Typability in Bounded Dimension

Andrej Dudenhefner
Faculty of Computer Science
Tecnical University Dortmund
Germany
Email: andrej.dudenhefner@cs.tu-dortmund.de

Jakob Rehof
Faculty of Computer Science
Technical University Dortmund
Germany
Email: jakob.rehof@cs.tu-dortmund.de

***Abstract*—Recently (authors, POPL 2017), a notion of dimensionality for intersection types was introduced, and it was shown that the bounded-dimensional inhabitation problem is decidable under a non-idempotent interpretation of intersection and undecidable in the standard set-theoretic model. In this paper we study the typability problem for bounded-dimensional intersection types and prove that the problem is decidable in both models. We establish a number of bounding principles depending on dimension. In particular, it is shown that dimensional bound on derivations gives rise to a bounded width property on types, which is related to a generalized subformula property for typings of arbitrary terms. Using the bounded width property we can construct a nondeterministic transformation of the typability problem to unification, and we prove that typability in the set-theoretic model is PSPACE-complete, whereas it is in NP in the multiset model.**

I. Introduction

A. Contribution and Related Work

The intersection type system [1], [2], [3] occupies a central position in the theory of typed λ-calculus (see [4] for an overview). Due to its enormous expressive power, both of the most fundamental type theoretical decision problems, *inhabitation* (given a type, is there a term having the type?) and *typability* (given a term, does it have a type?), are undecidable for the intersection type system [4]. It has therefore been a persistent topic of interest to search for bounding principles which would admit of algorithmic solutions to these problems while retaining interesting levels of expressiveness. In this paper we are concerned with the application of a new bounding principle for intersection types to the typability problem.

Of special interest are parametric bounding principles such that, by increasing a bounding parameter, say k, ever larger sections of the original system become expressible within the k-bounded fragments. Hitherto, the most prominent parametric bounding principle appears to have been that of type theoretical *rank* [5]. For the classical decision problems with intersection types, the main known results are that typability is decidable for all finite-rank fragments (rank k-typability), as was shown by Kfoury and Wells [6] with complexity growing exponentially in increasing rank [7], and that inhabitation is decidable and Expspace-complete for rank 2 and undecidable for all ranks $k \geq 3$, as was shown by Urzyczyn [8].

Recently [9], we introduced a concept of *dimensionality* for intersection types, which can be seen as a parametric bounding principle that is orthogonal to principles based on rank. Briefly, the dimension of a typed λ-term is given by the minimal *norm* of an *elaboration* (a proof theoretic decoration) necessary for typing the term at its type, and, intuitively, measures intersection introduction as a resource.

In [9] our focus was on the inhabitation problem. We could show that, while inhabitation remains undecidable under dimensional bound for the standard set-theoretic interpretation of intersection as an associative, commutative, idempotent operator, the problem is decidable and Expspace-complete parametric in bounded dimension when dimension is measured under a multiset (non-idempotent and nonlinear) interpretation of intersection. Because dimension is independent of rank and the rank 2-fragment of normal forms was shown to be subsumed under linear dimension, we thereby obtained a substantial extension of the rank 2-borderline [8] between decidability and undecidability, extending across all ranks within Expspace under dimensional bound.

In the present paper we turn our attention to the typability problem under dimensional bound d: given an untyped λ-term M and a dimension d, does M have a type in some type environment under dimensional bound d? We show that the problem is decidable both in the standard set-theoretic model and in the multiset model. More specifically, we show that typability in the standard model is Pspace-complete, whereas the problem is in NP (and type checking is shown NP-hard) in the multiset model. Interestingly, Pspace-hardness for the standard model holds already at fixed dimension 4, but, rather surprisingly, the problem stays in space bounded by a polynomial in the dimensional parameter d and input term size. Yet, every typing in the full intersection type system is expressible for sufficiently high dimension, and, for example, already dimension 1 contains much more than the simply typed λ-calculus. The upper bounds are constructed by nondeterministic transformations to standard unification, which leaves one hopeful that type inference could be engineered to reach interesting levels of efficiency in practice. From the perspective of typability and type inference, the set-theoretic system is the more interesting, because it is more expressive and corresponds directly to the standard intersection type system. However, from the perspective of inhabitation [9], the multiset system is of focal interest, because it contains a theory of decidable inhabitation for the intersection type system. Our study of typability in multiset dimension provides

978-1-5090-3018-7/17/$31.00

a companion algorithmic theory of typability, which improves our understanding of the behavior of multiset dimension on λ-terms and could be useful as type inference component in applications of inhabitation to program synthesis. Both dimension-bounded systems enjoy subject reduction [9], so each of the dimensional fragments constitutes a meaningful type system. Further information on general areas of related work, including synthesis and non-idempotent intersection types, can be found in [9].

Returning to the most closely related work on finite-rank fragments [6], [7], typability is decidable in all ranks, as mentioned above. Since dimension is independent of rank, dimension can be seen as an orthogonal principle of bounding for the typability problem. Because typability complexity increases exponentially in rank [7] but stays within PSPACE for each fixed dimension, dimensional bound can be seen to provide a more fine-grained parameter. It could also be combined with rank-bounding complementarily: while rank bounds intersections along type depth, dimension in a sense (as will be shown) bounds them in width. Some existing applications of rank-bounded type inference (e.g. [10]) report tractability problems that could be met by the more fine-grained approach of bounded dimension.

B. Overview

The paper is organized as follows.

Sec. II contains a compact presentation of the notions of elaboration, norm, and dimension as introduced in [9]. In order to minimize overlap with [9], where the reader can find a more detailed introduction, we give a condensed presentation in which both the set-theoretic and the multiset system emerge from a single set of rules which are easily mapped back to the standard intersection type system.

In Sec. III we develop various bounding properties from norm and dimension that are of relevance for the typability problem and provide insight into the typing power of the dimensional systems.

Sec. IV contains Theorem 14, bounded width theorem, which is used later for constructing the upper bounds for typability. We show that dimension can be used to induce a bound (also depending term size) on the maximal number of intersection type components ("width") needed globally in a type derivation. The proof of bounded width uses the idea of type filtration which also gives rise to a generalized form of subformula property (Theorem 16).

In Sec. V we prove PSPACE-completeness for the typability problem in bounded set-theoretic dimension. The lower bound proof uses techniques from [11], [12] to establish the rather striking result that PSPACE-hardness holds already in fixed dimension 4. The upper bound is constructed by exploiting the bounded width theorem to provide a nondeterministic transformation of typability into constraint systems which can be solved by standard unification.

The technical development is concluded in Sec. VI, where typability in bounded multiset dimension is shown to be in NP, and we prove also an NP lower bound for type checking.

Sec. VII concludes and contains discussion of future work. Some proofs were left out, they can be found in [13].

II. Elaborations, Norm, and Dimension

We give a compact presentation of dimensional intersection type calculus [9], fix notation, and define the typability problems we will be concerned with.[1] The basic idea is to introduce a *norm* on derivations by instrumenting the intersection type system with *elaborations*, $\mathbf{P}$, which are proof theoretic decorations of λ-terms M in which the usage of the intersection introduction rule ($\cap$I) is made explicit. Thus, a judgement $\Gamma \vdash \mathbf{P} : A_1 \cap \cdots \cap A_n$ means that M can be given type $A_1 \cap \cdots \cap A_n$ in the intersection type system, where the elaboration $\mathbf{P}$ records sets of types introduced by applications of ($\cap$I) in a type derivation for M, and where $\lceil \mathbf{P} \rceil$, the decoration erasure of $\mathbf{P}$, is M. We equip elaborations with a norm, $\|\mathbf{P}\|$, which measures the maximal size of decorating sets in $\mathbf{P}$, thereby yielding a measure of the amount of intersection introduction performed to type the term. The *dimension* of M (at Γ and A) can now be defined as the smallest d such that $\Gamma \vdash \mathbf{P} : A$ is derivable for some $\mathbf{P}$ with $\|\mathbf{P}\| = d$ and $\lceil \mathbf{P} \rceil \equiv M$. We consider two elaboration systems, one in which decorations are sets (as just mentioned) corresponding to the standard interpretation of intersection as an associative, commutative, idempotent operator, and one in which decorations are multisets, $\mathfrak{s}$, corresponding to a non-idempotent interpretation of intersection. Multiset elaborations are denoted $\Delta \vdash \mathbb{P} : \mathfrak{s}$. Because sets are just special multisets in which each member has multiplicity at most 1, we can present both elaboration systems compactly[2] by specifying the multiset calculus.

A. Preliminaries

Untyped λ-terms are ranged over by L, M, N:

$$M, N \quad ::= \quad x \mid (\lambda x.M) \mid (MN)$$

We let $\mathrm{fv}(M)$ denote the free variables in M. Unless otherwise stated we follow notions and notational conventions of [14]. We use the strict presentation of intersection types (see the overview in [15] for reference), which is a standard variant.

Types in the strict type system [15, Definition 5.1] are stratified into strict types and intersections of such. The set $\mathcal{M}_0$ of strict multiset types are ranged over by φ, ψ, etc., and the set $\mathcal{M}_1$ of multiset types are ranged over by $\mathfrak{s}, \mathfrak{t}$, etc., which are nonempty finite multisets $\mathfrak{s} : \mathcal{M}_0 \to \mathbb{N}_0$:

$$\begin{array}{lcll} \mathcal{M}_0 & \ni & \varphi, \psi & ::= \quad a \mid \mathfrak{s} \to \varphi \\ \mathcal{M}_1 & \ni & \mathfrak{s}, \mathfrak{t} & ::= \quad [\varphi_1, \ldots, \varphi_n] \end{array}$$

where $a, b, \ldots$ range over atoms, $[\varphi_1, \ldots, \varphi_n]$ is a multiset (unordered list), and $n > 0$. We let $\mathcal{M} = \mathcal{M}_0 \cup \mathcal{M}_1$. *Type environments*, ranged over by Δ, are finite sets of type assumptions $(x : \mathfrak{s})$, in which a term variable occurs at most once. We write

[1] For a more extensive introduction to dimensional calculus the reader is referred to [9], but all necessary definitions will be given here in order to fix notation and make the paper self contained.

[2] In [9] the systems are specified separately.

$\Delta \cup \{(x : \mathfrak{s})\}$ as $\Delta, x : \mathfrak{s}$ and let $\mathrm{dom}(\Delta) := \{x \mid \exists \mathfrak{s}.\ (x : \mathfrak{s}) \in \Delta\}$. *Multiset elaborations* are λ-terms decorated with elements of $\mathcal{M}_1$:

$$\mathbb{P}, \mathbb{Q} \quad ::= \quad x\langle \mathfrak{s}\rangle \mid (\lambda x.\mathbb{P})\langle \mathfrak{s}\rangle \mid (\mathbb{P}\mathbb{Q})\langle \mathfrak{s}\rangle$$

The syntactic parts $\langle \mathfrak{s}\rangle$ are referred to as (multiset) *decorations*. For $\mathfrak{s} = [\varphi_1, \ldots, \varphi_n]$ we sometimes write decorations $\langle \varphi_1, \ldots, \varphi_n\rangle$ as shorthand for $\langle [\varphi_1, \ldots, \varphi_n]\rangle$. *Partial elaborations*, ranged over by $\mathbb{T}$, are terms such that for some $\mathfrak{s}$, $\mathbb{T}\langle s\rangle$ is an elaboration (it is sometimes convenient to expose the outermost decoration of an elaboration $\mathbb{P}$ by writing $\mathbb{P} \equiv \mathbb{T}\langle \mathfrak{s}\rangle$).

We let $|\mathfrak{s}|$ denote the size of the multiset, $|\mathfrak{s}| = \sum\{\mathfrak{s}(\varphi) \mid \mathfrak{s}(\varphi) > 0\}$. We define the operation $\mathfrak{s} \cup \mathfrak{s}'$ by setting $(\mathfrak{s} \cup \mathfrak{s}')(\varphi) = \max\{\mathfrak{s}(\varphi), \mathfrak{s}'(\varphi)\}$, and we define multiset union $\mathfrak{s} \uplus \mathfrak{s}'$ as usual by $(\mathfrak{s} \uplus \mathfrak{s}')(\varphi) = \mathfrak{s}(\varphi) + \mathfrak{s}'(\varphi)$. Multiset containment is denoted $\Subset$, with $\mathfrak{s} \Subset \mathfrak{s}'$ if and only if $\mathfrak{s}(\varphi) \leq \mathfrak{s}'(\varphi)$ for all φ.

A *set* of types is a multiset $\mathfrak{s}$ such that $\mathfrak{s}(\varphi) \in \{0, 1\}$ for all φ, and for sets of types $\subseteq, \cap$ denote the subset relation and intersection operation on sets. Types in $\mathcal{M}$ in which all multisets appearing in subformulae are sets of types are called *set-theoretic types*, and such are just syntactic variants of standard (strict) intersection types [15, Definition 5.1]. Using standard syntax for intersection types we have the set $\mathcal{S}_0$ of strict types ranged over by A, B, etc., and the set $\mathcal{S}_1$ of intersection types ranged over by σ, τ, etc.,) defined by:

$$\begin{array}{lllll} \mathcal{S}_0 & \ni & A, B & ::= & a \mid \sigma \to A \\ \mathcal{S}_1 & \ni & \sigma, \tau & ::= & A_1 \cap \cdots \cap A_n \end{array}$$

where $a, b, \ldots$ range over atoms, and $n > 0$. We let $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1$. The operator $\cap$ in intersection types is tacitly regarded as associative, commutative, and idempotent.

