bar{\mathsf{w}}(d_k) \geq 2^{\text{tower}(k-1)-h} \; . \qquad (16)$$

Hence, by returning control to q_k^{init} as long as the value of d_k is at least 2, $\mathcal{B}_k$ can reach along every vertical path a node with state label q_k^{loop} at which the values of d_{k-1} and d_k are equal to $\text{tower}(k-1)$ and at least 1 (respectively). To complete the deduction tree to be $\{q^{\text{leaf}}\}$-leaf-covering, from every such node we let $\mathcal{B}_k$ decrement d_k and apply the induction hypothesis.

The interesting direction remains, so suppose $\mathcal{D}$ is a $(q_k^{\text{init}}, \bar{\mathsf{v}}_0)$-rooted $\{q^{\text{leaf}}\}$-leaf-covering deduction tree of $\mathcal{B}_k$. Since at every q_k^{loop}-labelled node in $\mathcal{D}$, the value of d_k must be at least 1, it suffices to establish the following claim and apply it for the maximum h:

Claim 18.1. For each $0 < h \leq \text{tower}(k-1)$, $\mathcal{D}$ contains 2^h incomparable nodes (i.e., none is a descendant of another) whose state label is q_k^{loop} and at which $d_{k-1} + d'_{k-1}$ has value at most h.

In turn, by induction on h, that claim is a straightforward consequence of the next one. (For the base case of that induction, i.e. $h = 1$, apply the next claim with $h' = 0$.)

Claim 18.2. For each node n in $\mathcal{D}$ whose state label is q_k^{init} and at which $d_{k-1} + d'_{k-1}$ has some value $h' < \text{tower}(k-1)$, there must be two incomparable descendants n_1 and n_2 whose state labels are q_k^{loop} and at which the values of $d_{k-1} + d'_{k-1}$ are at most $h' + 1$.

Consider a node n as in the latter claim. After the increment of d_{k-1} and the loops at q_k^1 and q_k^2, the value of $d_{k-1} + d'_{k-1}$ will be at most $2(h' + 1)$. If the first split divides $d_{k-1} + d'_{k-1}$ equally, we are done.

Otherwise, we have a q_k^{loop}-labelled descendant n' of n at which $d_{k-1} + d'_{k-1}$ has value at most h'. In particular, d_{k-1} is less than $\text{tower}(k-1)$ at n', so recalling the induction hypothesis regarding $\mathcal{B}_{k-1}$, the child n'' of n' cannot be q_{k-1}^{init}-labelled. Thus, n'' must be q_k^{init}-labelled, and the value of $d_{k-1} + d'_{k-1}$ at n'' is the same as at n', so at most h'. We can therefore repeat the argument with n'' instead of n, but since $\mathcal{D}$ is finite, two incomparable descendants as required eventually exist. □

Relying on the properties of the BVASS $\mathcal{B}_k$, we now establish the hardness of lossy reachability, matching the membership in TOWER in Theorem 16 already for BVASS. Although we do not match the upper bounds when the system dimension is fixed, we remark that our simulation uses a number of counters which is linear in the height of the tower of exponentials with coefficient 2.

Theorem 19. *Reachability for lossy BVASS is* TOWER*-hard.*

Proof. For a notion of Minsky machines that is similar to how $\text{ABVASS}_{\bar{0}}$ were defined in Section 3.1, let such a machine be given by a finite set of states Q, a finite set of counters C, and finite sets of increment rules "$q \xrightarrow{++c} q_1$," decrement rules "$q \xrightarrow{--c} q_1$" and zero-test rules "$q \xrightarrow{c \stackrel{?}{=} 0} q_1$." By simulating a tape using two stacks, and simulating a stack using two counters, it is straightforward to verify that the following problem is TOWER-hard:

> Given a Minsky machine $\mathcal{M}$ and two states q_0, q_H, does $\mathcal{M}$ have a computation that starts in q_0 with all counters having value 0, ends in q_H, and is such that all counter values are at most $\text{tower}(|\mathcal{M}|)$?

