

Living without Beth and Craig: Explicit Definitions and Interpolants in the Guarded Fragment

Jean Christoph Jung¹ and Frank Wolter²

¹University of Bremen, Germany

²University of Liverpool, UK

Abstract. The guarded fragment of FO fails to have the Craig Interpolation Property (CIP) and the Projective Beth Definability Property (PBDP). Thus, not every valid implication between guarded formulas has a guarded interpolant, and not every implicitly definable relation has an explicit guarded definition. In this article, we show that nevertheless the existence of guarded interpolants and explicit definitions is decidable. Moreover, it is 3EXPTIME-complete in general, and 2EXPTIME-complete if the arity of relation symbols is bounded by a constant $c \geq 3$. Deciding the existence of guarded interpolants and explicit definitions is thus by one exponential harder than validity in the guarded fragment.

1 Introduction

A logic enjoys the *Craig Interpolation Property* (CIP) if an implication $\varphi \Rightarrow \psi$ is valid if, and only if, there exists a formula χ using only the common symbols of φ and ψ such that $\varphi \Rightarrow \chi$ and $\chi \Rightarrow \psi$ are both valid. The formula χ is then called an interpolant for $\varphi \Rightarrow \psi$. The CIP is generally regarded as one of the most important and useful results in formal logic, with numerous applications [36,33,13,15]. One particularly interesting consequence of the CIP is the *Projective Beth Definability Property* (PBDP), which states that if a relation is implicitly definable over symbols in a signature τ , then it is explicitly definable over τ .

From an algorithmic viewpoint, the CIP and BDP are of interest because they translate existence problems to validity: an interpolant *exists* if, and only if, an implication is valid and an explicit definition *exists* if, and only if, a straightforward formula stating implicit definability is valid. The existence problems are thus not harder than validity.

In this article, we investigate the existence of interpolants and explicit definitions in a language that fails to have the CIP and PBDP, the guarded fragment (GF) of first-order logic (FO). GF has been introduced as a powerful generalization of modal logic that enjoys many of its attractive algorithmic and model-theoretic properties, including decidability, the finite model property, the tree-like model property, and preservation properties such as the Los-Tarski preservation theorem [1,16]. Since its introduction, the guarded fragment and variants of

it have been investigated extensively [19,17,34,6], not only as a natural generalisation of modal logic but also in databases and knowledge representation [7,20].

While GF is a “good” generalization of modal logic in many respects, it has at least one undesirable property: in contrast to standard modal logic, GF does not enjoy the CIP [23] nor the PBDP [4].¹ Our motivation for investigating the existence of interpolants and explicit definitions in GF despite the fact that GF does not enjoy the CIP nor PBDP stems from the following applications:

Strong separability of labeled data under GF-ontologies. There are several scenarios in which one aims to find a logical formula that separates positive from negative examples given in the form of labeled data items. Examples include concept learning in description logic [29], reverse engineering of database queries, also known as query by example (QBE) [31], and generating referring expression (GRE), where the aim is to find a formula that separates a single positive data item from all other data items [28]. In [26,25] an attempt is made to provide a unifying framework for these scenarios under the assumption that the data is given by a relational database and additional background information is available in the form of an ontology in first-order logic. A natural version of separability then asks whether for a first-order theory \mathcal{O} , a set of ground atoms \mathcal{D} , a signature τ of relation symbols, and sets P (of positive examples) and N (of negative examples) of tuples in \mathcal{D} of the same length whether there exists a formula φ over τ that separates P from N in the sense that $\mathcal{O} \cup \mathcal{D} \models \varphi(\mathbf{a})$ for all $\mathbf{a} \in P$ and $\mathcal{O} \cup \mathcal{D} \models \neg\varphi(\mathbf{b})$, for all $\mathbf{b} \in N$. For the fundamental case that \mathcal{O} is in GF and one asks for a separating formula in GF it is not difficult to see that there is a polynomial time reduction of separability to interpolant existence in GF. Moreover, interpolants give rise to separating formulas and vice versa.

Explicit definitions of relation symbols under GF-sentences. The existence and computation of explicit definitions of relations in GF has been proposed for ontology engineering [11] and for query rewriting under views and query-reformulation and compilation [32,35,9]. Thus, in these applications the focus shifts from interpolants to the existence of definitions over a signature. It is well known that the latter can be obtained from the former.

Our results are as follows. We show that interpolant and explicit definition existence in GF are both 3EXPTIME-complete, thus one exponential harder than validity in GF [16]. We also show that both are 2EXPTIME-complete if the arity of relation symbols is bounded by a constant $c \geq 3$, again one exponential harder than validity in GF [16] with a bound on the arity of relation symbols. We note that exactly one ternary relation symbol is needed to obtain 2EXPTIME-hardness and that it is known from [22] that the fragment of GF with only unary and binary relation symbols enjoys the CIP and the PBDP and so interpolant and explicit definition existence are EXPTIME-complete in that case.

¹ It should be noted that GF still has the Beth Definability Property (BDP) in which the signature τ of the implicit and explicit definitions contains all symbols except the relation that is defined [23].

The upper bound proofs start with a standard characterization of the non-existence of interpolants by the existence of certain guarded bisimulations between models. Guarded bisimulations were introduced in [1] as a generalization of bisimulations to characterize the expressive power of GF within FO, see also [18]. We then employ a mosaic based approach, using as mosaics sets of types over the input formulas which can be satisfied by tuples that are guarded bisimilar. Constraints for sets of such mosaics characterize when they can be linked together to construct, simultaneously, models of the input formulas and a guarded bisimulation between them. The triple exponential upper bound then follows from the observation that there are triple exponentially many mosaics. If the arity of relation symbols is bounded by a constant, then there are only double exponentially many mosaics. The lower bounds are proved by a reduction of the word problem for space-bounded alternating Turing machines.

2 Related Work

Over the past 20 years, a lot of progress has been made in understanding the CIP and PBDP for guarded fragments of FO. As mentioned above, the first results on GF itself and various fragments of GF are reported in [21,23,22]. In particular, after proving that GF fails to have the CIP it is shown that a natural modal version of the CIP holds for GF: if $\varphi \models \psi$ for guarded formulas φ, ψ , then there exists an interpolant for φ, ψ that may use in addition to the symbols shared by φ and ψ any relation symbol that occurs as a guard in φ or ψ .

More recently, the guarded negation fragment of FO (GNF) has been introduced. GNF extends GF by adding, in a careful way, unions of conjunctive queries [6]. Although GNF extends GF significantly, it is still decidable, has the finite model property, has the tree-like model property and enjoys various preservation theorems [6,5]. Importantly, and in contrast to GF, GNF enjoys the CIP and the PBDP [5,8]. Thus, the existence of Craig interpolants and explicit definitions reduces to validity checking which is 2EXPTIME-complete in GNF and even in EXPTIME if the arity of relation symbols is bounded by a constant. It follows that the existence of interpolants and explicit definitions is one exponential harder in GF than in GNF.

Also related is work on uniform interpolation for GF. As GF does not enjoy the CIP, it also does not enjoy the uniform interpolation property (UIP). However, in [14], the authors consider the same modal-like fragment as [22] and show that the CIP generalizes to the UIP for this fragment. Uniform interpolant existence for GF has been considered in [24]. In contrast to the decidability results obtained in this article, it is shown that uniform interpolant existence is undecidable for GF.

Also relevant for this work is the investigation of interpolation and definability in modal logic in general [30] and in hybrid modal logic in particular [2,10].

3 Preliminaries

Let τ range over relational signatures not containing function or constant symbols. Denote by $\text{FO}(\tau)$ the set of first-order (FO) formulas constructed from atomic formulas $x = y$ and $R(\mathbf{x})$, $R \in \tau$, using conjunction, disjunction, negation, and existential and universal quantification. As usual, we write $\varphi(\mathbf{x})$ to indicate that the free variables in FO-formula φ are all from \mathbf{x} and call a formula *open* if it has at least one free variable and a *sentence* otherwise. $\text{FO}(\tau)$ is interpreted in τ -structures $\mathfrak{A} = (\text{dom}(\mathfrak{A}), (R^{\mathfrak{A}})_{R \in \tau})$, where $\text{dom}(\mathfrak{A})$ is the non-empty *domain* of \mathfrak{A} , and each $R^{\mathfrak{A}}$ is a relation over $\text{dom}(\mathfrak{A})$ whose arity matches that of R . We often drop τ and simply speak of structures \mathfrak{A} . In the *guarded fragment* (GF) of FO [1,16], formulas are built from atomic formulas $R(\mathbf{x})$ and $x = y$ by applying the Boolean connectives and *guarded quantifiers* of the form

$$\forall \mathbf{y}(\alpha(\mathbf{x}, \mathbf{y}) \rightarrow \varphi(\mathbf{x}, \mathbf{y})) \text{ and } \exists \mathbf{y}(\alpha(\mathbf{x}, \mathbf{y}) \wedge \varphi(\mathbf{x}, \mathbf{y}))$$

where $\varphi(\mathbf{x}, \mathbf{y})$ is a guarded formula and $\alpha(\mathbf{x}, \mathbf{y})$ is an atomic formula that contains all variables in \mathbf{x}, \mathbf{y} . The formula α is called the *guard of the quantifier*. $\text{GF}(\tau)$ denotes the set of all guarded formulas (also called GF-formulas) over the signature τ . The signature $\text{sig}(\varphi)$ of a formula φ is the set of relation symbols used in it.

Let \mathfrak{A} be structure. It will be convenient to use the notation $[\mathbf{a}] = \{a_1, \dots, a_n\}$ to denote the set of components of the tuple $\mathbf{a} = (a_1, \dots, a_n) \in \text{dom}(\mathfrak{A})^n$. Similarly, for a tuple $\mathbf{x} = (x_1, \dots, x_n)$ of variables we use $[\mathbf{x}]$ to denote the set $\{x_1, \dots, x_n\}$.

4 Interpolants and Explicit Definitions in the Guarded Fragment

We introduce GF-interpolants and explicit GF-definitions and provide model-theoretic characterizations of the existence of GF-interpolants and explicit GF-definitions using guarded bisimulations.

Let $\varphi(\mathbf{x}), \psi(\mathbf{x})$ be GF-formulas with the same free variables \mathbf{x} . We call a GF-formula $\theta(\mathbf{x})$ a *GF-interpolant for φ, ψ* if $\text{sig}(\theta) \subseteq \text{sig}(\varphi) \cap \text{sig}(\psi)$, $\varphi(\mathbf{x}) \models \theta(\mathbf{x})$, and $\theta(\mathbf{x}) \models \psi(\mathbf{x})$. We are interested in *GF-interpolant existence*, the problem to decide for given $\varphi(\mathbf{x}), \psi(\mathbf{x})$ in GF whether a GF-interpolant for $\varphi(\mathbf{x}), \psi(\mathbf{x})$ exists.

In order to provide a model-theoretic characterization of when an interpolant exists, we introduce *guarded τ -bisimulations* [18]. A set $G \subseteq \text{dom}(\mathfrak{A})$ is *guarded* in \mathfrak{A} if G is a singleton or there exists R with $\mathfrak{A} \models R(\mathbf{a})$ such that $G = [\mathbf{a}]$. A tuple $\mathbf{a} \in \text{dom}(\mathfrak{A})^n$ is *guarded* in \mathfrak{A} if $[\mathbf{a}]$ is a subset of some guarded set in \mathfrak{A} .

For tuples $\mathbf{a} = (a_1, \dots, a_n)$ in \mathfrak{A} and $\mathbf{b} = (b_1, \dots, b_n)$ in \mathfrak{B} we call a mapping p from $[\mathbf{a}]$ to $[\mathbf{b}]$ with $p(a_i) = b_i$ for $1 \leq i \leq n$ (written $p : \mathbf{a} \mapsto \mathbf{b}$) a *partial τ -isomorphism* if p is an isomorphism from the τ -reduct of $\mathfrak{A}|_{[\mathbf{a}]}$ onto $\mathfrak{B}|_{[\mathbf{b}]}$.