Multiset types are mapped to standard intersection types by collapsing multisets to their underlying sets, by the function $(\_)^\circ : \mathcal{M} \to \mathcal{S}$ with definition $a^\circ \equiv a, (\mathfrak{s} \to \varphi)^\circ \equiv \mathfrak{s}^\circ \to \varphi^\circ, [\varphi_1, \ldots, \varphi_n]^\circ \equiv \varphi_1{}^\circ \cap \cdots \cap \varphi_n{}^\circ$. For a set-theoretic type $\mathfrak{s}$, the intersection type $\mathfrak{s}^\circ$ is just $\mathfrak{s}$ cast into standard intersection type syntax [3], [4], and for such types we will sometimes use the standard syntax tacitly. We let Γ range over type environments in which only set-theoretic types occur. For multiset types $\mathfrak{s} = [\varphi_1, \ldots, \varphi_n]$ we sometimes refer to the φ_i as *components* of $\mathfrak{s}$, and similarly, an intersection type $\sigma = A_1 \cap \cdots \cap A_n$ has the A_i as components.

Let $\lceil \mathbb{P} \rceil$ denote the untyped term arising from erasing all decorations from $\mathbb{P}$. For $x \in \mathrm{fv}(\lceil \mathbb{P} \rceil)$ we let $\mathrm{ve}_x(\mathbb{P})$ (the variable elaborations in $\mathbb{P}$) denote the set of multisets $\mathfrak{s}$ such that $x\langle \mathfrak{s}\rangle$ occurs as a subterm of $\mathbb{P}$.

Given a binary associative operation $\oplus : \mathcal{M}_1 \times \mathcal{M}_1 \to \mathcal{M}_1$ we define the operation $\mathbb{P} \langle \oplus \rangle \mathbb{Q}$ on elaborations $\mathbb{P}$ and $\mathbb{Q}$ with $\lceil \mathbb{P} \rceil \equiv \lceil \mathbb{Q} \rceil$ by pointwise application of $\oplus$ to decorating multisets as follows:

$$\begin{array}{lll} x\langle \mathfrak{s}\rangle \langle \oplus \rangle\, x\langle \mathfrak{s}'\rangle & \equiv & x\langle \mathfrak{s} \oplus \mathfrak{s}'\rangle \\ (\lambda x.\mathbb{P})\langle \mathfrak{s}\rangle \langle \oplus \rangle (\lambda x.\mathbb{Q})\langle \mathfrak{s}'\rangle & \equiv & (\lambda x.\mathbb{P} \langle \oplus \rangle \mathbb{Q})\langle \mathfrak{s} \oplus \mathfrak{s}'\rangle \\ (\mathbb{P}\mathbb{Q})\langle \mathfrak{s}\rangle \langle \oplus \rangle (\mathbb{P}'\mathbb{Q}')\langle \mathfrak{s}'\rangle & \equiv & ((\mathbb{P} \langle \oplus \rangle \mathbb{P}')(\mathbb{Q} \langle \oplus \rangle \mathbb{Q}'))\langle \mathfrak{s} \oplus \mathfrak{s}'\rangle \end{array}$$

We write $\left\langle \bigoplus \right\rangle_{i=1}^{n} \mathbb{P}_i$ for $\mathbb{P}_1 \langle \oplus \rangle \cdots \langle \oplus \rangle \mathbb{P}_n$. We define the *max-norm* $\|\bullet\|$ on elaborations:

$$\begin{array}{lll} \|x\langle \mathfrak{s}\rangle\| & = & |\mathfrak{s}| \\ \|(\lambda x.\mathbb{P})\langle \mathfrak{s}\rangle\| & = & \max\{\|\mathbb{P}\|, |\mathfrak{s}|\} \\ \|(\mathbb{P}\mathbb{Q})\langle \mathfrak{s}\rangle\| & = & \max\{\|\mathbb{P}\|, \|\mathbb{Q}\|, |\mathfrak{s}|\} \end{array}$$

By this definition, the set of elaborations of the same λ-term is organized as a normed space:

Lemma 1: For all elaborations $\mathbb{P}$, $\mathbb{Q}$ with $\lceil \mathbb{P} \rceil \equiv \lceil \mathbb{Q} \rceil$:

1) (non-negativity) $\|\mathbb{P}\| > 0$
2) (subadditivity) $\|\mathbb{P} \langle \oplus \rangle \mathbb{Q}\| \leq \|\mathbb{P}\| + \|\mathbb{Q}\|$ for $\oplus \in \{\cup, \uplus\}$.

B. Elaboration System

The elaboration calculus introduced in [9] can be presented by the following rules. Here we use a single parameterized rule system (instead of two in [9]). In rule ($\cap$I) we may fix the operation $\bigoplus$ to be either $\biguplus$ or $\bigcup$. The former choice defines the *multiset elaboration system* of [9], the latter choice defines the *set-theoretic elaboration system* of [9].

$$\frac{}{\Delta, x : [\varphi_1, \ldots, \varphi_n] \vdash x\langle [\varphi_i]\rangle : [\varphi_i]}(\mathrm{var})$$

$$\frac{\Delta, x : \mathfrak{s} \vdash \mathbb{P} : [\varphi]}{\Delta \vdash (\lambda x.\mathbb{P})\langle [\mathfrak{s} \to \varphi]\rangle : [\mathfrak{s} \to \varphi]}(\to\mathrm{I})$$

$$\frac{\Delta \vdash \mathbb{P} : [\mathfrak{s} \to \varphi] \qquad \Delta \vdash \mathbb{Q} : \mathfrak{s}}{\Delta \vdash (\mathbb{P}\mathbb{Q})\langle [\varphi]\rangle : [\varphi]}(\to\mathrm{E})$$

$$\frac{\Delta \vdash \mathbb{P}_i : [\varphi_i] \ \ (i = 1 \ldots n) \quad (\star)}{\Delta \vdash \left\langle \bigoplus \right\rangle_{i=1}^{n} \mathbb{P}_i : [\varphi_1, \ldots, \varphi_n]}(\cap\mathrm{I})$$

where $(\star)$ is the side condition:

1) $\lceil \mathbb{P}_i \rceil \equiv \lceil \mathbb{P}_j \rceil$ for all $i, j = 1 \ldots n$, and
2) for $\mathbb{P} \equiv \left\langle \bigoplus \right\rangle_{i=1}^{n} \mathbb{P}_i$ the following condition holds:

$$\forall x \in \mathrm{fv}(\lceil \mathbb{P} \rceil).\forall \mathfrak{s} \in \mathrm{ve}_x(\mathbb{P}).\forall \varphi \in \mathfrak{s}.\ \Delta(x)(\varphi) \geq \mathfrak{s}(\varphi)$$

That is, for each free variable x in $\mathbb{P}$, a type φ has a multiplicity in the assumption $\Delta(x)$ which is at least the maximum multiplicity with which φ occurs in any single decoration occurrence $x\langle \mathfrak{s}\rangle$ in $\mathbb{P}$.

A *multiset derivation* is a derivation in which $\bigoplus = \biguplus$ is used in rule ($\cap$I). We define the operation $\boxplus := \langle \biguplus \rangle$ on multiset elaborations. A *set-theoretic elaboration* $\mathbb{P}$ is one in which all multiset types $\mathfrak{s}$ appearing in decorations in $\mathbb{P}$ are set-theoretic types. We let $\mathbf{P}$ range over set-theoretic elaborations and $\mathbf{T}$ ranges over partial set-theoretic elaborations. We will feel free to write set-theoretic elaborations using standard intersection type syntax, so we may write a decoration as $\mathbf{T}\langle [A_1, \ldots, A_n]\rangle$, which we may also abbreviate as $\mathbf{T}\langle A_1, \ldots, A_n\rangle$ or $\mathbf{T}\langle \sigma\rangle$ for $\sigma = A_1 \cap \cdots \cap A_n$. A derivation of $\Delta \vdash \mathbb{P} : \mathfrak{s}$ in which all types appearing as subformulae in the derivation are set-theoretic types and which uses $\bigoplus = \bigcup$ in rule ($\cap$I) is called a *set-theoretic derivation*, and we write $\Gamma \vdash \mathbf{P} : \sigma$ with $\Gamma \equiv \Delta^\circ$, $\mathbf{P} \equiv \mathbb{P}^\circ$ and $\sigma \equiv \mathfrak{s}^\circ$. Notice that the rules of the elaboration system preserve the set-theoretic property of judgements when $\bigoplus = \bigcup$ is used in rule ($\cap$I). We define the operator $\bigsqcup := \langle\bigcup\rangle$ on set-theoretic elaborations. Notice that condition 2) of side condition ($\star$) is automatically satisfied in set-theoretic derivations (all components of elements in $\mathrm{ve}_x(\mathbf{P})$ must be among the components of $\Gamma(x)$ by rule (var)).

It is easy to verify that the elaboration system is but an instrumented version of the strict intersection type system [15, Definition 5.1]. Denoting that system as λ^S and its derivability relation as $\vdash_S$, we have $\Gamma \vdash_S M : \sigma$ if and only if $\Gamma \vdash \mathbb{P} : \sigma$ for some for some $\mathbb{P}$ with $\lceil\mathbb{P}\rceil \equiv M$.

Any multiset derivation can evidently be collapsed into a valid set-theoretic derivation, and the existence of the latter can be reduced to the former. Let us lift the map $(\_)^\circ$ to type environments, elaborations, judgements, and derivations by applying it to all types within these structures. Then we have:

Lemma 2: Derivability of $\Delta \vdash \mathbb{P} : \mathfrak{s}$ in the multiset system implies derivability of $\Delta^\circ \vdash \mathbb{P}^\circ : \mathfrak{s}^\circ$ in the set-theoretic system, and $\Gamma \vdash \mathbf{P} : \sigma$ if and only if there exist Δ, $\mathbb{P}$, and $\mathfrak{s}$ such that $\Delta \vdash \mathbb{P} : \mathfrak{s}$ and $\Gamma \equiv \Delta^\circ$, $\mathbf{P} \equiv \mathbb{P}^\circ$, and $\sigma \equiv \mathfrak{s}^\circ$.

We write $\Delta \vdash M \Longmapsto \mathbb{P} : \mathfrak{s}$ for $\Delta \vdash \mathbb{P} : \mathfrak{s}$ with $\lceil\mathbb{P}\rceil \equiv M$, and we write $\Gamma \vdash M \longmapsto \mathbf{P} : \sigma$ for $\Gamma \vdash \mathbf{P} : \sigma$ with $\lceil\mathbf{P}\rceil \equiv M$. These relations are identical to the identically notated relations of [9]. If $\mathcal{D}$ is a derivation with conclusion $\Delta \vdash \mathbb{P} : \mathfrak{s}$ we write $\mathcal{D} \triangleright \Delta \vdash \mathbb{P} : \mathfrak{s}$ and $\mathcal{D} \triangleright \Delta \vdash M \Longmapsto \mathbb{P} : \mathfrak{s}$ to indicate so.

Following [9] we define for $n > 0$ the *bounded-dimensional* relations $\vdash_n$ and $\Vdash_n$ by

- $\Gamma \vdash_n M : \sigma$ iff $\exists \mathbf{P}.\ \Gamma \vdash M \longmapsto \mathbf{P} : \sigma$ with $\|\mathbf{P}\| \leq n$
- $\Gamma \Vdash_n M : \sigma$ iff $\exists \Delta\mathbb{P}\mathfrak{s}.\ \Delta \vdash M \Longmapsto \mathbb{P} : \mathfrak{s}$ with $\Gamma = \Delta^\circ$, $\sigma = \mathfrak{s}^\circ$, and $\|\mathbb{P}\| \leq n$

The set-theoretic (resp. multiset) *dimension* of a term M at Γ and σ is the smallest $n > 0$ such that $\Gamma \vdash_n M : \sigma$ (resp. $\Gamma \Vdash_n M : \sigma$).

It should be emphasized that the relations $\vdash_n$ and $\Vdash_n$ are fundamentally different only in so far as the systems behave differently with respect to the norm $\|\bullet\|$: the multiset dimension of a term can be much larger than the set-theoretic counterpart (see Sec. III), and it was shown in [9] that the inhabitation problem for $\vdash_n$ is undecidable [9, Theorem 28], whereas it is decidable and Expspace-complete for $\Vdash_n$ [9, Theorem 34]. Both of the bounded-dimensional typing relations enjoy subject reduction, since terms can be elaborated under non-increasing norm under β-reduction [9, Theorem 18,19] and therefore constitute meaningful fragments of the intersection type system parametric in dimension.

We define the decision problems we will be concerned with. The problem of *typability in bounded set-theoretic dimension* is the following decision problem:

- Given a λ-term M and a dimension $d > 0$, does there exist Γ and A such that $\Gamma \vdash_d M : A$?

The problem of *typability in bounded multiset dimension* is the following decision problem:

- Given a λ-term M and a dimension $d > 0$, does there exist Γ and A such that $\Gamma \Vdash_d M : A$?

III. Some Bounding Properties of Norm and Dimension

In order to appreciate the nature of the results to be presented in later sections it is useful to discuss a few fundamental bounding properties of the dimensional system first (proofs are generally left out in this section, they are all by induction and can be found in [13]).

