A Polynomial Translation from the Two-Variable Guarded Fragment with Number Restrictions to the Guarded Fragment

Yevgeny Kazakov

MPI für Informatik, D-66123 Saarbrücken, Germany
ykazakov@mpi-sb.mpg.de

Abstract. We consider a two-variable guarded fragment with number restrictions for binary relations and give a satisfiability preserving transformation of formulas in this fragment to the three-variable guarded fragment. The translation can be computed in polynomial time and produces a formula that is linear in the size of the initial formula even for the binary coding of number restrictions. This allows one to reduce reasoning problems for many description logics to the satisfiability problem for the guarded fragment.

1 Introduction and Motivation

The guarded fragment $\mathcal{GF}$ has been introduced by Andréka, van Benthem & Németi (1998) as a first-order counterpart of modal and description logics. Its basic principle is to restrict the usage of quantifiers to the following bounded forms: $\forall\overline{x}.(G \rightarrow F)$ or $\exists\overline{x}.(G \wedge F)$, where G is an atom-"*guard*" that contains all free variables of F. The guarded fragment inherits many nice computational properties from modal logics, the most important of them are the (generalized) tree-model property and the decidability.

Although the guarded fragment covers a considerable part of logical formalisms, some decidable extensions of modal logics, in particular by functionality and transitivity, remain outside of the guarded fragment. Grädel (1999*b*) has shown that already the three-variable guarded fragment $\mathcal{GF}^3$ with one functional relation becomes undecidable. In this paper we study an extension of the two-variable guarded fragment $\mathcal{GF}^2$ by number restrictions, which we denote by $\mathcal{GF}^2\mathcal{N}$. Number restrictions generalize functionality restrictions. They are often used for representing configuration constraints, like, for example:

$$\texttt{Client}(x) \equiv \texttt{Computer}(x) \wedge \exists^{\geq 1}\ y.\texttt{Connection}(x,y),$$
$$\texttt{Server}(y) \equiv \texttt{Computer}(y) \wedge \exists^{<100} x.\texttt{Connection}(x,y).$$

$\mathcal{GF}^2\mathcal{N}$ relates to many known description logics, the most expressive of which is $\mathcal{ALCQI}b$. Cardinality and qualifying number restrictions are hard to handle, especially for big numbers, because not many efficient optimization techniques are known for them (the notable exceptions are algebraic methods developed by Haarslev, Timmann & Möller 2001, Haarslev & Möller 2001). This contrasts to the fact that $\mathcal{ALCQI}$ and even $\mathcal{ALCQI}b$ has the same complexity as $\mathcal{ALC}$, namely EXPTIME (Tobies 2001). In this paper we propose a translation which allows one to reason about number restrictions through the guarded fragment.

The contribution of this paper can be summarized as follows. First, we show that reasoning in $\mathcal{ALCQI}b$ knowledge bases can be polynomially reduced to satisfiability of $\mathcal{GF}^2\mathcal{N}$-formulas. Second, we give a satisfiability preserving translation from $\mathcal{GF}^2\mathcal{N}$ to $\mathcal{GF}^3$. The translation is computable in polynomial time and produces a guarded formula of linear size, even if number restrictions are coded in binary. This result has both theoretical and practical implications. From the theoretical side, we obtain a neat complexity result for satisfiability of $\mathcal{GF}^2\mathcal{N}$-formulas. Complexity results for $\mathcal{GF}^3$, imply that satisfiability of $F \in \mathcal{GF}^2\mathcal{N}$ can be decided in time $2^{O(|F|)}$. On the practical side, our translation provides a bridge between the fragment $\mathcal{GF}^2\mathcal{N}$ and the formalism for which optimization techniques are relatively well-studied. This makes it possible to use any decision procedure for the guarded fragment (Ganzinger & de Nivelle 1999, Hladik 2002), for deciding $\mathcal{GF}^2\mathcal{N}$ and reasoning in description logics.

