Small Substructures and Decidability Issues for First-Order Logic with Two Variables*

Emanuel Kieronski† Martin Otto‡

Abstract

We study first-order logic with two variables FO^2 and establish a small substructure property. Similar to the small model property for FO^2 we obtain an exponential size bound on embedded substructures, relative to a fixed surrounding structure that may be infinite. We apply this technique to analyse the satisfiability problem for FO^2 under constraints that require several binary relations to be interpreted as equivalence relations. With a single equivalence relation, FO^2 has the finite model property and is complete for non-deterministic exponential time, just as for plain FO^2. With two equivalence relations, FO^2 does not have the finite model property, but is shown to be decidable via a construction of regular models that admit finite descriptions even though they may necessarily be infinite. For three or more equivalence relations, FO^2 is undecidable.

0 Introduction

The undecidability of the satisfiability problem for first-order logic has inspired the classification of various syntactic fragments of first-order logic with a view to delineating the boundary of decidability as well as to finding useful decidable fragments. The main programme of this kind, which has led to a complete classification, concerned the taxonomy of prenex normal form formulae w.r.t. quantifier prefixes [3]. Along an orthogonal direction, one may investigate fragments defined not with reference to prenex normal form, but in terms of other uniform structural constraints on the quantification patterns. Modal logics, with their characteristic relativisation of all quantifiers by binary edge predicates, provide a typical example of a benign fragment of this kind. A more recent extension led to the guarded fragment [1, 11]. More crudely, a mere restriction of the number of distinct variable symbols leads to the finite variable fragments FO^k of first-order logic. Interestingly, both the modal families of logics and the finite variable fragments can also be motivated in terms of model theoretic games

---

*Full version of LICS 2005 paper [21].

†Institute of Computer Science, University of Wrocław, kiero@ii.uni.wroc.pl, supported by EPSRC grant GR/R11896/01 and Polish Ministry of Science and Higher Education research project nr N206 022 31/3660 2006/2009

‡Department of Mathematics, Technische Universität Darmstadt, otto@mathematik.tu-darmstsdt.de, research partially supported by EPSRC grant GR/R11896/01 and DFG grant OT 147/5-1

– namely bisimulation games for modal logics, and k-pebble games for FO^k. Correspondingly, and unlike prefix classes, these fragments enjoy natural closure properties which support some characteristic model theory. In particular, the finite variable fragments play a prominent role in finite model theory. In terms of satisfiability, FO^2 is decidable while FO^3 is undecidable. FO^2 here stands for the fragment of first-order logic with equality with only two variable symbols x and y, in finite relational vocabularies (without constants or function symbols). Without loss of generality, we also only consider vocabularies of width 2, without relation symbols of arities greater than 2.

The first decidability proof for FO^2 was given by Scott [30], via a reduction to the so-called Gödel prefix class $\exists^*\forall\forall\exists^*$, which however is only decidable in the absence of equality [10]. Full decidability for FO^2 with equality is due to Mortimer [25]. Mortimer shows FO^2 has the finite model property, and in fact every satisfiable FO^2 sentence has a model of size at most doubly exponential in the length of the sentence. This bound on the size of small models is improved to single exponential by Grädel, Kolaitis and Vardi in [12], which leads to their result that FO^2 is decidable in NEXPTIME, and in fact NEXPTIME-complete.

The study of FO^2 is also motivated by the fact that it embeds propositional modal logic K, via the standard translation. Numerous variants and extensions of modal logic find applications in various areas of computer science, including verification of software and hardware, distributed systems, knowledge representation and artificial intelligence. These applications are supported by the very good algorithmic and model-theoretic behaviour of modal logics, including their remarkably robust decidability which persists under various extensions towards greater expressiveness. Some of these extensions are equally well motivated in the context of two-variable logic. Description logics in particular naturally fit into the range between modal and two-variable logics, see [2] and in particular [6] for their connection with finite variable fragments. The extension of FO^2 by counting quantifiers [17], for instance, is decidable by [14, 28] although it does not have the finite model property. In analogy with graded modalities, it covers certain description logics with number constraints. But also in systematic terms the question naturally arises, to which extent FO^2 shares the good algorithmic behaviour of modal logics. The picture that emerged in [15] shows that, with the notable exception of the counting extension, most extensions of FO^2 are undecidable, compare also [13]. This includes e.g. various extensions by mechanisms for fixed points or transitive closures, in analogy with the modal μ-calculus or computation tree logics. In many cases, the results of these investigations can be phrased either for satisfiability of extensions of FO^2, or, alternatively, of FO^2 itself over restricted classes of structures. This interplay is fruitfully employed in [15, 26].

In connection with modal logics, or with applications of modal or two-variable logics in areas like knowledge representation or for description logics, a restriction of the underlying class of models is often very natural. Modal correspondence theory, for instance, associates transitivity of accessibility relations with the modal logic K4; equivalence relations with the modal logic S5. Multi-S5 systems with k equivalence relations among their accessibility relations can be used to model knowledge systems for k independent agents; linear orders as accessibility relations play an obvious role for linear temporal logics, etc. For FO^2 over such classes of structures, undecidability is established under several such constraints in [15, 13] and in particular in the presence of four equivalence relations. In the presence of a linear order, on the other hand, decidability is shown in [26]. Another interesting, related result is obtained in [5]: decidability is shown there for FO^2 with one equivalence relation and one linear order (accessible by both the order and the successor predicates) over so-called *data words*, i.e., words over a unary alphabet in which the linear order corresponds to the natural order on positions of letters. For some more results related to FO^2 over data words see [24, 29, 7, 27].

In this paper we concentrate on a complete analysis of the important case of models with several equivalence relations, by clarifying the situation for up to three equivalence relations.

We look at finite relational vocabularies $\tau = \tau_0 \dot\cup \tau_{\mathrm{eq}}$ where τ_{eq} consists of a finite number of distinguished binary relations (typically E or $E_i, i = 1, 2, 3$). We let $\mathcal{EQ}[\tau_0; \tau_{\mathrm{eq}}]$ denote the class of all $\tau_0 \dot\cup \tau_{\mathrm{eq}}$ structures $\mathfrak{A} = (\mathfrak{A}_0, (E^{\mathfrak{A}})_{E \in \tau_{\mathrm{eq}}})$ (with at least two elements) that interpret the relations $E \in \tau_{\mathrm{eq}}$ as equivalence relations. We refer to such structures as *equivalence structures*.

$\mathrm{SAT}(\mathcal{L}, \mathcal{C})$, the satisfiability problem for $\mathcal{L}$ over the class $\mathcal{C}$, is the decision problem, for sentences $\varphi \in \mathcal{L}$, whether φ has a model in $\mathcal{C}$. We say that $\mathcal{L}$ has the finite model property (or a small model property) over $\mathcal{C}$, if every sentence $\varphi \in \mathcal{L}$ that has a model in $\mathcal{C}$ also has a finite (small) model in $\mathcal{C}$. [1] $\mathrm{FINSAT}(\mathcal{L}, \mathcal{C})$ stands for the satisfiability problem for $\mathcal{L}$ in restriction to finite structures from $\mathcal{C}$, so that the finite model property for $\mathcal{L}$ over $\mathcal{C}$ is equivalent to $\mathrm{FINSAT}(\mathcal{L}, \mathcal{C}) = \mathrm{SAT}(\mathcal{L}, \mathcal{C})$. In these terms our main results for FO^2 over equivalence structures are the following.

Theorem 1 (i) FO^2 *has an exponential model property over* $\mathcal{EQ}[\tau_0; E]$. *Hence* $\mathrm{SAT}(\mathrm{FO}^2, \mathcal{EQ}[\tau_0; E])$ *and* $\mathrm{FINSAT}(\mathrm{FO}^2, \mathcal{EQ}[\tau_0; E])$ *are* NEXPTIME *complete.*

(ii) FO^2 *does not have the finite model property over* $\mathcal{EQ}[\tau_0; E_1, E_2]$. *However,* $\mathrm{SAT}(\mathrm{FO}^2, \mathcal{EQ}[\tau_0; E_1, E_2])$ *is decidable in* 3NEXPTIME.

(iii) $\mathrm{SAT}(\mathrm{FO}^2, \mathcal{EQ}[\tau_0; E_1, E_2, E_3])$ *and* $\mathrm{FINSAT}(\mathrm{FO}^2, \mathcal{EQ}[\tau_0; E_1, E_2, E_3])$ *are undecidable; in fact* FO^2 *over* $\mathcal{EQ}[\tau_0; E_1, E_2, E_3]$ *forms a conservative reduction class.*

Our decidability result for two equivalence relations should also be contrasted with another result by the first author, that FO^2 is undecidable in the presence of two transitive relations [20]. This fact was independently proved in [18]. A further improvement was obtained lately in [23] where undecidability of FO^2 in the presence of one equivalence and one transitive relation is shown. In [23] it is also shown that $\mathrm{FINSAT}(\mathrm{FO}^2, \mathcal{EQ}[\tau_0; E_1, E_2])$ is decidable.

En route to the decidability result we establish a *small substructure property* for FO^2 which does not directly focus on entire models of FO^2-sentences but rather on small substructures that are parts of the actual models. This technique proves to be applicable in the case of equivalence structures with two equivalence relations, where the finite model property for FO^2 fails. As we shall see, FO^2 sentences can force models in $\mathcal{EQ}[\tau_0; E_1, E_2]$ to have infinitely many equivalence classes as well as to have infinite equivalence classes. By our small substructure property the size of the

[1] With *small model property* we refer to the existence of some function that provides an upper bound on the size of small models for any satisfiable formula, in terms of the size of the formula or its vocabulary.

equivalence classes of the common refinement $E = E_1 \cap E_2$ of E_1 and E_2 can be exponentially bounded. This serves as a crucial step towards the construction of regular infinite models that admit finite descriptions. The following is a truncated version of our small substructure property, compare Proposition 4 for a full statement. By $\mathfrak{A} \Rightarrow_{\forall\forall} \mathfrak{A}'$ we denote the transfer property that any prenex $\forall\forall$-formula satisfied in $\mathfrak{A}$ is also satisfied in $\mathfrak{A}'$; similarly for $\mathfrak{A} \Rightarrow_{\forall\exists} \mathfrak{A}'$.

Theorem 2 *Let $\mathfrak{A}$ be a τ-structure with universe $A = B \dot\cup C$, $\mathfrak{B} := \mathfrak{A}{\restriction}B$. Then there is a τ-structure $\mathfrak{A}'$ with universe $A' = B' \dot\cup C$, for some set B' of size exponential in τ, such that $\mathfrak{A}'{\restriction}C = \mathfrak{A}{\restriction}C$, $\mathfrak{A} \Rightarrow_{\forall\forall} \mathfrak{A}'$ and $\mathfrak{A} \Rightarrow_{\forall\exists} \mathfrak{A}'$.*

Small substructure properties of this kind may be interesting in their own right. In settings where some parts of a larger structure are controlled by specifications in a logic $\mathcal{L}$ in their relationship to that surrounding structure, it may be natural to consider variations of these parts while the surrounding structure stays unchanged. One may think of descriptions of distributed systems, in which only some "local" components are amenable to modifications while the overall or remote environment is regarded as fixed and not locally controllable. Then the question arises whether the local components can be replaced by smaller equivalent components, without violating the global specification and possibly preserving some logical features of the interface between the local components and their environment. Our small small substructures property captures such a setting for FO^2 and yields exponential bounds on the size of crucial components, in a context where a global finite model is not available.

Plan of the paper. The paper is organised as follows. Section 1 introduces basic terminology and notation, and recall the well-known Scott normal form for FO^2; Section 2 contains the full statement and the proof of our small substructures property. The remainder of the paper concerns the satisfiability problem for FO^2 over structures with several equivalence relations.

In Section 3 we formally introduce the notion of equivalence structures and adapt Scott normal form to our purposes.

Section 4 deals with decidability and the finite model property for FO^2 in the presence of a single equivalence relation. This is achieved in two steps: we first show that every satisfiable formula has a model whose equivalence classes are exponentially size bounded; we then show that every formula has a model with exponentially many such classes.

Section 5 is devoted to FO^2 in the presence of two equivalence relations. We exhibit some key examples to illustrate the crucial difference that a second equivalence relation makes and introduce an auxiliary combinatorial problem to which satisfiability of FO^2 over structures with two equivalence relations can be reduced. We first show how to solve a simplified variant of this *coloured castles problem*, which allows us to concentrate on the main combinatorial issues and to hide some of the more wearisome detail. This approach is then extended to cover the *coloured castles problem* in the full generality required.

Finally, in Section 6, we show undecidability of FO^2 in the presence of three equivalence relations.

1 Quantifier-free types and Scott normal form

We use the term *type* to refer to quantifier-free types. Let $\mathfrak{A}$ be a τ-structure with universe A. For $a \in A$, the 1-type of a in $\mathfrak{A}$ is

$$\mathrm{tp}_{\mathfrak{A}}(a) = \big\{\varphi(x) \in \mathrm{FO}^2[\tau] \colon \varphi \text{ quantifier-free }, \mathfrak{A} \models \varphi[a]\big\}.$$

We let $\boldsymbol{\alpha}[\mathfrak{A}] = \{\mathrm{tp}_{\mathfrak{A}}(a) \colon a \in A\big\}$ be the set of all 1-types of $\mathfrak{A}$. Quantifier-free 2-types $\mathrm{tp}_{\mathfrak{A}}(a_1, a_2)$ of non-degenerate pairs, $a_1 \neq a_2$, are similarly defined. We let $\boldsymbol{\beta}[\mathfrak{A}] = \{\mathrm{tp}_{\mathfrak{A}}(a_1, a_2) \colon a_1 \neq a_2,\, a_1, a_2 \in A\big\}$ be the set of all 2-types of $\mathfrak{A}$.

We also write $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ for the sets of all 1-types and 2-types, across all $\mathfrak{A}$. We sometimes identify a type with a corresponding quantifier-free formula that determines it, which is just a conjunction of atoms or negated atoms. Note that the size of the sets $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is bounded by an exponential function in the size of the vocabulary.

We typically write $\alpha = \alpha(x)$ for a 1-type $\alpha \in \boldsymbol{\alpha}$, and $\beta = \beta(x, y)$ for a 2-type $\beta \in \boldsymbol{\beta}$. Then $\alpha(y)$ is the result of switching x for y in α. Also $\beta \restriction x$ is the 1-type consisting of all the $\varphi(x) \in \beta$; similarly for $\beta \restriction y$. However, we also write $\beta \restriction y = \alpha$ instead of the formally correct $\beta \restriction y = \alpha(y)$

Natural terminology with regard to types and their realisations applies. For instance, we say that an element $b \in \mathfrak{B}$ realises the type α if $\mathrm{tp}_{\mathfrak{B}}(b) = \alpha$ (or $\mathfrak{B} \models \alpha[b]$). For a type $\alpha \in \boldsymbol{\alpha}$, and a subset $S \subseteq A$ of a structure $\mathfrak{A}$ we denote as $\alpha[S] \subseteq A$ the set of all those $a \in S$ that have type α. Conversely $\boldsymbol{\alpha}[S]$, for a subset $S \subseteq \mathfrak{A}$, denotes the set of all 1-types realised in S, $\boldsymbol{\alpha}[S] = \big\{\mathrm{tp}_{\mathfrak{A}}(a) \colon a \in S\big\}$. Similarly $\boldsymbol{\beta}[S_1, S_2]$, for subsets $S_i \subseteq A$, is the set of all 2-types $\mathrm{tp}_{\mathfrak{A}}(a_1, a_2)$ with $a_i \in S_i$; $\boldsymbol{\beta}[a, S]$ for a subset $S \subseteq A$ is the set of all 2-types $\mathrm{tp}_{\mathfrak{A}}(a, a')$ with $a' \in S$, etc.

The following normal form is achieved through a natural process of relational Skolemisation, which essentially introduces for every quantified subformula of a given FO^2 formula in negation normal form a new binary relation symbol. The semantics of φ is then captured by the implicit definition of the new relations, which can naturally be given in $\forall\forall$ and $\forall\exists$ format. See, for instance, [12, 13] for expositions.

Proposition 3 (Scott normal form) *For $\varphi \in \mathrm{FO}^2[\tau]$ one can compute in polynomial time a* Scott normal form formula $\tilde{\varphi} \in \mathrm{FO}^2[\tilde{\tau}]$*, whose length is linear in the length of φ, of the form*

$$\tilde{\varphi} = \forall x \forall y\, \chi_0 \wedge \textstyle\bigwedge_{i=1}^{m} \forall x \exists y\, \chi_i$$

for quantifier-free formulae $\chi_i \in \mathrm{FO}^2[\tilde{\tau}]$, such that $\tilde{\varphi}$ is satisfiability equivalent to φ.

For normal form φ, whether or not $\mathfrak{A} \models \varphi$ is determined by the family of sets $\boldsymbol{\beta}[a, A] = \{\mathrm{tp}_{\mathfrak{A}}(a, b) \colon b \in A\big\}$ of types incident with a, for $a \in A$. Whether $\mathfrak{A}$ satisfies the $\forall\forall$ part of φ is determined by $\boldsymbol{\beta}[\mathfrak{A}] = \boldsymbol{\beta}[A, A]$; for the $\forall\exists$ parts, $\mathfrak{A}, a \models \exists \chi_i$ iff $\beta \models \chi_i$ for some $\beta \in \boldsymbol{\beta}[a, A]$.

2 A small substructure property for FO^2

Consider a structure $\mathfrak{A}$ and a fixed subset $B \subseteq A$ with induced substructure $\mathfrak{B} := \mathfrak{A} \restriction B$. We want to perform some surgery on $\mathfrak{A}$ which in effect replaces the substructure $\mathfrak{B}$ by some 'equivalent' $\mathfrak{B}'$ of bounded size.

