The Intersection Type Unification Problem

Andrej Dudenhefner[1], Moritz Martens[2], and Jakob Rehof[3]

1 **Department of Computer Science, Technical University of Dortmund, Dortmund, Germany**
andrej.dudenhefner@cs.tu-dortmund.de
2 **Department of Computer Science, Technical University of Dortmund, Dortmund, Germany**
moritz.martens@cs.tu-dortmund.de
3 **Department of Computer Science, Technical University of Dortmund, Dortmund, Germany**
jakob.rehof@cs.tu-dortmund.de

Abstract

The intersection type unification problem is an important component in proof search related to several natural decision problems in intersection type systems. It is unknown and remains open whether the unification problem is decidable. We give the first nontrivial lower bound for the problem by showing (our main result) that it is exponential time hard. Furthermore, we show that this holds even under rank 1 solutions (substitutions whose codomains are restricted to contain rank 1 types). In addition, we provide a fixed-parameter intractability result for intersection type matching (one-sided unification), which is known to be NP-complete.

We place the intersection type unification problem in the context of unification theory. The equational theory of intersection types can be presented as an algebraic theory with an ACI (associative, commutative, and idempotent) operator (intersection type) combined with distributivity properties with respect to a second operator (function type). Although the problem is algebraically natural and interesting, it appears to occupy a hitherto unstudied place in the theory of unification, and our investigation of the problem suggests that new methods are required to understand the problem. Thus, for the lower bound proof, we were not able to reduce from known results in ACI-unification theory and use game-theoretic methods for two-player tiling games.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases Intersection Type, Equational Theory, Unification, Tiling, Complexity

Digital Object Identifier 10.4230/LIPIcs.FSCD.2016.19

1 Introduction

Intersection type systems occupy a prominent place within the theory of typed λ-calculus [5]. As is well known, variants of such systems characterize deep semantic properties of λ-terms, including normalization and solvability properties [5]. As a consequence of the enormous expressive power of intersection types, standard type-theoretic decision problems are undecidable for general intersection type systems, including the problem of type checking (given a term and a type, does the term have the type?) and inhabitation (given a type, does there exist a term having the type?). A combinatorial problem centrally placed in many classical type-theoretic decision problems is that of *type unification*: given two types σ and τ, does there exist a substitution S of types for type variables such that $S(\sigma) = S(\tau)$ in a suitable equational theory $(=)$ of types? In this paper we wish to initiate a study of the problem of *intersection type unification* which we believe to be of considerable systematic interest.

We consider the standard equational theory of intersection types induced by a canonical subtyping relation for intersection types [4]. Although decidability of intersection type unification appears to be surprisingly difficult and remains open, the present paper provides the first nontrivial lower bound indicating that the problem is of very high complexity: we prove that the problem is EXPTIME-hard. Our proof uses game-theoretic methods, in the form of two-player tiling games, which we believe to be of intrinsic interest and potentially helpful towards understanding the problem of decidability. Moreover, as we will show, the intersection type unification problem occupies a natural but hitherto (so far as we are aware) unstudied place in the theory of unification. Thus, we hope with this paper to stimulate further work on a fascinating open problem in type theory as well as in unification theory.

We briefly summarize some of the most important algebraic properties of the equational theory of intersection types needed to appreciate the systematic placement of the unification problem (full details are given later in the paper). Intersection type systems are characterized by the presence of an associative, commutative, idempotent operator, $\cap$ (intersection), which allows the formation of types of the form $\sigma \cap \tau$. In addition, we have function types, $\sigma \to \tau$. The standard equational theory, denoted $=$, of intersection types [4] is induced from a partial order $\leq$ on types, referred to as subtyping, by taking type equality to be the relation $\leq \cap \leq^{-1}$. Conversely, as will be discussed in the paper, it is also possible to give a purely equational presentation of subtyping. Because intersection is greatest lower bound with respect to subtyping, the intersection type unification problem is equivalent to the subtype satisfiability problem: given σ and τ, does there exist a type substitution S such that $S(\sigma) \leq S(\tau)$? The latter is equivalent to $S(\sigma) \cap S(\tau) = S(\tau)$, hence satisfiability is reducible to unification. The equational theory includes right-distributivity of $\to$ over $\cap$: $\sigma \to (\tau_1 \cap \tau_2) = (\sigma \to \tau_1) \cap (\sigma \to \tau_2)$ and left-contravariance of $\to$ with respect to subtyping: $\sigma_1 \to \tau_1 \leq \sigma_2 \to \tau_2$ whenever $\sigma_2 \leq \sigma_1$ and $\tau_1 \leq \tau_2$. As a consequence, one has "half left-distributivity" of $\to$ over $\cap$: $(\sigma_1 \to \tau) \cap (\sigma_2 \to \tau) \leq (\sigma_1 \cap \sigma_2) \to \tau$ (but the symmetric relation does not hold). Altogether, we could say for short that $\to$ is "$1\frac{1}{2}$-distributive" over $\cap$. Axioms specific to a special largest type, ω, are added in some variants of the theory (both variants, with or without ω, are important in type theory), including the recursion axiom $\omega = \omega \to \omega$, and we have the derived equation $\sigma \to \omega = \omega$. Thus, ω is unit (neutral element) with respect to $\cap$ and right-absorbing element with respect to $\to$.

In the remainder of this section we consider the most closely related work within unification theory and type theory.

1.1 Related work in unification theory

The single most directly related piece of work in the literature is the study from 2004 by Anantharaman, Narendran, and Rusinowitch on unification modulo ACUI (associativity, commutativity, unit, idempotence) plus distributivity axioms [2]. They consider equational theories over a binary ACUI symbol, denoted $+$, together with a binary operator, $*$, which distributes (left, right, or both) over $+$. Indeed, since (as summarized above) we have an ACUI theory of $\cap$ together with $\to$ enjoying distributivity properties over $\cap$, it would seem that we are temptingly close to the theories studied in [2], by thinking of their $+$ as $\cap$ and their $*$ as $\to$. In particular, algebraically closest among the theories covered in that paper, ACUI-unification with one-sided (say, left) distributivity (ACUID$_l$) is shown to be EXPTIME-complete, using techniques from unification modulo homomorphisms [3]. But it turns out that there are fundamental obstacles to transferring results or techniques from ACUID$_l$-unification to intersection type unification, as will be summarized next.

With regard to any upper bound, the main obstacle is that, whereas decidability of the ACUID-problems can be relatively straight-forwardly obtained by appeal to an occurs-check (nontrivial cyclic equations have no solutions), this is very far from being clear in the case of intersection type unification. Indeed, even in the absence of the recursive type ω, we can solve nontrivial cyclic constraints, due to contravariance. For example, the constraint $\alpha \mathrel{\dot{\leq}} \alpha \to b$ (where $\dot{\leq}$ denotes a formal subtyping constraint, α is a type variable and b is a constant) can be solved, e.g., by setting $S(\alpha) = b \cap (b \to b)$. The theory of intersection types is *non-structural* in the sense that types with significantly different shapes (tree domains, when types are regarded as labeled trees) may be related, and this presents fundamental obstacles for bounding the depth of substitutions via any kind of standard occurs-check. Although "$1\frac{1}{2}$-distributivity" of $\to$ over $\cap$ may at first sight appear to be algebraically close to the ACUID-framework of [2], the contravariant "$\frac{1}{2}$-distributivity" makes the theory of intersection types significantly different. We cannot exclude that some kind of restricted occurs-check might be possible, but our investigations lead us to believe that, in case it exists, it is likely to be very complicated, and we have been unable to find such a bounding principle. Hence, decidability remains a challenging open problem.