Let us consider the natural attempt to obtain a decidable subsystem (with respect to typability) of the strict intersection type calculus by alternative principles of bounding the power of intersection introduction. The simplest such principle would probably be to bound the arity of the intersection type operator $\cap$ by a parameter k. In such a system, one would be able to form only intersections of the form $\bigcap_{i \in I} A_i$, where the size of the set I is bounded by k. This system is obviously more restrictive than dimensional restriction. For example, we have for $\omega \equiv \lambda x.(xx)$ the elaboration $\vdash \omega \longmapsto \mathbf{P} : (a \cap (a \to a)) \to a$ with

$$\mathbf{P} \equiv (\lambda x.(x\langle a \to a\rangle\ x\langle a\rangle)\langle a\rangle)\langle(a \cap (a \to a)) \to a\rangle$$

at dimension 1, whereas intersection arity is 2. In general, dimension does not bound the arity of the intersection type operator. For example, abstractions $\lambda x.M$ of type $\sigma \to A$ can have arbitrarily large arities in σ without increasing the norm (the term ω just considered provides an illustration). As we will see later (Theorem 14), dimension can in fact be used to indirectly bound maximal intersection arity ("width") for especially simple type derivations of importance for typability, but such bound will not be constant (it will depend on the size of the term as well as dimension).

A more interesting restriction is to bound the number n of premises in the intersection introduction rule ($\cap$I) by a parameter k. Let us refer to this restriction as *premise-bounded*. Interestingly, such restriction allows type derivations exceeding set-theoretic dimension k, since we can "chain" λ-terms in such a way that a single subterm occurrence gets retyped more than k times, as the following example illustrates.

Example 3: Let $A \equiv ((a \to a) \cap (b \to b)) \to e$, $B \equiv ((c \to c) \cap (d \to d)) \to f$, $\Gamma = \{x : (e \cap f) \to g, y : A \cap B\}$. Then we have a derivation $\mathcal{D} \triangleright \Gamma \vdash_4 x(y(\lambda z.z)) : g$ that arises as follows. Consider first the derivation $\mathcal{D}'$:

$$\dfrac{\dfrac{}{\Gamma \vdash_4 y : A} \qquad \dfrac{\Gamma \vdash_4 \lambda z.z : a \to a \quad \Gamma \vdash_4 \lambda z.z : b \to b}{\Gamma \vdash_4 \lambda z.z : (a \to a) \cap (b \to b)}}{\Gamma \vdash_4 y(\lambda z.z) : e}$$

We have a similar, independent derivation $\mathcal{D}''$ with conclusion $\Gamma \vdash_4 y(\lambda z.z) : f$. We compose the derivations $\mathcal{D}'$ and $\mathcal{D}''$ to obtain the derivation $\mathcal{D}$:

$$\dfrac{\dfrac{}{\Gamma \vdash_4 x : (e \cap f) \to g} \qquad \dfrac{\mathcal{D}' \triangleright \Gamma \vdash_4 y(\lambda z.z) : e \quad \mathcal{D}'' \triangleright \Gamma \vdash_4 y(\lambda z.z) : f}{\Gamma \vdash_4 y(\lambda z.z) : e \cap f}}{\Gamma \vdash_4 x(y(\lambda z.z)) : g}$$

The derivation $\mathcal{D}$ corresponds to the set-theoretic elaboration $\mathbf{P}$ (decorations only partially shown and using column vector notation for multisets):

$$x\langle e \cap f \to g\rangle(y\langle A, B\rangle\ (\lambda z.z\left\langle \begin{matrix} a \\ b \\ c \\ d \end{matrix} \right\rangle)\left\langle \begin{matrix} a \to a \\ b \to b \\ c \to c \\ d \to d \end{matrix} \right\rangle)$$

with $||\mathbf{P}|| = 4$. Still, $\mathcal{D}$ obeys the restriction that the operator $\cap$ is restricted to arity 2 and the rule ($\cap$I) is only used with two premises.

However, we can bound the norm for premise-bounded derivations as follows, implying that the restriction falls within the theory of bounded dimension. Let $|M|$ denote the number of nodes in the syntax tree of the λ-term M.

Proposition 4: If $\Gamma \vdash M \longmapsto \mathbf{P} : A$ by a premise-bounded derivation in which the number of premises in rule ($\cap$I) is bounded by k, then $||\mathbf{P}|| \leq k^{|M|}$.

Proof: By standard properties [15] we can assume that intersection introduction is only used as premise in applications of ($\to$E), so we can consider the system in which ($\cap$I) is restricted to:

$$\frac{\Gamma \vdash N \longmapsto \mathbf{P} : (A_1 \cap \cdots \cap A_n) \to B \qquad \Gamma \vdash L \longmapsto \mathbf{Q}_i : A_i\ (i = 1 \ldots n), n \leq k}{\Gamma \vdash (NL) \longmapsto (\mathbf{P} \bigsqcup_{i=1}^{n} \mathbf{Q}_i)\langle B\rangle : B}$$

We proceed by induction on the derivation, and in the case where the above rule is used we have by induction hypothesis $||\mathbf{P}|| \leq k^{|N|}$ and $||\mathbf{Q}_i|| \leq k^{|L|}$, and since

$$||\bigsqcup_{i=1}^{n} \mathbf{Q}_i|| \leq \sum_{i=1}^{n} ||\mathbf{Q}_i|| \leq k \cdot k^{|L|} = k^{|L|+1} \leq k^{|(NL)|}$$

we obtain $||(\mathbf{P} \bigsqcup_{i=1}^{n} \mathbf{Q}_i)\langle B\rangle|| = \max\{||\mathbf{P}||, ||\bigsqcup_{i=1}^{n} \mathbf{Q}_i||\} \leq k^{|(NL)|}$. Remaining cases are either obvious or analogous. ■

Next, let us consider some properties relating norm and dimension under set-theoretic and multiset elaboration. We define the *decoration size*, dsz, of an elaboration as

$$\begin{aligned} \mathrm{dsz}(x\langle[\varphi_1, \ldots, \varphi_n]\rangle) &= n \\ \mathrm{dsz}((\lambda x.\mathbb{P})\langle[\varphi_1, \ldots, \varphi_n]\rangle) &= n + \mathrm{dsz}(\mathbb{P}) \\ \mathrm{dsz}((\mathbb{P}\ \mathbb{Q})\langle[\varphi_1, \ldots, \varphi_n]\rangle) &= n + \mathrm{dsz}(\mathbb{P}) + \mathrm{dsz}(\mathbb{Q}) \end{aligned}$$

Clearly, $\mathrm{dsz}(\mathbb{P}) \leq ||\mathbb{P}|| \cdot |M|$, for $M \equiv \ulcorner\mathbb{P}\urcorner$.

Multiset dimension is indicative, up to an exponential order of magnitude, of the set-theoretic dimension of the β-normal form of a typable term:

Lemma 5: Let N be a normal form. If $\Gamma \vdash N \longmapsto \mathbf{P} : \sigma$ with $||\mathbf{P}|| \leq n$, then there exist Δ, $\mathfrak{s}$ and $\mathbb{P}$ such that $\Delta \vdash N \Longmapsto \mathbb{P} : \mathfrak{s}$ with $\Delta^\circ = \Gamma$, $\mathfrak{s}^\circ = \sigma$ and $||\mathbb{P}|| \leq n^{\mathrm{dsz}(\mathbf{P})}$.

Proposition 6: If $\Gamma \vdash_n M : \sigma$ and N is the normal form of M, then $\Gamma \Vdash_m N : \sigma$, where $m = n^{n \cdot |N|}$.

Proof: Assume $\Gamma \vdash_n M : \sigma$. By subject reduction in bounded dimension, $\Gamma \vdash_n N : \sigma$. There exists an elaboration $\mathbf{P}$ such that $\Gamma \vdash N \longmapsto \mathbf{P} : \sigma$ and $||\mathbf{P}|| \leq n$. By Lemma 5 we have $\Delta \vdash N \Longmapsto \mathbb{P} : \mathfrak{s}$ such that $\Delta^\circ \equiv \Gamma$, $\mathfrak{s}^\circ \equiv \sigma$ and $||\mathbb{P}|| \leq n^{\mathrm{dsz}(\mathbf{P})} \leq n^{n \cdot |N|}$, which proves the claim. ■

The following example shows that multiset dimension can undergo a non-Kalmar-elementary blow-up in term size while set-theoretic dimension remains constant.

Example 7: The Church numerals can be typed in simple types and hence, as is true of all simple typed terms, they are typable in multiset dimension 1 at those types. However, if we raise dimension by 1, we can get nonelementary blow-up in multiset dimension over set-theoretic dimension. Let $\mathbf{c}_2 \equiv \lambda f.\lambda x.f(fx)$ and $A_0 = a$, $B_0 = b$, $A_{i+1} = (A_i \cap B_i) \to A_i$, $B_{i+1} = (A_i \cap B_i) \to B_i$. We have $\vdash_2 \mathbf{c}_2 : A_n \cap B_n$ for arbitrary $n \geq 2$ in dimension 2 shown by the following set-theoretic elaborations

$$(\lambda f.(\lambda x.(f \left\langle \begin{matrix} A_{n-1} \\ B_{n-1} \end{matrix} \right\rangle (f \left\langle \begin{matrix} A_{n-1} \\ B_{n-1} \end{matrix} \right\rangle x \left\langle \begin{matrix} A_{n-2} \\ B_{n-2} \end{matrix} \right\rangle) \left\langle \begin{matrix} A_{n-2} \\ B_{n-2} \end{matrix} \right\rangle) \langle A_{n-2}\rangle) \langle A_{n-1}\rangle) \langle A_n\rangle$$

$$(\lambda f.(\lambda x.(f \left\langle \begin{matrix} A_{n-1} \\ B_{n-1} \end{matrix} \right\rangle (f \left\langle \begin{matrix} A_{n-1} \\ B_{n-1} \end{matrix} \right\rangle x \left\langle \begin{matrix} A_{n-2} \\ B_{n-2} \end{matrix} \right\rangle) \left\langle \begin{matrix} A_{n-2} \\ B_{n-2} \end{matrix} \right\rangle) \langle B_{n-2}\rangle) \langle B_{n-1}\rangle) \langle B_n\rangle$$

Since $A_n \equiv (A_{n-1} \cap B_{n-1}) \to \ldots \to (A_1 \cap B_1) \to (A_0 \cap B_0) \to A_0$, we have, for all terms $M_n \equiv \mathbf{c}_2\mathbf{c}_2 \ldots \mathbf{c}_2 f x$ (where $\mathbf{c}_2$ occurs n times), in set-theoretic dimension 2:

$$\{f : A_1 \cap B_1, x : A_0 \cap B_0\} \vdash_2 M_n : A_0$$

In contrast, the terms M_n have nonelementary multiset dimension at A_0. To get this lower bound, we reduce M_n to normal form and use subject reduction in bounded multiset dimension ([9, Theorem 18]): M_n reduces to $f\,(f\,(\ldots(f\,x)\ldots))$, where the number of occurrences of f is nonelementary in n and each occurrence of f requires intersection introduction to type its argument. Due to multiset union on intersection introduction, the multiset decoration of the innermost term x grows with each additional occurrence of f.

For a derivation $\mathcal{D}$ (either multiset derivation or set-theoretic derivation) let $|\mathcal{D}|$ denote the number of nodes in the proof tree $\mathcal{D}$. A characteristic of the multiset system is that elaborations are a measure of derivation size in the following sense:

Lemma 8: Suppose $\mathcal{D}$ is a multiset derivation such that $\mathcal{D} \triangleright \Delta \vdash M \Longmapsto \mathbb{P} : \mathfrak{s}$. Then

$$||\mathbb{P}|| \leq |\mathcal{D}| \leq \mathrm{dsz}(\mathbb{P}) \leq ||\mathbb{P}|| \cdot |M|$$

As can be seen from Example 7 together with Lemma 8, both norm and the size of multiset derivations can outgrow their set-theoretic counterparts by a nonelementary order of magnitude, when dimension is raised by 1. In distinction to multiset elaborations, set-theoretic elaborations are not a measure of proof (derivation) complexity. However, for normal forms we can show that the size of derivations are of the same order of magnitude in the two systems, as measured by multiset dimension, since we have:

Proposition 9: Let $\mathcal{D} \triangleright \Gamma \vdash N \longmapsto \mathbf{P} : \sigma$ be a set-theoretic derivation, where N is a normal form. Then there exist $\mathbb{P}$, Δ and $\mathfrak{s}$ such that $\Gamma \equiv \Delta^\circ$, $\mathbf{P} \equiv \mathbb{P}^\circ$, $\sigma \equiv \mathfrak{s}^\circ$ and

$$\Delta \vdash N \Longmapsto \mathbb{P} : \mathfrak{s} \text{ where } ||\mathbb{P}|| \leq |\mathcal{D}| \leq \mathrm{dsz}(\mathbb{P})$$

IV. Filtration and Bounded Width Theorem

The main result in this section is, informally, that derivations $\mathcal{D} \triangleright \Delta \vdash M \Longmapsto \mathbb{P} : \mathfrak{s}$ can be "tightened" in the sense that intersection types can be *globally* restricted to contain as components only such types that appear in decorations $\mathfrak{s}$ in subterms of $\mathbb{P}$. This property leads to the possibility of bounding the *intersection width* of types needed to type a term as a function of norm (dimension) and term size. Given a derivation $\mathcal{D}$ of an elaboration $\Delta \vdash M \Longmapsto \mathbb{P} : \mathfrak{s}$, let us divide the types appearing in $\mathcal{D}$ into two classes. First, we have the *decoration types*, which appear as components in decorations $\mathfrak{s}$ at subterms of $\mathbb{P}$. More precisely, we define the multiset of decoration types $\mathrm{T}(\mathbb{P})$ as the multiset union of all occurrences of such decorating multisets:

$$\begin{array}{lcl} \mathrm{T}(x\langle\mathfrak{s}\rangle) & = & \mathfrak{s} \\ \mathrm{T}((\lambda x.\mathbb{Q})\langle\mathfrak{s}\rangle) & = & \mathrm{T}(\mathbb{Q}) \uplus \mathfrak{s} \\ \mathrm{T}((\mathbb{Q}\mathbb{R})\langle\mathfrak{s}\rangle) & = & \mathrm{T}(\mathbb{Q}) \uplus \mathrm{T}(\mathbb{R}) \uplus \mathfrak{s} \end{array}$$