2 The Guarded Fragment and Number Restrictions

Throughout this paper we assume a standard logical notation for first-order logic with equality and description logics, and the correspondence between the syntax of description logics and the first-order syntax (that is, role names correspond to binary predicates, $\sqcap$ to conjunction, etc.). For an interpretation $\mathcal{I} = (D, \cdot^{\mathcal{I}})$, we denote by $\mathcal{I}|_S$ the interpretation induced by a subset $S \subseteq D$ of domain elements. We say that a formula F' is *conservative over* F if ***(i)*** every model of F can be expanded (by interpreting new predicate symbols) to a model of F' and ***(ii)*** F is a logical consequence of F'. A transformation $F \Longrightarrow F'$ is called *conservative* if F' is conservative over F. The *size* $|F|$ ($|C|$, $|T|$) of a formula F (a concept C or a TBox T) is its length when an appropriate coding of numbers (binary or unary) is assumed. The *width* $wd(F)$ of a *formula* F is the maximal number of free variables in subformulas of F.

The *guarded fragment* (with equality) can be defined by the grammar:

$$\mathcal{GF} ::= \mathtt{A} \mid \neg\mathtt{F}_1 \mid \mathtt{F}_1 \vee \mathtt{F}_2 \mid \mathtt{F}_1 \wedge \mathtt{F}_2 \mid \forall\overline{x}.(\mathtt{G} \rightarrow \mathtt{F}_1) \mid \exists\overline{x}.(\mathtt{G} \wedge \mathtt{F}_1). \qquad (1)$$

where $\mathtt{A}$ is an atom, $\mathtt{F}_i, i = 1, 2$ are guarded formulas, and $\mathtt{G}$ is an atom called *the guard* containing all free variables of $\mathtt{F}_1$. We assume that for $\mathtt{F}(x) \in \mathcal{GF}$, the formula $\forall x.\mathtt{F}(x)$ is also guarded, since it can be equivalently written in the form $\forall x.[(x \simeq x) \rightarrow \mathtt{F}(x)] \in \mathcal{GF}$. As usual we assume that implication and equivalence are expressible by means of the other boolean connectives. The bounded-variable variant $\mathcal{GF}^k$ of the guarded fragment is the set the guarded formulas that use at most k variable names. Note that for every $F \in \mathcal{GF}^k$, $wd(F) \leq k$.

2.1 The two-variable guarded fragment with number restrictions

In this paper we consider an extension of the two-variable guarded fragment $\mathcal{GF}^2$ with expressions of the form $\exists^{\geq n} y.e(x, y)$ and $\exists^{<n} y.e(x, y)$, where n is a natural

number and e is a binary atom. These expressions are called *"at-most"* and *"at-least" number restrictions* respectively. The semantics of number restrictions can be first-order defined as follows: $\exists^{<n}y.e(x,y) \equiv \neg[\exists^{\geq n}y.e(x,y)]$ where

$$\exists^{\geq n}y.e(x,y) \quad \equiv \quad \exists y_1 y_2 \ldots y_n.[\bigwedge_{1\,\leq\, i\,\leq\, n} e(x,y_i) \;\wedge \bigwedge_{1\,\leq\, i\,<\, j\,\leq\, n} y_i \not\simeq y_j].$$

The meaning of "at-most" and "at-least" number restrictions is that they require or respectively restrict the number of different elements in a model that can be connected by a binary relation with a given element.

The *two-variable guarded fragment with number restrictions* is defined by:

$$\mathcal{GF}^2\mathcal{N} ::= \mathtt{A} \mid \exists^{\bowtie n}y.e(x,y) \mid \neg\mathtt{F}_1 \mid \mathtt{F}_1 \mathbin{⩙} \mathtt{F}_2 \mid \forall\overline{x}.(\mathtt{G}\rightarrow\mathtt{F}_1) \mid \exists\overline{x}.(\mathtt{G}\wedge\mathtt{F}_1), \tag{2}$$

where $\exists^{\bowtie n}y.e(x,y)$ are number restrictions, $⩙$ stands for conjunction or disjunction and only variables from $\{x,y\}$ can be used in formulas. In the sequel we will work with conjunctions of number restrictions of the form:

$$\mathcal{N}(x) \equiv \bigwedge_{l\in L} \exists^{\geq n_l}y.e_l(x,y) \wedge \bigwedge_{m\in M} \exists^{<n_m}y.e_m(x,y). \tag{3}$$

where L and M are disjoint finite sets of indexes, n_l, n_m are natural numbers and e_l, e_m are binary predicate symbols for $l \in L$, $m \in M$.