Proposition 4 *Let $\mathfrak{A}$ be a τ-structure, $\mathfrak{B} = \mathfrak{A}{\restriction}B$ for some $B \subseteq A$, $C := A \setminus B$. Then there is a τ-structure $\mathfrak{A}'$ with universe $A' = B' \dot\cup C$ for some set B' of size polynomial in $|\boldsymbol{\beta}[\mathfrak{A}]|$ such that*

(i) $\mathfrak{A}'{\restriction}C = \mathfrak{A}{\restriction}C$.
(ii) $\boldsymbol{\alpha}[B'] = \boldsymbol{\alpha}[B]$, *whence* $\boldsymbol{\alpha}[\mathfrak{A}'] = \boldsymbol{\alpha}[\mathfrak{A}]$.
(iii) $\boldsymbol{\beta}[B'] = \boldsymbol{\beta}[B]$ *and* $\boldsymbol{\beta}[B', C] = \boldsymbol{\beta}[B, C]$, *whence* $\boldsymbol{\beta}[\mathfrak{A}'] = \boldsymbol{\beta}[\mathfrak{A}]$.
(iv) *for each* $b' \in B'$ *there is some* $b \in B$ *with* $\boldsymbol{\beta}[b', A'] \supseteq \boldsymbol{\beta}[b, A]$.
(v) *for each* $a \in C$: $\boldsymbol{\beta}[a, B'] \supseteq \boldsymbol{\beta}[a, B]$.

Note that (iii)–(v) imply that $\mathfrak{A} \models \varphi \;\Rightarrow \mathfrak{A}' \models \varphi$ *for all normal form* φ.

We point out that for plain FO^2 we reproduce the known small model property for FO^2 from [12]. Putting $B := A$ and observing that $\boldsymbol{\beta}[\mathfrak{A}] \subseteq \boldsymbol{\beta}$ is exponential in the vocabulary of (the normal form of) φ, we obtain an exponential size bound on $\mathfrak{A}' \models \varphi$. Also the proof of the proposition has crucial similarities with the proof by Grädel, Kolaitis, and Vardi for the small model property. However, our result is a proper extension, as will become apparent from its uses in settings where the surrounding structure cannot be made finite.

Proof Let $m := |\boldsymbol{\beta}[\mathfrak{A}]|$ be the number of 2-types realised in $\mathfrak{A}$, and enumerate these as $\boldsymbol{\beta}[\mathfrak{A}] := \{\beta_1, \ldots, \beta_m\}$. Note that m is bounded by $|\boldsymbol{\beta}|$, which is exponential in $|\tau|$. W.l.o.g. assume that $B \subseteq \alpha[A]$ consists of elements of the same 1-type α. The general case may be reduced to this one, by repeated application to the sets $\alpha[B]$ for all relevant α. Also assume that $B = \alpha[B]$ contains at least two elements so that $\boldsymbol{\beta}[\mathfrak{B}] \neq \emptyset$. Otherwise put $B' := B$.

We want to find a suitable set B' of new realisations of α, and links between B' and C and within B' in accordance with (iii)–(v). Technically $\mathfrak{A}'$ is specified by a consistent choice of 2-types for any pair involving at least one new element $b' \in B'$.

The general idea is to distinguish a small (exponentially bounded) set $M \subseteq C$, which, from the point of view of B, serves as a sufficiently rich representation of $\mathfrak{C}$, i.e., the number of realisations of each 1-type from $\boldsymbol{\alpha}[C]$ in M is sufficient to fulfil all $\boldsymbol{\beta}(b, C)$ requirements of elements $b \in B$. On the other hand the new set B' is large enough to fulfil all $\boldsymbol{\beta}[a, B]$ requirements of elements $a \in M$. The strategy of connecting B' and M without conflict is given in steps (1)-(3) below. The allocation of 2-types inside B' (step (5) below) is similar to the construction from [12]: B' is divided into three subsets, which we think of as arranged in a directed 3-cycle, such that the requirements of elements from one part are met by elements in the next part.

Let us give the details of the construction. For $\alpha' \in \boldsymbol{\alpha}[\mathfrak{A}]$ let

$$\begin{aligned} \boldsymbol{\beta}(\alpha') &:= \big\{\mathrm{tp}_{\mathfrak{A}}(b, a) \colon b \in B, a \in \alpha'[C]\big\} \subseteq \boldsymbol{\beta}[\mathfrak{A}], \\ N(\alpha') &:= |\boldsymbol{\beta}(\alpha')|, \\ n(\alpha') &:= \min\big(N(\alpha'), |\alpha'[C]|\big). \end{aligned}$$

So $\boldsymbol{\beta}(\alpha')$ is the set of 2-types linking elements of B with elements of type α' outside B. Note that $n(\alpha') \leqslant N(\alpha') \leqslant m$. For each $\alpha' \in \boldsymbol{\alpha}[C]$ choose a subset $M(\alpha') \subseteq \alpha'[C]$

of size $n(\alpha')$. Let M be the union of the (disjoint) sets $M(\alpha')$. We replace B by

$$B' = \{0,1,2\} \times \{1,\dots,m\} \times M,$$

consisting of $3m|M| \leqslant 3m^3$ elements that realise type α. It remains to allocate 2-types over $B' \times B'$ and $B' \times C$ in a consistent fashion and such that requirements (iii)–(v) are met. This is done in stages.

(1) We allocate 2-types for some pairs $(b', a) \in B' \times M$ to settle (v) for all $a \in M$. For each $a \in M(\alpha') \subseteq M$ and $\beta_k \in \boldsymbol{\beta}(\alpha')$, put $\mathrm{tp}((0,k,a),a) := \beta_k$.

Note that for each $b' \in B'$, $\mathrm{tp}(b', a')$ has been set for at most one element a' (necessarily $a' \in M$), and that in this case $\mathrm{tp}(b', a') = \mathrm{tp}_{\mathfrak{A}}(b, a)$ for some $(b, a) \in B \times C$.

(2) For each $b' \in B'$, we allocate 2-types to some further pairs (b', a) with $a \in M$, in such a way as to guarantee (iv) at b'. We treat one b' at a time.

– If no 2-type involving b' has been determined in stage (1), pick some element $b \in B$ and for each 2-type $\beta \in \boldsymbol{\beta}[b, C]$ and $\alpha' = \beta{\restriction}y$, select a fresh element $a = a(b', \beta) \in M(\alpha')$ and put $\mathrm{tp}(b', a) := \beta$. There are sufficiently many elements in $M(\alpha')$ by the definition of $n(\alpha')$.

– If one 2-type involving b' (and an element of M) has been determined in stage (1), $\mathrm{tp}_{\mathfrak{A}'}(b', a') = \mathrm{tp}_{\mathfrak{A}}(b, a)$ for some reference elements $b \in B$ and $a \in C$. We realise further 2-types $\beta \in \boldsymbol{\beta}[b, C] \subseteq \bigcup_{\alpha'} \boldsymbol{\beta}(\alpha')$ at b' with partner elements $a \in M(\alpha')$ for the appropriate α'. By the choice of $M(\alpha')$ there are sufficiently many distinct target elements available in each $M(\alpha')$ (just as there were in $\alpha'[C]$ for the reference element b). This makes sure that $\boldsymbol{\beta}[b', C] \supseteq \boldsymbol{\beta}[b, C]$.

(3) Allocation of all remaining 2-types for pairs $(b', a) \in B' \times M$. Choose $b_0 \in B$. For each $b' \in B'$ and $a \in M$, whose 2-type has not been attributed, put $\mathrm{tp}(b', a) := \mathrm{tp}_{\mathfrak{A}}(b_0, a)$. Since all types added in this step as well as in steps (1) and (2) belong to $\boldsymbol{\beta}[B, C]$, we have $\boldsymbol{\beta}[B', C] = \boldsymbol{\beta}[B, C]$.

(4) Allocation of 2-types to pairs $(b', a) \in B' \times (C \setminus M)$. For each $a \in C \setminus M$ of type α', pick $a_0 \in M(\alpha')$ and set $\mathrm{tp}(b', a) := \mathrm{tp}(b', a_0)$ for all $b' \in B$. Together with (1) this settles (v) for all $a \in C$.

(5) Allocation of 2-types to pairs in $B' \times B'$. For $(i, j, a) \in B'$ and $\beta_k \in \boldsymbol{\beta}[\mathfrak{B}]$ put $\mathrm{tp}((i,j,a),(i',k,a)) := \beta_k$ where $i' = (i+1) \bmod 3$. For any two distinct elements of B' whose type has not been allocated, put arbitrary $\beta \in \boldsymbol{\beta}[\mathfrak{B}]$. This settles (iv) and ensures $\boldsymbol{\beta}[B'] = \boldsymbol{\beta}[B]$, which completes (iii). □

3 Equivalence structures

We now restrict attention to finite relational vocabularies $\tau = \tau_0 \mathbin{\dot\cup} \tau_{\mathrm{eq}}$ where τ_{eq} consists of a finite number of distinguished binary relations to be interpreted as equivalence relations (we use E or $E_i, i = 1,2,3$ for these).

Definition 5 $\mathcal{EQ}[\tau_0; \tau_{\mathrm{eq}}]$ denotes the class of all *equivalence structures* in the vocabulary $\tau = \tau_0 \mathbin{\dot\cup} \tau_{\mathrm{eq}}$, i.e., the class of all $\tau_0 \mathbin{\dot\cup} \tau_{\mathrm{eq}}$ structures $\mathfrak{A} = (\mathfrak{A}_0, (E^{\mathfrak{A}})_{E \in \tau_{\mathrm{eq}}})$ with at least two elements that interpret the relations $E \in \tau_{\mathrm{eq}}$ as equivalence relations.

Types and Scott normal form for equivalence structures. When we are interested in models in $\mathcal{EQ}[\tau_0, \tau_{\rm eq}]$, we only admit types that are realisable in these. In the presence of one or more equivalence relations we want to distinguish 2-types according to equivalences/non-equivalences between x and y. For $\tau_{\rm eq} = \{E\}$, we distinguish $\boldsymbol{\beta}^+$ and $\boldsymbol{\beta}^-$ such that $\boldsymbol{\beta} = \boldsymbol{\beta}^+ \dot\cup \boldsymbol{\beta}^-$, where all $\beta \in \boldsymbol{\beta}^+$ contain the formula Exy, while those in $\boldsymbol{\beta}^-$ contain its negation. In the case of $\tau_{\rm eq} = \{E_1, E_2\}$ we correspondingly distinguish four sets: $\boldsymbol{\beta}^{++}, \boldsymbol{\beta}^{+-}, \boldsymbol{\beta}^{-+}, \boldsymbol{\beta}^{--}$, such that for instance $\beta \in \boldsymbol{\beta}^{-+}$ iff $(\neg E_1xy \wedge E_2xy) \in \beta$.

We also use superscripts $+$ and $-$ to indicate for an individual quantifier-free formula which equivalences/non-equivalences it stipulates. For instance, with $\tau_{\rm eq} = \{E_1, E_2\}$ and for a quantifier-free formula $\chi = \chi(x, y) \in \mathrm{FO}^2[\tau_0]$ we let

$$\chi^{+-}(x, y) := \chi(x, y) \wedge (\neg x = y \rightarrow (E_1xy \wedge \neg E_2xy)).$$

χ^+ and χ^- over $\tau_{\rm eq} = \{E\}$ are similarly defined. This decomposition of formulae leads to the following variant of the Scott normal form adapted to structures in $\mathcal{EQ}[\tau_0; \tau_{\rm eq}]$.

An $\mathcal{EQ}[\tau_0, E_1, E_2]$ Scott normal form sentence is of the form

$$\forall x \forall y\ \chi_0 \wedge \bigwedge_{i=1}^{m} \forall x \exists y\ \chi_i^{s_i},$$

for quantifier-free $\chi_0 \in \mathrm{FO}[\tau_0; E_1, E_2]$ and $\chi_i \in \mathrm{FO}[\tau_0]$ and $s_i \in \{+, -\} \times \{+, -\}$. For $\tau_{\rm eq} = \{E\}$, similarly, $\mathcal{EQ}[\tau_0, E]$ Scott normal form is $\forall x \forall y\ \chi_0 \wedge \bigwedge_{i=1}^{m} \forall x \exists y\ \chi_i^{s_i}$ for quantifier-free $\chi \in \mathrm{FO}[\tau_0; E]$ and $\chi_i \in \mathrm{FO}[\tau_0]$ and $s_i \in \{+, -\}$.

Proposition 6 ($\mathcal{EQ}$ Scott normal form) *For $\varphi \in \mathrm{FO}^2[\tau_0 \cup \tau_{eq}]$ there is a polynomial time computable $\mathcal{EQ}[\tau; \tau_{eq}]$ Scott normal form sentence $\tilde\varphi \in \mathrm{FO}^2[\tilde\tau_0 \cup \tau_{eq}]$, whose length is linear in the length of φ, such that φ and $\tilde\varphi$ are satisfiability equivalent over $\mathcal{EQ}[\tau; \tau_{eq}]$.*

In terms of this normal form, φ can be thought of as stipulating constraints on the sets of atomic 1-types and 2-types that may occur (the $\forall\forall$ requirements in χ_0) and requirements for certain *witnesses* to be provided at every element (the $\forall\exists$ requirements in the $\chi_i^{s_i}$ for $1 \geqslant 1$). Among the $\forall\exists$ requirements we distinguish, for instance in the case of two equivalence relations E_1 and E_2, by the superscripts $s_i \in \{+, -\}^2$ requirements that need to be met by witnesses

- within the same intersection of E_1- and E_2-classes ($++$),
- within the same E_1-class but in a distinct E_2-class ($+-$),
- within the same E_2-class but in a distinct E_1-class ($-+$), or
- in a distinct E_1-class and in a distinct E_2-class ($--$).

Correspondingly, such witnesses are linked to the base point by 2-types in $\boldsymbol{\beta}^{++}$, $\boldsymbol{\beta}^{+-}$, $\boldsymbol{\beta}^{-+}$, or $\boldsymbol{\beta}^{--}$, respectively. For instance, a witness at $a \in \mathfrak{A}$ for an $\forall\exists$ requirement $\forall x \exists y \chi_i^{+-}(x, y)$ in φ is an element a' such that $\mathfrak{A} \models \chi_i^{+-}[a, a']$, which means that $\mathrm{tp}_{\mathfrak{A}}(a, a') \in \boldsymbol{\beta}^{+-}$ contains $\chi_i^{+-}(x, y)$. If there is such a witness, we also say that that the requirement is *fulfilled* at a.

4 One equivalence relation

We consider models in $\mathcal{EQ}[\tau_0; E]$, with $\tau_{\text{eq}} = \{E\}$. Structures in $\mathcal{EQ}[\tau_0; E]$ are of the form $\mathfrak{A} = (A, E^{\mathfrak{A}}, \ldots)$, $E^{\mathfrak{A}}$ an equivalence relation over A. If $B \subseteq A$ is an equivalence class w.r.t. $E^{\mathfrak{A}}$, we refer also to the substructure $\mathfrak{B} = \mathfrak{A}{\restriction}B$ as an equivalence class of $\mathfrak{A}$.

Note that if $\mathfrak{A} \models \varphi$, where φ is in normal form according to Proposition 3, then each equivalence class $\mathfrak{B} = \mathfrak{A}{\restriction}B$ is a model of the $\forall\forall$ constituent as well as of all the $\forall\exists$ constituents of type χ^+.

4.1 An exponential model property

We shall find small models, whose size is exponential in terms of $|\tau_0|$ or in the length of φ, in two stages:

small classes: replacing each individual equivalence class $\mathfrak{A}{\restriction}B$ in $\mathfrak{A}$ by a small (exponential size) structure, while retaining the remainder of $\mathfrak{A}$ unchanged; for this we apply the small substructure property, which preserves any normal form φ.

few classes: building a new structure from an exponential number of isomorphic copies of classes from $\mathfrak{A}$, such that again any normal form φ is preserved.

As there are only exponentially many distinct 1-types, one can afford to realise exactly the same 1-types in the target structure as in the given model. At the level of equivalence classes, however, the given model may have doubly exponentially many types, distinguished by their composition in terms of 1-types. Out of these one needs to select a set of just exponentially many types to be used in the new small model.

Together these two steps allow us transform any given $\mathcal{EQ}[\tau_0; E]$ structure into an exponentially size bounded $\mathcal{EQ}[\tau_0; E]$ structure while preserving the truth of all normal form sentences. We thus get the following. The proof according to the two steps outlined above is provided in the following sections.

Proposition 7 *There is an exponential function f such that any normal form sentence $\varphi \in \mathrm{FO}^2[\tau_0; E]$ that is satisfiable in $\mathcal{EQ}[\tau_0; E]$ also has a model in $\mathcal{EQ}[\tau_0; E]$ of size bounded by $f(|\tau_0|)$.*

It follows that we can check satisfiability by guessing an exponential size model and verifying φ; the resulting complexity matches the known lower bound, which trivially transfers from FO^2 (we may add a dummy equivalence relation to any $\varphi \in \mathrm{FO}^2[\tau_0]$ with the clause $\forall x \forall y Exy$).

Corollary 8 $\mathrm{SAT}(\mathrm{FO}^2, \mathcal{EQ}[\tau_0; E])$ *is* NEXPTIME*-complete.*

4.1.1 Small classes

Lemma 9 *Every normal form φ that is satisfiable in an $\mathcal{EQ}[\tau_0, E]$ model, has an $\mathcal{EQ}[\tau_0, E]$ model with at most countably many equivalence classes each of which is bounded in size by an exponential function in $|\tau_0|$.*

Let $\mathfrak{A} \in \mathcal{EQ}[\tau_0; E]$. Towards the proof we first show how to replace a single equivalence class $\mathfrak{B}$ in $\mathfrak{A}$ by a new substructure $\mathfrak{B}'$ that also consists of a single E-class, of size exponential in $|\tau_0|$, and such that any normal form formula satisfied in $\mathfrak{A}$ is also satisfied in the resulting structure.

For a fixed class $\mathfrak{B}$, apply Proposition 4 to $\mathfrak{A}$ and $\mathfrak{B} \subseteq \mathfrak{A}$ to obtain $\mathfrak{A}'$ in which $\mathfrak{B}$ has been replaced by an exponential size $\mathfrak{B}'$ in such a way that in particular (cf. (iii) in the proposition)

$$\boldsymbol{\beta}[B'] = \boldsymbol{\beta}[B] \quad \text{and} \quad \boldsymbol{\beta}[B', A \backslash B'] = \boldsymbol{\beta}[B, A \backslash B].$$

$\boldsymbol{\beta}[B'] = \boldsymbol{\beta}[B] \subseteq \boldsymbol{\beta}^+$ and $\boldsymbol{\beta}[B', A \backslash B'] = \boldsymbol{\beta}[B, A \backslash B] \subseteq \boldsymbol{\beta}^-$ imply that $\mathfrak{B}' \subseteq \mathfrak{A}'$ also forms an equivalence class, and that $\mathfrak{A}' \in \mathcal{EQ}[\tau_0, E]$.