Notice that $|\,\mathrm{T}(\mathbb{P})| = \mathrm{dsz}(\mathbb{P})$. Second, we have types which are not decoration types. These are the types appearing as *subformulae* of decoration types, and which do not appear anywhere as decoration types. We refer to such types as *constituent types*. While the number of components of decoration types are bounded by norm, the problem we are concerned with is to somehow bound the number of components of constituent types. We define the function $[\![\bullet]\!]$ on types, which measures the *width* (maximal number of components in) intersections in a type. Write a finite multiset $\mathfrak{s}$ as $\mathfrak{s} = \left[\varphi_1^{n_1}, \ldots, \varphi_m^{n_m}\right]$, where it is understood that all elements φ_i appear with maximal multiplicities n_i for which $\mathfrak{s}(\varphi_i) = n_i$ with $n_i > 0$ (so, in particular, $\varphi_i \neq \varphi_j$ for $i \neq j$). Define

$$\begin{array}{lcl} [\![a]\!] & = & 1 \\ [\![\mathfrak{s} \to \varphi]\!] & = & \max\{[\![\mathfrak{s}]\!], [\![\varphi]\!]\} \\ [\![\left[\varphi_1^{n_1}, \ldots, \varphi_m^{n_m}\right]]\!] & = & \max\{\sum_{i=1}^m n_i, [\![\varphi_1]\!], \ldots, [\![\varphi_m]\!]\} \end{array}$$

The width function is lifted to type environments, elaborations, judgements, and derivations by taking the maximum over all values $[\![\mathfrak{s}]\!]$ such that $\mathfrak{s}$ occurs in the type environment, elaboration, judgement, or derivation. So, for a derivation $\mathcal{D}$, the quantity $[\![\mathcal{D}]\!]$ is the maximal width of any type occurring anywhere in $\mathcal{D}$.

Example 10: Consider the elaboration $\mathbf{P}$ of the term $M \equiv \omega I$ where $\omega \equiv \lambda z.zz$ and $I \equiv \lambda x.x$:

$$(\lambda z.(z\langle(a \to a) \to a \to a\rangle\; z\langle a \to a\rangle)\langle a \to a\rangle)\langle A\rangle\; \mathbf{Q})\langle a \to a\rangle$$

where $A \equiv \sigma \to a \to a$ with $\sigma \equiv (a \to a) \cap ((a \to a) \to a \to a) \cap (a \cap b \cap c \to a)$, and $\mathbf{Q} \equiv (\lambda x.x\langle a, a \to a\rangle)\langle\sigma\rangle$. The types $a \cap b \cap c$, b, c are constituent types. One has $\|\mathbf{P}\| = 2$, $[\![\mathbf{P}]\!] = 3$, and $[\![\vdash M \Longmapsto \mathbf{P} : a \to a]\!] = 3$.

The principal insight behind the following bounded width result is that constituent types are not needed for typing a term, and, as a consequence, we can globally bound the number of components of intersection types in a derivation by norm and size of the λ-term of its concluding judgement. The technical challenge for obtaining the bounded width result lies in showing that such width restriction can be imposed on types recursively, reaching into any depth and position (co- or contravariant) of constituent types, without losing typability and, a fortiori, without increasing norm. We solve the problem by introducing the notion of a *type filtration*. A filtration $\mathscr{T}_X$ on a multiset X of strict types is a function which recursively filters out constituent types among the components of types in a derivation with conclusion elaboration $\mathbb{P}$ which do not appear (with sufficient multiplicity) in X. The filtration property (Proposition 13) shows that filtration on X preserves derivability, provided $\mathrm{T}(\mathbb{P}) \Subset X$. Using this property we can bound intersection type width by $|\,\mathrm{T}(\mathbb{P})|$ by filtering an elaboration $\mathbb{P}$ "diagonally" on its own decoration types, taking $X = \mathrm{T}(\mathbb{P})$ (Theorem 14): if $\mathbb{P}$ is an elaboration for M, then so is $\mathscr{T}_{\mathrm{T}(\mathbb{P})}(\mathbb{P})$, and $\|\mathscr{T}_{\mathrm{T}(\mathbb{P})}(\mathbb{P})\| \leq \|\mathbb{P}\|$. The filtration property is interesting, we believe, in its own right. For it is akin to the *subformula property* of a type system (without being restricted to normal forms), which also holds for intersection types ([3, Lemma 4.5]). In fact, we can show that a strengthened subformula filtration on decoration types leads to a generalization of the subformula property (Theorem 16).

Fix an arbitrary atom b. For a multiset X of strict types we define the the *type filtration* function $\mathscr{T}_X$ by:

$$\begin{aligned} \mathscr{T}_X(a) &\equiv a \\ \mathscr{T}_X(\mathfrak{s} \to \varphi) &\equiv \mathscr{T}_X(\mathfrak{s}) \to \mathscr{T}_X(\varphi) \\ \mathscr{T}_X(\left[\varphi_1^{n_1}, \ldots, \varphi_m^{n_m}\right]) &\equiv \begin{cases} \biguplus_{i=1}^m \left[\mathscr{T}_X(\varphi_i)^{\min\{n_i, X(\varphi_i)\}}\right], \\ \qquad \text{if } X \cap \{\varphi_1, \ldots, \varphi_n\} \neq \emptyset \\ [b]\,, \quad \text{otherwise} \end{cases} \end{aligned}$$

In other words, a multiset $\mathfrak{s}$ is filtrated by filtrating each component with the minimal multiplicity it has in X and in $\mathfrak{s}$ (and in case $\min\{n_i, X(\varphi_i)\} = 0$ in the above definition, the element $\mathscr{T}_X(\varphi_i)$ is not included in the multiset).

A set-theoretic type σ can be filtrated taking X to be a set of set-theoretic types and by sending σ to $\mathscr{T}_X(\sigma)^\circ$ (applying $(\_)^\circ$ is needed because $\biguplus_{i=1}^m \left[\mathscr{T}_X(\varphi_i)^{\min\{n_i, X(\varphi_i)\}}\right]$ is not a set if two or more types map to the same type under $\mathscr{T}_X$).

Clearly, the cardinality of each multiset subformula is bounded by $|X|$:

Lemma 11: For any non-empty multisets X, $\mathfrak{s}$, and strict types φ, one has $[\![\mathscr{T}_X(\mathfrak{s})]\!] \leq |X|$ and $[\![\mathscr{T}_X(\varphi)]\!] \leq |X|$.

We lift filtrations to elaborations by the definition

$$\begin{array}{lcl} \mathscr{T}_X(x\langle\mathfrak{s}\rangle) & \equiv & x\langle\mathscr{T}_X(\mathfrak{s})\rangle \\ \mathscr{T}_X((\lambda x.\mathbb{Q})\langle\mathfrak{s}\rangle) & \equiv & (\lambda x.\mathscr{T}_X(\mathbb{Q}))\langle\mathscr{T}_X(\mathfrak{s})\rangle \\ \mathscr{T}_X((\mathbb{Q}\mathbb{R})\langle\mathfrak{s}\rangle) & \equiv & (\mathscr{T}_X(\mathbb{Q})\mathscr{T}_X(\mathbb{R}))\langle\mathscr{T}_X(\mathfrak{s})\rangle \end{array}$$

We lift $\mathscr{T}_X$ to type environments by pointwise application to types, to judgements by $\mathscr{T}_X(\Delta \vdash \mathbb{P} : \mathfrak{s}) \equiv \mathscr{T}_X(\Delta) \vdash \mathscr{T}_X(\mathbb{P}) : \mathscr{T}_X(\mathfrak{s})$, and to derivations by applying $\mathscr{T}_X$ to each judgement.

Lemma 12: Let X be a multiset.

1) If $\mathfrak{s} = \left[\varphi_1^{n_1}, \ldots, \varphi_m^{n_m}\right]$ and $\mathfrak{s} \Subset X$, then

$$\mathscr{T}_X(\mathfrak{s}) \equiv \biguplus_{i=1}^{m} [\mathscr{T}_X(\varphi_i)^{n_i}]$$

2) If $\mathfrak{s}_1 \uplus \mathfrak{s}_2 \Subset X$, then

$$\mathscr{T}_X(\mathfrak{s}_1 \uplus \mathfrak{s}_2) \equiv \mathscr{T}_X(\mathfrak{s}_1) \uplus \mathscr{T}_X(\mathfrak{s}_2)$$

3) If $\mathrm{T}(\mathbb{P}_1) \uplus \mathrm{T}(\mathbb{P}_2) \Subset X$, then

$$\mathscr{T}_X(\mathbb{P}_1 \boxplus \mathbb{P}_2) \equiv \mathscr{T}_X(\mathbb{P}_1) \boxplus \mathscr{T}_X(\mathbb{P}_2)$$

Proof: The first property follows easily from definitions, the second follows easily from the first, and the third from the second. ■

Proposition 13 (Filtration): Whenever $\mathcal{D} \triangleright \Delta \vdash \mathbb{P} : \mathfrak{s}$ is a derivation and $\mathrm{T}(\mathbb{P}) \Subset X$, then $\mathscr{T}_X(\mathcal{D})$ is a derivation. If $\mathcal{D}$ is a set-theoretic derivation, then $\mathscr{T}_X(\mathcal{D})^\circ$ is again a set-theoretic derivation.

Proof: Induction on the derivation $\mathcal{D}$. It is easy to check in each case that from a set-theoretic derivation the inductively obtained derivation $\mathscr{T}_X(\mathcal{D})^\circ$ is a set-theoretic derivation, and we will not mention this fact explicitly in each case.

Case $\mathcal{D}$ is

$$\frac{}{\Delta, x : [\varphi_1, \ldots, \varphi_n] \vdash x\langle[\varphi_i]\rangle : [\varphi_i]}(\mathrm{var})$$

In this case $\mathbb{P} \equiv x\langle[\varphi_i]\rangle$, and $\mathscr{T}_X(\mathcal{D})$ is

$$\frac{}{\mathscr{T}_X(\Delta), x : \mathscr{T}_X([\varphi_1, \ldots, \varphi_n]) \vdash x\langle\mathscr{T}_X([\varphi_i])\rangle : \mathscr{T}_X([\varphi_i])}(\mathrm{var})$$

Because $\varphi_i \in \mathrm{T}(\mathbb{P})$ and $\mathrm{T}(\mathbb{P}) \Subset X$, it follows that $\varphi_i \in X$, and hence by definition of $\mathscr{T}_X$ we have $\mathscr{T}_X([\varphi_1, \ldots, \varphi_n]) \equiv [\ldots, \mathscr{T}_X(\varphi_i), \ldots]$ and $\mathscr{T}_X([\varphi_i]) \equiv [\mathscr{T}_X(\varphi_i)]$, and so $\mathscr{T}_X(\mathcal{D})$ is a derivation.

Case $\mathcal{D}$ is

$$\frac{\mathcal{D}_1 \triangleright \Delta \vdash \mathbb{Q} : [\mathfrak{s} \to \varphi] \quad \mathcal{D}_2 \triangleright \Delta \vdash \mathbb{R} : \mathfrak{s}}{\Delta \vdash (\mathbb{Q}\mathbb{R})\langle[\varphi]\rangle : [\varphi]}(\to\mathrm{E})$$

Let $\mathbb{P} \equiv (\mathbb{Q}\mathbb{R})\langle[\varphi]\rangle$ with $\mathrm{T}(\mathbb{P}) \Subset X$ and let $\mathfrak{s} = [\varphi_1, \ldots, \varphi_n]$. We have $\mathfrak{s} \Subset \mathrm{T}(\mathbb{P})$ for $i = 1 \ldots n$, and hence $\mathfrak{s} \Subset X$. It follows from Lemma 12 that $\mathscr{T}_X(\mathfrak{s}) \equiv [\mathscr{T}_X(\varphi_1), \ldots, \mathscr{T}_X(\varphi_n)]$. Moreover, because $[\mathfrak{s} \to \varphi]$ and $[\varphi]$ appear in $\mathbb{P}$, we have by Lemma 12, $\mathscr{T}_X([\mathfrak{s} \to \varphi]) \equiv [\mathscr{T}_X(\mathfrak{s}) \to \mathscr{T}_X(\varphi)]$ and $\mathscr{T}_X([\varphi]) \equiv [\mathscr{T}_X(\varphi)]$. By induction hypothesis we then obtain derivations

$$\mathscr{T}_X(\mathcal{D}_1) \triangleright \mathscr{T}_X(\Delta) \vdash \mathscr{T}_X(\mathbb{Q}) : [\mathscr{T}_X(\mathfrak{s}) \to \mathscr{T}_X(\varphi)]$$

and

$$\mathscr{T}_X(\mathcal{D}_2) \triangleright \mathscr{T}_X(\Delta) \vdash \mathscr{T}_X(\mathbb{R}) : \mathscr{T}_X(\mathfrak{s})$$

and by $(\to\mathrm{E})$ we obtain a derivation of

$$\mathscr{T}_X(\mathcal{D}) \triangleright \mathscr{T}_X(\Delta) \vdash (\mathscr{T}_X(\mathbb{Q})\mathscr{T}_X(\mathbb{R}))\langle[\mathscr{T}_X(\varphi)]\rangle : [\mathscr{T}_X(\varphi)]$$

which shows the claim.