With regard to the exponential time lower bound, the results of [2] (in fact, both the Exptime upper and lower bounds) rely essentially on reductions from unification modulo a set H of noncommuting homomorphisms (ACUIDH), which was shown to be Exptime-complete in [3]. The basic idea is to represent unification with distributivity to unification modulo homomorphisms by replacing $s * t$ by $h_s(t)$ where h_s is a homomorphism with respect to the AC(U)I-theory. However, again, such techniques fail in our case due to contravariance. The equational presentation of the theory of intersection types captures contravariant subtyping by the absorption axiom (written in the algebraic notation of [2]): $s*t = s*t+(s+s')*t$. One could attempt to represent this axiom by $h_s(t) = h_s(t) + h_{s+s'}(t)$. But here the expression $h_{s+s'}(t)$ does not fall within the homomorphic format, and it is therefore not clear how the homomorphic framework could be applied. Moreover, the bounding problem discussed above leads to the problem that it is not clear how the theory could be adequately represented using only a finite set of homomorphisms. We concluded that we need new methods in order to make progress on understanding lower bounds for intersection type unification, and the route we present in this paper for the Exptime-lower bound is entirely different, relying on game theoretical results on tiling problems.

1.2 Related work in type theory

It may be surprising that computational properties (decidability, complexity) of the intersection type unification problem have not previously been systematically pursued *per se*. The theory of intersection type subtyping and its equational counterpart have rather been studied from semantic (operational and denotational) perspectives. Indeed, as mentioned already, the intersection type system captures deep operational properties of λ-terms, and undecidability of type checking and typability follows immediately. The theories of intersection type subtyping and equality studied here arose naturally out of model-theoretic considerations. For example, a fundamental result [14, 4] shows that intersection type subtyping and equality are sound and complete for set-theoretic containment in a class of λ-models: $\sigma \leq \tau$ holds, if and only if $[\![\sigma]\!]_v^{\mathcal{M}} \subseteq [\![\tau]\!]_v^{\mathcal{M}}$ for all models $\mathcal{M}$ in the class and valuations v. The intersection type unification problem can therefore also be endowed with semantic interpretations.

Several extensions and variations of the standard algebraic operations of unification studied here have been considered in connection with intersection type systems, foremostly motivated by questions related to notions of principality (principal types, principal typings, principal pairs) in such systems. Ronchi della Rocca, working from such motivations, defines a notion of unification in [21] and gives a semi-decision procedure for the corresponding unification problem. But that problem involves operations (chains of substitutions together with special expansion operations) which are not present in the algebraic notion of unification we consider here. Similarly, so-called expansion variables with associated operations have been used by Kfoury and Wells to characterize principality properties [17] and so-called β-unification involving expansion variables has been shown to characterize strong normalization in the λ-calculus [16], see also [10, 6]. The algebraic unification problem considered here is a centrally placed component in most forms of proof search related to intersection type systems. For example, it is not difficult to see that type checking parametric functions (or, combinatory expressions [15]) with intersection type schemes contains intersection type unification (we will give some concrete examples below in the paper). The problem is therefore likely to be involved as soon as one attempts to combine intersection types with usual notions of type instantiation. A recent example is the so-called type tallying problem of [7], which is not known to be decidable and is closely related to the intersection type satisfiability problem.

Summarizing the situation with regard to intersection type unification within type theory, it appears to hold an interesting and rather unexplored intermediate position: it is contained in many decision problems associated with intersection type systems, it is known to be expressive enough to capture certain restrictions of the type system, but it is not known whether it is decidable. It is therefore also a problem of importance for advancing our understanding of restrictions of the intersection type system and computational properties of associated decision problems.

Organization of the paper. The remainder of this paper is organized as follows. Intersection types are introduced in Sec. 2 together with the standard theory of subtyping [4]. In Sec. 3 we briefly study the matching problem (one-sided unification) as a natural preparation for considering the unification problem. The unification problem is studied in Sec. 4, which contains our main result. We first introduce the unification problem and the equational theory of intersection types (Sec. 4.1) and then turn to the proof of the EXPTIME-lower bound. We introduce tiling games (Sec. 4.2) and prove EXPTIME-completeness of a special form of such ("spiral tiling games"), which is then used (Sec. 4.3) in our reduction to unification and satisfiability. We conclude the paper in Sec. 5.

2 Intersection types

▶ **Definition 1** ($\mathbb{T}$)**.** The set $\mathbb{T}$ of intersection types, ranged over by σ, τ, ρ, is given by

$$\mathbb{T} \ni \sigma, \tau, \rho \ ::= \ a \mid \alpha \mid \omega \mid \sigma \to \tau \mid \sigma \cap \tau$$

where $a, b, c, \ldots$ range over type constants $\mathbb{C}$, ω is a special (universal) constant, and α, β, γ range over type variables $\mathbb{V}$.

As a matter of notational convention, function types associate to the right, and $\cap$ binds stronger than $\to$. A type $\tau \cap \sigma$ is said to have τ and σ as *components*.

▶ **Definition 2** (Subtyping $\leq$)**.** Subtyping $\leq$ is the least preorder (reflexive and transitive relation) over $\mathbb{T}$ (cf. [4]) such that

$$\sigma \leq \omega, \quad \omega \leq \omega \to \omega, \quad \sigma \cap \tau \leq \sigma, \quad \sigma \cap \tau \leq \tau, \quad (\sigma \to \tau_1) \cap (\sigma \to \tau_2) \leq \sigma \to \tau_1 \cap \tau_2,$$
$$\text{if } \sigma \leq \tau_1 \text{ and } \sigma \leq \tau_2 \text{ then } \sigma \leq \tau_1 \cap \tau_2, \quad \text{if } \sigma_2 \leq \sigma_1 \text{ and } \tau_1 \leq \tau_2 \text{ then } \sigma_1 \to \tau_1 \leq \sigma_2 \to \tau_2$$

Type equality, written $\sigma = \tau$, holds when $\sigma \leq \tau$ and $\tau \leq \sigma$, thereby making $\leq$ a partial order over $\mathbb{T}$. We use $\equiv$ for syntactic identity. By the axioms of subtyping, $\cap$ is associative, commutative, idempotent and has the following distributivity properties

$$(\sigma \to \tau_1) \cap (\sigma \to \tau_2) = \sigma \to (\tau_1 \cap \tau_2)\,,$$
$$(\sigma_1 \to \tau_1) \cap (\sigma_2 \to \tau_2) \leq (\sigma_1 \cap \sigma_2) \to (\tau_1 \cap \tau_2)\,.$$

We write $\bigcap_{i=1}^{n} \tau_i$ or $\bigcap_{i\in I} \tau_i$ or $\bigcap\{\tau_i \mid i \in I\}$ for an intersection of several components, where the empty intersection is identified with ω.

Using [4](Lemma 2.4.1) we syntactically define the set $\mathbb{T}^\omega$ of all types equal to ω.

▶ **Definition 3** ($\mathbb{T}^\omega$). The set $\mathbb{T}^\omega$ of types in $\mathbb{T}$ equal to ω is given by

$$\mathbb{T}^\omega \ni \sigma^\omega, \tau^\omega ::= \omega \mid \sigma \to \tau^\omega \mid \sigma^\omega \cap \tau^\omega\,.$$

▶ **Lemma 4.** *For $\tau \in \mathbb{T}$ we have $\tau \in \mathbb{T}^\omega$ iff $\tau = \omega$.*

▶ **Lemma 5** (Beta-Soundness [4, 5]). *Given $\sigma = \bigcap\limits_{i\in I}(\sigma_i \to \tau_i) \cap \bigcap\limits_{j\in J} a_j \cap \bigcap\limits_{k\in K} \alpha_k$, we have:*

(i) *If $\sigma \leq a$ for some $a \in \mathbb{C}$, then $a \equiv a_j$ for some $j \in J$.*
(ii) *If $\sigma \leq \alpha$ for some $\alpha \in \mathbb{V}$, then $\alpha \equiv \alpha_k$ for some $k \in K$.*
(iii) *If $\sigma \leq \sigma' \to \tau' \neq \omega$ for some $\sigma', \tau' \in \mathbb{T}$, then $I' = \{i \in I \mid \sigma' \leq \sigma_i\} \neq \emptyset$ and $\bigcap\limits_{i\in I'} \tau_i \leq \tau'$.*

▶ **Problem 6.** *(Subtyping) Given $\sigma, \tau \in \mathbb{T}$, does $\sigma \leq \tau$ hold?*

The subtyping relation is known to be decidable in polynomial time. The algorithm sketched in the proof of the following lemma gives an improved quadratic upper bound.