Let $\mathfrak{A}$ be a countable $\mathcal{EQ}[\tau_0, E]$-structure, with infinitely many equivalence classes enumerated as $(B_i)_{i \in \mathbb{N}}$ without repetition. We may apply the above process inductively to E-classes B_i, one at a time, to obtain a sequence of structures $\mathfrak{A}_n \in \mathcal{EQ}[\tau_0, E]$, starting form $\mathfrak{A}_0 = \mathfrak{A}$, such that for all $n \in \mathbb{N}$:

(i) the universe of $\mathfrak{A}_n$ consists of the disjoint union of old and new equivalence classes of the form $\bigcup_{i<n} B'_i \cup \bigcup_{i \geqslant n} B_i$, where the new B'_i are exponentially size bounded in $|\tau_0|$;

(ii) $\mathfrak{A}_n{\restriction}\bigcup_{i \geqslant n} B_i \simeq \mathfrak{A}{\restriction}\bigcup_{i \geqslant n} B_i$ and $\mathfrak{A}_n{\restriction}\bigcup_{i<n} B'_i = \mathfrak{A}_{n+1}{\restriction}\bigcup_{i<n} B'_i$;

(iii) $\boldsymbol{\alpha}[\mathfrak{A}_{n+1}] = \boldsymbol{\alpha}[\mathfrak{A}_n]$ and $\boldsymbol{\beta}[\mathfrak{A}_{n+1}] = \boldsymbol{\beta}[\mathfrak{A}_n]$;

(iv) for every $b' \in B'_n$ there is some $b \in B_n$ such that $\boldsymbol{\beta}[b', \mathfrak{A}_{n+1}] \supseteq \boldsymbol{\beta}[b, \mathfrak{A}_n]$;

(v) $\boldsymbol{\beta}[b, B'_n] \supseteq \boldsymbol{\beta}[b, B_n]$ for all $b \in \bigcup_{i<n} B'_i$.[2]

Let $\mathfrak{A}'$ be the natural limit structure induced by this sequence on $A' := \bigcup_{n \in \mathbb{N}} B'_i$. It is easy to see that $\mathfrak{A}'$ is an $\mathcal{EQ}[\tau_0, E]$-structure with equivalence classes B'_i, which are exponentially size bounded in $|\tau_0|$. Moreover, due to construction,

$$\mathfrak{A} \models \varphi \implies \mathfrak{A}_n \models \varphi \text{ for all } n \in \mathbb{N} \implies \mathfrak{A}' \models \varphi$$

for all normal form φ.

The first implication follows inductively, essentially from (iii), (iv) and (v), with the same argument as before. For the second implication, concerning the limit, note that $\boldsymbol{\alpha}[\mathfrak{A}'] = \boldsymbol{\alpha}[\mathfrak{A}]$ and $\boldsymbol{\beta}[\mathfrak{A}'] = \boldsymbol{\beta}[\mathfrak{A}]$ directly follow from the corresponding equalities at all intermediate $\mathfrak{A}_n$, because all 1- and 2-types are eventually static by (ii). For $\forall\exists$ requirements in normal form φ, (v) guarantees that witnesses for such requirements provided for some $b \in B_i$ in B_j for some $j > i$ will be reproduced in B'_j and then persist in the limit by (ii).

We have thus proved Lemma 9 in the more interesting case in which we start with a model that has infinitely many equivalence classes; the argument for the case of finitely many equivalence classes is analogous but simpler as we do not need to go to a limit.

[2]Note that the first set of 2-types may (by (ii)) be evaluated in any $\mathfrak{A}_m$ for $m > n$ while the second is evaluated in $\mathfrak{A}_n$.

4.1.2 Few classes

This section completes the proof of the exponential model property of Proposition 7. In view of Lemma 9 it remains to bound the number of equivalence classes in models with small equivalence classes. Fix a model $\mathfrak{A} \in \mathcal{EQ}[\tau_0; E]$, whose equivalence classes are of exponential size in $|\tau_0|$. There may still be doubly exponentially many non-isomorphic classes (distinguished even by their composition in terms of 1-types $\boldsymbol{\alpha}[B]$).

A special role is played by those classes which contain all realizations of some 1-type (a simple $\forall\forall$-sentence can force all realisations of 1-type $\alpha(x)$ to be connected by E).

Definition 10 An equivalence class B in $\mathfrak{A}$ is called *singular* if it contains all realisations of some 1-type $\alpha \in \boldsymbol{\alpha}[\mathfrak{A}]$. A 1-type $\alpha \in \boldsymbol{\alpha}[\mathfrak{A}]$ *singular* if it is realised in only singular classes.

Note that realisations of a singular type $\alpha(x)$ may appear in more than one singular class. The following Lemma is immediate.

Lemma 11 *For every non-singular $\alpha \in \boldsymbol{\alpha}[\mathfrak{A}]$:*

(i) *there exists at least one non-singular class B containing a realisation of α.*

(ii) *for every $\alpha' \in \boldsymbol{\alpha}[\mathfrak{A}]$ (singular or non-singular) there exists $\beta \in \boldsymbol{\beta}^{-}[\mathfrak{A}]$ with $\beta{\restriction}x = \alpha$ and $\beta{\restriction}y = \alpha'$.*

As there is an obvious injection from the set of all singular classes of $\mathfrak{A}$ into $\boldsymbol{\alpha}[\mathfrak{A}]$, we have the following bound.

Lemma 12 *The number of singular classes in any given $\mathfrak{A} \in \mathcal{EQ}[\tau_0; E]$ is bounded by $|\boldsymbol{\alpha}[\mathfrak{A}]| \leqslant |\boldsymbol{\alpha}|$, exponential in the size of the vocabulary.*

Towards Proposition 7 we still need to bound also the number of non-singular equivalence classes.

Lemma 13 *For any $\mathfrak{A} \in \mathcal{EQ}[\tau_0; E]$ there is some $\mathfrak{A}' \in \mathcal{EQ}[\tau_0; E]$ satisfying exactly the same normal form sentences and consisting of isomorphic copies of equivalence classes of $\mathfrak{A}$, such that $\mathfrak{A}'$ has the same singular classes as $\mathfrak{A}$ and an exponentially bounded number of non-singular classes. Overall therefore, the number of classes in $\mathfrak{A}'$ is exponentially bounded in the size of the vocabulary.*

Proof Let $\mathfrak{A}_s \subseteq \mathfrak{A}$ be the substructure formed by the union of the singular classes. Note that A_s is exponentially bounded. Let C be a union of equivalence classes of $\mathfrak{A}$ such that $A_s \subseteq C$ and for all $a \in A_s$ and $\beta \in \boldsymbol{\beta}^{-}[a, A]$ we have $\beta \in \boldsymbol{\beta}^{-}[a, C]$. As A_s and $\boldsymbol{\beta}^{-}[\mathfrak{A}]$ as well as all classes of $\mathfrak{A}$ are exponential, C can be chosen of exponential size. Let $\mathfrak{C} \subseteq \mathfrak{A}$ be the corresponding substructure of $\mathfrak{A}$, a union of classes and containing in particular all singular classes.

Let $\mathfrak{B} \subseteq \mathfrak{A}$ be a union of non-singular equivalence classes such that every non-singular α is realised by some b_α in $\mathfrak{B}$. Again $\mathfrak{B}$ can be chosen of exponential size. We fix such $\mathfrak{B}$ together with some choice of such $b_\alpha \in \mathfrak{B}$ for all non-singular α.

Observe that $\boldsymbol{\alpha}[\mathfrak{A}] = \boldsymbol{\alpha}[\mathfrak{C}] \cup \boldsymbol{\alpha}[\mathfrak{B}]$.

We construct $\mathfrak{A}'$ from the disjoint union of $\mathfrak{C}$ and an exponential number of isomorphic copies of $\mathfrak{B}$ in such a way that

(i) $\boldsymbol{\beta}[\mathfrak{A}'] \subseteq \boldsymbol{\beta}[\mathfrak{A}]$.

(ii) for each $c \in \mathfrak{C} \subseteq \mathfrak{A}'$, $\boldsymbol{\beta}[c, \mathfrak{A}'] \supseteq \boldsymbol{\beta}[c, \mathfrak{A}]$.

(iii) for each isomorphic copy $\mathfrak{B}' \subseteq \mathfrak{A}'$ of $\mathfrak{B}$, and for each $b \in \mathfrak{B}$, if $b' \in \mathfrak{B}'$ is the isomorphic image of b, then $\boldsymbol{\beta}[b', \mathfrak{A}'] \supseteq \boldsymbol{\beta}[b, \mathfrak{A}]$.

It follows that $\mathfrak{A}'$ satisfies any normal form sentence satisfied in $\mathfrak{A}$.

Let $\boldsymbol{\beta}^-[\mathfrak{A}] = \{\beta_1, \ldots, \beta_m\}$. We build $\mathfrak{A}'$ from

$$\mathfrak{C} \cup \{0, 1, 2\} \times \{1, \ldots, m\} \times \mathfrak{B}$$

by allocating 2-types $\beta \in \boldsymbol{\beta}^-[\mathfrak{A}]$ between any two elements from different parts. Again we proceed in several stages.

(1) For $b \in \mathfrak{B}$ and $\beta \in \boldsymbol{\beta}^-[b, A]$ such that $\alpha' = \beta \restriction y$ is singular, put $\mathrm{tp}_{\mathfrak{A}'}((i, k, b), c) := \beta$ for all $c \in A_s$ with $\mathrm{tp}_{\mathfrak{A}}(b, c) = \beta$ (there are such, as α' is only realised in A_s in $\mathfrak{A}$).

(2) For $c \in C \setminus A_s$ and $\beta_k \in \boldsymbol{\beta}^-[c, A]$, if $\beta_k \notin \boldsymbol{\beta}^-[c, C]$ then $\alpha = \beta_k \restriction y$ must be non-singular. Put $\mathrm{tp}_{\mathfrak{A}'}(c, (0, k, b_\alpha)) := \beta_k$. This settles (ii).

(3) For $b \in B$ and $\beta_k \in \boldsymbol{\beta}^-[b, A]$, if $\beta_k \notin \boldsymbol{\beta}^-[b, A_s]$ then $\alpha = \beta \restriction y$ must be non-singular. Put $\mathrm{tp}_{\mathfrak{A}'}((i, j, b), (i', k, b_\alpha)) := \beta_k$ for $i' = (i + 1) \bmod 3$. This settles (iii).

(4) For all remaining pairs of undeclared 2-type find $\beta \in \boldsymbol{\beta}[\mathfrak{A}]$ compatible with the given 1-types. This is possible by part (ii) of Lemma 11, since all such pairs involve at least one non-singular 1-type. ◻

Lemmas 9 and 13 together imply the exponential model property of Proposition 7.

5 Two equivalence relations

In this section we consider models in $\mathcal{EQ}[\tau_0; E_1, E_2]$, with $\tau_{\mathrm{eq}} = \{E_1, E_2\}$. Structures in $\mathcal{EQ}[\tau_0; E_1, E_2]$ are of the form $\mathfrak{A} = (A, E_1^{\mathfrak{A}}, E_2^{\mathfrak{A}}, \ldots)$, where $E_1^{\mathfrak{A}}, E_2^{\mathfrak{A}}$ are equivalence relations over $\mathfrak{A}$. If B is an equivalence class with respect to $E_1^{\mathfrak{A}}$ or $E_2^{\mathfrak{A}}$, we also refer to the substructure $\mathfrak{B} = \mathfrak{A} \restriction B$ as an equivalence E_1-class (E_2-class) of $\mathfrak{A}$; we call equivalence classes of $E_1^{\mathfrak{A}} \cap E_2^{\mathfrak{A}}$ as well as their induced substructures *intersections*. We say that intersections J, J' are in *a free position* if they are not connected by E_1 or E_2.

We will mostly work on the level of intersections, rather than individual elements. Correspondingly, we define notions of types of intersections and types of ordered pairs of intersections; they will play similar roles to 1-types and 2-types of individual elements.

The *type of an intersection* consists of the specification of the isomorphism type of the substructure induced by this intersection.

The *type of a pair of intersections* (J_1, J_2) consists of the full specification of the isomorphism type of the substructure induced by the union $J_1 \cup J_2$ of these two intersections and the identification of the parts, J_1 and J_2. In other words, the type of the (ordered) pair (J_1, J_2) adds to the information about the types of the two individual intersections the information about the 2-types of all pairs (a_1, a_2) for $a_i \in J_i$.

Note that the entire structure is fully specified through a listing of its intersections and the specification of the types of all pairs of these.

We start this section with a few examples showing how to enforce some interesting properties of models. In particular, we construct formulae satisfiable only in infinite models, and enforcing even infinite equivalence classes. Then we argue that, in contrast, infinite or even large intersections cannot be enforced: every satisfiable formula has a model with small (i.e., exponentially bounded) intersections.

Our decidability proof for the case of two equivalence relations is fairly involved. We therefore find it convenient to introduce an auxiliary combinatorial problem, which we call the *coloured castles problem*. We reduce satisfiability of FO^2 over $\mathcal{EQ}[\tau_0; E_1, E_2]$ to the new problem and show this new combinatorial problem to be decidable. This strategy should help to isolate the combinatorial core of the satisfiability issue more clearly.

To make the presentation even more readable we introduce two versions of the coloured castles problem: *simplified* and *regular*. To the simplified version we may reduce satisfiability of those normal form formulas which require no free witnesses. This simpler version allows the main ideas of the whole construction to be presented in a more conspicuous manner.

5.1 Examples

5.1.1 Failure of the finite model property

We show that even over $\mathcal{EQ}[\tau_0; E_1, E_2]$ for $\tau_0 = \{P, Q, S\}$ with unary P, Q, S, FO^2 does not have the finite model property. We exhibit an infinity axiom that enforces an infinite number of E_1-classes and of E_2-classes. Let $\lambda \in \mathrm{FO}^2[\tau_0; E_1, E_2]$ say that

(1) P and Q are disjoint and each E_2-class contains at most one element from P and one from Q; analogously for E_1-classes. The E_2-class of any element of S is trivial (a singleton).

(2) every element of P is E_1-equivalent to one in Q; every element of Q is E_2-equivalent to one in P.

(3) $S \cap P \neq \emptyset$.

It is easy to formalise these in FO^2 over $\mathcal{EQ}[P, Q, S; E_1, E_2]$, and in fact even in the guarded fragment[3] with two variables. Fig. 1 (a) shows a model of this sentence λ.

Conversely, any model of λ embeds such an infinite chain. Starting from an element of $S \cap P$, one finds new elements in Q and P along E_1- and E_2-links in an alternating fashion by appeal to condition (2); these indeed always have to be new elements, i.e., distinct from previous elements in the chain because of (1).

The straightforward formalisation of (1) requires equality. Use of equality, however, is not necessary to obtain an infinity axiom. Let us construct λ' by modifying (1)

[3] In the *guarded fragment* all quantifiers are relativised by atoms, as in $\forall \mathbf{x}(R\mathbf{x} \implies \ldots)$ and $\exists \mathbf{x}(R\mathbf{x} \wedge \ldots)$.

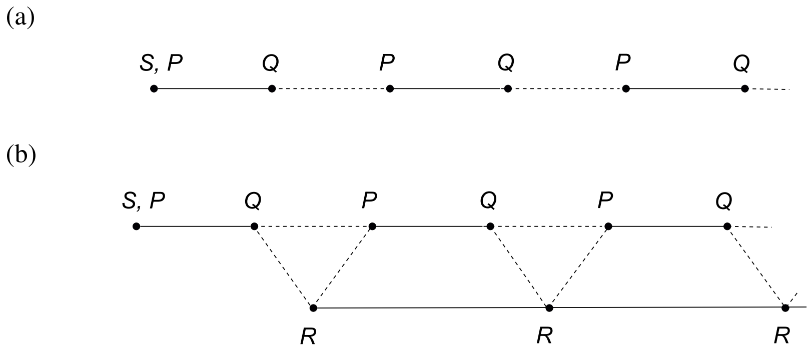

Figure 1: (a) A model of λ and λ'. (b) A model of λ". Solid segments represent E_1-connections, broken segments represent E_2-connections.

to

(1') P and Q are disjoint and S and Q are disjoint and not related by E_2-links;

and adding

(4) every pair of elements both of which belong to P or to Q is either connected by both, E_1 and E_2, or by neither.

Of course, the structure from Fig. 1 (a) is also a model of λ'. And again, every model has to embed this infinite chain. An attempt to reuse an earlier element leads (by an inductive argument) to an E_2-link between an element from S and an element from Q.

A simple modification λ'' of λ or λ' even enforces the existence of an infinite E_1-class. For a new unary predicate R, we can say that

(5) each E_2-class of an element from Q contains an element from R and all of R is contained in a single E_1-class.

One checks that this, together with λ or λ', forces the chain to have links to an infinite set of distinct elements in R, all contained in an infinite E_1-class as depicted in Fig. 1 (b).

5.1.2 Large finite classes

The tree-like structure in Fig. 2 contains an E_2-class with exactly 2^{k-1} elements satisfying S along its leaves. L and S are unary predicates of τ_0. The predicates P_i serve to number consecutive levels of the tree-like structure; they will be definable in terms of other basic predicates such that the number of levels k can exceed the size of $|\tau_0|$.

In the first instance, we describe how to axiomatise the given structure for $k = 2^n$ by a formula κ whose length is linear in n. For this we code the values i for the levels P_i in binary by means of n extra unary predicates $Q_0, \ldots, Q_{n-1}$. Let κ say that

(1) there is exactly one element in P_0.

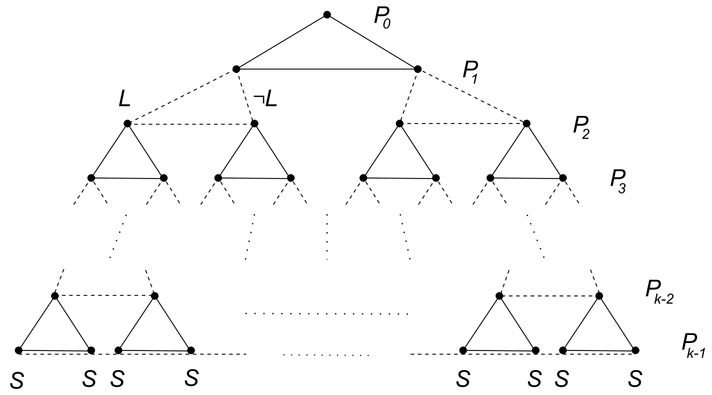

Figure 2: A large finite class. Solid segments represent E_1-connections, broken segments represent E_2-connections.