Case $\mathcal{D}$ is

$$\frac{\mathcal{D}_1 \triangleright \Delta, x : \mathfrak{s} \vdash \mathbb{Q} : [\varphi]}{\Delta \vdash (\lambda x.\mathbb{Q})\langle[\mathfrak{s} \to \varphi]\rangle : [\mathfrak{s} \to \varphi]}(\to\mathrm{I})$$

Let $\mathbb{P} \equiv (\lambda x.\mathbb{Q})\langle[\mathfrak{s} \to \varphi]\rangle$ with $\mathrm{T}(\mathbb{P}) \Subset X$. Then $\mathfrak{s} \to \varphi \in X$ and $\varphi \in X$. Therefore, Lemma 12 shows $\mathscr{T}_X([\mathfrak{s} \to \varphi]) = [\mathscr{T}_X(\mathfrak{s}) \to \mathscr{T}_X(\varphi)]$. By induction hypothesis, we have a derivation

$$\mathscr{T}_X(\mathcal{D}_1) \triangleright \mathscr{T}_X(\Delta), x : \mathscr{T}_X(\mathfrak{s}) \vdash \mathscr{T}_X(\mathbb{Q}) : \mathscr{T}_X([\varphi])$$

and by $(\to\mathrm{I})$ we obtain a derivation $\mathscr{T}_X(\mathcal{D})$ of

$$\mathscr{T}_X(\Delta) \vdash (\lambda x.\mathscr{T}_X(\mathbb{Q}))\langle[\mathscr{T}_X(\mathfrak{s}) \to \mathscr{T}_X(\varphi)]\rangle : [\mathscr{T}_X(\mathfrak{s}) \to \mathscr{T}_X(\varphi)]$$

which shows the claim.

Case $\mathcal{D}$ is

$$\frac{\mathcal{D}_i \triangleright \Delta \vdash \mathbb{P}_i : [\varphi_i] \;\; (i = 1 \ldots n) \quad (\star)}{\Delta \vdash \boxplus_{i=1}^n \mathbb{P}_i : [\varphi_1, \ldots, \varphi_n]}(\cap\mathrm{I})$$

where, by the side condition $(\star)$, we have $\lceil \mathbb{P}_i \rceil \equiv \lceil \mathbb{P}_j \rceil$ for all $i, j = 1 \ldots n$ and, writing $\mathbb{P} \equiv \boxplus_{i=1}^n \mathbb{P}_i$, we have

$$\forall x \in \mathrm{fv}(\lceil \mathbb{P} \rceil).\forall \mathfrak{s} \in \mathrm{ve}_x(\mathbb{P}).\ \Delta(x)(\varphi) \geq \mathfrak{s}(\varphi) \qquad (1)$$

Let $\mathrm{T}(\mathbb{P}) \Subset X$. By induction hypothesis we have derivations

$$\mathscr{T}_X(\mathcal{D}_i) \triangleright \mathscr{T}_X(\Delta) \vdash \mathscr{T}_X(\mathbb{P}_i) : [\mathscr{T}_X(\varphi_i)] \;\; (i = 1 \ldots n)$$

Consider any $\mathfrak{s} \in \mathrm{ve}_x(\mathbb{P})$ for arbitrary x and assume $\mathfrak{s} = \left[\psi_1^{n_1}, \ldots, \psi_m^{n_m}\right]$. By definition of $\mathrm{T}(\mathbb{P})$ and $\mathrm{T}(\mathbb{P}) \Subset X$, Lemma 12 shows $\mathscr{T}_X(\mathfrak{s}) \equiv \biguplus_{j=1}^m \left[\mathscr{T}_X(\psi_j)^{n_j}\right]$. It then follows from (1) that we have, for any $\psi \in \mathfrak{s}$, $\mathscr{T}_X(\Delta)(x)(\mathscr{T}_X(\psi)) \geq \mathscr{T}_X(\mathfrak{s})(\mathscr{T}_X(\psi))$, thereby showing that side condition $(\star)$ is satisfied for $\mathscr{T}_X(\Delta)$ and $\boxplus_{i=1}^n \mathscr{T}_X(\mathbb{P}_i)$. Moreover, Lemma 12 shows

$$\mathscr{T}_X(\biguplus_{i=1}^n [\varphi_i]) \equiv \biguplus_{i=1}^n [\mathscr{T}_X(\varphi_i)] \text{ and } \mathscr{T}_X(\boxplus_{i=1}^n \mathbb{P}_i) \equiv \boxplus_{i=1}^n \mathscr{T}_X(\mathbb{P}_i)$$

Therefore, by $(\cap\mathrm{I})$, we obtain a derivation of

$$\mathscr{T}_X(\Delta) \vdash \mathscr{T}_X(\boxplus_{i=1}^n \mathbb{P}_i) : \mathscr{T}_X([\varphi_1, \ldots, \varphi_n])$$

which shows the claim. ■

Let $|\mathfrak{s}|_T$ denote the number of nodes in the syntax tree of $\mathfrak{s}$ and let $|\Delta|_T = \sum\{|\Delta(x)|_T \mid x \in \mathrm{dom}(\Delta)\}$. Recall that $|M|$ denotes the number of nodes in the syntax tree of M.

Theorem 14 (Bounded Width Property): Let a derivation $\mathcal{D} \triangleright \Delta \vdash M \Longmapsto \mathbb{P} : \mathfrak{s}$ be given with $\|\mathbb{P}\| \leq d$. Then the following conditions are true:

1) There exists a derivation $\mathcal{D}' \triangleright \Delta' \vdash M \Longmapsto \mathbb{P}' : \mathfrak{s}'$ such that $\llbracket \mathcal{D}' \rrbracket \leq d \cdot |M|$ and $\|\mathbb{P}'\| = \|\mathbb{P}\|$.
2) There exists a derivation $\mathcal{D}'' \triangleright \Delta \vdash M \Longmapsto \mathbb{P}'' : \mathfrak{s}''$ such that $\llbracket \mathcal{D}'' \rrbracket \leq |\Delta|_T + d \cdot |M|$ and $\|\mathbb{P}''\| = \|\mathbb{P}\|$.
3) There exists a derivation $\mathcal{D}''' \triangleright \Delta \vdash M \Longmapsto \mathbb{P}''' : \mathfrak{s}$ such that $\llbracket \mathcal{D}''' \rrbracket \leq |\Delta|_T + |\mathfrak{s}|_T + d \cdot |M|$ and $\|\mathbb{P}'''\| = \|\mathbb{P}\|$.
4) In case the given derivation $\mathcal{D}$ is set-theoretic, then the derivations $(\mathcal{D}')^\circ$, $(\mathcal{D}'')^\circ$ and $(\mathcal{D}''')^\circ$ are set-theoretic, and the corresponding set-theoretic elaborations $\mathbf{P}$ all satisfy $\|\mathbf{P}\| \leq \|\mathbb{P}^\circ\|$ for $\mathbf{P} \equiv (\mathbb{P}')^\circ, (\mathbb{P}'')^\circ, (\mathbb{P}''')^\circ$.

Proof: To prove the first claim, we consider the "diagonal" filtrations $\mathscr{T}_{\mathrm{T}(\mathbb{P})}(\mathbb{P})$: Given a derivation $\mathcal{D} \triangleright \Delta \vdash M \Longmapsto \mathbb{P} : \mathfrak{s}$ take, by Proposition 13, as $\mathcal{D}'$ the derivation

$$\mathscr{T}_{\mathrm{T}(\mathbb{P})}(\mathcal{D}) \triangleright \mathscr{T}_{\mathrm{T}(\mathbb{P})}(\Delta) \vdash M \Longmapsto \mathscr{T}_{\mathrm{T}(\mathbb{P})}(\mathbb{P}) : \mathscr{T}_{\mathrm{T}(\mathbb{P})}(\mathfrak{s})$$

By Lemma 11 we have $\llbracket \mathscr{T}_{\mathrm{T}(\mathbb{P})}(\mathfrak{s}) \rrbracket \leq |\mathrm{T}(\mathbb{P})|$ for all $\mathfrak{s}$, which shows that $\llbracket \mathscr{T}_{\mathrm{T}(\mathbb{P})}(\mathcal{D}) \rrbracket \leq |\mathrm{T}(\mathbb{P})|$. Moreover, since $|T(\mathbb{P})| \leq \|\mathbb{P}\| \cdot |M|$, we have

$$|\mathrm{T}(\mathbb{P})| \leq \|\mathbb{P}\| \cdot |M| \leq d \cdot |M|$$

Finally, inspection of the definition of $\mathscr{T}_X$ shows that $\|\mathscr{T}_X(\mathbb{P})\| = \|\mathbb{P}\|$, since distinct types mapping to the same type under filtration appear multiply in multisets, whereas for set-theoretic derivations, we have $\|\mathscr{T}_X(\mathbb{P})^\circ\| \leq \|\mathbb{P}\|$.

To prove the second claim, let X_Δ be the multiset of strict type subformula occurrences of types in the range of Δ. Then we have $\mathscr{T}_{X_\Delta}(\mathfrak{s}) = \mathfrak{s}$ for all types $\mathfrak{s}$ appearing in Δ.

Let $X = X_\Delta \uplus \mathrm{T}(\mathbb{P})$. We apply Proposition 13 to $\mathcal{D}$ with the extended filtration $\mathcal{T}_X$ to obtain a derivation $\mathcal{D}''$

$$\mathcal{T}_X(\mathcal{D}) \triangleright \mathcal{T}_X(\Delta) \vdash M \Longmapsto \mathcal{T}_X(\mathbb{P}) : \mathcal{T}_X(\mathfrak{s})$$

Since we have $\mathcal{T}_X(\Delta) = \Delta$, the desired judgement is thereby obtained. Moreover, using $|T(\mathbb{P})| \leq \|\mathbb{P}\| \cdot |M|$, we have $|X| = |X_\Delta| + |\mathrm{T}(\mathbb{P})| \leq |\Delta|_T + d \cdot |M|$ and the claim follows from Lemma 11.

To prove the third claim, we consider the filtration set $X = X_\Delta \uplus X_\mathfrak{s} \uplus \mathrm{T}(\mathbb{P})$, where X_Δ is defined as above and $X_\mathfrak{s}$ is the multiset of subformula occurrences in $\mathfrak{s}$. We now have $\mathcal{T}_X(\Delta) = \Delta$ and $\mathcal{T}_X(\mathfrak{s}) = \mathfrak{s}$. We apply Proposition 13 as above, now using the extended filtration $\mathcal{T}_X$, and obtain the desired judgement with the claimed properties. ■

Example 15: Consider again the term with elaboration **P** from Example 10. The filtrated elaboration $\mathbf{P}' \equiv \mathcal{T}_{\mathrm{T}(\mathbf{P})}(\mathbf{P})$ is:

$$(\lambda z.(z\langle(a \to a) \to a \to a\rangle\ z\langle a \to a\rangle)\langle a \to a\rangle)\langle A'\rangle\ \mathbf{Q})\langle a\rangle$$

where $A' \equiv \sigma' \to a \to a$ with $\sigma' \equiv (a \to a) \cap ((a \to a) \to a \to a)$, and $\mathbf{Q} \equiv (\lambda x.x\langle a, a \to a\rangle)\langle\sigma'\rangle$. There are no constituent type components. The type $(a \to a) \cap ((a \to a) \to a \to a) \cap (a \cap b \cap c \to a)$ from **P** gets filtrated under $\mathcal{T}_{\mathrm{T}(\mathbf{P})}$ into $(a \to a) \cap ((a \to a) \to a \to a)$, because the constituent $(a \cap b \cap c \to a)$ is filtrated into $a \to a$ and is removed in σ' by idempotence.

Given a judgement $\Delta \vdash M \Longmapsto \mathbb{P} : \mathfrak{s}$, let us call $\mathfrak{s}$ and the types appearing in Δ *observable types*. The classical subformula property for intersection types [3, Lemma 4.5] can in our terminology be formulated as: If N is a normal form such that $\Delta \vdash N \Longmapsto \mathbb{P} : \mathfrak{s}$, then there is a derivation of this judgement in which every decoration type is a subformula of some observable type. A more radical form of filtration leads to the following theorem which holds for *all* derivations (not just those for normal forms). In the following we mean by a "strict type subformula" a strict type, which is a subformula.

Theorem 16 (Subformula Filtration): If $\Delta \vdash M \Longmapsto \mathbb{P} : \mathfrak{s}$ is derivable, then the following conditions are true:

- There is a derivation $\mathcal{D}'$ of $\Delta \vdash M \Longmapsto \mathbb{P}' : \mathfrak{s}$ such that every strict type subformula of every type in $\mathcal{D}'$ is either a decoration type or a subformula of an observable type.
- There is a derivation $\mathcal{D}''$ of $\Delta'' \vdash M \Longmapsto \mathbb{P}'' : \mathfrak{s}''$ such that every strict type subformula of every type in $\mathcal{D}''$ is also a decoration type.