2.2 A relationship between $\mathcal{GF}^2\mathcal{N}$ and $\mathcal{ALCQI}b$

The fragment $\mathcal{GF}^2\mathcal{N}$ is closely related to the description logic $\mathcal{ALCQI}b$ introduced by Tobies (2001). $\mathcal{ALCQI}b$ extends the description logic $\mathcal{ALCQI}$ with the "safe" boolean combinations of roles. The unrestricted boolean combinations of roles are defined from the atomic roles and their inverses by the grammar:[1]

$$\mathcal{R}_B ::= R_a \mid R_a^{-1} \mid \neg R_a \mid \neg R_a^{-1} \mid R_B^1 \sqcap R_B^2 \mid R_B^1 \sqcup R_B^2,$$

whereas the "safe" boolean combinations of roles are restricted to:[2]

$$\mathcal{R}_b ::= R_a \mid R_a^{-1} \mid R_b^1 \sqcap R_B^1 \mid R_b^1 \sqcup R_b^2 \quad \text{where } R_b^1, R_b^2 \in \mathcal{R}_b,\ R_B^1, R_B^2 \in \mathcal{R}_B.$$

The set of $\mathcal{ALCQI}b$-concepts is defined in the same way as in $\mathcal{ALCQI}$, except that now any "safe" role expressions R can be used as a role filler:

$$\mathcal{C} ::= \mathtt{A} \mid \neg C_1 \mid C_1 \sqcap C_2 \mid \exists^{\geq n}R.C_1, \tag{4}$$

where $\mathtt{A}$ is an atomic concept, $C_1, C_2 \in \mathcal{C}$, $R \in R_b$ and n is a natural number.

Tobies (2001) has demonstrated that concept satisfiability w.r.t. general $\mathcal{ALCQI}b$ TBoxes is EXPTIME-complete, even if the numbers in qualifying number restrictions are coded in binary. The result has been obtained by a reduction to the emptiness problem for looping tree automata. Given a TBox T and a concept C, an automaton L can be constructed such that L accepts a tree *iff* C has a T-model. The emptiness of L can be verified in polynomial time in the size of $|L|$, which gives an algorithm for checking T-satisfiability of C (see Fig. 1). However

[1] For convenience, we define boolean combinations that are in negation normal form.

[2] In the original definition (Tobies 2001), a boolean combination of roles is "safe" if its disjunction normal form contains a positive conjunct in every disjunct.

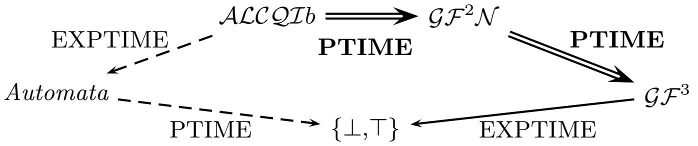

Fig. 1. The outline of decision procedures for $\mathcal{ALCQI}b$ and $\mathcal{GF}^2\mathcal{N}$: the dashed arrows represent an automata-based approach; the solid arrows represent a decision procedure through the guarded fragment. We give the translations shown by the double arrows

L can be exponentially large in $|C|+|T|$ in worst case and this blow-up cannot be avoided, since the problem is EXPTIME-hard. Moreover, the translation can be exponential in many trivial cases, say, when T contains lots of definitions that are not relevant to C, which is often the case in real knowledge bases. Therefore, a straightforward implementation of the automata-based decision procedure seems to be not very useful in practice. In this paper we propose an alternative decision procedure through the guarded fragment (see Fig. 1). This approach might be more efficient yet having the optimal complexity. As a first step of our procedure we establish a correspondence between $\mathcal{ALCQI}b$-knowledge bases and $\mathcal{GF}^2\mathcal{N}$.