▶ **Lemma 7.** *Problem 6 (Subtyping) is decidable in time $\mathcal{O}(n^2)$ where n is the sum of the sizes of the input types σ and τ.*

Proof. For a polynomial time decision algorithm with a quartic upper bound see [19]. For a different approach with a quintic upper bound using rewriting see [22]. However, a quadratic upper bound to decide $\sigma \leq \tau$ is achievable using Lemmas 4 and 5. First, in linear time, subterms of σ and τ of the shape defined by $\mathbb{T}^\omega$ are replaced by ω. Second, in linear time, nested intersection are flattened using associativity of $\cap$ and components equal to ω are dropped. Third, in quadratic time, Lemma 5 is applied recursively using the additional property $\rho \leq \bigcap_{i\in I} \tau_i$ iff $\rho \leq \tau_i$ for $i \in I$. The invariant that $\cap$ is not nested and does not contain ω as component is ensured in recursive calls using linked lists with constant time concatenation to store components of intersections. ◀

We recapitulate the notion of paths and organized types introduced in [13].

▶ **Definition 8** (Paths $\mathbb{P}$). The set $\mathbb{P}$ of paths in $\mathbb{T}$, ranged over by π, is given by

$$\mathbb{P} \ni \pi ::= a \mid \alpha \mid \tau \to \pi\,.$$

▶ **Definition 9** (Organized type). A type τ is *organized*, if $\tau \equiv \omega$ or $\tau \equiv \bigcap_{i\in I} \pi_i$ for some paths π_i for $i \in I$.

A type can be organized (transformed to an equivalent organized type) in polynomial time. Note that an organized type is not necessarily normalized [14]. Normalization can lead to an exponential blow-up of type size.

▶ **Lemma 10.** *Given two organized types $\sigma \equiv \bigcap_{i \in I} \pi_i$ and $\tau \equiv \bigcap_{j \in J} \pi_j$, we have $\sigma \leq \tau$ iff for all $j \in J$ there exists an $i \in I$ with $\pi_i \leq \pi_j$.*

▶ **Corollary 11.** *Given a path $\pi \in \mathbb{P}$ and types σ, τ, we have $\sigma \cap \tau \leq \pi$ iff $\sigma \leq \pi$ or $\tau \leq \pi$.*

For the sake of completeness, we outline the corresponding type assignment system [4], also called **BCD** in literature. A basis (also called context) is a finite set $\Gamma = \{x_1 : \tau_1, \dots, x_n : \tau_n\}$, where the variables x_i are pairwise distinct; we set $\text{dom}(\Gamma) = \{x_1, \dots, x_n\}$ and we write $\Gamma, x : \tau$ for $\Gamma \cup \{x : \tau\}$, where $x \notin \text{dom}(\Gamma)$.

▶ **Definition 12** (Type Assignment). **BCD** type assignment is given by the following rules

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau}\ (\text{Ax}) \qquad \frac{\Gamma, x : \sigma \vdash e : \tau}{\Gamma \vdash \lambda x.e : \sigma \to \tau}\ (\to\text{I}) \qquad \frac{\Gamma \vdash e : \sigma \to \tau \qquad \Gamma \vdash e' : \sigma}{\Gamma \vdash (e\ e') : \tau}\ (\to\text{E})$$

$$\frac{}{\Gamma \vdash e : \omega}\ (\omega) \qquad \frac{\Gamma \vdash e : \sigma \qquad \Gamma \vdash e : \tau}{\Gamma \vdash e : \sigma \cap \tau}\ (\cap\text{I}) \qquad \frac{\Gamma \vdash e : \sigma \qquad \sigma \leq \tau}{\Gamma \vdash e : \tau}\ (\leq)$$

3 Intersection type matching

In order to understand the unification problem it is useful first to investigate its restriction to matching (one-sided unification). Intersection type matching occurs naturally during proof search in intersection type systems and is known to be NP-complete [12]. We strengthen this result by showing that the problem remains so even when restricted to the fixed-parameter case where only a single type variable and only a single constant is used in the input.

For $\tau \in \mathbb{T}$ let $\text{Var}(\tau) \subseteq \mathbb{V}$ denote the set of variables occurring in τ.

▶ **Problem 13** (Matching). *Given a set of constraints $C = \{\sigma_1 \dot{\leq} \tau_1, \dots, \sigma_n \dot{\leq} \tau_n\}$, where for each $i \in \{1, \dots, n\}$ we have $\text{Var}(\sigma_i) = \emptyset$ or $\text{Var}(\tau_i) = \emptyset$, is there a substitution $S \colon \mathbb{V} \to \mathbb{T}$ such that $S(\sigma_i) \leq S(\tau_i)$ for $1 \leq i \leq n$?*

We say that a substitution S satisfies $\{\sigma_1 \dot{\leq} \tau_1, \dots, \sigma_n \dot{\leq} \tau_n\}$ if $S(\sigma_i) \leq S(\tau_i)$ for $1 \leq i \leq n$.

▶ **Problem 14** (One-Sided Unification). *Given a set of constraints $C = \{\sigma_1 \doteq \tau_1, \dots, \sigma_n \doteq \tau_n\}$, where for each $i \in \{1, \dots, n\}$ we have $\text{Var}(\sigma_i) = \emptyset$ or $\text{Var}(\tau_i) = \emptyset$, is there a substitution $S \colon \mathbb{V} \to \mathbb{T}$ such that $S(\sigma_i) = S(\tau_i)$ for $1 \leq i \leq n$?*

Note that any matching constraint set $C = \{\sigma_1 \dot{\leq} \tau_1, \dots, \sigma_n \dot{\leq} \tau_n\}$ can be reduced to a single matching (resp. one-sided unification) constraint $\sigma \dot{\leq} \tau$ (resp. $\sigma \cap \tau \doteq \sigma$) with $\text{Var}(\sigma) = \emptyset$ by fixing a type constant $\bullet \in \mathbb{C}$ to define

$$(\sigma_i', \tau_i') = \begin{cases} (\sigma_i \to \bullet, \tau_i \to \bullet) & \text{if } \text{Var}(\sigma_i) = \emptyset \\ (\tau_i, \sigma_i) & \text{if } \text{Var}(\tau_i) = \emptyset \end{cases} \quad \text{for } 1 \leq i \leq n$$

and $\sigma \equiv \sigma_1' \to \dots \to \sigma_n' \to \bullet$ and $\tau \equiv \tau_1' \to \dots \to \tau_n' \to \bullet$. By Lemma 5, for any substitution S we have $S(\sigma) \leq S(\tau)$ (resp. $S(\sigma \cap \tau) = S(\sigma)$) iff $S(\sigma_i) \leq S(\tau_i)$ for $1 \leq i \leq n$. Therefore, matching and one-sided unification remain NP-complete even restricted to single constraints.

In [12] the lower bound for matching is shown by reduction from 3-SAT and requires two type variables $\alpha_x, \alpha_{\neg x}$ for each propositional variable x. Since 3-SAT, parameterized by the number of propositional variables, is fixed parameter tractable, we naturally ask whether the same holds for matching (resp. one-sided unification) parameterized by the number of type variables.

▶ **Proposition 15.** *Problem 13 (Matching) is* NP*-hard even if only a single type variable and a single constant is used in the input.*

Proof. (Sketch) We fix a 3-SAT instance F containing clauses $(L_1 \vee L_2 \vee L_3) \in F$ over propositional variables V where L_i is either x or $\neg x$ for some $x \in V$. We reduce satisfiability of F to matching with one type variable α. First, we fix a set of type constants $B = V \cup \{\neg x \mid x \in V\}$ and the type constant $\bullet$. Let $\sigma_x \equiv \bigcap(B \setminus \{\neg x\})$ and $\sigma_{\neg x} \equiv \bigcap(B \setminus \{x\})$ for $x \in V$. We construct the set C containing following constraints

$$\begin{aligned}
&\text{for } x \in V \text{ (consistency)}:\\
&\quad ((\sigma_{\neg x} \to \bullet) \to (\neg x \to \bullet)) \cap ((\sigma_x \to \bullet) \to (x \to \bullet)) \mathrel{\dot{\leq}} (\alpha \to \bullet) \to (\alpha \to \bullet)\\
&\text{for } (L_1 \vee L_2 \vee L_3) \in F \text{ (validity)}:\\
&\quad (L_1 \to \bullet) \cap (L_2 \to \bullet) \cap (L_3 \to \bullet) \mathrel{\dot{\leq}} \alpha \to \bullet
\end{aligned}$$

If F is satisfied by a valuation v, then the substitution $\alpha \mapsto \bigcap_{v(x)=1} x \cap \bigcap_{v(x)=0} \neg x$ satisfies C. If C is satisfied by a substitution S, then by Corollary 11 and the consistency constraints we have either $\sigma_{\neg x} \leq S(\alpha) \leq \neg x$ or $\sigma_x \leq S(\alpha) \leq x$ for $x \in V$. A valuation v constructed according to these cases satisfies each clause in F due to Corollary 11 and the validity constraints.