Proof: For a multiset X of strict types we define the filtration function $\mathcal{T}^*_X$ as follows. Let $\psi \in X$ be a type of minimal size (if ambiguous take lexicographical minimum) such that any strict type subformula φ of ψ, excluding ψ itself, is not in X. Let $b \equiv \psi$ if ψ is a constant, otherwise fix any type constant b. Define $\mathcal{T}^*_X$ by applying the following cases in the order stated:

$$\mathcal{T}^*_X(\psi) \equiv b$$
$$\mathcal{T}^*_X(a) \equiv \begin{cases} a, & \text{if } a \in X \\ b, & \text{otherwise} \end{cases}$$
$$\mathcal{T}^*_X(\mathfrak{s} \to \varphi) \equiv \begin{cases} \mathcal{T}^*_X(\mathfrak{s}), \to \mathcal{T}^*_X(\varphi), & \text{if } \mathfrak{s} \to \varphi \in X \\ b, & \text{otherwise} \end{cases}$$
$$\mathcal{T}^*_X([\varphi_1^{n_1}, \ldots, \varphi_m^{n_m}]) \equiv \begin{cases} \biguplus_{i=1}^m \left[\mathcal{T}^*_X(\varphi_i)^{\min\{n_i, X(\varphi_i)\}}\right], \\ \quad \text{if } X \cap \{\varphi_1, \ldots, \varphi_n\} \neq \emptyset \\ [b], \text{ otherwise} \end{cases}$$

By construction $b \in \mathcal{T}^*_X(X)$, therefore for any type $\mathfrak{s}$ (resp. φ) and any strict type subformula φ' of $\mathcal{T}^*_X(\mathfrak{s})$ (resp. $\mathcal{T}^*_X(\varphi)$) there exists a type $\varphi'' \in X$ such that $\varphi' = \mathcal{T}^*_X(\varphi'')$.

Notice that the properties of Lemma 12 hold for $\mathcal{T}^*_X$. Proceed as in the proofs of Proposition 13 and Theorem 14, using the filtration $\mathcal{T}^*_X$ with X the appropriate extension of $\mathrm{T}(\mathbb{P})$ for the first claim and with $X = \mathrm{T}(\mathbb{P})$ for the second claim. Due to the diagonal choice of X the type ψ can neither appear in the first premise of rule ($\to$E) nor in the conclusion of rule ($\to$I). Therefore, if $\psi \not\equiv b$ the definition $\mathcal{T}^*_X(\psi) \equiv b$ does not remove any necessary arrow in the derivation. ■

Observe that the concept of subformula filtration differs from the concept of relevance [15, Definition 9.1]. For example, the type $[[a] \to a] \to [a] \to a$ can be assigned to the term $\lambda x.x$ in a relevant system, while the subformula a would not appear as a decoration. In general, relevance operates at the level of type environments, whereas filtration operates at the level of subformulae.

Example 17: In order to illustrate Theorem 16, consider types $\varphi_a = [a] \to a$, $\varphi_b = [b] \to b$, $\varphi_c = [c] \to c$, type environment $\Delta = \{x : [[\varphi_a] \to \varphi_b], y : [\varphi_a, \varphi_c]\}$ and elaboration $\mathbb{P} = (x\langle[[\varphi_a] \to \varphi_b]\rangle\, y\langle[\varphi_a]\rangle)\langle[\varphi_b]\rangle$ in the derivation $\mathcal{D}$:

$$\frac{\overline{\Delta \vdash x\langle[[\varphi_a] \to \varphi_b]\rangle : [[\varphi_a] \to \varphi_b]} \quad \overline{\Delta \vdash y\langle[\varphi_a]\rangle : [\varphi_a]}}{\Delta \vdash \mathbb{P} : [\varphi_b]}$$

Following the proof of Theorem 16 we take $X = \mathrm{T}(\mathbb{P}) = [[\varphi_a] \to \varphi_b, \varphi_a, \varphi_b]$. The smallest type $\psi \in X$ such that none of its subformulae (except itself) is in X is $\psi \equiv \varphi_a$, which is not a type constant, hence we fix some type constant d. Let $\varphi_d = [d] \to d$. Applying filtration $\mathcal{T}^*_X$ we obtain the filtrated derivation $\mathcal{T}^*_X(\mathcal{D}) =$

$$\frac{\overline{\mathcal{T}^*_X(\Delta) \vdash x\langle[[d] \to \varphi_d]\rangle : [[d] \to \varphi_d]} \quad \overline{\mathcal{T}^*_X(\Delta) \vdash y\langle[d]\rangle : [d]}}{\mathcal{T}^*_X(\Delta) \vdash \mathcal{T}^*_X(\mathbb{P}) : [\varphi_d]}$$

where $\mathcal{T}^*_X(\Delta) = \{x : [[d] \to \varphi_d], y : [d]\}$ and $\mathcal{T}^*_X(\mathbb{P}) = (x\langle[[d] \to \varphi_d]\rangle\, y\langle[d]\rangle)\langle[\varphi_d]\rangle$. We observe the following aspects

- The type φ_a is replaced by d, which does not invalidate the derivation.
- The type φ_b is replaced by φ_d because its subformulae do not occur in X and are replaced by d.
- The type φ_c is filtered out completely.
- $\mathrm{T}(\mathcal{T}^*_X(\mathbb{P})) = [[d] \to \varphi_d, d, \varphi_d] \ni d = \mathcal{T}^*_X(\varphi_a)$ and each subformula in any type in $\mathcal{T}^*_X(\mathcal{D})$ is in $\mathrm{T}(\mathcal{T}^*_X(\mathbb{P}))$.

V. Typability in Set-Theoretic Dimension

We prove (Theorem 25) Pspace-completeness of the typability problem in bounded set-theoretic dimension: given a λ-term M and a dimension $d > 0$ does there exist Γ and σ such that $\Gamma \vdash_d M : \sigma$?

A. Pspace *Lower Bound*

We reduce the Pspace-complete problem of quantified Boolean formula (QBF) satisfiability to typability in constant dimension 4 using tools from [11], [12].

Given an open QBF F over the signature $\forall, \wedge, \neg$ we use the embedding $[\![F]\!]$ into λ-calculus (cf. [11]) defined as follows

$$\begin{array}{ll} \mathbf{t} = \lambda xy.x \quad \mathbf{f} = \lambda xy.y & [\![x]\!] = x \\ [\![\neg]\!] = \lambda bxy.b\,y\,x & [\![\wedge]\!] = \lambda b_1 b_2 xy.b_1\,(b_2\,x\,y)\,y \\ [\![\forall]\!] = \lambda z.[\![\wedge]\!]\,(z\,\mathbf{t})\,(z\,\mathbf{f}) & [\![\neg F]\!] = [\![\neg]\!][\![F]\!] \\ [\![F \wedge G]\!] = [\![\wedge]\!]\,[\![F]\!]\,[\![G]\!] & [\![\forall x.F(x)]\!] = [\![\forall]\!]\,(\lambda x.[\![F(x)]\!]) \end{array}$$

Lemma 18 ([11], Thm. 5.2): Given a closed QBF F, F is true iff $[\![F]\!] \twoheadrightarrow_\beta \mathbf{t}$.

Next, given a QBF F we inspect dimensionality of typings of $[\![F]\!]$. Since $[\![F]\!]$ has the type $o \to o \to o$ in the simply typed λ-calculus, $[\![F]\!]$ is typable in dimension 1. However, to capture truth of F in the type of $[\![F]\!]$, we are interested in the question whether $[\![F]\!]$ can be assigned types of $\mathbf{t}$ that are not types of $\mathbf{f}$. Interestingly, bounded set dimension of 4 is enough to achieve such a distinction.

Lemma 19: Given a closed QBF F and types A, B we have

1) If F is true, then $\vdash_4 [\![F]\!] : A \to B \to A$
2) If F is false, then $\vdash_4 [\![F]\!] : A \to B \to B$

Proof: We show a more general claim: given an open QBF F, a valuation $v : \mathrm{var}(F) \to \{\mathbf{true}, \mathbf{false}\}$ and two types A, B, let $\top \equiv A \to B \to A$, $\bot \equiv A \to B \to B$ and define $[\![v]\!] = \{x : \top \mid v(x) = \mathbf{true}\} \cup \{y : \bot \mid v(y) = \mathbf{false}\}$. We show

1) If F is true under v, then $[\![v]\!] \vdash_4 [\![F]\!] : \top$
2) If F is false under v, then $[\![v]\!] \vdash_4 [\![F]\!] : \bot$

In the proof we ensure that embedded Boolean functions are decorated with corresponding logical function tables, i.e. $\mathbf{t}$ with $[\top]$, $\mathbf{f}$ with $[\bot]$, $[\![\neg]\!]$ with at most $[\top \to \bot, \bot \to \top]$, $[\![\wedge]\!]$ with at most $[\top \to \top \to \top, \top \to \bot \to \bot, \bot \to \top \to \bot, \bot \to \bot \to \bot]$ and $[\![\forall]\!]$ with at most $[((\top \to \top) \cap (\bot \to \top)) \to \top, ((\top \to \top) \cap (\bot \to \bot)) \to \bot, ((\top \to \bot) \cap (\bot \to \top)) \to \bot, ((\top \to \bot) \cap (\bot \to \bot)) \to \bot]$. We proceed by induction on F, the most interesting case being $F = \forall x.G(x)$: First, if F is true under v, then $G(x)$ is true under both $v[x := \mathbf{true}]$ and $v[x := \mathbf{false}]$. By induction hypothesis, $[\![v[x := \mathbf{true}]]\!] \vdash_4 [\![G(x)]\!] : \top$ and $[\![v[x := \mathbf{false}]]\!] \vdash_4 [\![G(x)]\!] : \top$. By typing $\vdash_4 [\![\forall]\!] : ((\top \to \top) \cap (\bot \to \top)) \to \top$ and $[\![v]\!] \vdash_4 \lambda x.[\![G(x)]\!] : (\top \to \top) \cap (\bot \to \top)$ we obtain $[\![v]\!] \vdash_4 [\![\forall]\!]\,(\lambda x.[\![G(x)]\!]) : \top$. Note that the typing of $\lambda x.[\![G(x)]\!]$ with both function types in dimension 4 relies on typing all subterms with corresponding (at most 4) function types. Second, if F is false under v, then $G(x)$ is false under $v[x := \mathbf{true}]$ or $v[x := \mathbf{false}]$ encompassing three subcases that are analogous to the previous one. ■

Lemma 19 shows that type checking in bounded set dimension is Pspace-hard. However, it does not necessarily establish a lower bound on typability because $[\![F]\!]$ can easily be typed by $o \to o \to o$ in the simply typed λ-calculus. To establish the lower bound we use subject reduction together with normalization.

Proposition 20 (Lower Bound for Typability): The typability problem in set-theoretic dimension 4 is Pspace-hard.

Proof: Given a closed QBF F we inspect typability of $G = ([\![F]\!]\,\omega\,I)\,([\![F]\!]\,\omega\,I)$, where $I = \lambda x.x$ and $\omega = \lambda x.x\,x$. Let $A = (a \cap (a \to a)) \to a$, $B_1 = a \to a$ and $B_2 = B_1 \to B_1$. Next, we show that F is false iff G is typable in dimension 4.

If F is false, then by Lemma 19 we have $\vdash_4 [\![F]\!] : A \to B_i \to B_i$ for $i = 1, 2$. Therefore, $\vdash_4 G : B_1$.

If F is true, then by Lemma 18 we have $[\![F]\!] \twoheadrightarrow_\beta \mathbf{t}$. Therefore, $G \twoheadrightarrow_\beta \omega\,\omega$ which is not normalizing and hence not typable in the intersection type system, particularly not in bounded dimension. By subject reduction, G is not typable. ■

B. Pspace *Upper Bound*

We show that the typability problem in bounded set-theoretic dimension is in Pspace. The key instruments for constructing the upper bound are the bounded width theorem (Theorem 14) together with a nondeterministic transformation of typability into a constraint resolution problem which can be solved by standard unification.

We assume all intersections are ordered lexicographically and are duplicate free. Therefore, for any types σ, τ we have $\sigma = \tau$ iff $\sigma \equiv \tau$. The use of normalized types is also wlog. imposed on derivations by sorting premises accordingly in the $(\cap\mathrm{I})$ rule and avoiding duplicates. An *elaboration schema* $\hat{\mathbf{P}}, \hat{\mathbb{P}}$ (resp. environment schema, judgement schema, type schema) is an elaboration (resp. environment, judgement, type) in which type variables, ranged over by α, β, γ etc. occur. The intention is that such variables always range over types in $\mathcal{M}_0$ and $\mathcal{S}_0$. We always use the explicit syntax $\hat{\Gamma} \vdash M \longmapsto \hat{\mathbf{P}} : [\alpha_1, \ldots, \alpha_n]$ to indicate intersections in judgement schemata, also using this notation for set-theoretic judgements to indicate the connection between judgement schemata and judgements.

Procedure 21 ($\mathscr{S}(M, d)$): Given a λ-term M and dimension d we decide typability of M in bounded set-theoretic dimension d as follows.

1) Construct an elaboration schema $\hat{\mathbf{P}}$ from M as follows. For each subterm of M choose $l \in \{1, \ldots, d\}$ and decorate the corresponding subterm of $\hat{\mathbf{P}}$ with $\langle[\alpha_1, \ldots, \alpha_l]\rangle$ where α_i are fresh for $i = 1 \ldots l$.
2) For each variable α occurring in $\hat{\mathbf{P}}$ choose $l \in \{0, \ldots, d \cdot |M|\}$ and let $\ulcorner\alpha\urcorner = \begin{cases} \alpha, & \text{if } l = 0 \\ (\beta_1 \cap \ldots \cap \beta_l) \to \beta, & \text{otherwise} \end{cases}$ where $\beta, \beta_1, \ldots, \beta_l$ are fresh.
3) Construct the set of constraints $C = \{\alpha \doteq \ulcorner\alpha\urcorner \mid \alpha \text{ occurs in } \hat{\mathbf{P}}\}$.
4) Construct a type environment schema $\hat{\Gamma}$ as follows. Starting with $\emptyset$, for each free variable x in M choose $l \in \{1, \ldots, d \cdot |M|\}$ and add $x : [\alpha_1, \ldots, \alpha_l]$ to $\hat{\Gamma}$ where α_i are fresh for $i = 1 \ldots l$.