We establish the theorem by working with the equivalent top-down coverability problem (see Section 3.2.2). We show that, given a Minsky machine $\mathcal{M}$ of size K and two states q_0, q_H, then a BVASS $\mathcal{A}(\mathcal{M})$, a state q_r and a finite set $Q_\ell \stackrel{\text{def}}{=} \{q_H, q^{\text{leaf}}\}$ are computable in logarithmic space, such that $\mathcal{M}$ has a 0-initialised $\text{tower}(K)$-bounded computation from q_0 to q_H if and only if $\mathcal{A}(\mathcal{M})$ has a $(q_r, \bar{0})$-rooted Q_ℓ-leaf-covering deduction tree.

For each counter c of $\mathcal{M}$, there are three counters in $\mathcal{A}(\mathcal{M})$ denoted $c, \hat{c}, c'$. The initial part of $\mathcal{A}(\mathcal{M})$ employs a "weak Petri net computer" [16] for the tower function, namely a constant VASS with a designated start state, input counter, finish state and output counter, which given a natural number m, can compute $\text{tower}(m)$ but non-deterministically may also compute a smaller value. (It is standard to construct such a VASS from weak routines for $2m$ and 2^m.) By means of the latter VASS, each counter $\hat{c}$ in $\mathcal{A}(\mathcal{M})$ is initialised to have value $\text{tower}(K)$ (or possibly smaller). Recalling that the auxiliary VASS is constant, a simple pattern for incorporating it into $\mathcal{A}(\mathcal{M})$ is to use fresh states and counters for each $\hat{c}$.

The main part of $\mathcal{A}(\mathcal{M})$ consists of simulating $\mathcal{M}$ from q_0, using the translations of increments, decrements and zero tests in Figure 5. For the increments and decrements, $\mathcal{A}(\mathcal{M})$ also performs the opposite operation on the hatted counter, thereby keeping the sums $c+\hat{c}$ constant. For the zero tests, $\mathcal{A}(\mathcal{M})$ attempts by two loops

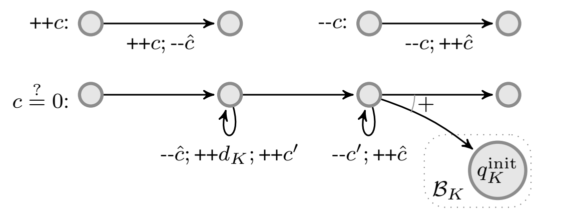

Figure 5. Simulating the Minsky operations.

and using the primed counter, to copy the hatted counter to d_K and then employ $\mathcal{B}_K$ (see Figure 4) to verify that the latter is maximal (i.e., has value $\text{tower}(K)$). Thus, $\mathcal{A}(\mathcal{M})$ also has counters d_i for $0 < i \leq K$ and d'_i for $0 < i < K$, and more precisely a variant of $\mathcal{B}_K$ is employed that has the same dimension as $\mathcal{A}(\mathcal{M})$ (and does not use the extra counters).

For each 0-initialised $\text{tower}(K)$-bounded computation of $\mathcal{M}$ from q_0 to q_H, it is straightforward to check that $\mathcal{A}(\mathcal{M})$ can simulate it as follows:

- each counter $\hat{c}$ is initialised to $\text{tower}(K)$;
- in every simulation of a zero test $c \stackrel{?}{=} 0$, the values of $c, \hat{c}, c', d_K$ are resp. $0, \text{tower}(K), 0, 0$ before the two loops, and $0, \text{tower}(K), 0, \text{tower}(K)$ before the split;
- at every start of $\mathcal{B}_K$, the value of d_K is $\text{tower}(K)$ and all other counters have value 0.

By Lemma 18, we obtain a $(q_r, \bar{0})$-rooted Q_ℓ-leaf-covering deduction tree of $\mathcal{A}(\mathcal{M})$.