(2) (top-down requirements) every $a \in P_{2i}$, $2i < k-1$, has E_1-links to at least two elements of P_{2i+1}, one in L and one not in L; similarly for $a \in P_{2i+1}$ and E_2-links to two successors in P_{2i+2}.

(3) distinct elements in P_{2i} are not E_1-connected; distinct elements in P_{2i+1} may be E_2-connected only if $2i+1 = k-1$; this ensures that elements fulfilling the requirements from (2) are not reused.

(4) (bottom-up requirements) dually to (2), every $a \in P_{i+1}$ is appropriately linked to some element on level P_i.

(5) distinct elements from every P_{2i} may be E_2-connected only if exactly one of them is in L; similarly, distinct elements from every P_{2i+1} may be E_1-connected only if exactly one of them is in L.

(6) an element satisfies S iff it is in P_{k-1}; all of S is contained in a single E_2-class.

The construction may be extended to cover $k = 2^{2^n}$. For this consider the black bullets from Fig. 2 as intersections, each intersection consisting of 2^n elements numbered by means of another set of unary predicates $U_1, \ldots, U_{n-1}$. The number encoded by these predicates in an element is called *U-value*. Values of P_i are encoded (in binary) by another unary predicate V (we call them *V-values*). An element whose U-value is ℓ is put into V if, and only if, the ℓ-th bit of i is 1. It is then possible to express, in a linear size FO^2 formula, the condition that a pair of intersections within the same E_i-class encodes a pair of consecutive values in $\{0, \ldots, 2^{2^n} - 1\}$.

We give some details. We say that an intersection is *even* (*odd*) if its V-value is even (odd). Note that to check if an intersection is even it is enough to see if the element whose U-value is $2^n - 1$ satisfies $\neg V$. We say that an intersection is *maximal* if its V-value is $2^{2^n} - 1$. Testing if an intersection is maximal can be done by checking if all its elements satisfy V. Now, we are ready to describe the formula:

(1) each intersection has exactly 2^n elements, whose U-values are $0, 1, \ldots, 2^n - 1$;

(2) all elements of an intersection agree on the predicate L;

(3) there exists exactly one intersection with the V-value 0 (all elements of the intersection satisfy $\neg V$);

(4) (top-down requirements) every element belonging to an even intersection is E_1-connected and not E_2-connected to at least two elements in some odd intersections, one in L, the other not in L; every element belonging to a non-maximal odd intersection is E_2-connected and not E_1-connected to at least two elements in some even intersections, one in L, the other not in L;

(5) if an even (odd) intersection J_1 is E_1-linked (E_2-linked) to an odd (even) intersection J_2, then the V-value encoded in J_2 is 1 plus that encoded in J_1;[4]

(6) distinct elements in P_{2i} of U-value 0 are not E_1-connected; distinct elements in P_{2i+1} of U-value 0 may be E_2-connected only if $2i+1 = 2^{2^n} - 1$;

(7) (bottom-up requirements) dually to (4), every element from P_i, $i > 0$ is appropriately linked to an element in P_{i-1};

(8) distinct elements from P_{2i} of U-value 0 may be E_2-connected only if exactly one of them is in L; similarly, distinct elements from P_{2i+1} of U-value 0 may be E_1-connected only if exactly one of them is in L (this ensures that intersections can have at most two successors);

(9) an element satisfies S iff its U-value is 0 and it is in $P_{2^{2^n}-1}$; all of S is contained in a single E_2-class.

It is worth mentioning that a similar construction may be used to encode computations of an exponential space alternating Turing machine. This yields a 2EXPTIME-lower bound for the satisfiability problem for FO^2 (and in fact even for the two-variable guarded fragment without equality) over $\mathcal{EQ}[\tau_0; E_1, E_2]$. This result is presented in [20].

5.1.3 Exactly two realisations of 1-types

In plain FO^2 we are able to enforce that some 1-type has to be realised exactly once in a structure (following [12] realisations of such types are called *kings*). In the previous example we enforced exactly 2^k realisations of some 1-type in a model, for some k. However, a king played a very important role in this construction, viz., the root element in P_0. In contrast, we now exhibit a simple satisfiable formula, whose models do not have kings but *exactly two* realisations of every 1-type. In fact, the following formula defines the structure depicted in Fig. 3 up to isomorphism. It says that

(1) P and Q form a partition of the universe and $P \times Q \subseteq E_1 \cup E_2$,

(2) there are at least two E_1-classes,

(3) P and Q each intersect every E_1-class and every E_2-class in precisely one point.

[4] This formula requires a high reuse of variables, but is rather standard: it says that the least significant (with respect to U-values) position at which J_1 encodes 0 (by the predicate V), is the least significant position at which J_2 encodes 1, and that both encodings agree on more significant positions.

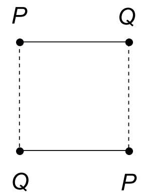

Figure 3: Two realisations of a type.

5.2 Small intersections

Let $\mathfrak{A} \in \mathcal{EQ}[\tau_0; E_1, E_2]$ be an equivalence structure. From the previous sections we know that, in contrast to the case of $\mathcal{EQ}[\tau_0; E]$, if we want to preserve (normal form) FO^2 sentences it is not always possible to replace $\mathfrak{A}$ by a structure with finite classes. However, we can transform $\mathfrak{A}$ into a structure with small intersections.

Lemma 14 *Every normal form φ that is satisfiable in an $\mathcal{EQ}[\tau_0, E_1, E_2]$ model has an $\mathcal{EQ}[\tau_0, E_1, E_2]$ model with at most countably many intersections each of which is bounded in size by an exponential function in $|\tau_0|$.*

Proof We first show how to replace a single intersection $\mathfrak{J}$ in $\mathfrak{A}$ by a new substructure $\mathfrak{J}'$ of size exponential in $|\tau_0|$. This is done in such a way that the E_i-links to the full E_i-classes of $\mathfrak{J}$ are reproduced by $\mathfrak{J}'$, which will then again be the intersection of these classes. We preserve any normal form formula in this modification.

Extend τ_0 by new unary symbols U_1, U_2 for the E_1- and E_2-classes of $\mathfrak{J}$, and expand $\mathfrak{A}$ accordingly. In other words put $U_i^{\mathfrak{A}} := B_i$ where B_i is the E_i-class of $\mathfrak{J}$ in $\mathfrak{A}$. We now apply Proposition 4 to this expansion $\hat{\mathfrak{A}} = (\mathfrak{A}, U_1^{\mathfrak{A}}, U_2^{\mathfrak{A}})$ and to the induced substructure $\hat{\mathfrak{J}} = \hat{\mathfrak{A}} \restriction J$, which is the trivial expansion of $\mathfrak{J}$ with $U_i^{\mathfrak{J}} = J$. By Proposition 4 we obtain a new structure $\hat{\mathfrak{A}}' = (\mathfrak{A}', U_1^{\mathfrak{A}'}, U_2^{\mathfrak{A}'})$ by replacing $\hat{\mathfrak{J}}$ in $\hat{\mathfrak{A}}$ by an exponential size $\hat{\mathfrak{J}}'$ in such a way that in particular (cf. (iii) in the proposition) $\boldsymbol{\beta}[J'] = \boldsymbol{\beta}[J]$ and $\boldsymbol{\beta}[J', A \setminus J'] = \boldsymbol{\beta}[J, A \setminus J]$.

Using the U_i, we see that this implies $\boldsymbol{\beta}[J'] \subseteq \boldsymbol{\beta}^{++}$, $\boldsymbol{\beta}[J', B_1 \setminus J'] \subseteq \boldsymbol{\beta}^{+-}$, $\boldsymbol{\beta}[J', B_2 \setminus J'] \subseteq \boldsymbol{\beta}^{-+}$, and $\boldsymbol{\beta}[J', A \setminus (B_1 \cup B_2)] \subseteq \boldsymbol{\beta}^{--}$. This guarantees that $\mathfrak{A}'$, which is obtained from $\hat{\mathfrak{A}}'$ just by dropping the interpretations of U_1 and U_2, is an $\mathcal{EQ}[\tau_0; E_1, E_2]$-structure, and that $\mathfrak{J}'$ is an intersection of $\mathfrak{A}'$.

We may apply the above process to all intersections of a countable $\mathfrak{A} \models \varphi$ in $\mathcal{EQ}[\tau_0, E_1, E_2]$ as in the construction of a model with small classes from Section 4.1.1. This finishes the proof of the lemma. $\square$

When we look for models for a formula φ, we may now restrict attention to structures whose intersections have a bounded size. From this point onward we consider only such structures.

5.3 Coloured castles problems

In our auxiliary problem we look for arrangements of coloured castles on the infinite chessboard, satisfying some constraints. In the simplified version of the problem the constraints are put on the multisets of colours of the castles in a single row or in a single column, and on the pairs of colours of castles in skew positions. Two chessboard positions are called *skew* if they are neither part of the same column nor of the same row.

The intuitive connection between chessboard setting and $\mathcal{EQ}[\tau_0, E_1, E_2]$ structures, which is to be made more precise in Section 5.3.1 below, is the following: rows and columns on the chessboard correspond to E_1- and E_2-equivalence classes so that chessboard positions are potential intersections; placement of a castle indicates that the corresponding intersection is non-empty, the colour of the castle determines the intersection type; colour multiplicities in rows and columns determine the $\boldsymbol{\beta}^{-+}$- and $\boldsymbol{\beta}^{+-}$-types; an additional stipulation of coloured arrows between castles in skew positions of the chessboard similarly takes care of $\boldsymbol{\beta}^{--}$-types.

We use the following natural terminology concerning multiplicities: the multiplicity of a colour $\delta \in \Delta$ in a Δ-coloured set B is the cardinality of the set $\{b \in B\colon\ b \text{ has colour } \delta\ \}$; we associate a multiplicity function $\theta_B\colon \Delta \to \mathbb{N} \cup \{\infty\}$ with these multiplicities and say that B realises some given multiplicity function $\theta\colon \Delta \to \mathbb{N} \cup \{\infty\}$ if $\theta_B = \theta$. In particular, for a partially Δ-coloured chessboard, we say that a row or column realises the multiplicity function θ if the set of Δ-coloured castles placed in this row or column realises θ.

Definition 15 Let Δ and E be two disjoint sets of colours called the set of *castle colours* and the set of *arrow colours*, respectively.

(i) A *multiplicity function* (*over* Δ) is a function of type $\theta\colon \Delta \to \mathbb{N} \cup \{\infty\}$; such θ is called *finite* if $\theta(\delta) \in \mathbb{N}$ for all $\delta \in \Delta$.

(ii) We say that a multiplicity function θ' *safely extends* a multiplicity function θ if for all $\delta \in \Delta$

- if $\theta(\delta) < 2$ then $\theta'(\delta) = \theta(\delta)$,
- if $\theta(\delta) \geq 2$ then $\theta'(\delta) \geq \theta(\delta)$.

(iii) A *placement* (*over* Δ, E) π is a pair of functions (π^Δ, π^E),

$$\begin{array}{rcl} \pi^\Delta\colon \mathbb{N} \times \mathbb{N} & \longrightarrow & \Delta \cup \{\perp\}, \\ \pi^E\colon (\mathbb{N} \times \mathbb{N}) \times (\mathbb{N} \times \mathbb{N}) & \longrightarrow & E \cup \{\perp\}, \end{array}$$

such that for $a, b \in \mathbb{N} \times \mathbb{N}$ if $\pi^E(a,b) \in E$ then $\pi^\Delta(a), \pi^\Delta(b) \in \Delta$, and if (i,j), (i',j') are such that $i = i'$ or $j = j'$ then $\pi^E((i,j),(i',j')) = \perp$.
If $\pi^E(a,b) = \perp$ for all (a,b), then we identify π with just π^Δ and call this placement *simple*.

We visualise a placement as an arrangement of Δ-coloured castles on the infinite $\mathbb{N} \times \mathbb{N}$ chessboard, with some castles linked by E-coloured arrows. $\pi^\Delta(i,j) = \delta \in \Delta$ corresponds to "putting a castle of colour δ" on field (i,j); $\pi^\Delta(i,j) = \perp$ indicates that there is no castle in field (i,j); we correspondingly refer to $a = (i,j)$ with $\pi^\Delta(i,j) \neq$

$\perp$ as a *castle* a. If $\pi^E(a,b) = \varepsilon \neq \perp$ for castles a, b, then we say that the castle a *sends an arrow of colour* ε *to* b. Note that a may send an arrow to b and, simultaneously, b may send another arrow, of possibly different colour, to a.

5.3.1 Correspondence

There is a natural correspondence between placements and (partially defined) structures with two equivalence relations.

Let $\tau = (\tau_0, E_1, E_2)$ be a signature. Let Δ be the set of possible types of intersections over τ, E the set of possible types of pairs of intersections in a free position. Consider the following translations between τ-structures and chessboard placements.

Structures to placements. From a given countable τ-structure $\mathfrak{A}$ we abstract a placement over (Δ, E). Every intersection J in $\mathfrak{A}$ corresponds to a castle, which is coloured by the type of J. Two castles are placed in the same row (column) if, and only if, they belong to the same E_1-class (E_2-class). For castles a, b in a skew position, $\pi^E(a,b)$ is set to the type of the corresponding pair of intersections. We remark that this translation does not specify a unique target placement, but that will also not be necessary.

Placements to structures. Let $\pi = (\pi^\Delta, \pi^E)$ be a placement satisfying the following conditions: for every a, b if $\pi^E(a,b) = \varepsilon \neq \perp$ then the type ε is consistent with the isomorphism types of intersections $\pi^\Delta(a), \pi^\Delta(b)$, i.e., there is a τ-structure built from two intersections J_1, J_2 in a free relation such that J_1 has type $\pi^\Delta(a)$, J_2 has type $\pi^\Delta(b)$ and the pair (J_1, J_2) has the type $\pi^E(a,b)$; moreover, if additionally $\pi^E(b,a) \neq \perp$, then the types of pairs of intersections $\pi^E(a,b)$ and $\pi^E(b,a)$ are also consistent with each other, i.e., there is a structure as above in which additionally the pair (J_2, J_1) has the type $\pi^E(b,a)$.

Under these conditions π can be transformed into a partially defined structure in the following way. Every castle a corresponds to an intersection of type $\pi^\Delta(a)$. Two intersections are made E_1-equivalent (E_2-equivalent) if their castles come from the same row (column). If $\pi^\Delta(a,b) = \varepsilon \neq \perp$ then the type of the pair of intersections corresponding to a, b is made to be ε. The connections within E_i-classes and between the remaining intersections in free positions remain unspecified.

5.3.2 The coloured castles problem

We give the formal definition of two variants of the coloured castles problem.

Coloured castles (simplified version). An instance of the simple version of the coloured castles problem is of the form $(\Delta, T_{\text{row}}, T_{\text{col}}, \text{Frb})$ where

- Δ is a finite set of colours for castles.
- T_{row} and T_{col} are finite collections of finite multiplicity functions over Δ.
- $\text{Frb} \subseteq \Delta \times \Delta$ is a set of *forbidden* pairs of colours.

The problem is to decide whether there exists some simple placement $\pi = \pi^\Delta$ with at least one castle such that

- every row (column) realises a (not necessarily finite) multiplicity function θ safely extending one of the functions from T_{row} (T_{col});
- if castles a, b are in a skew position then $(\pi^{\Delta}(a), \pi^{\Delta}(b)) \notin \text{Frb}$.

Coloured castles (regular version). An instance of the regular version is of the form

$$(\Delta, T_{\text{row}}, T_{\text{col}}, \text{Frb}, E, \text{Inv}^E, \text{Ends}^E, \text{SReq})$$

where $\Delta, T_{\text{row}}, T_{\text{col}}, \text{Frb}$ are as in the simple version and

- E is a finite set of colours for arrows;
- $\text{Inv}^E : E \to E$ is an involution, i.e., a function such that $\text{Inv}^E \circ \text{Inv}^E = \text{id}$;
- Ends^E is a function $\text{Ends}^E : E \to \Delta \times \Delta$;
- SReq is a function which for every colour $\delta \in \Delta$ returns a (possibly empty) finite collection of *skew requirements* $\text{SReq}(\delta) = \{S_1, \ldots, S_k\}$ where every S_i is a subset of E.

The problem is to decide whether there exists a placement $\pi = (\pi^{\Delta}, \pi^{E})$ with at least one castle such that:

- all requirements for a solution for the simple version are fulfilled;
- if $\pi^E(a, b), \pi^E(b, a) \neq \perp$ then $\pi^E(a, b) = \text{Inv}^E(\pi^E(b, a))$;
- if $\pi^E(a, b) = \varepsilon \neq \perp$ then $\text{Ends}^E(\varepsilon) = (\pi^{\Delta}(a), \pi^{\Delta}(b))$.
- the skew requirements are satisfied for every castle a, i.e., for every a there is some $S \in \text{SReq}(\pi^{\Delta}(a))$ such that a sends an arrow of colour ε to some castle a', for every $\varepsilon \in S$; note that $\text{SReq}(\delta) = \emptyset$ means that the skew-requirements of δ are unsatisfiable, and that if $\emptyset \in \text{SReq}(\delta)$ then δ has no skew requirements.

5.4 From satisfiability to the coloured castles problem

Let φ be in $\mathcal{EQ}[\tau_0, E_1, E_2]$-Scott normal form. For $s \in \{+, -\}^2$, we denote by φ^s the formula obtained from φ by removing all $\forall\exists$ requirements except those of type s. Note that, if $\mathfrak{A} \models \varphi$, then $\mathfrak{A}{\restriction}J \models \varphi^{++}$ for every intersection J; $\mathfrak{A}{\restriction}C \models \varphi^{+-} \wedge \varphi^{++}$ for every E_1-class C; and similarly $\mathfrak{A}{\restriction}C \models \varphi^{-+} \wedge \varphi^{++}$ for every E_2-class C.