Lemma 1. *For every $\mathcal{ALCQI}b$ TBox T one can construct in polynomial time a formula $F \in \mathcal{GF}^2\mathcal{N}$ that is conservative over T such that $|F| = O(|T|)$.*

Proof. Without loss of generality, we may assume that T contains only simple definitions, that is, definitions corresponding to one case in (4). Moreover, we introduce auxiliary role names $\mathtt{R}_\mathtt{b}$ for every compound subrole $R_b \in \mathcal{R}_b$ that occurs in T. We extend T with definitions for $\mathtt{R}_\mathtt{b}$ and use them in complex expressions instead of R_b. We also introduce additional binary predicate symbol $\mathtt{R}_\mathtt{C}$ for every definition of qualifying number restrictions $\mathtt{C} \doteq \exists^{\geq n}\mathtt{R}.\mathtt{C}_1$. The definitions from T and their first-order translations are given on Fig. 2. It is easy to see that the translation maps TBox T to a conjunction of formulas from $\mathcal{GF}^2\mathcal{N}$. □

Corollary 1. *The problem of concept satisfiability w.r.t. $\mathcal{ALCQI}b$ TBox-es is polynomially reducible to satisfiability of $\mathcal{GF}^2\mathcal{N}$-formulas.*

$\mathtt{C} \doteq \mathtt{A}$	$\Rightarrow \forall x.[\mathtt{C}(x) \leftrightarrow \mathtt{A}(x)]$	$\mathtt{R}_\mathtt{b} \doteq \mathtt{R}_\mathtt{a}$	$\Rightarrow \forall xy.[\mathtt{R}_\mathtt{b}(x,y) \leftrightarrow \mathtt{R}_\mathtt{a}(x,y)]$
$\mathtt{C} \doteq \neg\mathtt{C}_1$	$\Rightarrow \forall x.[\mathtt{C}(x) \leftrightarrow \neg\mathtt{C}_1(x)]$	$\mathtt{R}_\mathtt{b} \doteq {\mathtt{R}_\mathtt{a}}^{-1}$	$\Rightarrow \forall xy.[\mathtt{R}_\mathtt{b}(x,y) \leftrightarrow \mathtt{R}_\mathtt{a}(y,x)]$
$\mathtt{C} \doteq \mathtt{C}_1 \sqcap \mathtt{C}_2$	$\Rightarrow \forall x.(\mathtt{C}(x) \leftrightarrow [\mathtt{C}_1(x) \wedge \mathtt{C}_2(x)])$		

$\mathtt{C} \doteq \exists^{\geq n}\mathtt{R}.\mathtt{C}_1$	$\Rightarrow \forall x.(\mathtt{C}(x) \rightarrow \exists^{\geq n}y.\mathtt{R}_\mathtt{C}(x,y)) \wedge \forall xy.(\mathtt{R}_\mathtt{C}(x,y) \rightarrow [\mathtt{R}_\mathtt{C}(x,y) \wedge \mathtt{C}_1(y)]) \wedge$ $\wedge \forall x.(\exists^{\geq n}y.\mathtt{R}_\mathtt{C}(x,y) \rightarrow \mathtt{C}(x)) \wedge \forall xy.(\mathtt{R}(x,y) \rightarrow [\mathtt{C}_1(y) \rightarrow \mathtt{R}_\mathtt{C}(x,y)])$
$\mathtt{R}_\mathtt{b} \doteq \mathtt{R}^1_\mathtt{b} \sqcap R^1_B$	$\Rightarrow \forall xy.(\mathtt{R}_\mathtt{b}(x,y) \rightarrow [\mathtt{R}^1_\mathtt{b}(x,y) \wedge R^1_B(x,y)]) \wedge$ $\wedge \forall xy.(\mathtt{R}^1_\mathtt{b}(x,y) \rightarrow [\mathtt{R}_\mathtt{b}(x,y) \vee \neg R^1_B(x,y)])$
$\mathtt{R}_\mathtt{b} \doteq \mathtt{R}^1_\mathtt{b} \sqcup \mathtt{R}^2_\mathtt{b}$	$\Rightarrow \forall xy.(\mathtt{R}_\mathtt{b}(x,y) \rightarrow [\mathtt{R}^1_\mathtt{b}(x,y) \vee \mathtt{R}^2_\mathtt{b}(x,y)]) \wedge$ $\wedge \forall xy.[\mathtt{R}^1_\mathtt{b}(x,y) \rightarrow \mathtt{R}_\mathtt{b}(x,y)] \wedge \forall xy.[\mathtt{R}^2_\mathtt{b}(x,y) \rightarrow \mathtt{R}_\mathtt{b}(x,y)]$