Instead of using constants $\{a_1, \dots, a_k, \bullet\}$ for an instance of the matching problem, encode $[a_i] = \underbrace{\bullet \to \dots \to \bullet \to}_{i \text{ times}} \bullet$ for $1 \leq i \leq k$ in the proof. Using this technique it is easy to see that only one type constant $\bullet$ is sufficient. ◀

Combining Proposition 15 with the reduction in [12] we conclude that neither restricting substitutions to the shape $S : \{\alpha\} \to \mathbb{T}$ nor restricting to the shape $S : \mathbb{V} \to \mathbb{C}$ (atomic substitutions, mapping variables to type constants) reduces the complexity of matching.

4 Intersection type unification

4.1 The unification problem

▶ **Problem 16** (Satisfiability). *Given a set of constraints $C = \{\sigma_1 \mathrel{\dot{\leq}} \tau_1, \dots, \sigma_n \mathrel{\dot{\leq}} \tau_n\}$, is there a substitution $S \colon \mathbb{V} \to \mathbb{T}$ such that $S(\sigma_i) \leq S(\tau_i)$ for $1 \leq i \leq n$?*

▶ **Problem 17** (Unification). *Given a set of constraints $C = \{\sigma_1 \doteq \tau_1, \dots, \sigma_n \doteq \tau_n\}$, is there a substitution $S \colon \mathbb{V} \to \mathbb{T}$ such that $S(\sigma_i) = S(\tau_i)$ for $1 \leq i \leq n$?*

Since for any $\sigma, \tau \in \mathbb{T}$ and any substitution S we have $S(\sigma) \leq S(\tau) \iff S(\sigma) \cap S(\tau) = S(\sigma)$, satisfiability and unification are equivalent. Similarly to matching (resp. one-sided unification) restricting satisfiability (resp. unification) to single constraints does not reduce its complexity.

We now provide a number of observations that give some insight into the type-theoretical and combinatorial expressive power of unification.

Consider a combinatory logic with intersection types [15, 11] with arbitrary basis $\mathfrak{B}$, that is, a finite set of combinator symbols $F, G, \dots$ with type schemes $\tau_F, \tau_G, \dots$. Such a system is given by the rules (applicative fragment) $(\to\text{E}), (\cap\text{I}), (\leq)$ of Definition 12 together with a rule assigning types $S(\tau_F)$ to the combinator symbol F for any substitution S. Write $\mathfrak{B} \vdash E : \tau$ for derivability of the type τ for the combinatory expression E in this system.

▶ **Example 18.** Let $\mathfrak{B} = \{F : (\sigma \to \tau) \to \bullet, G : \alpha \to \alpha\}$, where wlog. $\alpha \notin \mathrm{Var}(\sigma) \cup \mathrm{Var}(\tau)$. In this scenario, type-checking $\mathfrak{B} \vdash F\,G : \bullet$ is equivalent to solving the satisfiability problem $\alpha \to \alpha \mathrel{\dot{\leq}} \sigma \to \tau$, equivalently, the unification problem $\sigma \cap \tau \doteq \sigma$, because we need to find substitutions $S, S_1, \dots S_n$ for some $n \in \mathbb{N}$ such that

$$\begin{aligned}
&\bigcap_{i=1}^{n} S_i(\alpha \to \alpha) \leq S(\sigma \to \tau)\\
\overset{\text{Lem. 5}}{\iff} &S(\sigma) \leq \bigcap_{i \in I} S_i(\alpha) \text{ and } \bigcap_{i \in I} S_i(\alpha) \leq S(\tau) \text{ for some } I \subseteq \{1, \dots, n\}\\
\iff &S(\alpha \to \alpha) \leq S(\sigma \to \tau) \text{ setting } S(\alpha) = \bigcap_{i \in I} S_i(\alpha)
\end{aligned}$$

Write $\mathfrak{B} \vdash^* E : \tau$ if $\mathfrak{B} \vdash E : \tau$ is derivable without the intersection introduction rule ($\cap$I). This restriction occupies an interesting 'intermediate' position: generalized to arbitrary bases $\mathfrak{B}$, it is the combinatory logic that subsumes the **BCD**-calculus without intersection introduction [18, 20]. For example, $\vdash^*$ is sufficient to type $S\,I\,I$, i.e. the SKI-combinatory logic equivalent of the λ-term $\lambda x.x\,x$ not typable in simple types.

▶ **Example 19.** Typability with respect to $\vdash^*$ is equivalent to unification. Let $\mathfrak{B} = \{F_1 : \tau_1, \dots F_n : \tau_n\}$, where wlog. $\mathrm{Var}(\tau_i) \cap \mathrm{Var}(\tau_j) = \emptyset$ for $i \neq j$, and let E be a combinatory term over $\mathfrak{B}$. We want to know whether there is a type τ such that $\mathfrak{B} \vdash^* E : \tau$.

For any combinatory term E' and type τ' we define

$$f(E', \tau') = \begin{cases} \{\tau_i \mathrel{\dot{\leq}} \tau'\} & \text{if } E' = F_i \text{ for some } i \in \{1, \dots, n\}\\ f(E_1, \alpha \to \beta) \cup f(E_2, \alpha) & \text{if } E' = E_1\,E_2 \text{ and } \alpha, \beta \text{ are fresh} \end{cases}$$

The unification problem instance $f(E, \alpha)$, where α is fresh, has a solution iff E is typable in the basis $\mathfrak{B}$, i.e. there exists a type τ such that $\mathfrak{B} \vdash^* E : \tau$. Conversely, given a satisfiability problem $\sigma \mathrel{\dot{\leq}} \tau$, we consider typability of $F\,G$ in the basis $\mathfrak{B} = \{F : \tau \to a, G : \sigma\}$.

The following example shows that unification can force exponential growth of the size of solutions.

▶ **Example 20.** Consider prime numbers $2, 3$ and the following unification constraints

$$a \to a \to (\beta_2 \cap b) \doteq \beta_2 \cap \alpha, \qquad a \to a \to a \to (\beta_3 \cap b) \doteq \beta_3 \cap \alpha\,.$$

The smallest substitution satisfying the above constraints is

$$\begin{aligned}
S(\beta_2) &= (a \to a \to b) \cap (a \to a \to a \to a \to b)\\
S(\beta_3) &= a \to a \to a \to b\\
S(\alpha) &= a \to a \to a \to a \to a \to a \to b
\end{aligned}$$

In particular, the size of $S(\alpha)$ is greater than the product of our initial primes. By adding an additional constraint $a \to a \to a \to a \to a \to (\beta_5 \cap b) \doteq \beta_5 \cap \alpha$, the size of $S(\alpha)$ becomes at least $2 \cdot 3 \cdot 5$, growing exponentially with additional constraints.

An axiomatization of the equational theory of intersection type subtyping (without ω) is derived in [22]. We add two additional axioms **(U)** and **(RE)** in the following Definition 21 to incorporate ω.