We construct an input $(\Delta, T_{\text{row}}, T_{\text{col}}, \text{Frb}, E, \text{Inv}^E, \text{Ends}^E, \text{SReq})$ to the coloured castles problem, which will have a solution if, and only if, φ is satisfiable. If φ^{--} is empty we will have $E = \emptyset$, $\text{SReq}(\delta) = \{\emptyset\}$ for every δ, and thus the associated instance may be regarded as an input to the simple version of the problem.

We define Δ to be the set of those isomorphism types of intersections which, from the local point of view, may be used in a model of φ: formally $\delta \in \Delta$ if intersections of type δ satisfy φ^{++}. Recall that we consider only types of intersections of size exponential in the size of the signature, so the number of possible types is bounded doubly exponentially in $|\varphi|$.

As shown by one of the examples, we sometimes require infinite classes in models. From a local point of view, however, finite classes are sufficient in the sense of the following lemma. The lemma says that every equivalence class can be approximated in a sense by a class of a bounded size.

Lemma 16 *Let $\mathfrak{C} \models \varphi^{+-} \wedge \varphi^{++} \wedge \forall xy E_1 xy$. Then there is a finite structure $\mathfrak{C}' \models \varphi^{+-} \wedge \varphi^{++} \wedge \forall xy E_1 xy$ such that*

(i) *$\mathfrak{C}'$ and $\mathfrak{C}$ realise the same types of intersections,*

(ii) *$\mathfrak{C}'$ and $\mathfrak{C}$ realise the same types of pairs of intersections,*

(iii) *for every type of an intersection δ the number of intersections of type δ in $\mathfrak{C}'$ is not greater then the number of intersections of type δ in $\mathfrak{C}$,*

(iv) *the overall number of intersections in $\mathfrak{C}'$ is bounded by $M = 3s^4$, where s is the number of types of pairs of intersections realised in $\mathfrak{C}$.*

We note that the bound in (iv) is doubly exponential in $|\tau_0|$. A symmetric claim obtains for E_2-classes and φ^{-+}.

Proof Consider a structure $\mathfrak{F}$ over a new language τ', whose universe is the set of intersections from $\mathfrak{C}$, 1-types of elements correspond to types of intersections and 2-types of pairs of elements correspond to types of pairs of the corresponding intersections. More precisely, $tp_{\mathfrak{F}}(J_1) = tp_{\mathfrak{F}}(J_2)$ if and only if types of intersections of J_1 and J_2 in $\mathfrak{C}$ are equal, and $tp_{\mathfrak{F}}(J_1, J_2) = tp_{\mathfrak{F}}(J_3, J_4)$ if and only if types of pairs of intersections (J_1, J_2) and (J_3, J_4) in $\mathfrak{C}$ are equal.

Let F_0 be the set of those elements in $\mathfrak{F}$ whose 1-types are realised more than $3s^3$ times (note that s is the number of 2-types in $\mathfrak{F}$). We apply Proposition 4 taking $\mathfrak{A} := \mathfrak{F}$ and $B := F_0$. We obtain a finite structure $\mathfrak{F}'$ in which every 1-type is realised at most $3s^3$ times. The structure $\mathfrak{F}'$ can be now transformed back in a natural manner into a τ-structure $\mathfrak{C}'$ with intersections corresponding to elements of $\mathfrak{F}$. The properties of $\mathfrak{F}'$ listed in Proposition 4 clearly guarantee that $\mathfrak{C}'$ is as required.

In the proof we apply our small substructure lemma rather than, e.g., the small model construction for FO^2 from [12] since the latter may increase the number of elements of some 1-type, which would violate (iii). $\square$

We are now ready to construct T_{row} and T_{col}. We enumerate all possible multisets of types of intersections of cardinality up to M, where M is as in Lemma 16. For each such multiset we nondeterministically check if it is possible to build a structure $\mathfrak{C}$ from intersections of the given types such that $\mathfrak{C} \models \varphi^{+-} \wedge \varphi^{++} \wedge \forall xy E_1 xy$ ($\mathfrak{C} \models \varphi^{-+} \wedge \varphi^{++} \wedge \forall xy E_2 xy$). If such $\mathfrak{C}$ exists then, clearly, every isomorphism type of an intersection from $\mathfrak{C}$ belongs to Δ. Let θ be the multiplicity function which for every $\delta \in \Delta$ returns the number of realisations of δ in $\mathfrak{C}$. We add θ to T_{row} (T_{col}).

To define Frb we consider all pairs $\delta, \delta' \in \Delta$. If there is no structure $\mathfrak{D}$ built from two intersections J of type δ and J' of type δ' in a free position such that $\mathfrak{D}$ satisfies the $\forall\forall$ constituents of φ, then we add (δ, δ') to Frb.

If φ^{--} is empty (φ requires no free witnesses) we set $E := \emptyset$, and $\mathrm{SReq}(\delta) := \{\emptyset\}$ for every $\delta \in \Delta$. In this case we have in fact constructed an instance of the simple version of the coloured castles problem.

If φ^{--} is not empty we proceed as follows.

For every $\delta, \delta' \in \Delta$ and every structure $\mathfrak{D}$ built from intersections J of type δ and J' of type δ' in a free position such that $\mathfrak{D}$ satisfies the $\forall\forall$ constituents of φ, let ε be the type of the pair (J, J') in $\mathfrak{D}$. We add ε to E, set $\mathrm{Inv}^E(\varepsilon)$ to be the type of the pair (J', J) and set $\mathrm{Ends}^E(\varepsilon) := (\delta, \delta')$. Note that Inv^E is an involution as required.

It remains do define SReq. For each $\delta \in \Delta$ and $k \leqslant N$, where N is the product of the maximal possible size of an intersection whose type is from Δ and the number of conjuncts of type $\forall\exists$ in φ^{--}, we repeat the following.

Generate all possible structures $\mathfrak{D}$ consisting of an intersection J_0 of type δ and intersections $J_1, \ldots, J_k$ of types from Δ with all pairs (J_0, J_i) in a free position. If all elements from J_0 have all the required free witnesses and the $\forall\forall$ constituent of φ is satisfied in each $\mathfrak{D}\restriction J_0 \cup J_i$, we add $\{\varepsilon_1, \ldots, \varepsilon_k\}$ to $\mathrm{SReq}(\delta)$, where ε_i is the type of the pair (J_0, J_i) in $\mathfrak{D}$.

Proposition 17 *Let φ be in $\mathcal{EQ}[\tau_0, E_1, E_2]$ normal form and let $\lambda = (\Delta, T_{\mathrm{row}}, T_{\mathrm{col}}, \mathrm{Frb}, E, \mathrm{Inv}^E, \mathrm{Ends}^E, \mathrm{SReq})$ be the input to the coloured castles problem as described above. Then λ has a solution if and only if φ is satisfiable in $\mathcal{EQ}[\tau_0, E_1, E_2]$.*

Proof ($\Leftarrow$) Let $\mathfrak{A} \models \varphi$ be a structure with exponentially bounded intersections. We obtain from $\mathfrak{A}$ a placement $\pi = (\pi^\Delta, \pi^E)$ as in Section 5.3.1. Let us check that it fulfils all the requirements of a solution.

Consider the multiplicity function θ of a row in π. Assume that this row corresponds to an E_1-class C in $\mathfrak{A}$. Let $\mathfrak{C}'$ be the structure whose existence is postulated by Lemma 16. Let θ' be the function returning for a given type δ the number of intersections of this type in $\mathfrak{C}'$. By the construction of the coloured castle instance we have $\theta' \in T_{\mathrm{row}}$. Observe that θ safely extends θ' since any intersection type which appears at most once in $\mathfrak{C}'$ cannot appear more times in $\mathfrak{C}$ (because $\mathfrak{C}$ and $\mathfrak{C}'$ have exactly the same types of pairs of intersections). Column multiplicities are treated analogously.

Consider now any pair of castles a of colour δ and a' of colour δ' in a skew position in π. Observe that intersections corresponding to a, a' are in a free position in $\mathfrak{A}$. Thus $(\delta, \delta') \notin \mathrm{Frb}$.

By the choice of functions Inv^E and Ends^E, the corresponding conditions on solutions are obviously satisfied.

Finally, if $E \neq \emptyset$, then consider any castle a of colour δ. Let J_0 be the intersection in $\mathfrak{A}$ corresponding to a. Let $J_1, \ldots, J_k$ be a minimal collection of intersections such that all elements from J find all of their required free witnesses in $\mathfrak{A}\restriction(J_0 \cup J_1 \cup \ldots \cup J_k)$. Let ε_i be the type of the pair (J_0, J_i). It is clear by the choice of SReq that $\{\varepsilon_1, \ldots, \varepsilon_k\} \in \mathrm{SReq}(\delta)$. Since a sends arrows of colours $\varepsilon_1, \ldots, \varepsilon_k$ to the castles corresponding to the intersections $J_1, \ldots, J_k$, the SReq-condition on solutions is also satisfied.

($\Rightarrow$) Let $\pi = (\pi^\Delta, \pi^E)$ be a solution to λ. Note that the construction of Inv^E and Ends^E allows us to build a partially defined structure $\mathfrak{A}$ as described in Section 5.3.1. We have still to define connections between some pairs of intersections of $\mathfrak{A}$.

Let J, J' be a pair of intersections in a free position such that the connection between them is not yet specified (this happens when $\pi^E(a, b) = \perp$ and $\pi^E(b, a) = \perp$ where a, b are castles corresponding to the intersections J, J'). Since $(\pi^\Delta(a), \pi^\Delta(b)) \notin \mathrm{Frb}$, there exists a type of a pair of intersections ε which may be given to (J, J') so as to be consistent with the $\forall\forall$ constituents of φ; we set the type of (J, J') to ε.

It remains to set the connection between intersections inside equivalence classes. Consider, for instance, an E_1-class C in $\mathfrak{A}$ corresponding to row i in π. Let θ be the multiplicity function of i and θ' a function in $T_{\rm row}$ which is safely extended by θ. We choose a set of castles in i consisting of exactly $\theta'(\delta)$ castles coloured by δ, for every $\delta \in \Delta$. By the definition of $T_{\rm row}$, we may define a structure on the intersections corresponding to the chosen castles and satisfying $\varphi^{++} \wedge \varphi^{+-} \wedge \forall xy E_1 xy$. Let us call this structure $\mathfrak{C}_0$. The remaining intersections from the class C can be joined to $\mathfrak{C}_0$ in the following way. Let J be an intersection in $C \setminus C_0$; its type is realised at least twice in $\mathfrak{C}_0$ (otherwise J would have to belong to $\mathfrak{C}_0$, because $\theta'(\delta) = 1$ implies that $\theta(\delta) = 1$ and that the only intersection of this type in C would thus be in $\mathfrak{C}_0$). Let J_1, J_2 be distinct realisations of δ in $\mathfrak{C}_0$. For any intersection J' in $\mathfrak{C}_0$ except J_1 we set the type of (J, J') to be equal to the type of (J_1, J'); the type of (J, J_1) to be equal to the type of (J_1, J_2). This ensures that the resulting class still satisfies $\varphi^{++} \wedge \varphi^{+-} \wedge \forall xy E_1 xy$.

The above steps guarantee that $\mathfrak{A} \models \varphi^{++} \wedge \varphi^{+-} \wedge \varphi^{-+}$. It remains to see that also $\mathfrak{A} \models \varphi^{--}$. Let $w \in A$ and let J be the intersection containing w. Let δ be the type of J, and let c be the castle in π corresponding to J. Let $S = \{\varepsilon_1, \ldots, \varepsilon_k\} \in \mathrm{SReq}(\delta)$ be the set whose existence in postulated by the conditions on the solution. Let $c_1, \ldots, c_k$ be such that c sends an arrow of colour ε_i to c_i. The construction of SRow ensures that all elements of J, and in particular w, have all the required free witnesses in the intersections corresponding to the castles $c_1, \ldots, c_k$. $\square$

5.5 Solving the coloured castles problem (simplified version)

5.5.1 Certificates (simplified version)

We introduce the notion of a *certificate* for a solution of an instance of the simplified coloured castles problem. We prove that a solution of the problem exists if, and only if, there exists a certificate of a bounded size.

Before presenting a precise definition of the notion of a certificate we describe it informally and roughly sketch how a solution is constructed from a certificate.

One of the elements of the certificate is a simple placement π_0 containing finite number of castles. Some of the rows and columns in π_0 have multiplicity functions safely extending those from $T_{\rm row}$ and $T_{\rm col}$. We will call such rows and columns, as well as empty rows and columns, *safe*. Each of the remaining contains at most one castle. The placement π_0 serves as an initial part of the whole solution. From there we proceed in stages: in the stage $k+1$ we construct a placement π_{k+1} by adding some finite number of castles to π_k in order to make previously non-safe rows and columns safe.

In each stage we consider non-safe rows or columns, which by our assumptions contain single castles each. Depending on the colour of such a castle we find a pattern in the certificate which tells how many castles of which colours are to be added to this row or column to make it safe. Most of the newly added castles in some previously non-safe row are put in new (i.e., previously empty) columns; some, however, are required by the pattern to be joined to existing columns, which were safe in π_0. Analogous provisions apply to to process of filling up some column so as to make it safe.

Definition 18 Let π be a (simple) placement over Δ, and $S_{\rm row}, S_{\rm col} \subseteq \mathbb{N}$ be finite subsets of $\mathbb{N}$. The *type* of a castle $a = (i, j)$ in π relative to sets $S_{\rm row}$ and $S_{\rm col}$ is the triple

$$\bar{\delta} = (\pi^\Delta(a), i', j') \in \Delta \times (S_{\mathrm{row}} \cup \{\bot\}) \times (S_{\mathrm{col}} \cup \{\bot\})$$

where $i' = i$ if $i \in S_{\mathrm{row}}$ and $i' = \bot$ else; and $j' = j$ if $j \in S_{\mathrm{col}}$ and $j' = \bot$ else.

If $i' \neq \bot$ and $j' \neq \bot$, then both the castle and its type are called *royal*.

Definition 19 A *simple placement generator* over Δ is a tuple $(\pi_0, S_{\mathrm{row}}, S_{\mathrm{col}}, \mathbb{P}_{\mathrm{row}}, \mathbb{P}_{\mathrm{col}})$ where

(a) $\pi_0 = \pi_0^\Delta$ is a simple placement with a finite number of castles.

(b) S_{row} and S_{col} are finite subsets of $\mathbb{N}$, called sets of *special rows* and *special columns*, respectively.

(c) $\mathbb{P}_{\mathrm{row}}$ and $\mathbb{P}_{\mathrm{col}}$ are partial functions that return a *row or column pattern* for a given colour δ.
A row pattern is of the form $(B, w, \nu^\Delta, \mathrm{Col})$, a column pattern is of the form $(B, w, \nu^\Delta, \mathrm{Row})$, where
   – $\nu^\Delta : B \to \Delta$ is a finite Δ-coloured set;
   – w is a distinguished element $w \in B$ with $\nu^\Delta(w) = \delta$;
   – Col (in row patterns) is a function $\mathrm{Col}\colon B \to S_{\mathrm{col}} \cup \{\bot\}$ with $\mathrm{Col}(w) = \bot$; similarly Row (in column patterns) is a function $\mathrm{Row}\colon B \to S_{\mathrm{row}} \cup \{\bot\}$ with $\mathrm{Row}(w) = \bot$.

We say that a type $\bar{\delta} = (\delta, \bot, j)$ *appears* in a row pattern $(B, w, \nu^\Delta, \mathrm{Col})$ if there exists $a \in B \setminus \{w\}$ such that $\nu^\Delta(a) = \delta$ and $\mathrm{Col}(a) = j$; similarly, $\bar{\delta} = (\delta, i, \bot)$ appears in a column pattern $(B, w, \nu^\Delta, \mathrm{Row})$ if $\nu^\Delta(a) = \delta$ and $\mathrm{Row}(a) = i$ for some $a \in B \setminus \{w\}$.

We say that a type $\bar{\delta} = (\delta, i', j')$ *appears* in the placement generator if there exists a castle of type $\bar{\delta}$ in π_0 or $\bar{\delta}$ appears in a row pattern or a column pattern of this placement generator.

Definition 20 A placement generator $\lambda = (\pi_0, S_{\mathrm{row}}, S_{\mathrm{col}}, \mathbb{P}_{\mathrm{row}}, \mathbb{P}_{\mathrm{col}})$ is a *certificate* for a solution of an instance $(\Delta, T_{\mathrm{row}}, T_{\mathrm{col}}, \mathrm{Frb})$ of the simplified coloured castles problem if the following are satisfied (we group these conditions under four separate headings).

Initial Placement π_0.

(I1) Each row (column) is safe or contains at most one castle.

(I2) If a row (column) belongs to S_{row} (S_{col}) then it is safe.

(I3) If castles a, a' are in a skew position then $(\pi^\Delta(a), \pi^\Delta(a')) \notin \mathrm{Frb}$.

(I4) Every castle from a row in S_{row} belongs to a safe column;
similarly, the rows of castles in special columns are safe.

Row Patterns.

(R0) If a type $(\delta, \bot, \bot)$ appears in λ then $\mathbb{P}_{\mathrm{row}}(\delta)$ is defined.

For all colours δ, if $\mathbb{P}_{\mathrm{row}}(\delta) = (B, w, \nu^\Delta, \mathrm{Col})$, then

(R1) the multiplicity function of B (with colouring ν^Δ) belongs to T_{row},

(R2) if $j \in \mathbb{N}$ then $\mathrm{Col}(a) = j$ for at most one $a \in B$, (R3) if $a \in B \setminus \{w\}$ is such that $\mathrm{Col}(a) = l \neq \bot$ and $\delta' = \nu^{\Delta}(a)$, then column l of placement π_0 contains at least two castles of colour δ'.

Column Patterns.

(C0) – (C4) strictly analogous to the above, with roles of rows and columns exchanged.

Forbidden Pairs of Colours.

(F1) For any pair (δ, i, j) and (δ', i', j') of types appearing in λ:
$i = i' \in \mathbb{N}$ or $j = j' \in \mathbb{N}$ or $(\delta, \delta') \notin \mathrm{Frb}$.

The next two sections will show that certificates are adequate in the sense of the following assertion.