A set I of partial τ -isomorphisms $p : \mathbf{a} \mapsto \mathbf{b}$ from guarded tuples \mathbf{a} in \mathfrak{A} to guarded tuples \mathbf{b} in \mathfrak{B} is called a *guarded τ -bisimulation* if the following hold for all $p : \mathbf{a} \mapsto \mathbf{b} \in I$:

- (i) for every guarded tuple \mathbf{a}' in \mathfrak{A} there exists a guarded tuple \mathbf{b}' in \mathfrak{B} and $p' : \mathbf{a}' \mapsto \mathbf{b}' \in I$ such that p' and p coincide on $[\mathbf{a}] \cap [\mathbf{a}']$.
- (ii) for every guarded tuple \mathbf{b}' in \mathfrak{B} there exists a guarded tuple \mathbf{a}' in \mathfrak{A} and $p' : \mathbf{a}' \mapsto \mathbf{b}' \in I$ such that p'^{-1} and p^{-1} coincide on $[\mathbf{b}] \cap [\mathbf{b}']$.

A pair \mathfrak{A}, \mathbf{a} with \mathbf{a} a tuple in \mathfrak{A} is called a *pointed structure*. Assume that \mathbf{a} and \mathbf{b} are (possibly not guarded) tuples in \mathfrak{A} and \mathfrak{B} . Then we say that the pointed structures \mathfrak{A}, \mathbf{a} and \mathfrak{B}, \mathbf{b} are *guarded τ -bisimilar*, in symbols $\mathfrak{A}, \mathbf{a} \sim_{\text{GF}, \tau} \mathfrak{B}, \mathbf{b}$, if there exists a partial τ -isomorphism $p : \mathbf{a} \mapsto \mathbf{b}$ and a guarded τ -bisimulation I such that Conditions (i) and (ii) hold for p . We write $\mathfrak{A}, \mathbf{a} \equiv_{\text{GF}, \tau} \mathfrak{B}, \mathbf{b}$ and call \mathfrak{A}, \mathbf{a} and \mathfrak{B}, \mathbf{b} *GF(τ)-equivalent* if $\mathfrak{A} \models \varphi(\mathbf{a})$ iff $\mathfrak{B} \models \varphi(\mathbf{b})$ holds for all formulas φ in $\text{GF}(\tau)$. The following equivalences are well known [18].

Lemma 1. *Let \mathfrak{A}, \mathbf{a} and \mathfrak{B}, \mathbf{b} be pointed structures and τ a signature. Then*

$$\mathfrak{A}, \mathbf{a} \sim_{\text{GF}, \tau} \mathfrak{B}, \mathbf{b} \quad \text{implies} \quad \mathfrak{A}, \mathbf{a} \equiv_{\text{GF}, \tau} \mathfrak{B}, \mathbf{b}$$

and, conversely, if \mathfrak{A} and \mathfrak{B} are ω -saturated, then

$$\mathfrak{A}, \mathbf{a} \equiv_{\text{GF}, \tau} \mathfrak{B}, \mathbf{b} \quad \text{implies} \quad \mathfrak{A}, \mathbf{a} \sim_{\text{GF}, \tau} \mathfrak{B}, \mathbf{b}$$

We are now in the position to characterize the existence of GF-interpolants. Call GF-formulas $\varphi(\mathbf{x}), \psi(\mathbf{x})$ *jointly GF(τ)-consistent* if there exist pointed models \mathfrak{A}, \mathbf{a} and \mathfrak{B}, \mathbf{b} with $\mathfrak{A} \models \varphi(\mathbf{a})$ and $\mathfrak{B} \models \psi(\mathbf{b})$ such that $\mathfrak{A}, \mathbf{a} \sim_{\text{GF}, \tau} \mathfrak{B}, \mathbf{b}$.

Lemma 2. *Let $\varphi(\mathbf{x}), \psi(\mathbf{x})$ be GF-formulas and let $\tau = \text{sig}(\varphi) \cap \text{sig}(\psi)$. Then the following conditions are equivalent:*

1. *there does not exist a GF-interpolant for $\varphi(\mathbf{x}), \psi(\mathbf{x})$;*
2. *$\varphi(\mathbf{x}), \neg\psi(\mathbf{x})$ are jointly GF(τ)-consistent.*

Proof. (\Leftarrow) Assume there is an interpolant $\theta(\mathbf{x})$ and let $\mathfrak{A}, \mathfrak{B}$ be structures and \mathbf{a}, \mathbf{b} be tuples such that $\mathfrak{A} \models \varphi(\mathbf{a})$ and $\mathfrak{B} \models \neg\psi(\mathbf{b})$. Suppose further that $\mathfrak{A}, \mathbf{a} \sim_{\text{GF}, \tau} \mathfrak{B}, \mathbf{b}$. Since $\varphi(\mathbf{x}) \models \theta(\mathbf{x})$, we have $\mathfrak{A} \models \theta(\mathbf{a})$. By Lemma 1, we obtain $\mathfrak{B} \models \theta(\mathbf{b})$. Finally, as $\theta(\mathbf{x}) \models \psi(\mathbf{x})$, we obtain $\mathfrak{B} \models \psi(\mathbf{b})$, a contradiction.

(\Rightarrow) Suppose that for all structures $\mathfrak{A}, \mathfrak{B}$ and tuples \mathbf{a}, \mathbf{b} such that $\mathfrak{A} \models \varphi(\mathbf{a})$ and $\mathfrak{B} \models \neg\psi(\mathbf{b})$ we have $\mathfrak{A}, \mathbf{a} \not\sim_{\text{GF}, \tau} \mathfrak{B}, \mathbf{b}$. Let Φ be defined by taking

$$\Phi = \{\varphi'(\mathbf{x}) \in \text{GF}(\tau) \mid \varphi(\mathbf{x}) \models \varphi'(\mathbf{x})\}$$

Clearly, $\varphi(\mathbf{x}) \models \Phi$. We claim that also $\Phi \models \psi(\mathbf{x})$. To see this, let \mathfrak{B}, \mathbf{b} such that $\mathfrak{B} \models \Phi(\mathbf{b})$. Let \mathfrak{B}' be an ω -saturated extension of \mathfrak{B} and let \mathfrak{A}, \mathbf{a} be an ω -saturated pointed structure realizing $\{\chi(\mathbf{x}) \in \text{GF}(\tau) \mid \mathfrak{B} \models \chi(\mathbf{b})\} \cup \{\varphi\}$ in \mathbf{a} . By definition of Φ and Lemma 1, we have $\mathfrak{A}, \mathbf{a} \sim_{\text{GF}, \tau} \mathfrak{B}', \mathbf{b}$. By the initial assumption, we cannot have $\mathfrak{B}' \models \neg\psi(\mathbf{b})$ and thus $\mathfrak{B} \models \psi(\mathbf{b})$. By compactness, there is a finite subset Φ' of Φ such that $\Phi' \models \psi(\mathbf{x})$. The conjunction of the formulas in Φ' is the required interpolant. \square

Let φ be a GF-sentence, $\theta(\mathbf{x})$ a GF-formula, and τ a signature. A GF(τ)-formula $\psi(\mathbf{x})$ is an *explicit GF(τ)-definition of θ under φ* if $\varphi \models \forall \mathbf{x}(\theta(\mathbf{x}) \leftrightarrow \psi(\mathbf{x}))$. We call θ *explicitly GF(τ)-definable under φ* if such an explicit GF(τ)-definition of θ under φ exists. We call θ *implicitly GF(τ)-definable under φ* if $\varphi \wedge \varphi' \models \forall \mathbf{x}(\theta(\mathbf{x}) \leftrightarrow \theta'(\mathbf{x}))$, where φ' and θ' are obtained from φ and θ , respectively, by renaming all non- τ symbols R to fresh R' of the same arity. Obviously, explicit GF(τ)-definability implies implicit GF(τ)-definability. However, as GF does not have the projective Beth definability property, the converse implication does not hold [23].

We consider the problem of *explicit GF-definability*, that is, the problem to decide for given $\varphi, \theta(\mathbf{x}), \tau$ whether there is an explicit GF(τ)-definition of $\theta(\mathbf{x})$ under φ . We first observe that explicit definition existence reduces to interpolant existence.

Lemma 3. *There is a polynomial time reduction of explicit GF-definability to GF-interpolant existence.*

Proof. Assume $\varphi, \theta(\mathbf{x})$, and τ are given. Then $\theta(\mathbf{x})$ is explicitly definable under φ iff there exists a GF-interpolant for $\varphi \wedge \theta(\mathbf{x}), \varphi' \rightarrow \theta'(\mathbf{x})$, where φ' and θ' are obtained from φ and θ , respectively, by renaming all non- τ symbols R to fresh R' of the same arity. \square

Lemma 3 suggests that there is a characterization of explicit definability in terms of joint GF(τ)-consistency as well. Indeed, we give this characterization next.

Lemma 4. *For every GF-sentence φ , every GF-formula $\theta(\mathbf{x})$, and signature τ , the following conditions are equivalent:*

1. *there does not exist an explicit GF(τ)-definition of $\theta(\mathbf{x})$ under φ ;*
2. *$\varphi \wedge \theta(\mathbf{x})$ and $\varphi \wedge \neg\theta(\mathbf{x})$ are jointly GF(τ)-consistent.*

We aim to prove the following result.

Theorem 1. *The explicit GF-definability and the GF-interpolant existence problems are both 3EXPTIME-complete in general, and 2EXPTIME-complete if the arity of relation symbols is bounded by a constant $c \geq 3$.*

As Lemma 2 provides a reduction of the complement of GF-interpolant existence to joint GF(τ)-consistency, that is, the problem of deciding whether given $\varphi(\mathbf{x}), \psi(\mathbf{x})$ are jointly GF(τ)-consistent, we will prove the complexity upper bounds for the latter problem. For the complexity lower bounds, we will also consider joint GF(τ)-consistency, but for an input of the form given in Lemma 4.

5 Deciding Joint GF(τ)-Consistency: Upper Bounds

To decide joint GF(τ)-consistency we pursue a mosaic approach based on types. Throughout the section, let $\varphi(\mathbf{x}_0), \psi(\mathbf{x}_0)$ be the input to joint GF(τ)-consistency,

for some signature τ . Let $\Xi = \{\varphi(\mathbf{x}_0), \psi(\mathbf{x}_0)\}$. Let $\text{width}(\Xi)$ denote the maximal arity of any relation symbol used in Ξ and let $\text{fv}(\Xi)$ be the number of free variables in Ξ . Let x_1, \dots, x_{2n} be fresh variables, where $n := \max\{\text{width}(\Xi), \text{fv}(\Xi)\}$. We use $\text{cl}(\Xi)$ to denote the smallest set of GF-formulas that is closed under taking subformulas and single negation, and contains:

- Ξ ,
- all formulae $x = y$ for distinct variables x, y ;
- all formulae $\exists \mathbf{x} R(\mathbf{x}\mathbf{y})$, where R is a relation symbol that occurs in Ξ and $\mathbf{x}\mathbf{y}$ is a tuple of variables.

Let \mathfrak{A} be a structure, \mathbf{a} a tuple of distinct elements from the domain of \mathfrak{A} , and \mathbf{x} a tuple of distinct variables in $\{x_1, \dots, x_{2n}\}$ of the same length as \mathbf{a} . Consider the bijection $\sigma : \mathbf{x} \mapsto \mathbf{a}$. Then the Ξ -type of \mathbf{a} in \mathfrak{A} defined through σ is

$$\text{tp}(\mathfrak{A}, \sigma : \mathbf{x} \mapsto \mathbf{a}) = \{\theta \mid \mathfrak{A} \models_{\sigma} \theta, \theta \in \text{cl}(\Xi)[\mathbf{x}]\},$$

where $\text{cl}(\Xi)[\mathbf{x}]$ is obtained from $\text{cl}(\Xi)$ by substituting in any formula $\theta \in \text{cl}(\Xi)$ the free variables of θ by variables in $[\mathbf{x}]$ in all possible ways. Note that the assumption that σ is bijective entails that $\neg(x = y) \in \text{tp}(\mathfrak{A}, \sigma : \mathbf{x} \mapsto \mathbf{a})$ for any two distinct $x, y \in [\mathbf{x}]$. We drop σ (and both σ and \mathbf{x}) and write $\text{tp}(\mathfrak{A}, \mathbf{x} \mapsto \mathbf{a})$ (and $\text{tp}(\mathfrak{A}, \mathbf{a})$, respectively), whenever they are obvious from the context. Any Ξ -type of some \mathbf{a} through some $\sigma : \mathbf{x} \mapsto \mathbf{a}$ is called a Ξ -type and simply denoted $t(\mathbf{x})$. The set of all Ξ -types is denoted $T(\Xi)$.