▶ **Definition 21** (ACIUD$_l$ReAb)**.** The equational theory ACIUD$_l$ReAb is given by
(A) $\sigma \cap (\tau \cap \rho) \sim (\sigma \cap \tau) \cap \rho$
(C) $\sigma \cap \tau \sim \tau \cap \sigma$
(I) $\sigma \cap \sigma \sim \sigma$
(U) $\sigma \cap \omega \sim \sigma$,
(D$_l$) $(\sigma \to \tau) \cap (\sigma \to \tau') \sim \sigma \to \tau \cap \tau'$
(RE) $\omega \sim \omega \to \omega$
(AB) $\sigma \to \tau \sim (\sigma \to \tau) \cap (\sigma \cap \sigma' \to \tau)$

The recursion axiom **(RE)** captures the recursive nature of ω and the absorption axiom **(AB)** captures contra-variance.

▶ **Lemma 22.** *Given $\sigma, \tau \in \mathbb{T}$ we have $\sigma = \tau$ iff $\sigma \sim \tau$.*

Proof.
(⇒) Induction on the depth of the derivation of $\sigma \leq \tau$ to show $\sigma \cap \tau \sim \sigma$. Therefore, $\sigma \leq \tau$ and $\tau \leq \sigma$ imply $\sigma \sim \sigma \cap \tau \sim \tau$.
(⇐) Each axiom of ACIUD$_l$ReAb is derivable using subtyping. ◀

The absorption axiom **(AB)** distinguishes the above theory ACIUD$_l$ReAb from theories studied in literature. As discussed in the introduction, the closest equational theory ACIUD$_l$ of [2], which assumes $\omega \to \sigma \sim \omega \sim \sigma \to \omega$ and has no equivalent of the absorption axiom **(AB)**, is Exptime-complete. Unfortunately, the absorption axiom prevents the approaches presented in [2, 1] as shown by the following examples.

▶ **Example 23.** Consider $\alpha \cap (\alpha \to a) \doteq \alpha$ (or equivalently $\alpha \mathrel{\dot{\leq}} \alpha \to a$). A DAG-based (or 'occurs-check'-based) approach cannot stratify such a constraint (even in the absence of ω) since any solution $S(\alpha)$ contains at least one subterm $S(\alpha) \to a$ and therefore a circular dependency. Interestingly, using absorption there is a solution $S(\alpha) = a \cap (a \to a)$.

▶ **Example 24.** Consider $\alpha \cap (((\alpha \to c) \cap b) \to a) \doteq \alpha$ (or equivalently $\alpha \mathrel{\dot{\leq}} ((\alpha \to c) \cap b) \to a)$. In contrast to the previous example, all occurrences of α are positive. Again, we have a circular dependency. Using absorption there is a solution $S(\alpha) = b \to a$.

4.2 Tiling games

In this section we introduce a special kind of domino tiling game, referred to as two-player corridor tiling games, for which Chlebus showed in 1986 that the problem of existence of winning strategies is Exptime-complete [9]. We then show that Exptime-completeness is preserved when tilings are restricted to a particular ("spiral") shape, which will be used to prove our Exptime-lower bound for intersection type unification in Sec. 4.3.

▶ **Definition 25** (Tiling System)**.** A *tiling system* is a tuple $(D, H, V, \bar{b}, \bar{t}, n)$, where:
- D is a finite set of tiles (also called dominoes)
- $H, V \subseteq D \times D$ are horizontal and vertical constraints
- $\bar{b}, \bar{t}$ are n-tuples of tiles
- n is a unary encoded natural number

▶ **Definition 26** (Corridor Tiling)**.** Given a tiling system $(D, H, V, \bar{b}, \bar{t}, n)$, a *corridor tiling* is a mapping $\lambda : \{1, \ldots, l\} \times \{1, \ldots, n\} \to D$ for some $l \in \mathbb{N}$ such that:
- $\bar{b} = (\lambda(1, 1), \ldots, \lambda(1, n))$ (correct bottom row)
- $\bar{t} = (\lambda(l, 1), \ldots, \lambda(l, n))$ (correct top row)

- for $i \in \{1, \ldots, l\}$ and $j \in \{1, \ldots, n-1\}$ we have $(\lambda(i,j), \lambda(i,j+1)) \in H$, i.e. the horizontal constraints are satisfied
- for $i \in \{1, \ldots, l-1\}$ and $j \in \{1, \ldots, n\}$ we have $(\lambda(i,j), \lambda(i+1,j)) \in V$, i.e. the vertical constraints are satisfied

Given a tiling system $(D, H, V, \bar{b}, \bar{t}, n)$, a *Two-Player Corridor Tiling* game consists of two players (*Constructor* and *Spoiler*). The game is played on an $\mathbb{N} \times \{1, \ldots, n\}$ board and starts with the bottom row $\bar{b}$. Each player places tiles in turn starting with Constructor. While Constructor tries to construct a corridor tiling, Spoiler tries to prevent it. Constructor wins if Spoiler makes an illegal move (with respect to H or V), or when a correct corridor tiling is completed. We say Constructor has winning strategy, if he can win no matter what Spoiler does.

▶ **Lemma 27** (Chlebus [9]). *The decision problem whether Constructor has a winning strategy in a given two-player corridor tiling game is* EXPTIME*-complete.*

Instead of directly encoding a Two-Player Corridor Tiling into intersection type satisfiability, we introduce a slightly different game that is played out as sequences instead of corridors. The main goal is to get rid of several structural constraints of corridors for a more accessible construction of a spiral where each new tile has a neighboring previous tile.

▶ **Definition 28** (Spiral Tiling). Given a tiling system $(D, H, V, \bar{b}, \bar{t}, n)$, a *spiral tiling* is a sequence $d_1 \ldots d_m \in D^m$ for some $m \in \mathbb{N}$ such that:

- $d_1 \ldots d_n = \bar{b}$
- $d_{m-n+1} \ldots d_m = \bar{t}$
- $(d_i, d_{i+1}) \in H$ for $1 \leq i \leq m-1$
- $(d_i, d_{i+n}) \in V$ for $1 \leq i \leq m-n$

Given a tiling system $(D, H, V, \bar{b}, \bar{t}, n)$ a *Two-Player Spiral Tiling* game, played by Constructor and Spoiler, starts with the sequence $\bar{b}$. Each player adds a tile to the end of the current sequence taking turns starting with Constructor. While Constructor tries to construct a spiral tiling, Spoiler tries to prevent it. Constructor wins if Spoiler makes an illegal move (with respect to H or V), or when a correct spiral tiling is completed. Again, we are interested in whether Constructor has a winning strategy.

The main differences between a corridor tiling and a spiral tiling is the lack of individual rows. While a tile at the beginning of the new row of a corridor is not constrained by the previously placed tile, in a spiral each new tile is constraint by the previously placed one. Additionally, a corridor tiling always contains $l \cdot n$ tiles for some l; a spiral tiling does not obey such a restriction.

▶ **Lemma 29.** *The decision problem whether Constructor has a winning strategy in a given two-player spiral tiling game is* EXPTIME*-complete.*

Proof.

Lower Bound: Given a tiling system $T = (D, H, V, (b_1, \ldots, b_n), (t_1, \ldots, t_n), n)$, let:

$$\begin{aligned}
D' &= D \,\dot\cup\, \{\#\} \\
H' &= H \,\dot\cup\, \{(d, \#) \mid d \in D'\} \cup \{(\#, d) \mid d \in D'\} \\
V' &= V \,\dot\cup\, \{(\#, \#)\} \\
T' &= (D', H', V', (b_1, \ldots, b_n, \#, \#), (t_1, \ldots, t_n, \#, \#), n+2)
\end{aligned}$$

We show that Constructor has a winning strategy for Two-Player Corridor Tiling in T iff he has a winning strategy for Two-Player Spiral Tiling in T'.

By construction, both players are allowed to and have to place the tile # at exactly the turns $i(n+2)-1$ and $i(n+2)$ for $i \geq 1$. Therefore, a winning strategy does not branch nor end at those turns. Additionally, a correct spiral tiling ends in two consecutive # tiles, therefore necessarily contains $i(n+2)$ tiles.