Lemma 21 *An instance of the simplified coloured castles problem has a solution if, and only if, it possesses a certificate.*

5.5.2 From a certificate to a solution (simplified version)

Let $(\Delta, T_{\mathrm{row}}, T_{\mathrm{col}}, \mathrm{Frb})$ be an instance to the simplified coloured castles problem and let $\lambda = (\pi_0, S_{\mathrm{row}}, S_{\mathrm{col}}, \mathbb{P}_{\mathrm{row}}, \mathbb{P}_{\mathrm{col}})$ be a certificate for this instance.

We construct a (possibly infinite) chain of simple placements $\pi_0, \pi_1 \ldots$, starting from the placement π_0 given in the certificate, such that π_{k+1} is obtained from π_k by adding some finite number of castles. By $\bigcup_{k \in \mathbb{N}} \pi_k$ we denote the natural limit of this chain, i.e., the simple placement which has a castle of colour δ in the field (i, j) if there is a castle of colour δ in field (i, j) of π_k for some $k \in \mathbb{N}$.

Every placement π_k will satisfy properties (I1) – (I4) from Definition 20 and will contain only castles of types appearing in λ. Placements are extended in such a way that each castle of π_k will be safe (i.e., its row and column will be safe) in π_n from $n = k + 1$ onward. It follows that the limit placement $\bigcup_{k \in \mathbb{N}} \pi_k$ is a correct solution. Note that it may contain infinite number of castles.

The inductive extension from π_k to π_{k+1} is performed as follows.

From π_k to π_{k+1}. Assume that π_k satisfies (I1) – (I4) and contains only castles of types appearing in λ.

For castle a in π_k in field (i, j) proceed as follows.

(a) If row i is not safe, add some castles to this row to make it safe. By (I1), a is the only castle in i; by (I2), $i \notin S_{\mathrm{row}}$; by (I4), $j \notin S_{\mathrm{col}}$. Thus the type $\bar{\delta}$ of a in π_k, which by the assumption appears in λ, is of the form $(\delta, \bot, \bot)$. Hence, by (R0), $\mathbb{P}_{\mathrm{row}}(\delta)$ is defined. Let $\mathbb{P}_{\mathrm{row}}(\delta) = (B, w, \nu^{\Delta}, \mathrm{Col})$ and enumerate the elements of $B \setminus \{w\}$ as $b_1, \ldots, b_l$. For each b_j put a new castle of colour $\nu^{\Delta}(b_j)$ in row i; if $\mathrm{Col}(b_j) = \bot$, this new castle is put in an empty column, otherwise it is put into column $\mathrm{Col}(b_j)$. By (R2) there is no danger that two castles are put in the same field.

(b) If column j is not safe, proceed analogously.

Note that for $k > 0$ we need to take actions according to (a) or (b) only for castles added in step k. Also, no castle needs to be treated in both (a) and (b). For some new castles, no further action is required; this is because every new castle comes with a safe row or column (built according to a pattern), and some of these castles are put in special columns or rows, whence both their row and column are safe already.

It should be clear that all types of castles in π_{k+1} appear in λ. We argue that π_{k+1} satisfies properties (I1) – (I4). Suppose castle a has been newly added in row i. If it was added in (a), then the multiplicity function of i is safe by (R1); otherwise, either $i \in S_{\mathrm{row}}$ (and in this case row i remains safe by (C4)), or a is the only castle in row i. Analogously we can show that the column of a is either safe or contains only a. This implies (I1) and (I2). For (I3) consider a pair of castles in a skew position in π_{k+1}: their types appear in λ and their pair of colours is not in Frb by (F1); hence (I3) is satisfied. (I4) is clearly preserved.

5.5.3 Extracting a certificate from a solution (simplified version)

Let $(\Delta, T_{\mathrm{row}}, T_{\mathrm{col}}, \mathrm{Frb})$ be an instance of the simplified coloured castles problem. Let $\pi = \pi^{\Delta}$ be a simple placement solving the problem. We show how to extract a certificate $(\pi_0, S_{\mathrm{row}}, S_{\mathrm{col}}, \mathbb{P}_{\mathrm{row}}, \mathbb{P}_{\mathrm{col}})$ from π.

Special rows and columns. We first distinguish a set S_{row} of special rows, a set S_{col} of special columns and a set $\Delta_0 \subseteq \Delta$ of *special colours*. Intuitively, a colour δ belongs to Δ_0 if it can appear only some specific number of times in a solution, or can only appear in some specific number of rows or columns. S_{row} and S_{col} are, respectively, rows and columns containing castles coloured with special colours.

The sets Δ_0, S_{row} and S_{col} are obtained by saturating initial instantiations recursively in order to achieve suitable closure conditions. We present a recursive process for their definition that will also facilitate the proof of an exponential size bound on S_{row} and S_{col} in Lemma 22 below.

Initialisation.

- (0) If a single row (column) i contains all castles of some colour $\delta \in \Delta$, then add δ to Δ_0 and i to S_{row} (S_{col}); in particular, if there is only one castle of colour δ in π, then both its row and its column become special in this step.
- (0') If a row i and a column j intersect in a castle and together contain all castles of some colour $\delta \in \Delta$, but if no single row or column contains all those castles, then add δ to Δ_0, i to S_{row} and j to S_{col}.

Recursive saturation.

- (1) If for some δ which is not yet in Δ_0, every castle of colour δ belongs to a row from S_{row} or to a column from S_{col}, then add δ to Δ_0.
- (2) If a is the only castle of colour $\delta \in \Delta_0$ in some row $i \in S_{\mathrm{row}}$ and the column j of a is not in S_{col}, then add j to S_{col}; similarly, if a is the only castle of colour $\delta \in \Delta_0$ in some column $j \in S_{\mathrm{col}}$ and its row i is not in S_{row}, add i to S_{row}.

We call castles coloured with special colours $\delta \in \Delta_0$ *special castles*. Note that special castles belong to special rows or special columns (or to both), but that special rows and columns may contain non-special castles.

Let us consider again models from our examples in Section 5.1. They naturally correspond to simple placements (see Section 5.3.1). Let us see which classes in those models are translated into special rows and columns (call those classes special as well). In Fig. 1 (a) the only special classes are the E_1- and E_2-classes of the leftmost element (the latter is a singleton). In Fig. 1 (b) we find one more special class, viz. the lower line of elements satisfying R. In both examples the classes are special already according to (0). In Fig. 3 all four classes are special according to (0').

The model from Fig. 2 is more interesting in that it illustrates a non-trivial saturation process. Initially, by step (0), only the E_2-class formed by the bottom line of elements in S, both classes of the root (its E_2-class is trivial) and two E_2-classes containing elements from P_1 are special.[5] By repeated applications of steps (1) and (2), however, all classes and intersections successively become special.

This shows that the number of special classes may be necessarily exponential in the number of colours[6] in Δ. The following Lemma establishes a corresponding upper bound.

Lemma 22 *The size of S_{row} and the size of S_{col} are at most exponential in $|\Delta|$.*

Proof Let $t = |\Delta|$. Step (1) can be executed at most t times. Let us denote by s_0 the sum of the number of elements in S_{row} and in S_{col} after steps (0) and (0'), and by s_k the sum of the number of elements in S_{row} and the number of elements in S_{col} after the k-th iteration of step (1). Clearly $s_0 \leq 4t$ initially.[7] Between the k-th and $(k+1)$-st iteration of (1), step (2) can be executed no more than ts_k times, since every $(\delta, i) \in \Delta_0 \times S_{\mathrm{row}}$ can produce at most one new member j of S_{col} and every $(\delta, i) \in \Delta_0 \times S_{\mathrm{col}}$ can produce at most one new $i \in S_{\mathrm{row}}$ (and if a row (column) i contains a castle a whose colour is a member of the current Δ_0, then the column (row) of a is already in S_{col} (S_{row})). Therefore $s_{k+1} \leq s_k + ts_k \leq (1+t)s_k$, and it follows that $s_t \leq (1+t)^t 4t$, which is exponential in t. ☐

The following simple observation will be helpful later.

Lemma 23 *Let $\delta = \pi^{\Delta}(a), \delta' = \pi^{\Delta}(a')$ for some castles a, a' that do not share a special row or a special column (but may belong to the same non-special row or non-special column). Then $(\delta, \delta') \notin \mathrm{Frb}$.*

Proof Let $a_1, a_2, \ldots$ be the list of all castles of colour δ'. If one of them is in a skew position with a in π then $(\delta, \delta') \notin \mathrm{Frb}$, since π is a correct solution. Otherwise, there are three possible cases: all a_i, including a', are in the row of a; all a_i are in the column of a; or all a_i are in the union of both. In the first case, the row of a becomes special in step (0); in the second case, the column of a becomes special in step (0); and in the third case, both the row and the column of a become special in step (0'). Thus a and a' share a special row or a special column, contrary to the assumption of the lemma. ☐

[5]Note that the elements in P_1 have different colours, since only one of them is in L.

[6]And triply exponential in terms of formula size, if we translate the problem

[7]We bound s_0 by $4t$, rather than $2t$, because sometimes an castle can produce two special rows and two special columns, cf. Fig. 3.

The initial placement π_0. We construct π_0 from π in three steps.

(1) For every row $i \in S_{\mathrm{row}}$ (column $i \in S_{\mathrm{col}}$), for every $\delta \in \Delta$, mark exactly $\theta(\delta)$ castles in i, where $\theta \in T_{\mathrm{row}}$ ($\theta \in T_{\mathrm{col}}$) is a function which is safely extended by the multiplicity function of row or column i.

(2) For every castle (i, j) marked in step (1) if $i \notin S_{\mathrm{row}}$ ($j \notin S_{\mathrm{col}}$) then mark exactly $\theta(\delta)$ castles in row i (column j), where $\theta \in T_{\mathrm{row}}$ ($\theta \in T_{\mathrm{col}}$) is a function which is safely extended by the multiplicity function of i (j).

(3) Let π' be the placement obtained from π by removing all castles that have not been marked in the previous steps. If a row i (column j) in π' is not safe then for every castle in this row (column) move it to an empty row (column) without changing its column (row); this is always possible since we have only finitely many marked castles. Let π_0 be the structure obtained at the end of this process.

It is readily checked that π_0 meets conditions (I1), (I2) and (I4). For (I3) consider two castle which were in the same row (column) in π but were moved to skew positions in step (3): in this case, their original row (column) was not special, and by Lemma 23 their pair of colours is not forbidden.

Patterns. We describe the process of defining row patterns. The process for column patterns is analogous. Let $\bar{\delta} = (\delta, \perp, \perp)$ be a type realised in π by a castle a. Thus $\delta \in \Delta \setminus \Delta_0$. Let i be the row of a; according to $\bar{\delta}$, $i \notin S_{\mathrm{row}}$. Let $\theta \in T_{\mathrm{row}}$ be a function which is safely extended by the multiplicity function of row i. We mark exactly $\theta(\delta')$ castles in i for every $\delta' \in \Delta$. This is done in such a way that a is one of the marked elements. Let B consists of all the marked castles.

Let $\nu^{\Delta}(b) = \pi^{\Delta}(b)$ for all $b \in B$. Consider $b \in B$ and let $(\delta', \perp, j)$ be its type in π. If the type $(\delta', \perp, \perp)$ is realized in π (by b or some other castle) then we set $\mathrm{Col}(b) = \perp$, otherwise we set $\mathrm{Col}(b) = j$ (which may happen to be $\perp$ as well). We define $\mathbb{P}_{\mathrm{row}}(\bar{\delta})$ to be $(B, a, \nu^{\Delta}, \mathrm{Col})$.

We repeat the process for all appropriate δ.

Let us now consider the conditions for certificates related to row patterns. We note that all types in patterns or in π_0 appear in π. Since we build a pattern for all types of the form $(\delta, \perp, \perp)$, (R0) is satisfied. (R1) and (R2) are straightforward. Considering (R3), let δ be the colour of a. Note that $\delta \in \Delta_0$. The pattern B is constructed from a non-special row i in π which intersects column $\mathrm{Col}(a)$ in an element of colour δ. Column $\mathrm{Col}(a)$ in π has to contain at least two castles of colour δ, since otherwise step (2) in the collection of special rows would have made row i special; at least two such castles are retained in π_0, by our choice of π_0.

Forbidden pairs of colours. Property (F1) for certificates is satisfied: all types appearing in the certificate are types from π so that (F1) is a direct consequence of Lemma 23.

5.5.4 Size of certificates (simplified version)

Let $(\Delta, T_{\mathrm{row}}, T_{\mathrm{col}}, \mathrm{Frb})$ be an instance of the simplified coloured castles problem. Let $\max(T_{\mathrm{row}}) = \max\{\sum_{\delta \in \Delta} \theta(\delta) : \theta \in T_{\mathrm{row}}\}$, i.e., $\max(T_{\mathrm{row}})$ is the maximal size of a set whose multiplicity function is taken from T_{row}; similarly define $\max(T_{\mathrm{col}})$. Let $\lambda = (\pi_0, S_{\mathrm{row}}, S_{\mathrm{col}}, \mathbb{P}_{\mathrm{row}}, \mathbb{P}_{\mathrm{col}})$ be a certificate extracted from a solution π. The number of castles in the initial placement π_0 can be bounded by $|S_{\mathrm{row}}| \max(T_{\mathrm{row}}) \max(T_{\mathrm{col}}) + |S_{\mathrm{col}}| \max(T_{\mathrm{col}}) \max(T_{\mathrm{row}})$. $\mathbb{P}_{\mathrm{row}}$ and $\mathbb{P}_{\mathrm{col}}$ are defined for at most $|\Delta|$ colours, and each of the patterns has at most $\max(T_{\mathrm{row}})$ or $\max(T_{\mathrm{col}})$ elements. Finally, $|\mathrm{Frb}|$ can be bounded by $|\Delta|^2$.

Recall that, by Lemma 22, $|S_{\mathrm{row}}|$ and $|S_{\mathrm{col}}|$ are at most exponential in $|\Delta|$. Hence, the whole certificate can be described in size that is exponential with respect to the input size.

5.6 Solving the coloured castles problem (regular version)

The construction for the regular version goes along the same lines as the simplified version. We emphasise the most important differences.

5.6.1 Certificates (regular version)

The new component of the certificate is a pattern function $\mathbb{P}_{\mathrm{skw}}$ indicating a strategy to satisfy skew requirements. For technical reasons, pattern functions now take as an input not only the colour of a castle but also the information whether and to which special row or column it belongs. We do not maintain property (I4) from the simplified version; instead we impose additional restrictions (I4) – (I7) on π_0 related to skew requirements and consistency of arrow connections. Obviously some conditions on skew patterns also appear.

The new complete definitions of placement generators and certificates are as follows.

Definition 24 A *placement generator* over (Δ, E) is a tuple $(\pi_0, S_{\mathrm{row}}, S_{\mathrm{col}}, \mathbb{P}_{\mathrm{row}}, \mathbb{P}_{\mathrm{col}}, \mathbb{P}_{\mathrm{skw}})$ where

(a) $\pi_0 = (\pi_0^{\Delta}, \pi_0^{E})$ is a placement with a finite number of castles.

(b) S_{row} and S_{col} are finite subsets of $\mathbb{N}$, called sets of *special rows* and *special columns*, respectively.

(c) $\mathbb{P}_{\mathrm{row}}, \mathbb{P}_{\mathrm{col}}$ and $\mathbb{P}_{\mathrm{skw}}$ are partial functions that return a *row, column, or skew pattern* for a given type (δ, i, j).
A row pattern is of the form $(B, w, \nu^{\Delta}, \mathrm{Col})$, a column pattern is of the form $(B, w, \nu^{\Delta}, \mathrm{Row})$, and a skew pattern is of the form $(B, \nu^{\Delta}, \nu^{E}, \mathrm{Row}, \mathrm{Col})$, where
   - $\nu^{\Delta} : B \to \Delta$ is a finite Δ-coloured set;
   - w is a distinguished element $w \in B$ with $\nu^{\Delta}(w) = \delta$;
   - Col (in row or skew patterns) is a function $\mathrm{Col}\colon B \to S_{\mathrm{col}} \cup \{\bot\}$ with $\mathrm{Col}(w) = j$; similarly Row (in column or skew patterns) is a function $\mathrm{Row}\colon B \to S_{\mathrm{row}} \cup \{\bot\}$ with $\mathrm{Row}(w) = i$.

We say that a type $\bar{\delta} = (\delta, \bot, j)$ *appears* in a row pattern $(B, w, \nu^{\Delta}, \mathrm{Col})$ if there exists $a \in B \setminus \{w\}$ such that $\nu^{\Delta}(a) = \delta$ and $\mathrm{Col}(a) = j$; similarly, $\bar{\delta} = (\delta, i, \bot)$

appears in a column pattern $(B, w, \nu^\Delta, \mathrm{Row})$ if $\nu^\Delta(a) = \delta$ and $\mathrm{Row}(a) = i$ for some $a \in B \setminus \{w\}$; and a type $\bar{\delta} = (\delta, i, j)$ appears in a skew pattern $(B, \nu^\Delta, \nu^E, \mathrm{Row}, \mathrm{Col})$ if $\nu^\Delta(a) = \delta$, $\mathrm{Row}(a) = i$ and $\mathrm{Col}(a) = j$ for some $a \in B$.

We say that a type $\bar{\delta} = (\delta, i, j)$ *appears* in the placement generator if there exists a castle of type $\bar{\delta}$ in π_0 or $\bar{\delta}$ appears in a row pattern, a column pattern or a skew pattern of this placement generator.

A certificate for an instance of the regular version of the coloured castles problem serves the same purpose as in the simple version. There are a few differences in the actual format:

- (I4) from the simplified version is replaced by new conditions (I4) – (I7), which are related to skew requirements and arrow connections.
- (R0) and (C0) are slightly modified.
- There is a whole new group of conditions, (S0) – (S5), related to skew patterns.

Definition 25 A placement generator $\lambda = (\pi_0, S_{\mathrm{row}}, S_{\mathrm{col}}, \mathbb{P}_{\mathrm{row}}, \mathbb{P}_{\mathrm{col}}, \mathbb{P}_{\mathrm{skw}})$ is a *certificate* for a solution of an instance $(\Delta, T_{\mathrm{row}}, T_{\mathrm{col}}, \mathrm{Frb}, E, \mathrm{Inv}^E, \mathrm{Ends}^E, \mathrm{SReq})$ of the regular version of the coloured castles problem if it satisfies the following.