The other direction is more involved: we show that, if $\mathcal{A}(\mathcal{M})$ has a $(q_r, \bar{0})$-rooted Q_ℓ-leaf-covering deduction tree $\mathcal{D}$, then $\mathcal{M}$ has a 0-initialised $\text{tower}(K)$-bounded computation from q_0 to q_H. By construction, $\mathcal{D}$ consists of a path π from which there are branchings to deduction trees of $\mathcal{B}_K$. The main part of π consists of the simulations of increments, decrements and zero tests as in Figure 5. From it, we obtain a 0-initialised $\text{tower}(K)$-bounded computation of $\mathcal{M}$ from q_0 to q_H, after observing the following for every counter c of $\mathcal{M}$:

- After $\hat{c}$ is initialised in $\mathcal{D}$, the value of $c + \hat{c} + c'$ is always at most $\text{tower}(K)$.
- For each simulation of a zero test of c, we have by Lemma 18 that the value of d_K is $\text{tower}(K)$ before the split and is 0 after the split on the path π, and consequently that the values of $c, \hat{c}, c'$ are $0, \text{tower}(K), 0$ (respectively) before the two loops.
- The value of c may erroneously decrease due to the branchings, but since that makes the value of $c + \hat{c} + c'$ smaller than $\text{tower}(K)$, such losses may occur only after the last simulation of a zero test of c, and so cannot result in an erroneous such simulation.
- Similarly, only the last transfer of c' to $\hat{c}$ may be incomplete (i.e., it does not empty c'). □

Since lossy reachability reduces to reachability and by Proposition 10 and Proposition 11, this entails:

Corollary 20. *Provability in MELL, MELLW, and LLW is* TOWER*-hard.*

7. Concluding Remarks

Although connections between propositional linear logic and families of counter machines have long been known, they have rarely been exploited for complexity-theoretic results. Using a model of

	MELL	LL
	TOWER-hard, Σ^0_1-easy	Σ^0_1-c. [14]
with W	TOWER-c.	TOWER-c.
with C	2EXP-c. [22]	ACK-c.

Table 1. The complexity of provability in fragments and variants of LL.

	AVASS	BVASS	$\text{ABVASS}_{\bar{0}}$
Reachability	Σ^0_1-c. [14]	TOWER-hard, Σ^0_1-easy	Σ^0_1-c.
Lossy reach.	2EXP-c. [4]	TOWER-c.	TOWER-c.
Incr. reach.	ACK-c. [23]	2EXP-c. [6]	ACK-c.

Table 2. The complexity of reachability problems in $\text{ABVASS}_{\bar{0}}$.

alternating branching VASS, we have unified several of these connections, and derived complexity bounds for provability in substructural logics from the (old and new) bounds on $\text{ABVASS}_{\bar{0}}$ reachability, summarized in Table 1 and Table 2 respectively.

Our main results in this regard are the TOWER-completeness of provability in LLW and the new TOWER lower bound for MELL: the latter has consequences on numerous problems mentioned in Section 3, and entails for instance that the satisfiability problem for FO^2 on data trees is non-elementary [2, 7]. The ACKERMANN-completeness of MALLC and LLC is perhaps less surprising in the light of Urquhart's results, but we take it as a testimony of the versatility of the $\text{ABVASS}_{\bar{0}}$ model.

The main open question remains whether BVASS reachability, or equivalently MELL provability, is decidable.

References

[1] A. Asperti and L. Roversi. Intuitionistic light affine logic. *ACM Trans. Comput. Logic*, 3(1):137–175, 2002.

[2] M. Bojańczyk, A. Muscholl, T. Schwentick, and L. Segoufin. Two-variable logic on data trees and XML reasoning. *J. ACM*, 56(3):1–48, 2009. ISSN 0004-5411.

[3] A. Bouajjani and M. Emmi. Analysis of recursively parallel programs. *ACM Trans. Prog. Lang. Syst.*, 35(3):10:1–10:49, 2013.