To decide joint GF(τ)-consistency of $\varphi(\mathbf{x}_0), \psi(\mathbf{x}_0)$ we determine all sets $\Phi \subseteq T(\Xi)$ using at most n variables from $\{x_1, \dots, x_{2n}\}$ that can be satisfied in guarded τ -bisimilar models in the following sense: there are models \mathfrak{A}_t , $t \in \Phi$, realizing t in tuples \mathbf{a}_t through assignments σ_t that are mutually guarded τ -bisimilar on the images of shared variables between types. Such sets Φ will be called τ -mosaics and are the main ingredient of our approach. We can check whether $\varphi(\mathbf{x}_0), \psi(\mathbf{x}_0)$ are jointly GF(τ)-consistent by simply checking whether there are types $t_1(\mathbf{x}), t_2(\mathbf{x})$ in a single Φ such that we can replace the variables \mathbf{x}_0 by variables in $[\mathbf{x}]$ and $\varphi' \in t_1(\mathbf{x}_1)$, $\psi' \in t_2(\mathbf{x}_2)$ for the resulting formulas φ', ψ' . Thus, in what follows we aim to determine the characteristic properties of τ -mosaics and show that they can be enumerated in triple exponential time in general. If $\text{width}(\Xi)$ is fixed, we perform a closer analysis of the set of mosaics and show that double exponential time is sufficient.

To formulate the characteristic properties of τ -mosaics, we require some notation. The *restriction* $t(\mathbf{x})|_X$ of a Ξ -type $t(\mathbf{x})$ to a set X of variables is the set of $\theta \in t(\mathbf{x})$ with free variables among X . The *restriction* $\Phi|_X$ of a set Φ of Ξ -types to X is defined as $\{t(\mathbf{x})|_X \mid t(\mathbf{x}) \in \Phi\}$. Types $t(\mathbf{x})$ and $t'(\mathbf{x}')$ *coincide on* X if $t(\mathbf{x})|_X = t'(\mathbf{x}')|_X$ and sets Φ, Φ' of Ξ -types *coincide on* X if $\Phi|_X = \Phi'|_X$. A variable x is *free* in a Φ if Φ contains a type in which x is free.

A formula $Q(\mathbf{x})$ of the form $x = x$ or $\exists \mathbf{y} R(\mathbf{x}\mathbf{y})$ with $R \in \tau$ is called a τ -guard (for \mathbf{x}). It is called a *strict τ -guard* if it is of the form $x = x$ or \mathbf{y} is empty, respectively. We call a set $\Phi \subseteq T(\Xi)$ a τ -mosaic if it satisfies the following conditions:

- Φ is τ -uniform: for all τ -guards $Q(\mathbf{z})$ and all $t(\mathbf{x}), s(\mathbf{y}) \in \Phi$ with $[\mathbf{z}] \subseteq [\mathbf{x}] \cap [\mathbf{y}]$, $Q(\mathbf{z}) \in t(\mathbf{x})$ iff $Q(\mathbf{z}) \in s(\mathbf{y})$;
- *closed under restrictions*: if $t(\mathbf{x}) \in \Phi$ and $X \subseteq [\mathbf{x}]$, then $t(\mathbf{x})|_X \in \Phi$;
- τ -bisimulation saturated: for all $t(\mathbf{x}) \in \Phi$, all strict τ -guards $Q(\mathbf{y}) \in t(\mathbf{x})$, and all $t'(\mathbf{z}) \in \Phi$ with $[\mathbf{z}] \subseteq [\mathbf{y}]$, there is an $s(\mathbf{y}') \in \Phi$ such that $t'(\mathbf{z}) \subseteq s(\mathbf{y}')$ and $[\mathbf{y}'] = [\mathbf{y}]$.

Intuitively, τ -uniformity reflects that guarded τ -bisimulations preserve all τ -guards and τ -bisimulation saturatedness reflects the first condition for guarded τ -bisimulations. In addition to the properties above, we have to ensure that τ -mosaics can be linked together. The next two conditions state when this is the case. We say that τ -mosaics Φ_1, Φ_2 are *compatible* if for $\{i, j\} = \{1, 2\}$:

1. for every $t(\mathbf{x}) \in \Phi_i$ there is an $s(\mathbf{y}) \in \Phi_j$ such that $t(\mathbf{x})$ and $s(\mathbf{y})$ coincide on $[\mathbf{x}] \cap [\mathbf{y}]$;
2. if there are $t(\mathbf{x}) \in \Phi_i$ and $s(\mathbf{y}) \in \Phi_j$ and a τ -guard $Q(\mathbf{z}) \in t(\mathbf{x})$ with $[\mathbf{z}] \subseteq [\mathbf{x}] \cap [\mathbf{y}]$, then Φ_i and Φ_j coincide on $[\mathbf{z}]$.

Note that compatibility is a reflexive and symmetric relation. Let \mathcal{M} be a set of τ -mosaics. We call $\Phi \in \mathcal{M}$ *existentially saturated in \mathcal{M}* if for every $t(\mathbf{x}) \in \Phi$ and every formula $\exists \mathbf{y}(R(\mathbf{x}', \mathbf{y}) \wedge \lambda(\mathbf{x}', \mathbf{y})) \in t(\mathbf{x})$ there is a some $\Phi' \in \mathcal{M}$ such that Φ, Φ' are compatible and $R(\mathbf{x}', \mathbf{y}') \wedge \lambda(\mathbf{x}', \mathbf{y}') \in t'(\mathbf{z})$ for some $t'(\mathbf{z}) \in \Phi'$ which coincides with $t(\mathbf{x})$ on $[\mathbf{x}] \cap [\mathbf{z}]$. \mathcal{M} is called *good* if every $\Phi \in \mathcal{M}$ is existentially saturated in \mathcal{M} .

Lemma 5. *Assume \mathcal{M} is good and let $t_1(\mathbf{x}_1), t_2(\mathbf{x}_2) \in \Psi \in \mathcal{M}$. Then there are pointed models $\mathfrak{A}_1, \mathbf{a}_1$ and $\mathfrak{A}_2, \mathbf{a}_2$ and $\sigma_i : \mathbf{x}_i \mapsto \mathbf{a}_i$ such that $\mathfrak{A}_i \models t_i(\mathbf{a}_i)$, $i = 1, 2$, and for $Y = [\mathbf{x}_1] \cap [\mathbf{x}_2]$, $\mathfrak{A}_1, \sigma_1(\mathbf{x}_1|_Y) \sim_{GF, \tau} \mathfrak{A}_2, \sigma_2(\mathbf{x}_2|_Y)$.*

Proof. Let $\Psi \in \mathcal{M}$. We assume w.l.o.g. that \mathcal{M} is *closed under restrictions* in the sense that for any $\Phi \in \mathcal{M}$ and subset X of the free variables of Φ , $\Phi|_X \in \mathcal{M}$. (If it is not closed under restrictions simply add all $\Phi|_X$ with $\Phi \in \mathcal{M}$ to \mathcal{M} . The resulting set is still good.) Define $\widehat{\Psi} := \Psi|_{\emptyset}$, that is, $\widehat{\Psi}$ contains all Ξ -types in Ψ without free variables. By closure under restrictions of \mathcal{M} , we have $\widehat{\Psi} \in \mathcal{M}$. Assume $\widehat{\Psi} = \{\hat{t}_1, \dots, \hat{t}_m\}$. We construct structures \mathfrak{A}_i , $i = 1, \dots, m$, with \mathfrak{A}_i satisfying \hat{t}_i . For the construction, it is useful to employ notation for tree decompositions. A *tree decomposition* of a structure \mathfrak{A} is a triple (T, E, bag) with (T, E) a tree and bag a function that assigns to every $t \in T$ a set $\text{bag}(t) \subseteq \text{dom}(\mathfrak{A})$ such that

1. $\mathfrak{A} = \bigcup_{t \in T} \mathfrak{A}|_{\text{bag}(t)}$;
2. $\{t \in T \mid a \in \text{bag}(t)\}$ is connected in (T, E) , for every $a \in \text{dom}(\mathfrak{A})$.

We construct the structures \mathfrak{A}_i , $i = 1, \dots, m$ by giving a tree decomposition (T_i, E_i, bag_i) of \mathfrak{A}_i . To this end, we define (T_i, E_i, bag_i) and structures $\text{Bag}_i(t)$ with domain $\text{bag}_i(t)$, $t \in T_i$, and then show that (T_i, E_i, bag_i) is a tree decomposition of the union \mathfrak{A}_i of all $\text{Bag}_i(t)$, $t \in T_i$. We start with the definition of (T_i, E_i) . Let T_i be the set of all sequences

$$\sigma_n = (t_0(\mathbf{y}_0), \Phi_0), \dots, (t_n(\mathbf{y}_n), \Phi_n)$$

such that $t_0 = \hat{t}_i$, $\Phi_0 = \hat{\Psi}$, $t_j(\mathbf{y}_j) \in \Phi_j$ for all $j \leq n$, and for all $j < n$:

- Φ_j, Φ_{j+1} are compatible, and
- $t_j(\mathbf{y}_j)$ and $t_{j+1}(\mathbf{y}_{j+1})$ coincide on $[\mathbf{y}_j] \cap [\mathbf{y}_{j+1}]$.

Let E_i be the induced prefix-order on T_i . We call $(t_n(\mathbf{y}_n), \Phi_n)$ the *tail* of σ_n . It remains to define the functions bag_i and Bag_i . We give an inductive definition with the aim to achieve the following: for all $\sigma_n \in T_i$ of the form above the Ξ -type $t_n(\mathbf{y}_n)$ is satisfied in \mathfrak{A}_i under a canonical assignment v_{σ_n} into the set $\text{bag}_i(\sigma_n)$. For the construction, it is important to note that we have $\neg(x = y) \in t$ for any two distinct free variables x, y in any Ξ -type t . Thus we can essentially use (copies of) the variables \mathbf{y}_n to define $\text{bag}_i(\sigma_n)$.

For the inductive definition, start by setting $\text{bag}_i(\sigma_0) = \emptyset$ and $v_{\sigma_0} = \emptyset$ for $\sigma_0 = (\hat{t}_i, \Phi_0)$. In the inductive step, assume that bag_i , $v_{\sigma_{n-1}}$, and Bag_i have been defined on σ_{n-1} , where

$$\sigma_{n-1} = (t_0(\mathbf{y}_0), \Phi_0), \dots, (t_{n-1}(\mathbf{y}_{n-1}), \Phi_{n-1}).$$

Then $\text{bag}_i(\sigma_n)$ contains

- fresh copies y' of the variables $y \in [\mathbf{y}_n] \setminus [\mathbf{y}_{n-1}]$ and
- $v_{\sigma_{n-1}}(y)$ for every $y \in [\mathbf{y}_n] \cap [\mathbf{y}_{n-1}]$,

and $v_{\sigma_n}(y)$ is defined as the copy y' of y for $y \in [\mathbf{y}_n] \setminus [\mathbf{y}_{n-1}]$ and by setting $v_{\sigma_n}(y) := v_{\sigma_{n-1}}(y)$ for $y \in [\mathbf{y}_n] \cap [\mathbf{y}_{n-1}]$. Finally, we define $\text{Bag}_i(\sigma_n)$ by interpreting any relation symbol R in such a way that the atomic formulas in $t_n(\mathbf{y}_n)$ are satisfied under v_{σ_n} , that is, such that $\text{Bag}_i(\sigma_n)$ satisfies $R(v_{\sigma_n}(\mathbf{y}))$ iff $R(\mathbf{y}) \in t_n(\mathbf{y}_n)$.

Let \mathfrak{A}_i be the union of all $\text{Bag}_i(t)$, $t \in T_i$. It is easy to see that (T_i, E_i, bag_i) is a tree decomposition of \mathfrak{A}_i . In fact, in the inductive step above, $t_n(\mathbf{y}_n)$ and $t_{n-1}(\mathbf{y}_{n-1})$ coincide on $[\mathbf{y}_n] \cap [\mathbf{y}_{n-1}]$. Thus, the interpretation of any relation symbol R coincides on the intersection of $\text{bag}_i(\sigma_n)$ and $\text{bag}_i(\sigma_{n-1})$. We proceed to show that the guarded τ -bisimulation mentioned in Lemma 5 indeed exists. To this end, we prove the following auxiliary lemma. We call a tuple \mathbf{a} τ -guarded in \mathfrak{A} if there exists a τ -guard $Q(\mathbf{x})$ such that $\mathfrak{A} \models Q(\mathbf{a})$.