Initial Placement π_0.

(I1) Each row (column) is safe or contains at most one castle.

(I2) If a row (column) belongs to S_{row} (S_{col}) then it is safe.

(I3) If castles a, a' are in a skew position then $(\pi_0^\Delta(a), \pi_0^\Delta(a')) \notin \mathrm{Frb}$.

(I4) Each royal castle has its skew requirements satisfied in π_0.

(I5) Each castle either has its skew requirements satisfied in π_0 or has no arrows to or from royal castles.

(I6) If $\pi_0^E(a, b) = \varepsilon \neq \perp$ for castles a, b, then $\mathrm{Ends}^E(\varepsilon) = (\pi_0^\Delta(a), \pi_0^\Delta(b))$.

(I7) If $\pi_0^E(a, b) = \varepsilon \neq \perp$ and $\pi_0^E(b, a) = \varepsilon' \neq \perp$ for castles a, b, then $\mathrm{Inv}^E(\varepsilon) = \varepsilon'$.

Row Patterns. (unchanged)

(R0) If a type $\bar{\delta} = (\delta, \perp, j)$ appears in λ then $\mathbb{P}_{\mathrm{row}}(\bar{\delta})$ is defined.

For all types $(\delta, \perp, j)$, if $\mathbb{P}_{\mathrm{row}}(\delta, \perp, j) = (B, w, \nu^\Delta, \mathrm{Col})$, then:

(R1) the multiplicity function of B belongs to T_{row},

(R2) if $j' \in \mathbb{N}$ then $\mathrm{Col}(a) = j'$ for at most one $a \in B$,

(R3) if $a \in B \setminus \{w\}$ is such that $\mathrm{Col}(a) = l \neq \perp$ and $\delta' = \nu^\Delta(a)$, then column l of π_0 contains at least two castles of colour δ'.

Column Patterns. (unchanged)

(C0) – (C4) strictly analogous to the above, with roles of rows and columns exchanged.

Skew Patterns. (new)

(S0) $\mathbb{P}_{\mathrm{skw}}$ is defined for all non-royal types (δ, i, j) appearing in λ.

If $\mathbb{P}_{\mathrm{skw}}(\delta, i, j) = (B, \nu^\Delta, \nu^E, \mathrm{Row}, \mathrm{Col})$ then

(S1) $\mathrm{Ends}^E(\nu^E(a)) = (\delta, \nu^\Delta(a))$ for all $a \in B$;

(S2) $\mathrm{Row}(a) \neq i$ or $\mathrm{Row}(a) = \perp$ for all $a \in B$; similarly, $\mathrm{Col}(a) \neq j$ or $\mathrm{Col}(a) = \perp$;

(S3) if for some $a \in B$ the colour $\nu^{\Delta}(a) = \delta'$ is not royal and $\mathrm{Row}(a) = l \neq \perp$, then row l of π_0 contains at least two castles of colour δ'; similarly, for columns;

(S4) there is some $S \in \mathrm{SReq}(\delta)$ such that every $\varepsilon \in S$ is attained as value $\nu^{E}(a) = \varepsilon$ for some $a \in B$;

(S5) If a royal type appears in a skew pattern then it also appears in π_0.

Forbidden Pairs of Colours. (unchanged)

(F1) For any pair (δ, i, j) and (δ', i', j') of types appearing in λ:
$i = i' \in \mathbb{N}$ or $j = j' \in \mathbb{N}$ or $(\delta, \delta') \notin \mathrm{Frb}$.

We establish the following in Sections 5.6.2 and 5.6.3 below.

Lemma 26 *There exists a solution for an instance to the regular version of the coloured castles problem if and only if it possesses a certificate.*

5.6.2 From a certificate to a solution (regular version)

Let $(\Delta, T_{\mathrm{row}}, T_{\mathrm{col}}, \mathrm{Frb}, E, \mathrm{Inv}^{E}, \mathrm{Ends}^{E}, \mathrm{SReq})$ be an instance to the regular version of the coloured castles problem and let $\lambda = (\pi_0, S_{\mathrm{row}}, S_{\mathrm{col}}, \mathbb{P}_{\mathrm{row}}, \mathbb{P}_{\mathrm{col}}, \mathbb{P}_{\mathrm{skw}})$ be a certificate for this instance.

We construct a (possibly infinite) chain of finite placements $\pi_0, \pi_1 \ldots$, starting from the initial placement π_0 provided by the certificate, such that π_{k+1} is obtained from π_k by adding some finite number of castles. Every placement π_k will satisfy properties (I1) – (I7) from Definition 25 and will contain only castles of types appearing in λ. We extend placements in such a way that every castle from π_k will have its row and its column made safe and its skew requirements satisfied in π_{k+1}. Thus the natural limit of the chain of placements, defined as in the simple case, will be a correct solution.

The inductive extension from π_k to π_{k+1} is performed as follows.

From π_k to π_{k+1}. Assume that π_k satisfies (I1) – (I7), contains only castles of types appearing in λ and contains no new royal types except those realised in π_0. We extend this placement to π_{k+1}.

For castle a in π_k in field (i, j) proceed as follows.

(a) If row i of a is not safe, add some castles to this row. By (I1), a is the only castle in i; by (I2) $i \notin S_{\mathrm{row}}$. Thus the type $\bar{\delta}$ of a in π_k is of the form $(\delta, \perp, j)$ and by the inductive assumption it appears in λ. By (R0), $\mathbb{P}_{\mathrm{row}}$ is defined for $\bar{\delta}$. Let $\mathbb{P}_{\mathrm{row}}(\bar{\delta}) = (B, w, \nu^{\Delta}, \mathrm{Col})$ and let $b_1, \ldots, b_l$ enumerate the elements of $B \setminus \{w\}$. For each b_j put a new castle of colour $\nu^{\Delta}(b_j)$ in row i; if $\mathrm{Col}(b_j) = \perp$ the new castle is put into an empty column, otherwise into column $\mathrm{Col}(b_j)$. By (R2) there is no danger that two castles are put in the same field.

(b) If column j of a is not safe proceed analogously.

(c) If a has its skew requirements not already satisfied in π_k, then we satisfy them in π_{k+1} as follows. By (I4) and the inductive assumption about royal types, a is not royal; by (S0) this implies that $\mathbb{P}_{\mathrm{skw}}$ is defined for the type $\bar{\delta} = (\delta, i, j)$

of a. Let $\mathbb{P}_{\rm skw}(\bar{\delta}) = (B, \nu^{\Delta}, \nu^{E}, \mathrm{Row}, \mathrm{Col})$ and enumerate the elements in B as $b_1, \ldots, b_l$. If $\mathrm{Row}(b_j) \neq \perp$ and $\mathrm{Col}(b_j) \neq \perp$, find a royal castle c in π_k of type $(\nu^{\Delta}(b_j), \mathrm{Row}(b_j), \mathrm{Col}(b_j))$; such a castle exists by (S5); by (I5) it is not yet connected to a by an arrow, so we may put an arrow of colour $\nu^{E}(b_j)$ from a to c.
For each of the remaining b_j put a new castle of colour $\nu^{\Delta}(b_j)$ in π_{k+1}; if $\mathrm{Row}(b_j) = \mathrm{Col}(b_j) = \perp$ the new castle is put it in an arbitrary intersection of an empty row with and empty column; if $\mathrm{Row}(b_j) = i \in \mathbb{N}$ into row i (and a fresh column); if $\mathrm{Col}(b_j) = j \in \mathbb{N}$ into column j (and a fresh row).

Note that, similarly to the simplified version, we need to take some actions only for some castles added in step k. This time, however, there may be castles for which we have to perform some actions in all of (a), (b), (c), since some castles added according to skew patterns come without a safe row or a safe column. Observe also that, in contrast to the simplified version, we sometimes have to construct a safe row (column) for a castle in a special column (row); again, this is because some castles are put in special columns or rows according to skew patterns.

It should be clear that all types of castles in π_{k+1} appear in λ. Our strategy of extending placements and (S5) guarantee also that no new royal types are realised in π_{k+1}.

It is also not hard to see that π_{k+1} satisfies properties (I1) – (I5). Consider the row of a newly added castle a. If it was added in (a) then its multiplicity function is safe by (R1). In the other case the row either belongs to $S_{\rm row}$ (and in this case it remains safe by (C3) or (S3)), or contains only a. Analogously we can show that the column of a is either safe or contains only a. This implies (I1) and (I2). (I3) is satisfied due to (F1) (cf. the simple version). (I4) and (I5) obviously remain true. (I6) is preserved due to (S1). (I7) is maintained since there may be arrows simultaneously from a to b and from b to a only if a, b are castles from π_0.

5.6.3 Extracting a certificate from a solution (regular version)

Let $(\Delta, T_{\rm row}, T_{\rm col}, \mathrm{Frb}, E, \mathrm{Inv}^{E}, \mathrm{Ends}^{E}, \mathrm{SReq})$ be an instance of the regular version of the coloured castles problem and let $\pi = (\pi^{\Delta}, \pi^{E})$ be a solution. We extract a valid certificate $\lambda = (\pi_0, S_{\rm row}, S_{\rm col}, \mathbb{P}_{\rm row}, \mathbb{P}_{\rm col}, \mathbb{P}_{\rm skw})$ from π.

Special rows and columns. These are defined exactly as in the simple version.

The initial placement. We construct π_0 from π in the following steps.

(1) In every row $i \in S_{\rm row}$ (column $i \in S_{\rm col}$), for every $\delta \in \Delta$, mark exactly $\theta(\delta)$ many castles where $\theta \in T_{\rm row}$ ($\theta \in T_{\rm col}$) is a function that is safely extended by the multiplicity function of i. Additionally mark all royal castles.

(2) For every royal castle c mark a minimal number of castles to ensure that all skew requirements of c are satisfied.

(3) For every non-royal castle a marked in step (2) mark a minimal number of castles to ensure that all skew requirements of a are satisfied.

(4) For every royal castle a retain the arrows from a to castles added in (2) to satisfy its skew requirements. Similarly, for every castle a added in step (2), retain the arrows from a to castles added in (3) to satisfy its skew requirements. All remaining arrows and all non-marked castles are removed; let π_0' be the placement thus obtained.

(5) If a row i (column j) in π_0' is not safe, we move every castle in this row (column) to an empty row (column) without changing its column (row). This is always possible since there are only finitely many marked castles. Every castle moved in this process retains all arrow links with other castles.

Let π_0 be the placement thus obtained.

It is readily checked that π_0 meets conditions (I1) and (I2). The only potential problem with (I3) arises if some castles, which originally were in the same row or column, are moved to skew positions in step (5). But in this case, their original shared row or column was not special and by Lemma 23 their pair of colours is not forbidden. (I4) is ensured in steps (2) and (4); (I5) in steps (3) and (4). For (I6) and (I7) observe that in π_0' all castles and arrows are taken from the original solution π (we only removed some castles and arrows). So π_0' satisfies (I6) and (I7). Moving some castles to another positions according to (5) cannot spoil these properties.

Row and column patterns. We describe the process of defining row patterns. The process for column patterns is analogous.

Let $\bar{\delta} = (\delta, \perp, j)$ be a type realised in π by a castle a. Let i be the row of a. Let $\theta \in T_{\rm row}$ be a function which is safely extended by the multiplicity function of the row i. We mark exactly $\theta(\delta')$ castles in i for every $\delta' \in \Delta$; we do this in such a way that a is one of the marked elements. Let B consist of all the marked castles. We put $\nu^{\Delta}(b) := \pi^{\Delta}(b)$ for all $b \in B$. Put $\mathrm{Col}(a) := j$. Consider some $b \in B \setminus \{a\}$, of type $(\delta', \perp, j)$ according to π. If the type $(\delta', \perp, \perp)$ is realized in π (by b or some other castle) then we set $\mathrm{Col}(b) = \perp$, otherwise we set $\mathrm{Col}(b) = j$ (which may happen to be $\perp$). We define $\mathbb{P}_{\rm row}(\bar{\delta})$ to be $(B, a, \nu^{\Delta}, \mathrm{Col})$.

We repeat the process for all appropriate $\bar{\delta}$.

Let us now consider requirements (R0) – (R3) on certificates. All types that appear in patterns or in π_0 are realised in π; thus (R0) is satisfied. (R1) and (R2) are straightforward.

Consider now (R3). Let $(B, w, \nu^{\Delta}, \mathrm{Col})$ be the row pattern returned for some $(\delta, \perp, j)$. Let $a \in B \setminus \{w\}$, $\mathrm{Col}(a) = l \neq \perp$ and let $\delta' = \nu^{\Delta}(a)$. Note that δ' is a special colour, since otherwise we would have a castle coloured by δ' in a non-special row and a non-special column, and thus, when constructing the pattern, we would set $\mathrm{Col}(a) = \perp$. The pattern B is constructed from a non-special row i in π which intersects column $\mathrm{Col}(a)$ in an element of colour δ'. The column $\mathrm{Col}(a)$ in π has to contain at least two realisations of δ, since otherwise step (2) of distinguishing special rows and column would have made row i special; at least two such castles are retained in π_0, by our choice of π_0.

Skew patterns. Let $\bar{\delta} = (\delta, i, j)$ with $i = \perp$ or $j = \perp$ be a type realised in π. Let a be a realisation of $\bar{\delta}$ in π. Let $S \in \mathrm{SReq}(\delta, i, j)$ be a set such that a sends arrow of colour ε for all $\varepsilon \in S$. For every $\varepsilon \in S$ choose some castle b such that a sends an arrow of colour ε to b. Add b to B and set $\nu^{\Delta}(b) := \pi^{\Delta}(b)$. Let i' be the row of b and j' its column. If $i' \in S_{\mathrm{row}}$ and $j' \in S_{\mathrm{col}}$ (i.e., the type of b is royal) or $i' \notin S_{\mathrm{row}}$ and $j' \notin S_{\mathrm{col}}$ or if there is no realisation of $(\delta, \perp, \perp)$ in π, then set $\mathrm{Row}(b) = i'$ and $\mathrm{Col}(b) = j'$, otherwise set $\mathrm{Row}(b) = \mathrm{Col}(b) = \perp$. Set $\nu^{E}(b) = \pi^{E}(a, b)$.

Properties (S0) – (S2) are straightforward to check. (S3) holds because of the same arguments as used for (R3). (S4) and (S5) easily follow from the construction.

5.6.4 Size of certificates (regular version)

The analysis of the size of certificates is very similar to the simple case: the number of castles in the initial placement is still at most exponential with respect to the size of the input. Pattern functions $\mathbb{P}_{\mathrm{row}}, \mathbb{P}_{\mathrm{col}}, \mathbb{P}_{\mathrm{skw}}$ are defined for an at most exponential number of types. So the whole certificate can still be described in exponential size.

5.7 Complexity issues

Consider an FO^2-formula φ . In polynomial time we can transform it into $\mathcal{EQ}[\tau_0, E_1, E_2]$-normal form formula φ', such that φ and φ' are satisfiability equivalent over $\mathcal{EQ}[\tau_0, E_1, E_2]$. We reduce satisfiability of φ' to the coloured castles problem as described in Section 5.4. This reduction requires generating some objects of size doubly exponential with respect to $|\varphi|$ and checking whether they satisfy certain properties, which are simple to verify. Thus it works in in doubly exponential time.

Given an instance of the coloured castles problem we check whether it is solvable in a straightforward manner by guessing a pattern generator of at most exponential size and checking the requirements for a valid certificate. This procedure works in nondeterministic exponential time with respect to the size of the instance.

Combining the above, we finally obtain a 3NEXPTIME upper bound on the satisfiability problem of FO^2 over the class of models with two equivalence relations. This proves part (ii) of Theorem 1.

6 Three Equivalence Relations

It is shown in [13] that FO^2 is undecidable over equivalence structures with four equivalence relations. We sharpen this result by reducing the number of equivalence relations to three, thus completing the picture. In fact we obtain a stronger result. Its proof uses a special adaptation of a reduction of the domino tiling problem [3], which is particularly suited for FO^2, as presented in [26] based on work in [13, 15].

A *conservative reduction* from one logic to another is a reduction that simultaneously translates both the satisfiability problem (satisfiability over all structures) and the finite satisfiability problem (satisfiability in finite structures). The existence of a conservative reduction from FO to some other logic $\mathcal{L}$ implies not just that $\mathrm{SAT}(\mathcal{L})$ and $\mathrm{FINSAT}(\mathcal{L})$ are both undecidable, but even that the complement of $\mathrm{SAT}(\mathcal{L})$ is recursively inseparable from $\mathrm{FINSAT}(\mathcal{L})$, cf. [3]. A logic (fragment of FO) admitting a conservative reduction from FO is classically referred to as a *conservative reduction class*. These notions and this observation extend naturally to the setting with additional semantic restrictions on the class of (finite) models admitted for $\mathcal{L}$. In our case, a conservative reduction to FO^2 over equivalence structures with three equivalence relations in particular implies an analogous inseparability result concerning $\mathrm{SAT}(\mathrm{FO}^2, \mathcal{EQ}[\tau_0; E_1, E_2, E_3])$ and $\mathrm{FINSAT}(\mathrm{FO}^2, \mathcal{EQ}[\tau_0; E_1, E_2, E_3])$.

Proposition 27 FO^2 *over* $\mathcal{EQ}[\tau_0; E_1, E_2, E_3]$ *forms a conservative reduction class.*

A closer inspection of the formulae used in the basic reductions given below, or ramifications of these, show that FO^2 can be further restricted to the two-variable guarded fragment without equality and in a vocabulary consisting of just unary predicates apart from E_1, E_2, E_3.

Corollary 28 *Satisfiability and finite satisfiability in* $\mathcal{EQ}[\tau_0; E_1, E_2, E_3]$ *are undecidable even for the following fragment of* FO^2 *in vocabularies* τ_0 *consisting only of unary predicates: conjunctions of sentences of the forms*

(a) $\forall x \forall y (E_i xy \rightarrow \chi(x, y))$*, and*

(b) $\forall x (\alpha(x) \rightarrow \exists y (E_i xy \wedge \alpha'(y)))$*,*

for quantifier-free and equality-free formulae χ *and* α *and* α'*. This fragment is in particular contained in the two-variable guarded fragment (note however, that equivalence symbols are allowed in* χ*).*

Our underlying reduction of the tiling problem to FO^2 satisfiability in restricted classes of structures, which serves to establish a conservative reduction, is closely based on the framework of [13, 15] and its ramifications in [26]. We include the following basic definitions and lemmas (without proofs) from [26] for convenience.