[4] J.-B. Courtois and S. Schmitz. Alternating vector addition systems with states. Manuscript, 2014. URL http://hal.inria.fr/hal-00980878.

[5] Ph. de Groote, B. Guillaume, and S. Salvati. Vector addition tree automata. In *LICS 2004*, pages 64–73. IEEE Computer Society, 2004. ISBN 0-7695-2192-4.

[6] S. Demri, M. Jurdziński, O. Lachish, and R. Lazić. The covering and boundedness problems for branching vector addition systems. *J. Comput. Syst. Sci.*, 79(1):23–38, 2013.

[7] J. Dimino, F. Jacquemard, and L. Segoufin. $\text{FO}^2(<, +1, \sim)$ on data trees, data tree automata and an extension of BVASS. Manuscript, 2013. URL http://hal.inria.fr/hal-00769249.

[8] D. Figueira, S. Figueira, S. Schmitz, and Ph. Schnoebelen. Ackermannian and primitive-recursive bounds with Dickson's Lemma. In *LICS 2011*, pages 269–278. IEEE Computer Society, 2011.

[9] M. I. Kanovich. Petri nets, Horn programs, linear logic and vector games. *Ann. Pure App. Logic*, 75(1–2):107–135, 1995.

[10] A. P. Kopylov. Decidability of linear affine logic. *Inform. and Comput.*, 164(1):173–198, 2001.

[11] Y. Lafont. The finite model property for various fragments of linear logic. *J. Symb. Log.*, 62(4):1202–1208, 1997.

[12] D. Larchey-Wendling and D. Galmiche. Nondeterministic phase semantics and the undecidability of Boolean BI. *ACM Trans. Comput. Logic*, 14(1):6:1–6:41, 2013.

[13] R. Lazić. The reachability problem for branching vector addition systems requires doubly-exponential space. *Inf. Process. Lett.*, 110 (17):740–745, 2010.

[14] P. Lincoln, J. Mitchell, A. Scedrov, and N. Shankar. Decision problems for propositional linear logic. *Ann. Pure App. Logic*, 56(1–3):239–311, 1992.

[15] R. Lipton. The reachability problem requires exponential space. Technical Report 62, Yale University, 1976.

[16] E. W. Mayr and A. R. Meyer. The complexity of the finite containment problem for Petri nets. *J. ACM*, 28(3):561–576, 1981.

[17] M. Okada and K. Terui. The finite model property for various fragments of intuitionistic linear logic. *J. Symb. Log.*, 64(2):790–802, 1999.

[18] C. Rackoff. The covering and boundedness problems for vector addition systems. *Theor. Comput. Sci.*, 6(2):223–231, 1978.

[19] O. Rambow. Multiset-valued linear index grammars: imposing dominance constraints on derivations. In *ACL'94*, pages 263–270. ACL Press, 1994.

[20] S. Schmitz. On the computational complexity of dominance links in grammatical formalisms. In *ACL 2010*, pages 514–524. ACL Press, 2010.

[21] S. Schmitz. Complexity hierarchies beyond Elementary. Manuscript, 2013. URL http://arxiv.org/abs/1312.5686.

[22] S. Schmitz. Implicational relevance logic is 2-EXPTIME-complete. In G. Dowek, editor, *RTA-TLCA 2014*, LNCS. Springer, 2014. URL http://arxiv.org/abs/1402.0705. To appear.

[23] A. Urquhart. The complexity of decision procedures in relevance logic II. *J. Symb. Log.*, 64(4):1774–1802, 1999.

[24] A. Urquhart. The complexity of linear logic with weakening. In S. Buss, P. Hajek, and P. Pudlak, editors, *Logic Colloquium '98*, volume 13 of *Lect. Notes in Logic*, pages 500–515. ASL, 2000.

[25] K. N. Verma and J. Goubault-Larrecq. Karp-Miller trees for a branching extension of VASS. *Disc. Math. Theor. Comput. Sci.*, 7(1):217–230, 2005.