Lemma 6. *For all i, j with $1 \leq i, j \leq m$, we have:*

1. *For every $\sigma \in T_i$ with $\text{tail}(\sigma) = (t(\mathbf{y}), \Phi)$, we have $\mathfrak{A}_i \models t(v_\sigma(\mathbf{y}))$;*
2. *Let $H_{i,j}$ be the set of all mappings $p_{\sigma, \sigma', \mathbf{z}}$, where*
 - $\sigma \in T_i, \sigma' \in T_j$, $\text{tail}(\sigma) = (t(\mathbf{y}), \Phi)$, and $\text{tail}(\sigma') = (t'(\mathbf{y}'), \Phi)$;
 - \mathbf{z} is a tuple with $[\mathbf{z}] \subseteq [\mathbf{y}] \cap [\mathbf{y}']$ and $v_\sigma(\mathbf{z})$ is τ -guarded in \mathfrak{A}_i (or, equivalently, $v_{\sigma'}(\mathbf{z})$ is τ -guarded in \mathfrak{A}_j);
 - $p_{\sigma, \sigma', \mathbf{z}} : v_\sigma(\mathbf{z}) \mapsto v_{\sigma'}(\mathbf{z})$.

Then $H_{i,j}$ is a guarded τ -bisimulation between \mathfrak{A}_i and \mathfrak{A}_j .

Proof. For Point 1, we prove by induction that, for all $\sigma \in T_i$ with $\text{tail}(\sigma) = (t(\mathbf{y}), \Phi)$ and all formulas $\varphi(\mathbf{z})$ with $[\mathbf{z}] \subseteq [\mathbf{y}]$, we have:

$$\varphi(\mathbf{z}) \in t(\mathbf{y}) \quad \text{iff} \quad \mathfrak{A}_i \models \varphi(v_\sigma(\mathbf{z}))$$

The induction base is given by the definition of $\text{bag}_i(\sigma)$. If φ is of the shape $\neg\varphi'$, $\varphi' \wedge \varphi''$, or $\varphi' \vee \varphi''$, the statement is immediate from the hypothesis. Consider now $\varphi(\mathbf{z}) = \exists \mathbf{x}(R(\mathbf{z}, \mathbf{x}) \wedge \lambda(\mathbf{z}, \mathbf{x}))$.

(\Rightarrow) Since \mathcal{M} is existentially saturated, there is a $\Phi' \in \mathcal{M}$ such that Φ, Φ' are compatible and $R(\mathbf{z}, \mathbf{x}') \wedge \lambda(\mathbf{z}, \mathbf{x}') \in t'(\mathbf{y}')$ for some $t'(\mathbf{y}') \in \Phi'$ such that $t(\mathbf{y})$ and $t'(\mathbf{y}')$ coincide on $[\mathbf{y}] \cap [\mathbf{y}']$. By definition of T_i and compatibility of Φ, Φ' , we have $\sigma' = \sigma \cdot (t'(\mathbf{y}'), \Phi') \in T_i$. Moreover, by induction, we obtain that \mathfrak{A}_i satisfies $R(\mathbf{z}, \mathbf{x}') \wedge \lambda(\mathbf{z}, \mathbf{x}')$ under $v_{\sigma'}$. By definition of $\text{bag}_i(\sigma)$ and $\text{bag}_i(\sigma')$, we get $\mathfrak{A}_i \models \varphi(v_\sigma(\mathbf{z}))$.

(\Leftarrow) Conversely, assume $\mathfrak{A}_i \models \varphi(v_\sigma(\mathbf{z}))$. By construction, there is some $\sigma' \in T_i$ such that $v_\sigma(\mathbf{z}) = v_{\sigma'}(\mathbf{z})$ and \mathfrak{A}_i satisfies $R(\mathbf{z}, \mathbf{x}') \wedge \lambda(\mathbf{z}, \mathbf{x}')$ under $v_{\sigma'}$, for some \mathbf{x}' . By induction hypothesis, $R(\mathbf{z}, \mathbf{x}') \wedge \lambda(\mathbf{z}, \mathbf{x}') \in t'(\mathbf{y}')$, where $\text{tail}(\sigma') = (t'(\mathbf{y}'), \Phi')$. Thus, $\exists \mathbf{x}(R(\mathbf{z}, \mathbf{x}) \wedge \lambda(\mathbf{z}, \mathbf{x})) = \varphi(\mathbf{z}) \in t'(\mathbf{y}')$. As $v_\sigma(\mathbf{z}) = v_{\sigma'}(\mathbf{z})$, the construction of T_i implies that $t'(\mathbf{y}')$ and $t(\mathbf{y})$ coincide on all subformulas over \mathbf{z} , hence $\varphi(\mathbf{z}) \in t(\mathbf{y})$.

For Point 2, observe first that the $p_{\sigma, \sigma', \mathbf{z}}$ are partial τ -isomorphisms between τ -guarded tuples since all $\Phi \in \mathcal{M}$ are τ -uniform. (In addition, the observation that $v_\sigma(\mathbf{z})$ is τ -guarded in \mathfrak{A}_i iff $v_{\sigma'}(\mathbf{z})$ is τ -guarded in \mathfrak{A}_j follows from the condition that Φ is τ -uniform.) By symmetry, it suffices to prove the first condition for guarded τ -bisimulations.

Let $p \in H_{i,j}$. Then we have $\sigma \in T_i, \sigma' \in T_j$ with $\text{tail}(\sigma) = (t(\mathbf{y}), \Phi)$ and $\text{tail}(\sigma') = (t'(\mathbf{y}'), \Phi')$ and we have a tuple \mathbf{z} such that $[\mathbf{z}] \subseteq [\mathbf{y}] \cap [\mathbf{y}']$ and $v_\sigma(\mathbf{z})$ is τ -guarded in \mathfrak{A}_i and $p = p_{\sigma, \sigma', \mathbf{z}}$. Thus, there is a τ -guard $Q(\mathbf{z})$ with $\mathfrak{A}_i \models Q(v_\sigma(\mathbf{z}))$ and $\mathfrak{A}_j \models Q(v_{\sigma'}(\mathbf{z}))$. Consider any tuple \mathbf{b} with $\mathfrak{A}_i \models R(\mathbf{b})$ for some $R \in \tau$. We have to show that there exists a mapping $p_{\rho, \rho', \mathbf{z}'} \in H_{i,j}$ with domain $[\mathbf{b}]$ which coincides with $p_{\sigma, \sigma', \mathbf{z}}$ on $[v_\sigma(\mathbf{z})] \cap [\mathbf{b}]$. We distinguish on whether or not that intersection is empty.

Case 1. $[v_\sigma(\mathbf{z})] \cap [\mathbf{b}] = \emptyset$. The existence of such a mapping follows from τ -bisimulation saturatedness: to see this, observe that, as we have a tree decomposition, there exists $\rho_0 \in T_i$ such that $[\mathbf{b}] \subseteq \text{dom}(\text{bag}(\rho_0))$. Let $\text{tail}(\rho_0) = (s(\mathbf{x}_0), \Omega)$. Then there exists a tuple \mathbf{y}_0 with $[\mathbf{y}_0] \subseteq [\mathbf{x}_0]$ such that $v_{\rho_0}(\mathbf{y}_0) = \mathbf{b}$. We have $R(\mathbf{y}_0) \in s(\mathbf{x}_0)$. As $\hat{t}_j \in \Omega$, by τ -bisimulation saturatedness of Ω , there exists $s'(\mathbf{y}'_0) \in \Omega$ such that $\hat{t}_j \subseteq s'(\mathbf{y}'_0)$ and $[\mathbf{y}'_0] = [\mathbf{y}_0]$. But then $R(\mathbf{y}_0) \in s'(\mathbf{y}'_0)$. Also $\rho = (\hat{t}_j, \hat{\Psi}) \cdot (s'(\mathbf{y}'_0), \Omega) \in T_j$. Thus $p_{\rho_0, \rho, \mathbf{y}'_0}$ is as required.

Case 2. $[v_\sigma(\mathbf{z})] \cap [\mathbf{b}] \neq \emptyset$. As we have a tree decomposition, there exists $\rho_0 \in T_i$ such that $[\mathbf{b}] \subseteq \text{dom}(\text{bag}(\rho_0))$. Let $\text{tail}(\rho_0) = (s(\mathbf{x}_0), \Omega)$. Then there exists a tuple \mathbf{z}' with $[\mathbf{z}'] \subseteq [\mathbf{x}_0]$ such that $v_{\rho_0}(\mathbf{z}') = \mathbf{b}$. We distinguish the following cases:

- (a) $\rho_0 = \sigma$;
- (b) $\rho_0 \neq \sigma$.

Assume first that (a) holds. Then $(s(\mathbf{x}_0), \Omega) = (t(\mathbf{y}), \Phi)$ and $\mathbf{b} = v_\sigma(\mathbf{z}')$. We use τ -bisimulation saturatedness of Φ . Consider the restriction \mathbf{z}'' of \mathbf{z}' to $[\mathbf{z}] \cap [\mathbf{z}']$ and the restriction $t'(\mathbf{y}')|_{[\mathbf{z}'']}$ of $t'(\mathbf{y}')$ to $[\mathbf{z}'']$. Then there exists $s'(\mathbf{z}'_0) \in \Phi$ such

that $t'(\mathbf{y}')|_{[\mathbf{z}'']} \subseteq s'(\mathbf{z}'_0) \in \Phi$ and $[\mathbf{z}'_0] = [\mathbf{z}']$. Let $\sigma'' = \sigma' \cdot (s'(\mathbf{z}'_0), \Phi) \in T_j$. Then $p_{\sigma, \sigma'', \mathbf{z}'_0}$ is as required, as Φ is τ -uniform.

Assume now that Point (b) holds. Consider the restriction \mathbf{z}'' of \mathbf{z}' to $[\mathbf{z}] \cap [\mathbf{z}']$ and the restriction $t'(\mathbf{y}')|_{[\mathbf{z}'']}$ of $t'(\mathbf{y}')$ to $[\mathbf{z}'']$. Consider the restriction $\Phi|_{[\mathbf{z}'']}$ of Φ to $[\mathbf{z}'']$. By closure under restrictions, $\Phi|_{[\mathbf{z}'']} \in \mathcal{M}$. Observe that $\Phi, \Phi|_{[\mathbf{z}'']}$ and $\Phi|_{[\mathbf{z}'']}, \Omega$ are compatible: indeed, in the tree decomposition all bags on the path from σ to ρ_0 have a tail (\cdot, Ω') satisfying $\Phi|_{[\mathbf{z}'']} \subseteq \Omega'$. Thus $t'(\mathbf{y}')|_{[\mathbf{z}'']} \in \Omega$. Using the fact that Ω is τ -bisimulation saturated, one can now show that there exists $s'(\mathbf{z}'_0) \in \Omega$ such that $t'(\mathbf{y}')|_{[\mathbf{z}'']} \subseteq s'(\mathbf{z}'_0)$ and $[\mathbf{z}'_0] = [\mathbf{z}']$. We then have

$$\rho = \sigma' \cdot (t'(\mathbf{y}')|_{[\mathbf{z}'']}, \Phi|_{[\mathbf{z}'']}) \cdot (s'(\mathbf{z}'_0), \Omega) \in T_j$$

and $p_{\rho_0, \rho, \mathbf{z}'_0}$ is as required. \square

To complete the proof of Lemma 5, assume w.l.o.g. that $\hat{t}_i \subseteq t_i(\mathbf{x}_i)$ for $i = 1, 2$. Take $\rho_i = (\hat{t}_i, \hat{\Psi}) \cdot (t_i(\mathbf{x}_i), \Psi) \in T_i$, for $i = 1, 2$. Consider the tuples $\mathbf{a}_i := v_{\rho_i}(\mathbf{x}_i)$. By Lemma 6, $\mathfrak{A}_i \models t_i(\mathbf{a}_i)$. Also by Lemma 6, for any tuple \mathbf{z} with $[\mathbf{z}] \subseteq [\mathbf{x}_1] \cap [\mathbf{x}_2]$ and such that $v_{\rho_1}(\mathbf{z})$ is τ -guarded in \mathfrak{A}_1 or \mathfrak{A}_2 , we have $p_{\rho_1, \rho_2, \mathbf{z}} : v_{\rho_1}(\mathbf{z}) \mapsto v_{\rho_2}(\mathbf{z}) \in H_{1,2}$. But then, as any two $p_{\rho_1, \rho_2, \mathbf{z}}$ coincide on the intersection of their domains, we have for $Y = [\mathbf{x}_1] \cap [\mathbf{x}_2]$, $\mathfrak{A}_1, v_{\rho_1}(\mathbf{x}_1|_Y) \sim_{\text{GF}, \tau} \mathfrak{A}_2, v_{\rho_2}(\mathbf{x}_2|_Y)$, as required. \square