Fig. 2. A translation of $\mathcal{ALCQI}b$ TBox-es to $\mathcal{GF}^2\mathcal{N}$

It is not clear whether the expressive power of $\mathcal{ALCQI}b$ reaches $\mathcal{GF}^2\mathcal{N}$, since $\mathcal{ALCQI}b$ does not have a built-in equality. However, we conjecture, that $\mathcal{GF}^2\mathcal{N}$ without equality has the same expressive power as $\mathcal{ALCQI}b$.

Remark 1. In some papers on the guarded fragment, e.g. in (Grädel 1999*b*), atoms are allowed to contain individual constants. In this paper, however, we do not assume this. Reasoning with individual constants in the presence of number restrictions becomes much harder, since so-called *cardinality restrictions* can be expressed using them: The formula $\forall x.[a(x) \rightarrow e(c,x)] \wedge \forall x.\exists^{<n+1} y.e(x,y)$ has only models where a is satisfied in at most n elements. The extension of $\mathcal{ALCQI}$ with cardinality restrictions becomes NEXPTIME-complete already for the unary coding of numbers and the exact complexity for binary coding is still an open problem (for details see Tobies 2001). $\diamondsuit$

2.3 A normal form for formulas in $\mathcal{GF}^2\mathcal{N}$

To prove properties for guarded formulas it is convenient to have them in a simple normal form. We show that $\mathcal{GF}^2\mathcal{N}$-formulas have a Scott-like normal form that is similar to the one found by Grädel (1999*b*) for $\mathcal{GF}$.

Lemma 2. *For every formula $F \in \mathcal{GF}^2\mathcal{N}$ there is a formula F' of the form:*

$$\bigwedge_{i \in I} \forall xy.[g_i(x,y) \rightarrow \varphi_i(x,y)] \wedge \forall x.\mathcal{N}(x) \tag{5}$$

where $g_i(x,y)$ are atoms, $i \in I$, $\varphi_i(x,y)$ are quantifier-free formulas and $\mathcal{N}(x)$ are number restrictions of the form (3), such that: ***(i)*** *F' is conservative over F,* ***(ii)*** *$|F'| = O(|F|)$ and F' is computable in polynomial time from F.*

Proof. Given a guarded formula $F \in \mathcal{GF}^2\mathcal{N}$, first, we put F into *negation normal form (NNF)* by pushing all negations inside to atoms using the usual *de Morgan*'s laws. The resulting formula $[F]^{nnf}$ belongs to the following fragment:

$$[\mathcal{GF}^2\mathcal{N}]^{nnf} ::= (\neg)\mathtt{A} \mid \exists^{\bowtie n} y.e(x,y) \mid \mathtt{F}_1 \mathbin{\wedge\!\!\!\!\vee} \mathtt{F}_2 \mid \forall\overline{x}.(\mathtt{G} \rightarrow \mathtt{F}_1) \mid \exists\overline{x}.(\mathtt{G} \wedge \mathtt{F}_1).$$