5) Starting with the judgement schema $\hat{\Gamma} \vdash M \longmapsto \hat{\mathbf{P}} : [\alpha]$, where α is fresh, extend the set of constraints C using the following steps:
   - From a judgement schema $\hat{\Gamma}', x : [\alpha_1, \dots, \alpha_l] \vdash x \longmapsto x\langle[\beta]\rangle : [\gamma]$ choose an $i \in \{1, \dots, l\}$, and add $\beta \doteq \gamma$ and $\alpha_i \doteq \gamma$ to C.
   - From a judgement schema $\hat{\Gamma}' \vdash \lambda x.N \longmapsto (\lambda x.\hat{\mathbf{Q}})\langle[\beta]\rangle : [\gamma]$ where $\ulcorner\beta\urcorner = (\alpha_1 \cap \dots \cap \alpha_l) \to \alpha$, add $\beta \doteq \gamma$ to C (if $\ulcorner\beta\urcorner = \beta$, reject). Proceed recursively with $\hat{\Gamma}', x : [\alpha_1, \dots, \alpha_l] \vdash N \longmapsto \hat{\mathbf{Q}} : [\alpha]$.
   - From a judgement schema $\hat{\Gamma}' \vdash N\,L \longmapsto (\hat{\mathbf{Q}}\,\hat{\mathbf{R}})\langle[\beta]\rangle : [\gamma]$ where $\hat{\mathbf{Q}} \equiv \hat{\mathbf{T}}\langle[\delta]\rangle$, $\ulcorner\delta\urcorner = (\alpha_1 \cap \dots \cap \alpha_l) \to \alpha$ and $l \leq d$, add $\beta \doteq \gamma$ and $\alpha \doteq \beta$ to C (if $\ulcorner\delta\urcorner = \delta$, reject). Choose a decomposition $\hat{\mathbf{R}}_1, \dots, \hat{\mathbf{R}}_l$ such that $\hat{\mathbf{R}} \equiv \bigsqcup_{i=1}^{l} \hat{\mathbf{R}}_i$. Proceed recursively with $\hat{\Gamma}' \vdash N \longmapsto \hat{\mathbf{Q}} : [\delta]$ and $\hat{\Gamma}' \vdash L \longmapsto \hat{\mathbf{R}}_i : [\alpha_i]$ for $i = 1 \dots l$.
6) If the syntactic unification instance C has no solution, then fail. Otherwise, compute the most general unifier U of C.
7) Fix a type constant a and let T be a substitution with $T(\alpha) = a$ for each variable α in the codomain of U. If $T(U(\alpha))$ is a strict type for each α in C, then succeed, otherwise, fail.

We show that procedure $\mathscr{S}$ is sound (Lemma 22) and complete (Lemma 23), and in Proposition 24 we establish the PSPACE upper bound.

Lemma 22 (Soundness of $\mathscr{S}$): Given a λ-term M and a dimension d, if $\mathscr{S}(M, d)$ succeeds, then there exists a type environment Γ and a type A such that $\Gamma \vdash_d M : A$.

Proof: Let $S = T \circ U$ be the substitution constructed in step 7 with a strict type codomain. We verify by induction on M that for each intermediate judgement schema $\hat{\Gamma}' \vdash N \longmapsto \hat{\mathbf{Q}} : [\beta]$ in step 5 we have $S(\hat{\Gamma}') \vdash N \longmapsto S(\hat{\mathbf{Q}}) : [S(\beta)]$ with $\|S(\hat{\mathbf{Q}})\| \leq d$, where $S(\hat{\Gamma}')$ and $S(\hat{\mathbf{Q}})$ are defined by replacing each occurring variable with the corresponding strict type. Since the codomain of S is strict, cardinalities of annotations do not increase. We proceed by induction on M.

Case x. Follows immediately from rule (var).

Case $\lambda x.N$. By induction hypothesis we have $S(\hat{\Gamma}'), x : [S(\alpha_1), \dots, S(\alpha_l)] \vdash N \longmapsto S(\hat{\mathbf{Q}}) : [S(\alpha)]$. By construction $S(\beta) = S(\gamma) = S((\alpha_1 \cap \dots \cap \alpha_l) \to \alpha)$. Using rule ($\to$I) we obtain $S(\hat{\Gamma}') \vdash \lambda x.N \longmapsto (\lambda x.S(\hat{\mathbf{Q}}))\langle[S(\beta)]\rangle : [S(\gamma)]$. Additionally, $\|S((\lambda x.\hat{\mathbf{Q}})\langle[\beta]\rangle)\| = \|S(\hat{\mathbf{Q}})\| \leq d$.

Case $N\,L$. By induction hypothesis we have $S(\hat{\Gamma}') \vdash N \longmapsto S(\hat{\mathbf{Q}}) : [S((\alpha_1 \cap \dots \cap \alpha_l) \to \alpha)]$ and $S(\hat{\Gamma}') \vdash L \longmapsto S(\hat{\mathbf{R}}_i) : [S(\alpha_i)]$. Since $S(\hat{\mathbf{R}}) = S(\bigsqcup_{i=1}^{l} \hat{\mathbf{R}}_i) = \bigsqcup_{i=1}^{l} S(\hat{\mathbf{R}}_i)$ we can apply ($\cap$I) followed by ($\to$E) to obtain $S(\hat{\Gamma}') \vdash N\,L \longmapsto (S(\hat{\mathbf{Q}})\ S(\hat{\mathbf{R}}))\langle[S(\beta)]\rangle : [S(\gamma)]$ with $\|S(\hat{\mathbf{R}})\| \leq d$. Therefore, $\|S((\hat{\mathbf{Q}}\,\hat{\mathbf{R}})\langle[\beta]\rangle)\| \leq d$. ■

Lemma 23 (Completeness of $\mathscr{S}$): Given a λ-term M and a dimension d, if there exists a type environment Γ and a type A such that $\Gamma \vdash_d M : A$, then $\mathscr{S}(M, d)$ succeeds.

Proof: Assume that M is typable in bounded set dimension d. By the bounded width property (Theorem 14) there exists a derivation $\mathcal{D}' \triangleright \Gamma' \vdash M \longmapsto \mathbf{P}' : [A']$ such that $\llbracket\mathcal{D}'\rrbracket \leq d \cdot |M|$ and $\|\mathbf{P}'\| \leq d$.

Wlog. $\mathcal{D}'$ uses only normalized types (wrt. ACI of $\cap$) and the domain of Γ' is restricted to the free variables in M.

We prove $\mathscr{S}(M, d)$ completeness by explicitly giving a strict substitution S that satisfies the set of constraints C.

In step 1 for each subterm of M with corresponding annotation $\langle[A_1, \dots, A_l]\rangle$ in $\mathbf{P}'$ and $l \leq d$ the procedure may choose the annotation $\langle[\alpha_1, \dots, \alpha_l]\rangle$ in order to construct the elaboration schema $\hat{\mathbf{P}}$. Additionally, we fix $S(\alpha_i) = A_i$ for $i = 1 \dots l$.

In step 2 for each variable α occurring in $\hat{\mathbf{P}}$ we have either fixed $S(\alpha) = a$ for some type constant a or $S(\alpha) = (B_1 \cap \dots \cap B_l) \to B$ with $l \leq \llbracket\mathcal{D}'\rrbracket \leq d \cdot |M|$. In the first case, the procedure may choose $\ulcorner\alpha\urcorner = \alpha$. In the second case, the procedure may choose $\ulcorner\alpha\urcorner = (\beta_1 \cap \dots \cap \beta_l) \to \beta$. Additionally, we fix $S(\beta) = B$ and $S(\beta_i) = B_i$ for $i = 1 \dots l$. In both cases the respective constraint $\alpha \doteq \ulcorner\alpha\urcorner$ in step 3 is satisfied by S.

In step 4 for each free variable x in M with $\Gamma'(x) = [A_1, \dots, A_l]$ and $l \leq \llbracket\mathcal{D}'\rrbracket \leq d \cdot |M|$ the procedure $\mathscr{S}(M, d)$ may choose $x : [\alpha_1, \dots, \alpha_l]$ in order to construct the type environment schema $\hat{\Gamma}$. Additionally, we fix $S(\alpha_i) = A_i$ for $i = 1 \dots l$.

In step 5 we fix $S(\alpha) = A'$, which is the last fresh variable occurring in C.

Next, we show that in step 5 the recursive descent may exactly follow $\mathcal{D}'$. Therefore, any additional constraints in C are satisfied by S. In case of the initial judgement schema, $S(\hat{\Gamma} \vdash M \longmapsto \hat{\mathbf{P}} : [\alpha])$ is by construction the root of $\mathcal{D}'$.

Case $S(\hat{\Gamma}', x : [\alpha_1, \dots, \alpha_n] \vdash x \longmapsto x\langle[\beta]\rangle : [\gamma])$ in $\mathcal{D}'$. Clearly, the procedure may choose and $i \in \{1, \dots, l\}$ such that $S(\alpha_i) = S(\beta) = S(\gamma)$ satisfying $\beta \doteq \gamma$ and $\alpha_i \doteq \gamma$.

Case $S(\hat{\Gamma}' \vdash \lambda x.N \longmapsto (\lambda x.\hat{\mathbf{Q}})\langle[\beta]\rangle : [\gamma])$ in $\mathcal{D}'$. By inversion of ($\to$I) we have

- $S(\beta) = (B_1 \cap \dots \cap B_l) \to B$ with $l \leq \llbracket\mathcal{D}'\rrbracket$, therefore $\ulcorner\beta\urcorner = (\alpha_1 \cap \dots \cap \alpha_l) \to \alpha$, $S(\alpha) = B$ and $S(\alpha_i) = B_i$ for $i = 1 \dots l$
- $S(\beta) = S(\gamma)$, therefore S satisfies $\beta \doteq \gamma$
- the judgement $S(\hat{\Gamma}'), x : [B_1, \dots, B_l] \vdash N \longmapsto \hat{\mathbf{Q}} : [B]$ is in $\mathcal{D}'$, therefore $S(\hat{\Gamma}, x : [\alpha_1, \dots, \alpha_l] \vdash N \longmapsto \hat{\mathbf{Q}} : [\alpha])$ is in $\mathcal{D}'$

Case $S(\hat{\Gamma}' \vdash N\,L \longmapsto (\hat{\mathbf{T}}\langle[\delta]\rangle\,\hat{\mathbf{R}})\langle[\beta]\rangle : [\gamma])$ in $\mathcal{D}'$. By inversion of ($\to$E) potentially preceded by ($\cap$I) we have

- $S(\delta) = (B_1 \cap \dots \cap B_l) \to B$ with $l \leq d$, therefore $\ulcorner\delta\urcorner = (\alpha_1 \cap \dots \alpha_l) \to \alpha$, $S(\alpha) = B$ and $S(\alpha_i) = B_i$ for $i = 1 \dots l$
- the judgement $S(\hat{\Gamma}') \vdash N \longmapsto S(\hat{\mathbf{T}})\langle[S(\delta)]\rangle : [S(\delta)]$ is in $\mathcal{D}'$, therefore $S(\hat{\Gamma} \vdash N \longmapsto \hat{\mathbf{T}}\langle[\delta]\rangle : [\delta])$ is in $\mathcal{D}'$
- $S(\alpha) = S(\beta) = S(\gamma) = B$, therefore S satisfies $\beta \doteq \gamma$ and $\alpha \doteq \beta$
- $S(\hat{\mathbf{R}}) = \bigsqcup_{i=1}^{l} \mathbf{R}_i$ such that $S(\hat{\Gamma}) \vdash L \longmapsto \mathbf{R}_i : [B_i]$ is in $\mathcal{D}'$, therefore the procedure may choose a decomposition $\hat{\mathbf{R}} = \bigsqcup_{i=1}^{l} \hat{\mathbf{R}}_i$ such that the judgements $S(\hat{\Gamma} \vdash L \longmapsto \hat{\mathbf{R}}_i : [\beta_i])$ are in $\mathcal{D}'$ for $i = 1 \dots l$

Overall the substitution S satisfies C and has a strict type codomain. Therefore, there exists a most general unifier U of C such that $S = T' \circ U$ for some substitution T'. Fix a type constant a and let T be a substitution with $T(\alpha) = a$ for each variable α in the codomain of U. We have for each variable α in C that $T(U(\alpha))$ can be obtained by replacing subterms of $T'(U(\alpha))$ by a, therefore $T(U(\alpha))$ is a strict type. ■

Proposition 24 (Upper Bound for Typability): The typability problem in bounded set-theoretic dimension is in PSPACE.

Proof: We show that procedure $\mathscr{S}$ is polynomially bounded in nondeterministic space. More precisely, there is a polynomial p such that the runs of $\mathscr{S}(M, d)$ can be bounded in nondeterministic space by $p(d \cdot |M|)$.

The total number of required fresh variables in order to construct C is quadratic in $d \cdot |M|$ because no fresh variables are introduced during the recursive descent in step 5. Constraints in C are only of the shape $\alpha \doteq \beta$ or $\alpha \doteq \ulcorner\alpha\urcorner$ where the size of $\ulcorner\alpha\urcorner$ is linear in $d \cdot |M|$. Therefore, the size of C is polynomial in $d \cdot |M|$. Clearly, the size of each intermediate judgement schema $\hat{\Gamma} \vdash M \longmapsto \hat{\mathbf{P}} : [\alpha]$ is polynomial in $d \cdot |M|$. Recursive descent in step 5 can be done in polynomial space depth first by maintaining a stack of judgement schemata, because recursion depth is at most $|M|$ and the number of new judgement schemata put onto the stack is at most $d + 1$. Since C is of size polynomial in $d \cdot |M|$, computation of the MGU in step 6 as a directed acyclic graph of polynomial size (cf. [16]) can be done in polynomial space (in fact, polynomial time) as well as verification of strictness in step 7. Finally, the claim follows by Lemma 22 and Lemma 23 together with the identity PSPACE = NPSPACE. ■

Theorem 25 (Typability in bounded set-theoretic dimension): The typability problem in bounded set-theoretic dimension is PSPACE-complete.