From any correct corridor tiling $\lambda : \{1,\dots,l\} \times \{1,\dots,n\}$ for T we construct a spiral tiling $d_1 \dots d_{l(n+2)}$ for T' by

$$d_k = \begin{cases} \lambda(i,j) & \text{if } k = (i-1)(n+2)+j \text{ and } i \geq 1 \text{ and } 1 \leq j \leq n \\ \# & \text{if } k = (i-1)(n+2)+j \text{ and } i \geq 1 \text{ and either } j = 0 \text{ or } j = n+1 \end{cases}$$

From a correct spiral tiling $d_1 \dots d_{l(n+2)}$ for T' we construct a corridor tiling $\lambda : \{1,\dots,l\} \times \{1,\dots,n\}$ for T by $\lambda(i,j) = d_{(i-1)(n+2)+j}$.

In particular, Constructor's winning strategy (skipping/adding the forced # turns) is exactly the same for both games.

Upper Bound: Computation in APSPACE = EXPTIME (similar to Two-Player Corridor Tiling). To continue the game only the n previously placed tiles have to be considered. ◀

4.3 Exptime lower bound

We now prove our main result, that the intersection type unification problem is EXPTIME-hard. The proof will be by reduction from spiral tiling games (Lemma 29) to the intersection type satisfiability problem.

Let $T = (D, H, V, \bar{b} = b_1 \dots b_n, \bar{t} = t_1 \dots t_n, n)$ be a tiling system. Wlog. $(b_i, b_{i+1}) \in H$ and $(t_i, t_{i+1}) \in H$ for $1 \leq 1 < n$. We fix the set of type constants $\mathbb{C} = D \dot\cup \{\bullet\}$ and variables $\mathbb{V} = \{\alpha\} \cup \{\beta_d \mid d \in D\}$ and construct the following set of constraints $\mathcal{C}_T$:

(i) $\sigma_\perp^H \cap \sigma_\perp^V \cap \sigma_t \cap \bigcap_{d \in D} \beta_d \dot\leq \sigma_b \cap \bigcap_{d' \in D} \bigcap_{d \in D} (d' \to d \to \beta_d)$ (Game moves)

(ii) $\bigcap_{(d',d) \in H} (d \to d' \to \alpha) \dot\leq \bigcap_{d \in D} (d \to \beta_d)$ (d respects H)

(iii) $\bigcap_{(d',d) \in V} (d \to \underbrace{\omega \to \dots \to \omega \to}_{n-1 \text{ times}} d' \to \alpha) \dot\leq \bigcap_{d \in D} (d \to \beta_d)$ (d respects V)

where

$$\begin{aligned} \sigma_b &\equiv b_n \to \dots \to b_1 \to \bullet && \text{(Initial state)} \\ \sigma_t &\equiv (t_n \to \dots \to t_1 \to \alpha) \cap (\omega \to t_n \to \dots \to t_1 \to \alpha) && \text{(Final states)} \\ \sigma_\perp^H &\equiv \bigcap_{(d,d') \in D \times D \setminus H} (d' \to d \to \alpha) && (d' \text{ violates } H) \\ \sigma_\perp^V &\equiv \bigcap_{(d,d') \in D \times D \setminus V} (d' \to \underbrace{\omega \to \dots \to \omega \to}_{n-1 \text{ times}} d \to \alpha) && (d' \text{ violates } V) \end{aligned}$$

Intuitively, we want to use Lemma 10 to realize alternation. The rhs of (i) represents an intersection of all board positions which Constructor may face. Therefore, for all such position he needs to find a suitable move by picking a path on the lhs of (i). He can either state that the Spoilers last move violates H or V choosing $\sigma_\perp^H$ or $\sigma_\perp^V$ or state that the game is finished choosing σ_t or pick his next move $d \in D$ choosing β_d. Intuitively, β_d captures all board positions in which Constructors decides to place d next. Note that on the rhs of (i) in the type $d' \to d \to \beta_d$ the tile d' is not constrained (representing all possible moves of Spoiler) while the tile d is constrained to the index of β_d, i.e. Constructors previous choice.

Therefore, by picking a move d Constructor is faced with all board positions that arise from the previous position extended by d and each possible d'. Constraints (ii) and (iii) ensure that whenever Constructor picks his next move $d \in D$ choosing β_d he has to respect H and V.

We show that Constructor has a winning strategy for two-player spiral tiling in T iff the constraint system $\mathcal{C}_T$ is satisfiable. To represent game positions as types, we define the mapping $[\cdot] : D^* \to \mathbb{T}$ such that $[\epsilon] = \bullet$ and $[\bar{s}d] = d \to [\bar{s}]$ for $\bar{s} \in D^*$ and $d \in D$. To improve readability, we use the notation $\sigma \overset{\phi}{\leq} \tau$, where ϕ is a hint why the inequality holds.

▶ **Lemma 30.** *Let T be a tiling system. If Constructor has a winning strategy in a two-player spiral tiling game in T, then the constraint system $\mathcal{C}_T$ is satisfiable.*

Proof. Assume that Constructor has a winning strategy that is represented by a labeled tree $f : \mathrm{dom}(f) \to \{C, S\}$ where

- $\mathrm{dom}(f) \subseteq D^*$ is finite and prefix-closed, i.e. $\bar{u}\bar{v} \in \mathrm{dom}(f)$ implies $\bar{u} \in \mathrm{dom}(f)$.
- $\mathrm{depth}(f) = \max\{k \mid d_1 \dots d_k \in \mathrm{dom}(f)\}$.
- For $\bar{s} = d_1 \dots d_k \in \mathrm{dom}(f)$ we have $f(\bar{s}) = C$ if k is even and $f(\bar{s}) = S$ if k is odd, i.e. C places a tile after an even number of turns and S after an odd number of turns.
- For $\bar{s} \in \mathrm{dom}(f)$ such that $f(\bar{s}) = S$ we have $\bar{s}d' \in \mathrm{dom}(f)$ for all $d' \in D$, i.e. the strategy has to consider all (possibly illegal) Spoilers moves.
- For $\bar{s} \in \mathrm{dom}(f)$ such that $f(\bar{s}) = C$ we have either
  - There exists exactly one $d \in D$ such that $\bar{s}d \in \mathrm{dom}(f)$ and $\bar{b}\bar{s} = \bar{u}d_1 \dots d_n$ for some $\bar{u} \in D^*$ and $d_1, \dots, d_n \in D$ with $(d_n, d) \in H$ and $(d_1, d) \in V$, i.e. Constructors next move is d which respects H and V.
  - $\bar{s}d \notin \mathrm{dom}(f)$ for all $d \in D$ and either
    - $\bar{b}\bar{s} = \bar{u}\bar{t}$ for some $\bar{u} \in D^*$, i.e. Constructor states that the game is finished.
    - $\bar{b}\bar{s} = \bar{u}\bar{t}d'$ for some $\bar{u} \in D^*$ and $d' \in D$, i.e. Constructor states that Spoilers last move d' is illegal because the game already ended.
    - $\bar{b}\bar{s} = \bar{u}dd'$ for some $\bar{u} \in D^*$, $d, d' \in D$ such that $(d, d') \notin H$, i.e. Constructor states that Spoilers last move d' violates H.
    - $\bar{b}\bar{s} = \bar{u}d\bar{v}d'$ for some $\bar{u} \in D^*$, $\bar{v} \in D^{n-1}$, $d, d' \in D$ such that $(d, d') \notin V$, i.e. Constructor states that Spoilers last move d' violates V.

We construct the following substitution S

$$S(\alpha) = \bigcap_{\substack{d_1 \dots d_k = \bar{s} \in D^* \\ k \leq \mathrm{depth}(f)+n}} [\bar{s}] \quad \text{and} \quad S(\beta_d) = \bigcap_{\substack{\bar{s} \in f^{-1}(C) \\ \bar{s}d \in \mathrm{dom}(f)}} [\bar{b}\bar{s}] \quad \text{for } d \in D\,.$$

We verify that the individual inequalities hold.