Lemma 7. *Let $\mathfrak{A}_1, \mathbf{a}_1$ and $\mathfrak{A}_2, \mathbf{a}_2$ be pointed structures with \mathbf{a}_1 and \mathbf{a}_2 tuples with pairwise distinct elements of length $m \leq \text{fv}(\Xi)$ and let τ be a signature. Consider assignments $\mathbf{x}_0 \mapsto \mathbf{a}_i$ with $[\mathbf{x}_0] \subseteq \{x_0, \dots, x_{2n}\}$. If $\mathfrak{A}_1, \mathbf{a}_1 \sim_{\text{GF}, \tau} \mathfrak{A}_2, \mathbf{a}_2$, then there exists a good set \mathcal{M} and some $\Psi \in \mathcal{M}$ such that*

- all $\Phi \in \mathcal{M}$ with $\Phi \neq \Psi$ use at most $\text{width}(\Xi)$ many free variables;
- there exist types $t_1(\mathbf{x}_0), t_2(\mathbf{x}_0) \in \Psi$ such that $t_i(\mathbf{x}_0) = \text{tp}(\mathfrak{A}_i, \mathbf{x}_0 \mapsto \mathbf{a}_i)$ for $i = 1, 2$ and all types $t(\mathbf{y}) \in \Psi \setminus \{t_1(\mathbf{x}_0), t_2(\mathbf{x}_0)\}$ use at most $\text{width}(\Xi)$ free variables among $[\mathbf{x}_0]$.

Proof. Assume w.l.o.g. that \mathfrak{A}_1 and \mathfrak{A}_2 are disjoint. For any tuples \mathbf{b}_1 in \mathfrak{A}_i and \mathbf{b}_2 in \mathfrak{A}_j with $i, j \in \{1, 2\}$, we use $\text{tp}(\mathbf{x}_1 \mapsto \mathbf{b}_1)$ to denote $\text{tp}(\mathfrak{A}_i, \mathbf{x}_1 \mapsto \mathbf{b}_1)$ and we write $\mathbf{b}_1 \sim_{\text{GF}, \tau} \mathbf{b}_2$ if $\mathfrak{A}_i, \mathbf{b}_1 \sim_{\text{GF}, \tau} \mathfrak{A}_j, \mathbf{b}_2$. Define \mathcal{M} as follows. Take any tuple \mathbf{a} of distinct elements in \mathfrak{A}_i , $i \in \{1, 2\}$. Take a tuple \mathbf{x} from $\{x_1, \dots, x_{2n}\}$ such that $\sigma : \mathbf{x} \mapsto \mathbf{a}$ is a bijection. Then let $\Phi_{\mathbf{a}, \mathbf{x}}$ contain all types $\text{tp}(\sigma' : \mathbf{x}|_Y \mapsto \mathbf{b})$ with $Y \subseteq [\mathbf{x}]$ and \mathbf{b} in either \mathfrak{A}_1 or \mathfrak{A}_2 such that $\sigma(\mathbf{x}|_Y) \sim_{\text{GF}, \tau} \sigma'(\mathbf{x}|_Y)$.

Let \mathcal{M} contain all such $\Phi_{\mathbf{a}, \mathbf{x}}$ with \mathbf{a} of length at most $\text{width}(\Xi)$ and \mathbf{x} from $\{x_1, \dots, x_{2n}\}$. Moreover, if $m > \text{width}(\Xi)$, then add $\hat{\Phi}_{\mathbf{a}_1, \mathbf{x}_0}$ to \mathcal{M} , where $\hat{\Phi}_{\mathbf{a}_1, \mathbf{x}_0}$ is obtained from $\Phi_{\mathbf{a}_1, \mathbf{x}_0}$ by removing all t distinct from $t_1(\mathbf{x}_0)$ and $t_2(\mathbf{x}_0)$ using more than $\text{width}(\Xi)$ many free variables.

We show that \mathcal{M} is as required. By definition, $\text{tp}(\mathfrak{A}_1, \mathbf{x}_0 \mapsto \mathbf{a}_1), \text{tp}(\mathfrak{A}_2, \mathbf{x}_0 \mapsto \mathbf{a}_2) \in \Phi_{\mathbf{a}_1, \mathbf{x}_0} \in \mathcal{M}$.

For the next steps we first assume that instead of $\hat{\Phi}_{\mathbf{a}_1, \mathbf{x}_0}$ we have $\Phi_{\mathbf{a}_1, \mathbf{x}_0}$ in \mathcal{M} . Then observe that if we have any $\Phi \in \mathcal{M}$ and $t(\mathbf{x}'), s(\mathbf{x}'') \in \Phi$, then we can

assume that $\Phi = \Phi_{\mathbf{a}, \mathbf{x}}$, we have a bijection σ from \mathbf{a} to \mathbf{x} , $\mathbf{x}' = \mathbf{x}_{|Y'}$ and $\mathbf{x}'' = \mathbf{x}_{|Y''}$ for appropriate sets of variables $Y', Y'' \subseteq [\mathbf{x}]$, and there are $\sigma' : \mathbf{x}_{|Y'} \mapsto \mathfrak{A}_i$ and $\sigma'' : \mathbf{x}_{|Y''} \mapsto \mathfrak{A}_j$ such that $\sigma'(\mathbf{x}_{|Y'}) \sim_{\text{GF}, \tau} \sigma(\mathbf{x}_{|Y'})$ and $\sigma''(\mathbf{x}_{|Y''}) \sim_{\text{GF}, \tau} \sigma(\mathbf{x}_{|Y''})$. Then $\sigma'(\mathbf{x}_{|Y' \cap Y''}) \sim_{\text{GF}, \tau} \sigma''(\mathbf{x}_{|Y' \cap Y''})$. We show that each $\Phi_{\mathbf{a}, \mathbf{x}}$ is τ -uniform and τ -bisimulation saturated.

1. Every $\Phi_{\mathbf{a}, \mathbf{x}} \in \mathcal{M}$ is τ -uniform: let $t(\mathbf{x}'), s(\mathbf{x}'') \in \Phi_{\mathbf{a}, \mathbf{x}}$ be as above and assume that $Q(\mathbf{z})$ is a τ -guard with $[\mathbf{z}] \subseteq [\mathbf{x}'] \cap [\mathbf{x}'']$. Then $[\mathbf{z}] \subseteq Y' \cap Y''$ and so $Q(\mathbf{z}) \in t(\mathbf{x}')$ iff $Q(\mathbf{z}) \in s(\mathbf{x}'')$ since $\sigma'(\mathbf{x}_{|Y' \cap Y''}) \sim_{\text{GF}, \tau} \sigma''(\mathbf{x}_{|Y' \cap Y''})$, as required.
2. To show τ -bisimulation saturatedness let $\Phi_{\mathbf{a}, \mathbf{x}} \in \mathcal{M}$ and $t(\mathbf{x}'), s(\mathbf{x}'') \in \Phi_{\mathbf{a}, \mathbf{x}}$ be as above and let $R(\mathbf{y}) \in t(\mathbf{x}')$ with $[\mathbf{x}''] \subseteq [\mathbf{y}]$ be a strict τ -guard. We have $Y'' \subseteq [\mathbf{y}] \subseteq Y'$ and $\sigma'(\mathbf{x}_{|Y''}) \sim_{\text{GF}, \tau} \sigma''(\mathbf{x}_{|Y''})$. Let H be the guarded τ -bisimulation witnessing this. By the definition of guarded τ -bisimulations, there exists $p \in H$ with domain $\sigma'(\mathbf{x}_{|[\mathbf{y}]})$ such that $p \circ \sigma'_{|Y''} = \sigma''$. Now we expand σ'' to the domain $[\mathbf{y}]$ by setting $\hat{\sigma} := p \circ \sigma'_{|[\mathbf{y}]}$. Let \mathbf{b}' be the image of $\mathbf{x}_{|[\mathbf{y}]}$ under $\hat{\sigma}$. Then the type $\text{tp}(\hat{\sigma} : \mathbf{x}_{|[\mathbf{y}]} \mapsto \mathbf{b}')$ is as required.

Finally we show that every $\Phi \in \mathcal{M}$ is existentially saturated in \mathcal{M} . Assume $\Phi_{\mathbf{a}, \mathbf{x}}$ is given. Assume $\exists \mathbf{y}(R(\mathbf{x}', \mathbf{y}) \wedge \lambda(\mathbf{x}', \mathbf{y})) \in t(\mathbf{x}_{|Y}) = \text{tp}(\sigma' : \mathbf{x}_{|Y} \mapsto \mathbf{b})$ with $Y \subseteq [\mathbf{x}]$ and \mathbf{b} w.l.o.g. in \mathfrak{A}_1 . Then $\mathfrak{A}_1 \models_{\sigma'} \exists \mathbf{y}(R(\mathbf{x}', \mathbf{y}) \wedge \lambda(\mathbf{x}', \mathbf{y}))$. Then we find an assignment σ'' for the variables in $[\mathbf{x}'\mathbf{y}]$ which coincides with σ' on $[\mathbf{x}']$ such that $\mathfrak{A}_1 \models_{\sigma''} R(\mathbf{x}', \mathbf{y}) \wedge \lambda(\mathbf{x}', \mathbf{y})$. Take a tuple \mathbf{c} of distinct elements with $[\mathbf{c}] = [\sigma''(\mathbf{x}'\mathbf{y})]$ and a tuple \mathbf{y}' of variables in $\{x_1, \dots, x_{2n}\}$ such that $[\mathbf{x}'] = [\mathbf{x}] \cap [\mathbf{y}']$ and we have a bijection $\rho : \mathbf{y}' \mapsto \mathbf{c}$ which coincides with σ' on $[\mathbf{x}']$. Then $\rho(\mathbf{y}'_{|[\mathbf{x}']}) \sim_{\text{GF}, \tau} \sigma(\mathbf{x}_{|[\mathbf{x}']})$ and so $\Phi_{\mathbf{a}, \mathbf{x}}$ and $\Phi_{\mathbf{c}, \mathbf{y}'}$ are compatible and $\Phi_{\mathbf{c}, \mathbf{y}'}$ is as required.

For the proof with $\hat{\Phi}_{\mathbf{a}_1, \mathbf{x}_0}$ instead of $\Phi_{\mathbf{a}_1, \mathbf{x}_0}$ in \mathcal{M} observe that $\hat{\Phi}_{\mathbf{a}_1, \mathbf{x}_0}$ is τ -uniform and τ -bisimulation saturated as $\hat{\Phi}_{\mathbf{a}_1, \mathbf{x}_0}$ behaves in exactly the same way as $\Phi_{\mathbf{a}_1, \mathbf{x}_0}$ regarding τ -guarded $Q(\mathbf{y})$. For the same reason all elements of \mathcal{M} are still existentially saturated in \mathcal{M} . \square

It follows from Lemmas 5 and 7 that the following two conditions are equivalent.

1. $\varphi(\mathbf{x}_0), \psi(\mathbf{x}_0)$ are jointly $\text{GF}(\tau)$ -consistent;
2. there is a good set $\mathcal{M} = \{\Phi_0\} \cup \mathcal{M}'$ and Ξ -types $t_1(\mathbf{x}), t_2(\mathbf{x}) \in \Phi_0$ such that:
 - (a) $t_1(\mathbf{x}), t_2(\mathbf{x})$ have $\text{fv}(\Xi)$ free variables and one can replace the variables in $[\mathbf{x}_0]$ by variables in \mathbf{x} such that $\varphi' \in t_1(\mathbf{x}), \psi' \in t_2(\mathbf{x})$ for the resulting formulas φ', ψ' ;
 - (b) all Ξ -types $t(\mathbf{y}) \in \Phi_0 \setminus \{t_1(\mathbf{x}), t_2(\mathbf{x})\}$ use at most $\text{width}(\Xi)$ free variables among $[\mathbf{x}]$;
 - (c) all mosaics in \mathcal{M}' use at most $\text{width}(\Xi)$ free variables.