Let $\mathfrak{G}_{\mathbb{Z}}$ be the canonical grid structure on $\mathbb{Z} \times \mathbb{Z}$:

$$\mathfrak{G}_{\mathbb{Z}} = (\mathbb{Z}^2, H, V),$$

$$H = \{((p, q), (p+1, q)) \colon p, q \in \mathbb{Z}\},$$
$$V = \{((p, q), (p, q+1)) \colon p, q \in \mathbb{Z}\},$$

and let $\mathfrak{G}_{\mathbb{N}}$ denote the standard grid on $\mathbb{N} \times \mathbb{N}$, which is just the restriction of $\mathfrak{G}_{\mathbb{Z}}$ to $\mathbb{N} \times \mathbb{N}$. Let $\mathfrak{G}_m$ denote the standard grid on a finite $m \times m$ torus:

$$\mathfrak{G}_m = (\mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}, H, V),$$

$$H = \{((p, q), (p', q)) \colon p' - p \equiv 1 \mod m\},$$
$$V = \{((p, q), (p, q')) \colon q' - q \equiv 1 \mod m\}.$$

Let $\mathfrak{A}_i = (A_i, H_i, V_i), i = 1, 2$. $\mathfrak{A}_1$ is *homomorphically embeddable* into $\mathfrak{A}_2$ if there is a homomorphism $\pi : \mathfrak{A}_1 \rightarrow \mathfrak{A}_2$, i.e., a mapping π such that for all $v, v' \in A_1$: $(v, v') \in H_1 \Rightarrow (\pi(v), \pi(v')) \in H_2$ and $(v, v') \in V_1 \Rightarrow (\pi(v), \pi(v')) \in V_2$. For instance, $\mathfrak{G}_{\mathbb{N}}$ is homomorphically embeddable into $\mathfrak{G}_{\mathbb{Z}}$, and both are homomorphically embeddable into every $\mathfrak{G}_m$.

In order to allow for a simultaneous reduction of the infinite and periodic tiling problems to satisfiability and finite satisfiability, one wants to work over a sufficiently rich class of grid-like structures in the following sense.

Definition 29 An infinite structure $\mathfrak{G} = (G, H, V)$ is called *grid-like* if $\mathfrak{G}_{\mathbb{N}}$ is homomorphically embeddable into $\mathfrak{G}$; a finite $\mathfrak{G}$ is grid-like if some $\mathfrak{G}_m$ is homomorphically embeddable into $\mathfrak{G}$.

A class $\mathcal{G}$ of grid-like structures is called *rich* if at least one of $\mathfrak{G}_{\mathbb{N}}$ or $\mathfrak{G}_{\mathbb{Z}} \in \mathcal{G}$, and if for all $n \geq 1$ there is some k such that $\mathfrak{G}_{k \cdot n} \in \mathcal{G}$.

The following Lemma gives a simple and sufficient local criterion for grid-likeness (cf. Lemma 2.4 in [26]). We say that H is *complete over* V in $\mathfrak{G} = (G, H, V)$ if $\mathfrak{G}$ satisfies

$$\forall x \forall y \forall x' \forall y' ((Hxy \wedge Vxx' \wedge Vyy') \rightarrow Hx'y').$$

Lemma 30 *Let $\mathfrak{G} = (G, H, V)$ satisfy the* FO^2*-axiom $\forall x(\exists y Hxy \wedge \exists y Vxy)$. If H is complete over V, then $\mathfrak{G}$ is grid-like.*

Lemma 31 *If a rich class $\mathcal{G}$ of grid-like structures is* FO^2*-axiomatisable over $\mathcal{C}$, then* $\mathrm{SAT}[\mathrm{FO}^2, \mathcal{C}]$ *and* $\mathrm{FINSAT}[\mathrm{FO}^2, \mathcal{C}]$ *are undecidable. In fact,* FO^2 *over $\mathcal{C}$ induces a conservative reduction class so that the complement of* $\mathrm{SAT}[\mathrm{FO}^2, \mathcal{C}]$ *is recursively inseparable from* $\mathrm{FINSAT}[\mathrm{FO}^2, \mathcal{C}]$.

See [4, 16, 3] for the undecidability and recursive inseparability of corresponding tiling problems, which serve as the natural intermediaries here, and [13] and Lemma 2.3 in [26] in particular for the last lemma.

For our present purposes it therefore suffices to show the following.

Proposition 32 *There is a rich class $\mathcal{G}$ of grid-like structures that is* FO^2*-axiomatisable over $\mathcal{EQ}[\tau_0; E_1, E_2, E_3]$. In fact there is an* FO^2*-sentence η such that*

(a) *$\mathfrak{G}_{\mathbb{N}}$ and all $\mathfrak{G}_{3m}$ for $m \geq 1$ admit expansions to $\mathcal{EQ}[\tau_0; E_1, E_2, E_3]$ models of η.*

(b) *Every $\mathcal{EQ}[\tau_0; E_1, E_2, E_3]$ model of η is grid-like.*

Proof Let us expand the standard grid $\mathfrak{G}_{\mathbb{N}}$ to the structure $\bar{\mathfrak{G}}_{\mathbb{N}}$ illustrated in Figure 4. Besides grid relations H, V (omitted from the picture) and equivalence relations E_i, $i = 1, 2, 3$, we use unary predicates H_i, V_i, for $i = 0, 1, 2$, which periodically mark rows and columns of the grid. Note that the whole expansion is periodic and that the size of all equivalence classes is bounded (a class consists of at most nine elements); these two facts are crucial for the proof.

We capture some properties of $\bar{\mathfrak{G}}_{\mathbb{N}}$ by the sentence η, observe that every $\mathfrak{G}_{3m}$ can also be expanded to a model of η and prove that every model of η which interprets E_1, E_2, E_3 as equivalence relations is grid-like.

The sentence η is the conjunction of the following formulae:

(1) The initial formulae

$$\exists x (H_0 x \wedge V_0 x),$$

$$\forall x (\exists y Hxy \wedge \exists y Vxy),$$

$$\forall x \Big(\big(\dot{\bigvee_{0 \leq i \leq 2}} H_i x \big) \wedge \big(\dot{\bigvee_{0 \leq i \leq 2}} V_i x \big) \Big).$$

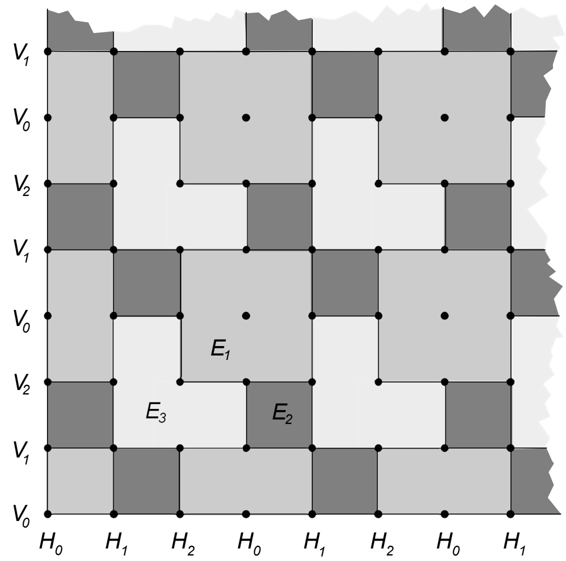

Figure 4: Expansion of the grid $\mathfrak{G}_{\mathbb{N}}$. Elements on the borders and inside the lightgray, darkgray and white areas form, respectively, E_1-, E_2- and E_3-classes.

(2) A formula axiomatising H, which has the following shape

$$\forall xy\ (Hxy \rightarrow (\bigvee_{0 \leq i,j \leq 2} \varphi_{ij}^{H})),$$

where φ_{ij}^{H} describes values of unary predicates and equivalences on H-related vertices $(3k+i, 3l+j)$ and $(3k+i+1, 3l+j)$, for $k, l \in \mathbb{N}$, e.g.,

$$\varphi_{00}^{H} \equiv E_1xy \wedge H_0x \wedge V_0x \wedge H_1y \wedge V_0y, \quad \text{and}$$
$$\varphi_{11}^{H} \equiv E_2xy \wedge E_3xy \wedge H_1x \wedge V_1x \wedge H_2y \wedge V_1y.$$

(3) A formula axiomatising V, which is built similarly to the one for H.

(4) A group of nine formulae (one for each combination of values of H_i, V_j on horizontally adjacent vertices) stating that some elements that are connected by one equivalence relation are linked by H. Sample formulae from this group are

$$\forall xy\ ((E_2xy \wedge H_1x \wedge V_1x \wedge H_2y \wedge V_1y) \rightarrow Hxy).$$

$$\forall xy\ ((E_3xy \wedge H_1x \wedge V_2x \wedge H_2y \wedge V_2y) \rightarrow Hxy).$$

It should be clear that $\bar{\mathfrak{G}}_{\mathbb{N}}$ is a model of η. In particular, note that equivalence relations connect only close elements, so (4) does not impose unwanted H-connections between elements which are distant from each other in the grid. It is also not hard to see that every $\mathfrak{G}_{3m}$ can be expanded to a model of η. In fact, a natural quotient of the above expansion of $\mathfrak{G}_{\mathbb{N}}$ serves this purpose.

We sketch the argument for grid-likeness of a model $\mathfrak{G} \models \eta$. By (1), $\mathfrak{G}$ satisfies $\forall x(\exists y Hxy \wedge \exists y Vxy)$ so we can use Lemma 30. We show that H is complete over V. Assume that for a, a', b, b'

$$\mathfrak{G} \models Hab \wedge Vaa' \wedge Vbb'.$$

We show that then $\mathfrak{G} \models Ha'b'$.

We have to consider nine cases distinguished by values of the H_i and V_i on a. Let us go through one of them, for instance, $\mathfrak{G} \models H_1a \wedge V_1a$.

By (2), we have

$$\mathfrak{G} \models H_2b \wedge V_1b \wedge E_2ab \wedge E_3ab.$$

Similarly (3) implies

$$\mathfrak{G} \models H_1a' \wedge V_2a' \wedge E_2aa' \wedge E_3aa' \quad \text{and} \quad \mathfrak{G} \models H_2b' \wedge V_2b' \wedge E_3bb'.$$

From transitivity of E_3 it follows that $\mathfrak{G} \models E_3a'b'$. Now an appropriate formula of the form (4) guarantees $\mathfrak{G} \models Ha'b'$, which finishes the proof for this case. The remaining eight cases can be treated in the same way. $\square$

Remark. The technique we used in the proof was provided in [26] for extensions of FO^2. It is not difficult to adapt it to extensions of the two-variable guarded fragment GF^2 without equality. All the formulae in our proof are essentially guarded. Moreover relations H and V do not play a crucial role. They can be defined in terms of E_1, E_2 and E_3 and extra unary predicates. This means that we can show Corollary 28. Similar result was obtained independently in [18]. For more about the satisfiability of the guarded fragment in restricted classes of models see [9, 31, 19, 20, 18, 22].

7 Conclusion

We considered the satisfiability problem for two-variable logic FO^2 in the class of structures in which some designated subset of the binary relation symbols are required to be interpreted as equivalence relations. We gave a complete classification regarding decidability with respect to the number of equivalence relations. If just one relation is required to be an equivalence relation, FO^2 retains its finite model property and its satisfiability problem is still decidable and NEXPTIME-complete. Over structures with two equivalence relations FO^2 does have infinity axioms, but the satisfiability problem is still decidable – and we give a 3NEXPTIME upper bound. Over structures with three equivalence relations the satisfiability problem for FO^2 becomes undecidable.

There is one particular auxiliary result of independent interest: an arbitrary substructure $\mathfrak{B}$ of a structure $\mathfrak{A}$ can be replaced by a substructure $\mathfrak{B}'$, whose size is exponentially bounded in the signature, in such a way that the new structure satisfies all normal form FO^2 sentences that are true in $\mathfrak{A}$.

Since FO^2 lacks the finite model property in the presence of two equivalence relations, it is natural to ask about satisfiability in finite models in this case. The finite satisfiability problem is shown to be decidable in [23]. That paper uses integer programming and a more involved analysis of models. Roughly speaking, the technique there consists in guessing a doubly exponential special fragment of a model (whereas our construction of π_0 is exponential in the number of colours, and thus triply exponential in the size of $|\varphi|$) and describing the remainder by a system of linear inequalities. The authors of [23] conjecture that, by some Carathéodory-like results on integer programming from [8], this approach can be extended to yield a 2NEXPTIME upper bound for both the satisfiability and the finite satisfiability problem. A corresponding lower bound has recently been obtained by Ian Pratt-Hartmann[8], using ideas similar to those from Section 5.1.2,

Acknowledgement. We would like to thank the anonymous referees for their comments and criticism of an earlier version, which helped to improve the presentation.

References

[1] H. Andréka, J. van Benthem and I. Németi, *Modal languages and bounded fragments of predicate logic*, Journal of Philosophical Logic, 27, 1998, 217-274.

[2] F. Baader, D. Calvanese, D. McGuinness, D. Nardi and P. Patel-Schneider (eds), *The Description Logic Handbook*, Cambridge University Press, 2003.

[3] E. Börger, E. Grädel and Y. Gurevich, *The Classical Decision Problem*, Springer, 1997.

[4] R. Berger, *The Undecidability Of The Domino Problem*, Mem. AMS, 66, 1966.

[5] M. Bojańczyk, C. David, A. Muscholl, T. Schwentick and L. Segoufin, *Two-variable logic on words with data*, Proc. of 21st IEEE Symp. on Logic in Computer Science, LICS 2006, 7-16.

[6] A. Borgida, *On the relative expressive power of description logics and predicate logic*, Artificial Intelligence, 82, 1996, 353-367.

[7] C. David, L. Libkin and T. Tan, *On the satisfiability of two-variable logic over data words*, Proc. of 17th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning, LPAR 2010, Lecture Notes in Computer Science 6397, 248-262.

[8] F. Eisenbrand and G. Shmonin, *Caratheodory bounds for integer cones*, Operations Research Letters, vol. 34, no. 5, 2006, 564-568.

[9] H. Ganzinger, C. Meyer and M. Veanes, *The two-variable guarded fragment with transitive relations*, Proc. of 14th IEEE Symp. on Logic in Computer Science, LICS 1999, 24-34.

[8] Private communication, 2011.

[10] W. Goldfarb, *The unsolvability of the Gödel's class with identity*, Journal of Symbolic Logic, 49, 1984, 1237-1252.

[11] E. Grädel, *On the restraining power of guards*, Journal of Symbolic Logic, 64, 1999, 1719-1742.

[12] E. Grädel, P. Kolaitis and M. Vardi, *On the decision problem for two-variable first order logic*, Bulletin of Symbolic Logic, 3, 1997, 53-96.

[13] E. Grädel and M. Otto, *On logics with two variables*, Theoretical Computer Science, 224, 1999, 77-113.

[14] E. Grädel, M. Otto and E. Rosen, *Two-variable logic with counting is decidable*, Proc. 12th IEEE Symp. on Logic in Computer Science, LICS 1997, 306-317.

[15] E. Grädel, M. Otto and E. Rosen, *Undecidability results on two-variable first-order logic*, Archive of Mathematical Logic, 38, 1999, 313-354.

[16] Y. Gurevich and I. Koryakov, *Remarks on Berger's paper on the domino problem*, Siberian Mathematical Journal, 13, 1972, 319–321.

[17] N. Immerman and E. Lander, *Describing graphs: a first-order approach to graph canonization*, in: A. Selman (ed), Complexity Theory Retrospective, Springer, 1990, 59-81.

[18] Y. Kazakov, *Saturation-based decision procedures for extensions of the guarded fragment*, PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 2006.

[19] E. Kieroński, *On the complexity of the two-variable guarded fragment with transitive guards*, Information and Computation, 204, 2006, 1663-1703.

[20] E. Kieroński, *Results on the guarded fragment with equivalence or transitive relations*, Proc. of Computer Science Logic, 19th International Workshop, 14th Annual Conference of the EACSL, CSL 2005, Lecture Notes in Computer Science 3634, 309-324.

[21] E. Kieroński and M. Otto, *Small substructures and decidability issues for first-order logic with two variables*, Proc. of 20th IEEE Symp. on Logic in Computer Science, LISC 2005, 448-457.

[22] E. Kieroński, L. Tendera, *On finite satisfiability of the guarded fragment with equivalence or transitive guards*, Proc. of 14th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning, LPAR 2007, Lecture Notes in Computer Science 4790, 318-322.

[23] E. Kieroński and L. Tendera, *On finite satisfiability of two-variable first order logic with equivalence relations.*, Proc. of 23rd IEEE Symp. on Logic in Computer Science, LICS 2009, 123-132.

[24] A. Manuel, *Two variables and two successors*, Proc. of 35th International Symposium on Mathematical Foundations of Computer Science, MFCS 2010, Lecture Notes in Computer Science 6281, 513-524.

[25] M. Mortimer, *On languages with two variables*, Zeitschr. f. Logik und Grundlagen d. Mathematik, 21, 1975, 135-140.

[26] M. Otto, *Two variable first-order logic over ordered domains*, Journal of Symbolic Logic, 66, 2001, 685-702.

[27] M. Niewerth and T. Schwentick, *Two-variable logic and key constraints on data words*, Proc. of 14th International Conference on Database Theory, ICDT 2011, 138-149.

[28] L. Pacholski, W. Szwast and L. Tendera, *Complexity of two-variable logic with counting*, Proc. of 12th IEEE Symp. on Logic in Computer Science, LICS 1997, 318-327.

[29] T. Schwentick and T. Zeume, *Two-variable logic with two order relations*, Proc. of Computer Science Logic, 24th International Workshop, 19th Annual Conference of the EACSL, CSL 2010, Lecture Notes in Computer Science 6247, 499-513.

[30] D. Scott, *A decision method for validity of sentences in two variables*, Journal of Symbolic Logic, 27, 1962, 377.

[31] W. Szwast and L. Tendera *On the decision problem for the guarded fragment with transitivity*, Proc. of 16th IEEE Symp. on Logic in Computer Science, LICS 2001, 147-156.