- $S(\sigma_\perp^H \cap \sigma_\perp^V \cap \sigma_t \cap \bigcap_{d \in D} \beta_d) \leq \sigma_b$:
  if $\bar{b} = \bar{t}$, then $S(\sigma_t) \leq \sigma_b$. Otherwise, according to f, there exists a $d \in D$ such that $d \in \mathrm{dom}(f)$. Therefore, $S(\beta_d) \leq [\bar{b}] \equiv \sigma_b$.
- $S(\sigma_\perp^H \cap \sigma_\perp^V \cap \sigma_t \cap \bigcap_{d \in D} \beta_d) \leq S(d' \to d \to \beta_d)$ for all $d, d' \in D$:
  we show that for any $\pi = d_k \to \ldots \to d_1 \to \sigma_b = [\bar{b}\bar{s}]$ such that $d_1 \dots d_k = \bar{s} \in f^{-1}(C)$ and $\bar{s}d \in \mathrm{dom}(f)$ we have $S(\sigma_\perp^H \cap \sigma_\perp^V \cap \sigma_t \cap \bigcap_{d \in D} \beta_d) \leq d' \to d \to \pi \equiv [\bar{b}\bar{s}dd']$. Since $\bar{s}d \in \mathrm{dom}(f)$ and $f(\bar{s}) = C$ we have $\bar{s}dd' \in \mathrm{dom}(f)$ and $f(\bar{s}dd') = C$. According to f we have either
  - $\bar{s}dd'd'' \in \mathrm{dom}(f)$ for some $d'' \in D$. Therefore, $S(\beta_{d''}) \leq d' \to d \to \pi \equiv [\bar{b}\bar{s}dd']$
  - or

  - $\bar{b}\bar{s}dd' = \bar{u}\bar{t}$ for some $\bar{u} \in D^*$, then $S(\sigma_t) \leq [\bar{u}\bar{t}] \equiv [\bar{b}\bar{s}dd']$.
  - $\bar{b}\bar{s}dd' = \bar{u}\bar{t}d'$ for some $\bar{u} \in D^*$, then $S(\sigma_t) \leq [\bar{u}\bar{t}d'] \equiv [\bar{b}\bar{s}dd']$.
  - $(d, d') \notin H$, then $S(\sigma_\perp^H) \leq [\bar{b}\bar{s}dd']$.
  - $\bar{b}\bar{s}dd' = \bar{u}d''\bar{v}d'$ for some $\bar{u} \in D^*$, $\bar{v} \in D^{n-1}$, $d'' \in D$ such that $(d'', d') \notin V$, then $S(\sigma_\perp^V) \leq [\bar{u}d''\bar{v}d'] \equiv [\bar{b}\bar{s}dd']$.

- $S(\bigcap_{(d',d)\in H} (d \to d' \to \alpha)) \leq S(d \to \beta_d)$ for all $d \in D$:
  fix any $\pi = d_k \to \ldots \to d_1 \to \sigma_b = [\bar{b}\bar{s}]$ such that $d_1 \ldots d_k = \bar{s} \in f^{-1}(C)$ and $\bar{s}d \in \mathrm{dom}(f)$. Since $f(\bar{s}) = C$ and $\bar{s}d \in \mathrm{dom}(f)$, we have $\bar{b}\bar{s}d = \bar{u}d'd$ for some $\bar{u} \in D^*$ and $d' \in D$ such that $(d', d) \in H$. We have:

$$S(\bigcap_{(d',d)\in H} (d \to d' \to \alpha)) \overset{S(\alpha)\leq[\bar{u}]}{\leq} [\bar{u}d'd] \equiv [\bar{b}\bar{s}d] \equiv d \to \pi\,.$$

- $S(\bigcap_{(d',d)\in V} (d \to \underbrace{\omega \to \ldots \to \omega \to}_{n-1 \text{ times}} d' \to \alpha)) \leq S(d \to \beta_d)$ for all $d \in D$:
  fix any $\pi = d_k \to \ldots \to d_1 \to \sigma_b = [\bar{b}\bar{s}]$ such that $d_1 \ldots d_k = \bar{s} \in f^{-1}(C)$ and $\bar{s}d \in \mathrm{dom}(f)$. Since $f(\bar{s}) = C$ and $\bar{s}d \in \mathrm{dom}(f)$, we have $\bar{b}\bar{s}d = \bar{u}d'\bar{v}d$ for some $\bar{u} \in D^*$, $\bar{v} \in D^{n-1}$ and $d' \in D$ such that $(d', d) \in V$. We have

$$S(\bigcap_{(d',d)\in V} (d \to \underbrace{\omega \to \ldots \to \omega \to}_{n-1 \text{ times}} d' \to \alpha)) \overset{S(\alpha)\leq[\bar{u}]}{\leq} [\bar{u}d'\bar{v}d] \equiv [\bar{b}\bar{s}d] \equiv d \to \pi\,. \qquad ◀$$

▶ **Lemma 31.** *Let T be a tiling system. If the constraint system $\mathcal{C}_T$ is satisfiable, then Constructor has a winning strategy in a two-player spiral tiling game in T.*

Proof. Assume that there exists a substitution S that satisfies the constraints $\mathcal{C}_T$ and wlog. uses only organized types. Constructor wins the game regardless of Spoilers moves as follows:

Let $\tau \equiv S(\sigma_b \cap \bigcap_{d'\in D} \bigcap_{d\in D} (d' \to d \to \beta_d))$, i.e. the rhs of (i). The initial game position is $\bar{b}$. Note that $\tau \leq \sigma_b \equiv [\bar{b}]$. We consider a single turn of Constructor from any game position $\bar{s} \in D^*$ that he may face.

Assume ($\star$) that the current game position $\bar{s}$ satisfies $\tau \leq [\bar{s}]$. Due to (i) and Corollary 11 we have the following cases

- If $S(\sigma_\perp^H) \leq [\bar{s}]$, then there exist $d, d' \in D$ such that $(d, d') \notin H$ and for some path π we have $d' \to d \to \pi \leq [\bar{s}]$. Therefore, $\bar{s} = \bar{u}dd'$ for some $\bar{u} \in D^*$ and Constructor wins because Spoilers last move d' violates H. Note that this is not possible for $\bar{s} = \bar{b}$ since $\bar{b}$ is horizontally consistent.
- If $S(\sigma_\perp^V) \leq [\bar{s}]$, then there exist $d, d' \in D$ such that $(d, d') \notin V$ and for some path π we have $d' \to \underbrace{\omega \to \ldots \to \omega \to}_{n-1 \text{ times}} d \to \pi \leq [\bar{s}]$. Therefore, $\bar{s} = \bar{u}d\bar{v}d'$ for some $\bar{u} \in D^*$, $\bar{v} \in D^{n-1}$ and Constructor wins because Spoilers last move d' violates V. Note that this is not possible for $\bar{s} = \bar{b}$ since $\bar{b}$ is too short.
- If $S(\sigma_t) \leq [\bar{s}]$, then Constructor wins because $t_n \to \ldots \to t_1 \to \pi \leq [\bar{s}]$ for some path π implies the winning condition $\bar{s} = \bar{u}\bar{t}$ for some $\bar{u} \in D^*$. Alternatively $\omega \to t_n \to \ldots \to t_1 \to \pi \leq [\bar{s}]$ for some path π implies $\bar{s} = \bar{u}\bar{t}d'$ for some $\bar{u} \in D^*$ and $d' \in D$, therefore Spoilers last move d' was illegal because the game already ended.
- If $S(\beta_d) \leq [\bar{s}]$ for some $d \in D$, then Constructor may safely place d as the next tile. We verify consistency wrt. H and V. First, due to (ii) there exists a path $d \to d' \to \pi$ for

some $d' \in D$ with $(d', d) \in H$ such that

$$d \to d' \to \pi \overset{(ii)}{\leq} S(d \to \beta_d) \overset{S(\beta_d)\leq[\bar{s}]}{\leq} [\bar{s}d]\,.$$

Therefore, $\bar{s}d = \bar{u}d'd$ for some $\bar{u} \in D^*$ and placing d does not violate H. Second, due to (iii) there exists a path $d \to \underbrace{\omega \to \ldots \to \omega \to}_{n-1 \text{ times}} d' \to \pi'$ for some $d' \in D$ with $(d', d) \in V$ such that

$$d \to \underbrace{\omega \to \ldots \to \omega \to}_{n-1 \text{ times}} d' \to \pi' \overset{(iii)}{\leq} S(d \to \beta_d) \overset{S(\beta_d)\leq[\bar{s}]}{\leq} [\bar{s}d]\,.$$