Hence, it suffices to provide an algorithm deciding Condition 2.

Lemma 8. *On input $\varphi(\mathbf{x}_0), \psi(\mathbf{x}_0)$, Condition 2 can be decided in time triple exponential in the size of $\varphi(\mathbf{x}_0), \psi(\mathbf{x}_0)$ in general, and double exponential in the size of $\varphi(\mathbf{x}_0), \psi(\mathbf{x}_0)$ if $\text{width}(\Xi)$ is bounded by a constant.*

Proof. We proceed as follows to identify a good set \mathcal{M} that satisfies Condition 2. For every pair $t_1(\mathbf{x}), t_2(\mathbf{x})$ of Ξ -types that satisfies Condition 2(a) enumerate all τ -mosaics Φ_0 satisfying Condition 2(b), that is, $t_1(\mathbf{x}), t_2(\mathbf{x}) \in \Phi_0$ and all types in Φ_0 except $t_1(\mathbf{x}), t_2(\mathbf{x})$ use at most $\text{width}(\Xi)$ free variables among $[\mathbf{x}]$. For each such Φ_0 start with the set \mathcal{M}_0 consisting of Φ_0 and all mosaics Φ with at most $\text{width}(\Xi)$ free variables. Then exhaustively remove mosaics from \mathcal{M}_0 that are not existentially saturated. It can be verified that the fixpoint is good. Accept if the fixpoint still contains Φ_0 . Reject if for no choice of Φ_0 this is the case.

Correctness of the algorithm is straightforward, so it remains to analyze its run time. For this purpose, let r be the number of subformulas (of formulas) in Ξ and $\ell \geq 0$. Observe that a subformula with ℓ free variables has at most $(2n)^\ell$ instantiations with variables from x_1, \dots, x_{2n} . Since for every such instantiated formula either the formula itself or its negation is contained in any type, there are at most $2^{r(2n)^\ell}$ many types with ℓ free variables. Thus, there are only double exponentially many choices for $t_1(\mathbf{x}), t_2(\mathbf{x})$ and Φ_0 . Moreover, the initial sets \mathcal{M}_0 are of size triple exponential in the size of $\varphi(\mathbf{x}_0), \psi(\mathbf{x}_0)$ in general, and double exponential in the size of $\varphi(\mathbf{x}_0), \psi(\mathbf{x}_0)$ if $\text{width}(\Xi)$ is bounded by a constant. Checking whether some Φ is existentially saturated in some set of mosaics \mathcal{M} can be done in time polynomial in the size of \mathcal{M} . Since in every round at least one mosaic is removed, the overall run time is as stated in the Lemma. \square

From the equivalence of Conditions 1 and 2, and Lemma 8 we finally obtain:

Theorem 2. *Joint $\text{GF}(\tau)$ -consistency of $\varphi(\mathbf{x}_0), \psi(\mathbf{x}_0)$ can be decided in time triple exponential in the size of $\varphi(\mathbf{x}_0), \psi(\mathbf{x}_0)$ in general, and double exponential in the size of $\varphi(\mathbf{x}_0), \psi(\mathbf{x}_0)$ if the arity of relation symbols is bounded by a constant.*

6 Deciding Joint $\text{GF}(\tau)$ -Consistency: Lower Bounds

We reduce the word problem for exponentially and double exponentially space bounded alternating Turing machines (ATMs), respectively. We actually use a slightly unusual ATM model which is easily seen to be equivalent to the standard model.

An *alternating Turing machine (ATM)* is a tuple $M = (Q, \Theta, \Gamma, q_0, \Delta)$ where $Q = Q_\exists \uplus Q_\forall$ is the set of states that consists of *existential states* in Q_\exists and *universal states* in Q_\forall . Further, Θ is the input alphabet and Γ is the tape alphabet that contains a *blank symbol* $\square \notin \Theta$, $q_0 \in Q_\exists$ is the *starting state*, and the *transition relation* Δ is of the form $\Delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{L, R\}$. The set $\Delta(q, a) := \{(q', a', M) \mid (q, a, q', a', M) \in \Delta\}$ must contain exactly two or zero elements for every $q \in Q$ and $a \in \Gamma$. Moreover, the state q' must be from Q_\forall if $q \in Q_\exists$ and from Q_\exists otherwise, that is, existential and universal states alternate. Note that there is no accepting state. The ATM accepts if it runs forever and rejects otherwise. Starting from the standard ATM model, this can be achieved by assuming that exponentially space bounded ATMs terminate on any input and then modifying them to enter an infinite loop from the accepting state.

A *configuration* of an ATM is a word wqw' with $w, w' \in \Gamma^*$ and $q \in Q$. We say that wqw' is *existential* if q is, and likewise for *universal*. *Successor configurations* are defined in the usual way. Note that every configuration has exactly two successor configurations. A *computation tree* of an ATM M on input w is an infinite tree whose nodes are labeled with configurations of M such that

- the root is labeled with the initial configuration q_0w ;
- if a node is labeled with an existential configuration wqw' , then it has a single successor and this successor is labeled with a successor configuration of wqw' ;
- if a node is labeled with a universal configuration wqw' , then it has two successors and these successors are labeled with the two successor configurations of wqw' .

An ATM M *accepts* an input w if there is a computation tree of M on w . It is well-known that the word problem for 2^n -space bounded and 2^{2^n} -space bounded ATMs is 2EXPTIME-hard and 3EXPTIME-hard, respectively [12].

6.1 2EXPTIME lower bound

For didactic reasons, we start with 2EXPTIME-hardness for the bounded arity case. Let M be a 2^n -space bounded ATM and w an input. The idea of the reduction is as follows. We set

$$\begin{aligned}\theta(x) &= A(x) \\ \tau &= \{R, S, X, Z, B_\forall, B_\exists^1, B_\exists^2\} \cup \{A_\sigma \mid \sigma \in \Gamma \cup (Q \times \Gamma)\},\end{aligned}$$

where R, S are binary relation symbols, and the remaining symbols are unary. We aim to construct φ such that M accepts w iff $\varphi \wedge A(x)$ and $\varphi \wedge \neg A(x)$ are jointly GF(τ)-consistent. The sentence φ is a conjunction of several sentences. The first conjunct, φ_0 below, enforces that every element satisfying A is involved in a three-element R -loop:

$$\varphi_0 = \forall x (A(x) \rightarrow \exists yz (G(x, y, z) \wedge X(x) \wedge R(x, y) \wedge R(y, z) \wedge R(z, x)))$$

Now, if $\varphi \wedge A(x)$ and $\varphi \wedge \neg A(x)$ are jointly GF(τ)-consistent, there exist models \mathfrak{A} and \mathfrak{B} of φ and elements a, b such that $a \in A^{\mathfrak{A}}$, $b \notin A^{\mathfrak{B}}$, and $\mathfrak{A}, a \sim_{\text{GF}, \tau} \mathfrak{B}, b$. If the latter holds, then from $a \in A^{\mathfrak{A}}$ and φ_0 it follows that b has an infinite outgoing path ρ along R on which every third element satisfies X and is guarded τ -bisimilar to a . Let us call these elements the *X-elements*. As guarded bisimilarity is an equivalence relation, all X -elements are actually guarded τ -bisimilar. The other conjuncts of φ will enforce that along the X -elements on ρ , a counter counts modulo 2^n using relation symbols not in τ . Moreover, in every X -element of ρ starts an infinite tree along symbol S that is supposed to mimic the computation tree of M . Along this tree, two counters are maintained:

- one counter starting at 0 and counting modulo 2^n to divide the tree in subpaths of length 2^n ; each such path of length 2^n represents a configuration;

- another counter starting at the value of the counter along ρ and also counting modulo 2^n .

To link successive configurations we use the fact that all X -elements on ρ are guarded τ -bisimilar and thus each X -element is the starting point of trees along S with identical τ -decorations. As on the m th such tree the second counter starts at all nodes at distances $k \cdot 2^n - m$, for all $k \geq 1$, we are in the position to coordinate all positions at all successive configurations.

In detail, let $w = a_0, \dots, a_{n-1}$ be an input to M of length n . We will be using unary symbols A_i, U_i, V_i , $1 \leq i \leq n$ to represent the aforementioned binary counters; we will refer to them with A -counter, U -counter, and V -counter, respectively.

The sentences below enforce that the A -counter along the R -path ρ is incremented (precisely) at every X -element. In order to avoid that the counter is stipulated at a (which would lead to a contradiction), we use an additional symbol $I \notin \tau$ that is satisfied along the entire path and acts as an additional guard:²

$$\begin{aligned} & \forall xy (R(x, y) \rightarrow (\neg A(x) \rightarrow I(x))) \\ & \forall xy (R(x, y) \rightarrow (I(x) \leftrightarrow I(y))) \\ & \forall xy (R(x, y) \wedge I(x) \rightarrow (X(x) \wedge \neg X(y) \rightarrow \text{Eq}(x, y))) \\ & \forall xy (R(x, y) \wedge I(x) \rightarrow (\neg X(x) \wedge \neg X(y) \rightarrow \text{Eq}(x, y))) \\ & \forall xy (R(x, y) \wedge I(x) \rightarrow (\neg X(x) \wedge X(y) \rightarrow \text{Succ}(x, y))) \end{aligned}$$

Here, atoms $\text{Eq}(x, y)$ and $\text{Succ}(x, y)$ are abbreviations for formulas that express that the A -counter value at x equals (respectively, is the predecessor of) the A -counter value at y , that is:

$$\begin{aligned} \text{Eq}(x, y) &= \bigwedge_{i < n} A_i(x) \leftrightarrow A_i(y) \\ \text{Succ}(x, y) &= \bigvee_{i < n} (A_i(y) \wedge \neg A_i(x) \wedge \bigwedge_{j < i} (\neg A_j(y) \wedge A_j(x)) \wedge \bigwedge_{j > i} (A_j(y) \leftrightarrow A_j(x))) \end{aligned}$$

Now, we start a tree along S from all X -elements on the infinite R -path. Along the path, we maintain the U - and V -counter, which are initialized to 0 and the value of the A -counter, respectively:

$$\begin{aligned} & \forall x \exists y S(xy) \\ & \forall xy (S(xy) \rightarrow (I(x) \leftrightarrow I(y))) \\ & \forall x (I(x) \wedge X(x) \rightarrow \text{Min}_U(x)) \\ & \forall x (I(x) \wedge X(x) \rightarrow \bigwedge_{i < n} (V_i(x) \leftrightarrow A_i(x))) \end{aligned}$$

² Some formulas are not syntactically guarded but can easily be rewritten.

Here, $\text{Min}_U(x)$ is an abbreviation for the formula that expresses that the U -counter is 0 at x ; we use similar abbreviations such as $\text{Max}_V(x)$ below. The U and V -counters are incremented along S analogously to how the A -counter is incremented along R , but on every S -step; we omit details. Configurations of M are represented between two consecutive elements having U -counter value 0. We next enforce the structure of the computation tree, assuming that $q_0 \in Q_V$.

$$\begin{aligned}
& \forall x (I(x) \wedge X(x) \rightarrow B_V(x)) \\
& \forall xy (S(x, y) \wedge I(x) \rightarrow (\neg \text{Max}_U(x) \rightarrow (B_V(x) \leftrightarrow B_V(y)))) \\
& \forall xy (S(x, y) \wedge I(x) \rightarrow (\neg \text{Max}_U(x) \rightarrow (B_{\exists}^i(x) \leftrightarrow B_{\exists}^i(y)))) \quad i \in \{1, 2\} \\
& \forall xy (S(x, y) \wedge I(x) \rightarrow (\text{Max}_U(x) \rightarrow (B_V(x) \leftrightarrow \neg B_V(y)))) \\
& \forall x (I(x) \wedge \text{Max}_U(x) \rightarrow \exists y (S(x, y) \wedge Z(y)) \wedge \exists y (S(x, y) \wedge \neg Z(y))) \\
& \forall x (I(x) \wedge \neg B_V(x) \rightarrow (B_{\exists}^1(x) \leftrightarrow \neg B_{\exists}^2(x)))
\end{aligned}$$

These sentences enforce that all points which represent a configuration satisfy exactly one of $B_V, B_{\exists}^1, B_{\exists}^2$, indicating the kind of configuration and, if existential, also a choice of the transition function. The symbol $Z \in \tau$ enforces the branching.