Proof: By Proposition 20 and Proposition 24. ■

Corollary 26: Premise-bounded typability is in EXPSPACE.

Proof: Using Proposition 4, apply Proposition 24. ■

VI. Typability in Multiset Dimension

We show that type checking in bounded multiset dimension is NP-hard in dimension 1, and that the typability problem in bounded multiset dimension is in NP. We use a construction in [17] to establish the NP lower bound on type checking in dimension 1, and the NP upper bound on typability is established by adjusting procedure $\mathscr{S}$.

A. Lower Bound On Type Checking

We consider the type checking problem in bounded multiset dimension 1: given Γ, σ and M, is it the case that $\Gamma \Vdash_1 M : \sigma$? Intuitively, bounded dimension 1 disallows the use of intersection introduction. Therefore, any typing of the λ-term $(\lambda x.M)\,z$ in type-environment $\{z : [t, f]\}$ necessarily chooses either the type t or f (but not both) for x. This is essentially the same as overloading resolution [17], which is NP-complete and is related to typability in a low-rank intersection type system. Following [17], we reduce the Monotone One-in-Three-3SAT problem to $\Vdash_1$ type checking.

Problem 27 (Monotone One-in-Three-3SAT): Given a set of variables X and a set of clauses $\{(y_j^1, y_j^2, y_j^3) \mid j = 1 \dots m\}$ where $y_j^l \in X$ for $j = 1 \dots m$ and $l = 1, 2, 3$, decide whether there exists a valuation $v : X \to \{t, f\}$ such that v assigns the value t to exactly one variable in each clause.

Lemma 28 ([17, Thm. 3]): Monotone One-in-Three-3SAT is NP-complete.

Proposition 29: Type checking in multiset dimension 1 is NP-hard.

Proof: Following the proof of Theorem 4 in [17] we reduce Monotone One-in-Three-3SAT problem to $\Vdash_1$ type checking. Given clauses $\{(y_j^1, y_j^2, y_j^3) \mid j = 1 \dots m\}$ over variables $X = \{x_1, \dots, x_n\}$, i.e. $y_j^l \in X$ for $j = 1 \dots m$ and $l = 1, 2, 3$, we construct the type environment $\Gamma = \{z : [t, f], g : \left[\underbrace{t \to \dots \to t}_{m \text{ times}} \to t\right], h : [t \to f \to f \to t, f \to t \to f \to t, f \to f \to t \to t]\}$ and the λ-term $M = (\lambda x_1 \dots x_n.g\,(h\,y_1^1\,y_1^2\,y_1^3) \dots (h\,y_m^1\,y_m^2\,y_m^3))\,\underbrace{z \dots z}_{n \text{ times}}$.

By the same argumentation as in [17], the given Monotone One-in-Three-3SAT instance is satisfiable iff $\Gamma \Vdash_1 M : t$ holds. In bounded dimension 1 the variables $x_1, \dots, x_n$ are assigned either the type t or f and are therefore used consistently. Additionally, in bounded dimension 1 each occurrence of h is assigned at most one of the types of h in Γ (similar to the overloading restriction), therefore exactly one variable is assigned the type t in each clause represented by $h\,y_j^1\,y_j^2\,y_j^3$ for $j = 1 \dots m$. ■

B. Upper Bound on Typability

To decide typability in bounded multiset dimension we define the procedure $\mathscr{M}$ by modifying procedure $\mathscr{S}$ (cf. Procedure 21) to construct schematic multiset judgements $\hat{\Delta} \vdash M \Longrightarrow \hat{\mathbb{P}} : \hat{\mathfrak{s}}$ and modifying step 5 of procedure $\mathscr{S}$ as follows: instead of choosing a decomposition $\hat{\mathbf{R}}_1, \dots, \hat{\mathbf{R}}_l$ such that $\hat{\mathbf{R}} \equiv \bigsqcup_{i=1}^{l} \hat{\mathbf{R}}_i$, choose a decomposition $\hat{\mathbb{R}}_1, \dots, \hat{\mathbb{R}}_l$ such that $\hat{\mathbb{R}} \equiv \biguplus_{i=1}^{l} \hat{\mathbb{R}}_i$. Additionally to ensure condition $(\star)$, for each free variable occurrence $x\langle[\beta_1, \dots, \beta_n]\rangle$ in $\hat{\mathbb{R}}$, where $x : [\alpha_1, \dots, \alpha_m] \in \hat{\Delta}$, for each $i = 1 \dots n$ choose a distinct $j \in \{1, \dots, m\}$ and add the constraint $\beta_i \doteq \alpha_j$ to C.

Lemma 30 (Soundness of $\mathscr{M}$): Given a λ-term M and a dimension d, if procedure $\mathscr{M}(M, d)$ succeeds, then there exists a type environment Γ and a type A such that $\Gamma \Vdash_d M : A$.

Proof: Analogous to proof of Lemma 22, of course using multiset union ($\biguplus$) instead of union ($\bigsqcup$). Due to the disjoint decomposition in step 5, the size of decorations cannot exceed dimensional parameter d. Side condition $(\star)$ in rule $(\cap\mathrm{I})$ is ensured explicitly by the second modification. ■

Lemma 31 (Completeness of $\mathscr{M}$): Given a λ-term M and a dimension d, if there exists a type environment Γ and a type A such that $\Gamma \Vdash_d M : A$, then procedure $\mathscr{M}(M, d)$ succeeds.

Proof: Assume that M is typable in bounded multiset dimension d. Similar to the proof of Lemma 23, by the bounded width property (Theorem 14) there exists a derivation $\mathcal{D}' \triangleright \Delta' \vdash M \Longrightarrow \mathbb{P}' : \mathfrak{s}'$ such that $\llbracket \mathcal{D}' \rrbracket \leq d \cdot |M|$ and $\|\mathbb{P}'\| \leq d$. We proceed exactly as in the proof of Lemma 23 with $\mathcal{D}'$ using Δ' and the elaboration $\mathbb{P}'$ to construct a substitution S that maps variables in C to strict types. The only difference is the disjoint decomposition that follows the inversion of the ($\cap$I) rule in the multiset system. Completeness of the second modification is ensured by side condition ($\star$) in rule ($\cap$I). ∎

Proposition 32: The typability problem in bounded multiset dimension is in NP.

Proof: Similar to the proof of Proposition 24, the size of C is polynomial in $d \cdot |M|$. Therefore, MGU computation and verification is in polynomial time using directed acyclic graphs. Inspecting the number of recursive calls, due to the disjoint decomposition in step 5, two different leaves $\hat{\Delta} \vdash x \Longmapsto x\langle[\beta]\rangle : [\gamma]$ and $\hat{\Delta}' \vdash x' \Longmapsto x'\langle[\beta']\rangle : [\gamma']$ cannot share the same decorating variable, i.e. $\beta \not\equiv \beta'$. Since recursion depth is bounded by $|M|$ and recursion width is bounded by $d \cdot |M|$ (max. number of decorating variables), recursive descent in step 5 can be done in non-deterministic polynomial time. ∎

In addition to the proven NP upper bound for typability in bounded multiset dimension we conjecture an NP lower bound (as we have shown for type checking).

VII. Conclusion and Future Work

In this paper we have studied the "other half" of the decision problems associated with bounded-dimensional intersection types, complementing our study of inhabitation [9] with a study of the typability problem. We thereby obtain a fairly complete theory of fundamental decision problems in dimensional intersection type calculus. We believe that the dimensional framework is natural, providing a quantitative theory of elaborations as normed spaces of derivations, and that it complements existing results in a fruitful way, both in the theory of intersection types and towards practical applications of the intersection type system.

We are hopeful that interesting levels of efficiency could be reached for the typability problem. One focus in near future will be to engineer and implement the algorithms. An immediate next step will be to implement the Pspace typability algorithm for type inference for dimension-bounded intersection types. We also foresee integrating the inhabitation algorithm of [9] and type inference into the CLS synthesis framework based on type theoretic inhabitation [18], [19].

On the theoretical side, we would like to consider the expressiveness of bounded-dimensional typability in more depth. An interesting avenue could be to analyze the complexity of equivalence as a measure of expressiveness, as was done for finite rank in [7], and further investigation of the relation to rank-bounding would be of interest. The subterm filtration theorem (Theorem 16) is likely to be helpful here, as it could be a useful tool for relating type depth to width. A further topic is typability in the presence of subtyping and the universal type (empty intersection). Broader topics of theoretical interest (see also [9]) include the relation to non-idempotent intersection using concepts of linear logic (our system is not linear, as discussed in [9]), the relation between dimension and operational (reduction) complexity, and model theory of the dimensional calculus.

References

[1] M. Coppo and M. Dezani-Ciancaglini, "An Extension of Basic Functionality Theory for Lambda-Calculus," *Notre Dame Journal of Formal Logic*, vol. 21, pp. 685–693, 1980.

[2] M. Coppo, M. Dezani-Ciancaglini, and B. Venneri, "Functional Characters of Solvable Terms," *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, pp. 45–58, 1981.

[3] H. P. Barendregt, M. Coppo, and M. Dezani-Ciancaglini, "A Filter Lambda Model and the Completeness of Type Assignment," *Journal of Symbolic Logic*, vol. 48, no. 4, pp. 931–940, 1983.

[4] H. P. Barendregt, W. Dekkers, and R. Statman, *Lambda Calculus with Types*. Perspectives in Logic, Cambridge University Press, 2013.

[5] D. Leivant, "Polymorphic Type Inference," in *POPL 1983, Proceedings of the 10th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. ACM, 1983, pp. 88–98.

[6] A. J. Kfoury and J. B. Wells, "Principality and decidable type inference for finite-rank intersection types," in *POPL 1999, Proceedings of the 26th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, San Antonio, TX, USA, January 20-22, 1999*, 1999, pp. 161–174.

[7] A. J. Kfoury, H. G. Mairson, F. A. Turbak, and J. B. Wells, "Relating typability and expressiveness in finite-rank intersection type systems (extended abstract)," in *Proceedings of the fourth ACM SIGPLAN International Conference on Functional Programming (ICFP '99), Paris, France, September 27-29, 1999.*, D. Rémi and P. Lee, Eds. ACM, 1999, pp. 90–101.

[8] P. Urzyczyn, "Inhabitation of Low-Rank Intersection Types," in *TLCA 2009, Proceedings of Typed Lambda Calculus and Applications*, ser. LNCS, vol. 5608. Springer, 2009, pp. 356–370.

[9] A. Dudenhefner and J. Rehof, "Intersection Type Calculi of Bounded Dimension," in *POPL 2017, Proceedings of the 44th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Paris, France, January 2017*, 2017.

[10] F. Damiani and F. Prost, "Detecting and removing dead-code using rank 2 intersection," in *Types for Proofs and Programs, International Workshop TYPES'96, Aussois, France, December 15-19, 1996, Selected Papers*, 1996, pp. 66–87.

[11] A. Schubert, "The complexity of beta-reduction in low orders," in *TLCA, Typed Lambda Calculi and Applications, 5th International Conference, Krakow, Poland, May 2-5, 2001*, ser. LNCS, vol. 2044. Springer, 2001, pp. 400–414.

[12] K. Terui, "Semantic evaluation, intersection types and complexity of simply typed lambda calculus," in *23rd International Conference on Rewriting Techniques and Applications (RTA'12) , RTA 2012, May 28 - June 2, 2012, Nagoya, Japan*, 2012, pp. 323–338.

[13] A. Dudenhefner and J. Rehof, "Typability in Bounded Dimension (Extended Version)," Faculty of Computer Science (TU Dortmund), Tech. Rep. 858, 2017, http://www-seal.cs.tu-dortmund.de/seal/downloads/research/cls/TR858-DIM-Typability.pdf.

[14] H. P. Barendregt, *The Lambda Calculus. Its Syntax and Semantics*. Studies in Logic and the Foundations of Mathematics, 2nd Edition. Elsevier Science Publishers, 1984.

[15] S. van Bakel, "Strict Intersection Types for the Lambda Calculus," *ACM Computing Surveys*, vol. 43, no. 3, 2011.

[16] F. Baader and W. Snyder, *Handbook of Automated Reasoning*. Elsevier, 2001, ch. Chapter 8, Unification Theory, pp. 439–526.

[17] J. Palsberg, "Overloading Is NP-Complete - A Tutorial Dedicated to Dexter Kozen," in *Logic and Program Semantics - Essays Dedicated to Dexter Kozen on the Occasion of His 60th Birthday*, 2012, pp. 204–218.

[18] B. Düdder, M. Martens, and J. Rehof, "Staged Composition Synthesis," in *ESOP 2014, Proceedings of European Symposium on Programming, Grenoble, France 2014*, ser. LNCS, vol. 8410. Springer, 2014, pp. 67–86.

[19] J. Bessai, A. Dudenhefner, B. Düdder, M. Martens, and J. Rehof, "Combinatory Logic Synthesizer," in *Leveraging Applications of Formal Methods, Verification and Validation, 6th International Symposium ISoLA 2014, Corfu, Greece, October 8-11, 2014*, 2014, pp. 26–40.