Therefore, $\bar{s}d = \bar{u}d'\bar{v}d$ for some $\bar{u} \in D^*$, $\bar{v} \in D^{n-1}$ and placing d does not violate V. Note that in neither case Constructor loses. If Constructor placed the tile $d \in D$, in which case we have $S(\beta_d) \leq [\bar{s}]$, Spoiler may place any tile $d' \in D$. The new game position is $\bar{s}dd'$. Note that our initial assumption $(\star)$ is inductively satisfied

$$\tau \leq S(d' \to d \to \beta_d) \overset{S(\beta_d)\leq[\bar{s}]}{\leq} [\bar{s}dd']\,.$$

Therefore, we may apply our argument inductively. Additionally, the game necessarily ends after a finite number of turns: if $\tau = \bigcap_{i \in I}(\sigma_1^i \to \ldots \sigma_{l_i}^i \to c_i)$ (for some index set I, integers $l_i \geq 0$ for $i \in I$, types σ_j^i for $i \in I$ and $1 \leq j \leq l_i$ and type constants c_i for $i \in I$), then $(\star)$, i.e. $\tau \leq [\bar{s}]$, cannot be satisfied by any $\bar{s} \in D^k$ with $k > \max\{l_i \mid i \in I\}$. ◀

▶ **Theorem 32.** *The intersection type satisfiability problem and the intersection type unification problem are* EXPTIME*-hard.*

Proof. Immediate from Lemma 29, Lemma 30 and Lemma 31, since all reduction steps are evidently computable in polynomial time. Moreover, satisfiability is polynomial time equivalent to unification. ◀

▶ **Corollary 33.** *Satisfiability and unification are* EXPTIME*-hard even in the presence of only one constant.*

Proof. Instead of using constants $\{d_1, \ldots, d_k, \bullet\}$, encode $[d_i] = \underbrace{\bullet \to \ldots \to \bullet \to}_{i \text{ times}} \bullet$ for $1 \leq i \leq k$ in the original proofs. Therefore, only one type constant $\bullet$ is sufficient. ◀

Note that without any constants satisfiability and unification are trivial by mapping all type variables to ω.

▶ **Corollary 34.** *Satisfiability and unification are* EXPTIME*-hard even if the codomain of substitutions is restricted to types of rank 1, i.e. intersections of simple types.*

Proof. In the proof of Lemma 30 each variable is substituted by an intersection of simple types, i.e. a rank 1 intersection type. ◀

Interestingly, the axiom **(RE)** $\omega \sim \omega \to \omega$ (resp. $\omega \leq \omega \to \omega$) is not necessary for the EXPTIME lower bound proof, while the axioms **(U)** and **(AB)** (resp. $\sigma \leq \omega$ and $\omega \to \tau \leq \sigma \to \tau$ derived from contravariance) play a crucial role in the construction of $\sigma_\perp^V$ to capture an exponential number of cases.

5 Conclusion and future work

We have positioned the intersection type unification problem as a natural object of study within unification theory and type theory, and we have provided the first nontrivial lower bound showing that the problem is of high complexity. Our EXPTIME-lower bound uses game-theoretic methods which may be useful for making further progress on the main open question for future work, that of decidability. Next steps include exploring variants and restrictions. Variants of intersection type subtyping theories (see [5]) give rise to a *family* of intersection type unification problems yet to be studied, e.g., the ω-free theory. We conjecture an NEXPTIME-upper bound for rank 1 restricted unification, in which variables are substituted by intersections of simple types. Since organized rank 1 subtyping corresponds to set inclusion, one can reduce a rank 1 unification problem to satisfiability of set constraints with projections [8] in finite sets. Unfortunately, standard set constraint interpretations may contain infinite sets, which is why this approach needs further investigation.

Acknowledgements. We would like to thank Boris Düdder (Dortmund), Paweł Urzyczyn, Aleksy Schubert, and Marcin Benke (Warsaw) and Mariangiola Dezani and the Torino λ-calculus group for helpful discussions. Thanks are also due to our reviewers for useful comments.

References

1 S. Anantharaman, P. Narendran, and M. Rusinowitch. Acid-unification is NEXPTIME-decidable. In *Mathematical Foundations of Computer Science 2003*, pages 169–178. Springer, 2003.

2 S. Anantharaman, P. Narendran, and M. Rusinowitch. Unification Modulo ACUI Plus Distributivity Axioms. *Journal of Automated Reasoning*, 33(1):1–28, 2004.

3 F. Baader and P. Narendran. Unification of Concept Terms in Description Logics. *Journal of Symbolic Computation*, 31:277–305, 2001.

4 H. Barendregt, M. Coppo, and M. Dezani-Ciancaglini. A Filter Lambda Model and the Completeness of Type Assignment. *Journal of Symbolic Logic*, 48(4):931–940, 1983. doi:10.2307/2273659.

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

6 G. Boudol and P. Zimmer. On type inference in the intersection type discipline. *Electr. Notes Theor. Comput. Sci.*, 136:23–42, 2005. doi:10.1016/j.entcs.2005.06.016.

7 G. Castagna, K. Nguyen, Z. Xu, and P. Abate. Polymorphic functions with set-theoretic types: Part 2: Local type inference and type reconstruction. In *Principles of Programming Languages POPL 2015*, pages 289–302. ACM, 2015.

8 W. Charatonik and L. Pacholski. Set constraints with projections are in NEXPTIME. In *35th Annual Symposium on Foundations of Computer Science*, pages 642–653. IEEE, 1994. doi:10.1109/SFCS.1994.365727.

9 B. S. Chlebus. Domino-tiling games. *Journal of Computer and System Sciences*, 32(3):374–392, 1986.

10 M. Coppo and P. Giannini. Principal types and unification for a simple intersection type system. *Inf. Comput.*, 122(1):70–96, 1995. doi:10.1006/inco.1995.1141.

11 M. Dezani-Ciancaglini and J. R. Hindley. Intersection Types for Combinatory Logic. *Theoretical Computer Science*, 100(2):303–324, 1992.

12 B. Düdder, M. Martens, and J. Rehof. Intersection Type Matching with Subtyping. In *Proceedings of TLCA'13*, Springer LNCS, 2013.

13 B. Düdder, M. Martens, J. Rehof, and P. Urzyczyn. Bounded Combinatory Logic. In *Proceedings of CSL'12*, volume 16 of *LIPIcs*, pages 243–258, 2012.

14 J. R. Hindley. The Simple Semantics for Coppo-Dezani-Sallé Types. In *International Symposium on Programming*, volume 137 of *LNCS*, pages 212–226. Springer, 1982.

15 J. R. Hindley and J. P. Seldin. *Lambda-calculus and Combinators, an Introduction.* Cambridge University Press, 2008.

16 A. J. Kfoury. Beta-reduction as unification. *Banach Center Publications*, 46(1):137–158, 1999.

17 A. J. Kfoury and J. B. Wells. Principality and Type Inference for Intersection Types Using Expansion Variables. *Theor. Comput. Sci.*, 311(1-3):1–70, 2004.

18 T. Kurata and M. Takahashi. Decidable properties of intersection type systems. In *TLCA*, volume 902 of *LNCS*, pages 297–311. Springer, 1995.

19 J. Rehof and P. Urzyczyn. Finite Combinatory Logic with Intersection Types. In *Proceedings of TLCA'11*, volume 6690 of *LNCS*, pages 169–183. Springer, 2011.

20 J. Rehof and P. Urzyczyn. The Complexity of Inhabitation with Explicit Intersection. In *Kozen Festschrift*, volume 7230 of *LNCS*, pages 256–270. Springer, 2012.

21 S. Ronchi Della Rocca. Principal Type Scheme and Unification for Intersection Type Discipline. *Theor. Comput. Sci.*, 59:181–209, 1988.

22 R. Statman. A finite model property for intersection types. In *Proceedings Seventh Workshop on Intersection Types and Related Systems, ITRS 2014, Vienna, Austria, 18 July 2014.*, pages 1–9, 2015. doi:10.4204/EPTCS.177.1.