We next set the initial configuration, for input $w = a_0, \dots, a_{n-1}$. Below, we use $\forall y (S^i(x, y) \rightarrow \psi(y))$ to abbreviate the GF formula that enforces ψ at all elements y that are reachable in i steps via S from x .

$$\begin{aligned}
& \forall x (I(x) \wedge X(x) \rightarrow A_{q_0, a}(x)) \\
& \forall x (I(x) \wedge X(x) \rightarrow \forall y (S^k(x, y) \rightarrow A_{a_k}(y)) \quad 0 < k < n \\
& \forall x (I(x) \wedge X(x) \rightarrow \forall y (S^{n+1}(x, y) \rightarrow \text{Blank}(y))) \\
& \forall x (\text{Blank}(x) \rightarrow A_{\square}(x)) \\
& \forall x (\text{Blank}(x) \wedge \neg \text{Max}_U(x) \rightarrow \forall y (S(x, y) \rightarrow \text{Blank}(y)))
\end{aligned}$$

To coordinate consecutive configurations, we associate with M functions f_i , $i \in \{1, 2\}$ that map the content of three consecutive cells of a configuration to the content of the middle cell in the i -th successor configuration (we assume an arbitrary order on the $\Delta(q, a)$).³ Clearly, for each possible such triple $(\sigma_1, \sigma_2, \sigma_3) \in \Gamma \cup (Q \times \Gamma)$ there is a GF formula $\psi_{\sigma_1, \sigma_2, \sigma_3}(x)$ that evaluates to true at an element a of the computation tree iff a is labeled with A_{σ_2} , a 's S -predecessor is labeled with A_{σ_1} , and a 's S -successor is labeled with A_{σ_3} . Now, in each configuration, we synchronize elements with V -counter 0, by including for every $\sigma = (\sigma_1, \sigma_2, \sigma_3)$ and $i \in \{1, 2\}$ the following sentences:

$$\begin{aligned}
& \forall x (I(x) \wedge \text{Min}_V(x) \wedge \neg \text{Min}_U(x) \wedge \neg \text{Max}_U(x) \wedge \psi_{\sigma}(x) \wedge B_V(x) \rightarrow \\
& \quad A_{f(\sigma)}^1(x) \wedge A_{f(\sigma)}^2(x)) \\
& \forall x (I(x) \wedge \text{Min}_V(x) \wedge \neg \text{Min}_U(x) \wedge \neg \text{Max}_U(x) \wedge \psi_{\sigma}(x) \wedge B_{\exists}^i(x) \rightarrow A_{f(\sigma)}^i(x)).
\end{aligned}$$

³ We ignore the cornercases that occur at the border of a configuration; they are treated in a similar way.

The unary symbols A_σ^i are used as markers (not in τ) and are propagated along S for 2^n steps, exploiting the V -counter. The superscript $i \in \{1, 2\}$ determines the successor configuration that the symbol is referring to. After crossing the end of a configuration, the symbol σ is propagated using further unary symbols A'_σ (the superscript is not needed anymore because the branching happens at the end of the configuration, based on Z):

$$\begin{aligned}
& \forall x (\neg \text{Max}_U(x) \wedge A_\sigma^i(x) \rightarrow \forall y (S(x, y) \rightarrow A_\sigma^i(y))) \\
& \forall x (\text{Max}_U(x) \wedge B_\forall(x) \wedge A_\sigma^1(x) \rightarrow \forall y (S(x, y) \rightarrow (Z(y) \rightarrow A'_\sigma(y)))) \\
& \forall x (\text{Max}_U(x) \wedge B_\forall(x) \wedge A_\sigma^2(x) \rightarrow \forall y (S(x, y) \rightarrow (\neg Z(y) \rightarrow A'_\sigma(y)))) \\
& \forall x (\text{Max}_U(x) \wedge B_\exists^i(x) \wedge A_\sigma^i(x) \rightarrow \forall y (S(x, y) \rightarrow A'_\sigma(y))) \quad i \in \{1, 2\} \\
& \forall x (\neg \text{Max}_V(x) \wedge A'_\sigma(x) \rightarrow \forall y (S(x, y) \rightarrow A'_\sigma(x))) \\
& \forall x (\text{Max}_V(x) \wedge A'_\sigma(x) \rightarrow \forall y (S(x, y) \rightarrow A_\sigma(x)))
\end{aligned}$$

For those (q, a) with $\Delta(q, a) = \emptyset$, we add the sentence

$$\forall xy (S(x, y) \rightarrow \neg A_{q,a}(x)).$$

The following Lemma establishes correctness of the reduction.

Lemma 9. *M accepts the input w iff there exists models $\mathfrak{A}, \mathfrak{B}$ of φ and elements $a \in A^\mathfrak{A}$, $b \notin A^\mathfrak{B}$ such that $\mathfrak{A}, a \sim_{GF, \tau} \mathfrak{B}, b$.*

Proof. (\Rightarrow) If M accepts w , there is a computation tree of M on w . We construct a single model \mathfrak{A} of φ as follows. Let \mathfrak{A}^* be the infinite tree-shaped structure that represents the computation tree of M on w as described above, that is, configurations are represented by sequences of 2^n elements linked by S . Moreover, all elements of a configuration are labeled with B_\forall , B_\exists^1 , or B_\exists^2 depending on whether the configuration is universal or existential, and in the latter case the superscript indicates which choice has been made for the existential state. Finally, the first element of the first successor configuration of a universal configuration is labeled with Z . In particular, \mathfrak{A}^* only interprets the symbols in τ non-empty. Now, we obtain structures \mathfrak{A}_k , $k < 2^n$ from \mathfrak{A}^* by interpreting non- τ -symbols as follows:

- the entire domain of \mathfrak{A}_k satisfies I ;
- the root of \mathfrak{A}_k has A -counter k ;
- the U -counter starts at 0 at the root and counts modulo 2^n along each S -path;
- the V -counter starts at k at the root and counts modulo 2^n along each S -path;
- the auxiliary concept names of the shape A_σ^i and A'_σ are interpreted in a minimal way so as to satisfy the sentences listed above. Note that the sentences are Horn, thus there is no choice.

Now obtain \mathfrak{A} from \mathfrak{A}^* and the \mathfrak{A}_k as follows. First, create a both side infinite R -path

$$\dots b_{-2} R b_{-1} R b_0 R b_1 R b_2 \dots$$

and realize the corresponding A -counter along the path and label every b_{3k} , $k \in \mathbb{Z}$, with X . Then, add all \mathfrak{A}_k^* to every node b_{3k} , $k \in \mathbb{Z}$, on the path by identifying the roots of the \mathfrak{A}_k with the respective node on the path. Moreover, add to \mathfrak{A} three elements a_0, a_1, a_2 such that $(a_0, a_1, a_2) \in G^{\mathfrak{A}}$, $(a_0, a_1), (a_1, a_2), (a_2, a_0) \in R^{\mathfrak{A}}$ and $a_0 \in X^{\mathfrak{A}}$. Finally, add a copy of \mathfrak{A}^* to \mathfrak{A} by identifying the root of \mathfrak{A}^* with a_0 . We claim that \mathfrak{A} is as required. In particular, \mathfrak{A} is a model of φ and the set S of all mappings

- $(a_i, a_{i+1}) \mapsto (b_{i+3k}, b_{i+3k})$ with $k \in \mathbb{Z}$, $i \in \{0, 1, 2\}$, and $a_3 := a_0$,
- $(e, f) \mapsto (e', f')$ with $(e, f) \in S^{\mathfrak{B}}$ and e', f' copies of e, f in some \mathfrak{A}_k , and
- all restrictions of the above,

is a guarded τ -bisimulation on \mathfrak{A} with $a_0 \mapsto b_0 \in S$.

(\Leftarrow) Let $\mathfrak{A}, \mathfrak{B}$ be a models of φ such that $\mathfrak{A}, a \sim_{\text{GF}, \tau} \mathfrak{B}, b$ for some elements a, b with $a \in A^{\mathfrak{A}}$, $b \notin A^{\mathfrak{B}}$. As it was argued above, due to the three-element R -loop enforced at a via φ_0 , from b there has to be an outgoing infinite R -path on which all S -trees are guarded τ -bisimilar. (There is also an incoming infinite R -path with this property, but it is not relevant for the proof.) All those S -trees are additionally labeled with some auxiliary relation symbols not in τ , depending on the distance from b . However, it can be shown using the arguments that accompanied the construction of φ that all S -trees contain a computation tree of M on input w . Hence, M accepts w . \square

6.2 3EXPTIME lower bound

We reduce the word problem of 2^{2^n} -space bounded ATMs using the very same idea as in the previous section. However, we need double exponential counters (instead of single exponential ones above). These counters are encoded in a way similar to the 2EXPTIME-hardness proof for satisfiability in the guarded fragment [16]. The mentioned encoding is based on *pairs of elements*, so we “lift” the above reduction to pairs of elements and consequently double the arity of all involved symbols. More precisely, we set

$$\begin{aligned} \theta(x) &= A(x, y) \\ \tau &= \{R, S, X, Z, B_{\forall}, B_{\exists}^1, B_{\exists}^2\} \cup \{A_{\sigma} \mid \sigma \in \Gamma \cup (Q \times \Gamma)\}, \end{aligned}$$

where R, S are 4-ary relation symbols, and the remaining symbols are binary. The sentence φ is a conjunction of several sentences. The first conjunct, φ_0 below, enforces that every pair of elements satisfying A is involved in a three-element R -loop as follows:⁴

$$\begin{aligned} \varphi_0 &= \forall xx' (A(xx') \rightarrow \exists yy' zz' (G(xx' yy' zz') \wedge X(xx') \wedge \\ &\quad R(xx' yy') \wedge R(yy' zz') \wedge R(zz' xx'))) \end{aligned}$$

⁴ We drop the commas in atoms whenever no confusion can arise.

Similar to above, we aim to construct φ such that M accepts w iff there exist models \mathfrak{A} and \mathfrak{B} of φ and tuples \mathbf{a}, \mathbf{b} such that $\mathbf{a} \in A^{\mathfrak{A}}, \mathbf{b} \notin A^{\mathfrak{B}}$, and $\mathfrak{A}, \mathbf{a} \sim_{\text{GF}, \tau} \mathfrak{B}, \mathbf{b}$. If the latter holds then from $\mathbf{a} \in A^{\mathfrak{A}}$ and φ_0 it follows that \mathbf{b} has an infinite outgoing path ρ along R on which every third pair of element satisfies X and is guarded τ -bisimilar to \mathbf{a} . Let us call these pairs the X -pairs. Observe that all X -pairs are guarded τ -bisimilar.

The main difference to the reduction above is the realization of the counters, so we will concentrate on this and leave the (straightforward) remainder of the proof to the reader. For realizing the A -counter, we use an n -ary relation symbol D and associate a counter to every pair of elements (a, a') as follows. We assume the order $a < a'$ which induces an order $<$ on tuples $\mathbf{a} \in \{a, a'\}^n$. Thus, every tuple $\mathbf{a} \in \{a, a'\}^n$ corresponds to a number $r(\mathbf{a}) < 2^n$, the rank of \mathbf{a} according to $<$. Now the sequence of truth values on all these tuples in D can be viewed as the binary representation of a number $< 2^{2^n}$.

The A -counter along the R -path ρ is enforced by the following sentences:

$$\begin{aligned} & \forall xx'yy' (R(xx'yy') \rightarrow (\neg A(xx') \rightarrow I(xx'))) \\ & \forall xx'yy' (R(xx'yy') \rightarrow (I(xx') \leftrightarrow I(yy'))) \\ & \forall xx'yy' (R(xx'yy') \wedge I(xx') \rightarrow (X(xx') \wedge \neg X(yy') \rightarrow \text{Eq}(xx'yy'))) \\ & \forall xx'yy' (R(xx'yy') \wedge I(xx') \rightarrow (\neg X(xx') \wedge \neg X(yy') \rightarrow \text{Eq}(xx'yy'))) \\ & \forall xx'yy' (R(xx'yy') \wedge I(xx') \rightarrow (\neg X(xx') \wedge X(yy') \rightarrow \text{Succ}(xx'yy'))) \end{aligned}$$

Again, the I acts as an additional guard that disables the counting at \mathbf{a} . It remains to define the formulas $\text{Eq}(xx'yy')$ and $\text{Succ}(xx'yy')$. For doing so, assume for the moment that we have available a $(4n + 4)$ -ary predicate E such that, for pairs a, a' and b, b' where b, b' represents a successor node of a, a' , and for $\mathbf{a}, \mathbf{a}' \in \{a, a'\}^n$ and $\mathbf{b}, \mathbf{b}' \in b, b'^n$, we have

$$E(\mathbf{a}\mathbf{a}'\mathbf{a}\mathbf{a}'\mathbf{b}\mathbf{b}'\mathbf{b}\mathbf{b}') \text{ holds if and only if } r(\mathbf{a}) = r(\mathbf{b}) \text{ and } r(\mathbf{a}') = r(\mathbf{b}'). \quad (1)$$

Then the formulas Eq and Succ are defined by:

$$\begin{aligned} \text{Eq}(xx'yy') &= \forall \mathbf{x}\mathbf{y}\mathbf{x}'\mathbf{y}' (E(\mathbf{x}\mathbf{x}'xx'\mathbf{y}\mathbf{y}'bb') \rightarrow (D(\mathbf{x}) \leftrightarrow D(\mathbf{y}))) \\ \text{Succ}(xx'yy') &= \exists \mathbf{x}\mathbf{y} (E(\mathbf{x}\mathbf{x}xx'\mathbf{y}\mathbf{y}bb') \wedge \neg D(\mathbf{x}) \wedge D(\mathbf{y}) \\ &\quad \wedge \forall \mathbf{x}'\mathbf{y}' (E(\mathbf{x}\mathbf{x}'xx'\mathbf{y}\mathbf{y}'bb') \rightarrow (\text{less}(\mathbf{x}'\mathbf{x}xx') \rightarrow D(\mathbf{x}') \wedge \neg D(\mathbf{y}')))) \\ &\quad \wedge \forall \mathbf{x}'\mathbf{y}' (E(\mathbf{x}\mathbf{x}'xx'\mathbf{y}\mathbf{y}'bb') \rightarrow (\text{less}(\mathbf{x}\mathbf{x}'xx') \rightarrow (D(\mathbf{x}') \leftrightarrow D(\mathbf{y}'))))) \end{aligned}$$

where, for $\mathbf{x} = x_0 \dots x_{n-1}$ and $\mathbf{x}' = x'_0 \dots x'_{n-1}$, we have

$$\text{less}(\mathbf{x}\mathbf{x}'xx') = \bigvee_{i < n} (x'_i = x' \wedge x_i = x \wedge \bigwedge_{j > i} x_j = x'_j).$$

Thus, $\text{less}(\mathbf{x}\mathbf{x}'xx')$ compares the positions of \mathbf{x} and \mathbf{x}' according to the order $x < x'$. Moreover, $\text{Eq}(xx'yy')$ is true iff the counters stipulated by x, x' and y, y' have precisely the same bits set. Finally, $\text{Succ}(xx'yy')$ asserts the existence of a

position k such that (i) in the counter stipulated by x, x' bit k is set to 0 while in the counter stipulated by y, y' bit k is set to 1, (ii) on all positions k' greater than k , the bits in the former counter are 1 while the bits in the latter counter are 1, and (iii) on all positions k' smaller than k the counters agree on their bits.

Let us finally axiomatize the predicate E as announced above. We abbreviate the tuples $\mathbf{x}x'$ and $\mathbf{y}y'$ with \mathbf{u} and \mathbf{v} , respectively; thus $\mathbf{u} = u_0 \dots u_{2n-1}$ and $\mathbf{v} = v_0 \dots v_{2n-1}$ are tuples of length $2n$. Moreover, let Σ be the set of all substitutions $[u_i/x, v_i/y]$ and $[u_i/x', v_i/y']$, for all $i < 2n$. Now, add the following sentences:

$$\begin{aligned} & \forall xx'yy' (R(xx'yy') \rightarrow E(x^{2n}xx'y^{2n}yy')) \\ & \forall \mathbf{u}xx'\mathbf{v}yy' (E(\mathbf{u}xx'\mathbf{v}yy') \rightarrow \bigwedge_{\sigma \in \Sigma} E(\sigma(\mathbf{u})xx'\sigma(\mathbf{v})yy')) \\ & \forall \mathbf{u}xx'\mathbf{v}yy' (E(\mathbf{u}xx'\mathbf{v}yy') \rightarrow \bigwedge_{i < 2n} (u_i = x \wedge v_i = y) \vee (u_i = x' \wedge v_i = y')) \end{aligned}$$

These sentences axiomatize E as required, since the last sentence enforces “only if” of Property (1) while the first and second sentence together enforce “if”.

The proof now proceeds along the lines of the proof given in the previous section, always replacing single elements/variables with pairs of elements/variables as exemplified above. Additionally, we need another set of sentences with copies D_U, D_V of D and E_U, E_V of E for the U and V counters. We finish noting that $\text{Min}_U(xx')$ can be expressed by

$$\text{Min}_U(xx') = \forall \mathbf{x} (D_U(\mathbf{x}) \rightarrow \bigvee_{i < n} (x_i \neq x \wedge x_i \neq x')).$$

7 Conclusion

We have determined the computational complexity of deciding the existence of interpolants and explicit definitions in the guarded fragment. To the best of our knowledge, this is the first study of the complexity of interpolant existence and explicit definition existence for a logic without the CIP and PBDP, respectively; the only exception being a recent article considering both problems for description logics with nominals [3]. Interestingly, also for DLs with nominals interpolant existence and explicit definition existence are one exponential harder than validity. Is this the rule for logics that do not enjoy the CIP nor PBDP? An interesting candidate to consider is the two-variable fragment of FO that also does not enjoy the CIP nor PBDP. Other problems that arise from this work include: what is the size of interpolants/explicit definitions if they exist? Note that recently the size and computation of interpolants in the guarded negation fragment (GNF) has been studied in depth in [8]. In contrast to GF, GNF enjoys the CIP and PBDP and it is not difficult to show using the complexity lower bound proof given above that in GF minimal interpolants/explicit definitions are, in the worst case, at least by one exponential larger than in GNF. We conjecture that the techniques introduced in the upper bound proof can be used to

prove that this bound is tight. It would also be of interest to see what happens if constants are added to GF (in this case even GNF does not enjoy the CIP and no decidable extension of GF does [10]) and whether the Horn fragment of GF introduced in [27] behaves better with regards to interpolants and explicit definitions than full GF. Note that the formulas constructed in the lower bound proof are not in the Horn fragment of GF.

References

1. Andréka, H., Németi, I., van Benthem, J.: Modal languages and bounded fragments of predicate logic. *J. Philosophical Logic* 27(3), 217–274 (1998)
2. Areces, C., Blackburn, P., Marx, M.: Hybrid logics: Characterization, interpolation and complexity. *J. Symb. Log.* 66(3), 977–1010 (2001)
3. Artale, A., Jung, J.C., Mazzullo, A., Ozaki, A., Wolter, F.: Living without beth and craig: Explicit definitions and interpolants in description logics with nominals. In: submitted (2020)
4. Bárány, V., Benedikt, M., ten Cate, B.: Rewriting guarded negation queries. In: *Proc. of MFCS*. pp. 98–110. Springer (2013)
5. Bárány, V., Benedikt, M., ten Cate, B.: Some model theory of guarded negation. *J. Symb. Log.* 83(4), 1307–1344 (2018)
6. Bárány, V., ten Cate, B., Segoufin, L.: Guarded negation. *J. ACM* 62(3), 22:1–22:26 (2015)
7. Bárány, V., Gottlob, G., Otto, M.: Querying the guarded fragment. *Logical Methods in Computer Science* 10(2) (2014)
8. Benedikt, M., ten Cate, B., Vanden Boom, M.: Effective interpolation and preservation in guarded logics. *ACM Trans. Comput. Log.* 17(2), 8:1–8:46 (2016)
9. Benedikt, M., Leblay, J., ten Cate, B., Tsamoura, E.: *Generating Plans from Proofs: The Interpolation-based Approach to Query Reformulation*. Synthesis Lectures on Data Management, Morgan & Claypool Publishers (2016)
10. ten Cate, B.: Interpolation for extended modal languages. *J. Symb. Log.* 70(1), 223–234 (2005)
11. ten Cate, B., Franconi, E., Seylan, I.: Beth definability in expressive description logics. *J. Artif. Intell. Res.* 48, 347–414 (2013)
12. Chandra, A.K., Kozen, D.C., Stockmeyer, L.J.: Alternation. *J. ACM* 28, 114–133 (1981)
13. Cimatti, A., Griggio, A., Sebastiani, R.: Interpolant generation for UTVPI. In: *Proc. of CADE*. pp. 167–182. Springer (2009)
14. D’Agostino, G., Lenzi, G.: Bisimulation quantifiers and uniform interpolation for guarded first order logic. *Theor. Comput. Sci.* 563, 75–85 (2015)
15. Goel, A., Krstic, S., Tinelli, C.: Ground interpolation for combined theories. In: *Proc. of CADE*. pp. 183–198. Springer (2009)
16. Grädel, E.: On the restraining power of guards. *J. Symb. Log.* 64(4), 1719–1742 (1999)
17. Grädel, E., Hirsch, C., Otto, M.: Back and forth between guarded and modal logics. In: *Proc. of LICS*. pp. 217–228. IEEE Computer Society (2000)
18. Grädel, E., Otto, M.: The freedoms of (guarded) bisimulation. In: *Johan van Benthem on Logic and Information Dynamics*, pp. 3–31. Springer International Publishing (2014)

19. Grädel, E., Walukiewicz, I.: Guarded fixed point logic. In: Proc. of LICS. pp. 45–54. IEEE Computer Society (1999)
20. Hernich, A., Lutz, C., Papacchini, F., Wolter, F.: Dichotomies in ontology-mediated querying with the guarded fragment. *ACM Trans. Comput. Log.* 21(3), 1–47 (2020)
21. Hoogland, E., Marx, M.: Interpolation in guarded fragments. Technical report, University of Amsterdam (1999)
22. Hoogland, E., Marx, M.: Interpolation and definability in guarded fragments. *Studia Logica* 70(3), 373–409 (2002)
23. Hoogland, E., Marx, M., Otto, M.: Beth definability for the guarded fragment. In: Proc. of LPAR. pp. 273–285. Springer (1999)
24. Jung, J.C., Lutz, C., Martel, M., Schneider, T., Wolter, F.: Conservative extensions in guarded and two-variable fragments. In: Proc. of ICALP. pp. 108:1–108:14. Schloss Dagstuhl – LZI (2017)
25. Jung, J.C., Lutz, C., Pulcini, H., Wolter, F.: Separating positive and negative data examples by concepts and formulas: The case of restricted signatures. In: submitted (2020)
26. Jung, J.C., Lutz, C., Pulcini, H., Wolter, F.: Logical separability of incomplete data under ontologies. In: Proc. of KR (2020)
27. Jung, J.C., Papacchini, F., Wolter, F., Zakharyashev, M.: Model comparison games for horn description logics. In: Proc. of LICS. pp. 1–14. IEEE (2019)
28. Krahmer, E., van Deemter, K.: Computational generation of referring expressions: A survey. *Computational Linguistics* 38(1), 173–218 (2012)
29. Lehmann, J., Hitzler, P.: Concept learning in description logics using refinement operators. *Machine Learning* 78, 203–250 (2010)
30. Maksimova, L., Gabbay, D.: Interpolation and Definability in Modal and Intuitionistic Logics. Clarendon Press (2005)
31. Martins, D.M.L.: Reverse engineering database queries from examples: State-of-the-art, challenges, and research opportunities. *Information Systems* (2019)
32. Marx, M.: Queries determined by views: Pack your views. In: Proc. of PODS. p. 2330. ACM (2007)
33. McMillan, K.L.: Interpolation and sat-based model checking. In: Proc. of CAV. pp. 1–13. Springer (2003)
34. Otto, M.: Highly acyclic groups, hypergraph covers, and the guarded fragment. *J. ACM* 59(1), 5:1–5:40 (2012), <https://doi.org/10.1145/2108242.2108247>
35. Toman, D., Weddell, G.E.: Fundamentals of Physical Design and Query Compilation. Synthesis Lectures on Data Management, Morgan & Claypool Publishers (2011)
36. Van Benthem, J.: The many faces of interpolation. *Synthese* pp. 451–460 (2008)