we extend this fragment by dropping the restrictions for the existential part:

$$[\mathcal{GF}^2\mathcal{N}]^{nnf}_w ::= (\neg)\mathtt{A} \mid \exists^{\bowtie n} y.e(x,y) \mid \mathtt{F}_1 \mathbin{\wedge\!\!\!\!\vee} \mathtt{F}_2 \mid \forall\overline{x}.(\mathtt{G} \rightarrow \mathtt{F}_1) \mid \exists\boldsymbol{y}.\mathbf{F_1}.$$

thus, the existential closure $F^n := \exists\overline{x}.[F]^{nnf} \in [\mathcal{GF}^2\mathcal{N}]^{nnf}_w$. Note that F^n is conservative over F. After that, we apply a so-called *structural transformation* for the sentence F^n by introducing *definitions* for its subformulas. We assume that to every subformula $\mathtt{F}$ of F^n corresponding to a case in the recursive definition, a unique predicate $P_{\mathtt{F}} = p_{\mathtt{F}}(\overline{x})$ is assigned, where $\overline{x} = \mathit{free}[\mathtt{F}] \subseteq \{x,y\}$. If $\mathtt{F}$ is a number restriction $\exists^{\bowtie n} y.e(x,y)$, then we also introduce an auxiliary binary predicate $e_{\mathtt{F}}(x,y)$. The result of the structural transformation for F^n is the formula $P_{F^n} \wedge [F^n]^{st}$, where $[\mathtt{F}]^{st}$ is defined recursively for $\mathtt{F} \in [\mathcal{GF}]^{nnf}_w$ as follows:

$$\begin{array}{rrll}
[\mathtt{F}]^{st} := & [(\neg)\mathtt{A}]^{st} : & \forall\overline{x}.(P_{\mathtt{F}}\rightarrow(\neg)\mathtt{A}) & | \\
 & [\exists^{\geq n}y.e(x,y)]^{st} : & \forall x.\exists^{\geq n}y.e_{\mathtt{F}}(x,y)\wedge\forall xy.[e_{\mathtt{F}}(x,y)\rightarrow(p_{\mathtt{F}}(x)\rightarrow e(x,y))] & | \\
 & [\exists^{<n}y.e(x,y)]^{st} : & \forall xy.[e(x,y)\rightarrow(p_{\mathtt{F}}(x)\rightarrow e_{\mathtt{F}}(x,y))]\wedge\forall x.\exists^{<n}y.e_{\mathtt{F}}(x,y) & | \\
 & [\mathtt{F}_1\mathbin{\wedge\!\!\!\!\vee}\mathtt{F}_2]^{st} : & \forall\overline{x}.(P_{\mathtt{F}}\rightarrow[P_{\mathtt{F}_1}\mathbin{\wedge\!\!\!\!\vee}P_{\mathtt{F}_2}])\wedge[\mathtt{F}_1]^{st}\wedge[\mathtt{F}_2]^{st} & | \\
 & [\forall\overline{y}.(\mathtt{G}\rightarrow\mathtt{F}_1)]^{st} : & \forall\overline{xy}.(\mathtt{G}\rightarrow[P_{\mathtt{F}}\rightarrow P_{\mathtt{F}_1}])\wedge[\mathtt{F}_1]^{st} & | \\
 & [\exists y.\mathtt{F}_1]^{st} : & \forall\overline{x}.(P_{\mathtt{F}}\rightarrow\exists y.P_{\mathtt{F}_1})\wedge[\mathtt{F}_1]^{st}. &
\end{array}$$

The function $[\mathtt{F}]^{st}$ is defined recursively over the definition of $[\mathcal{GF}^2\mathcal{N}]^{nnf}_{w}$. On each step, subformulas of $\mathtt{F}$ are replaced by fresh atoms. These atoms are defined in separate conjuncts by means of subformulas that they replace. It is easy to see that the result of the transformation can be captured by a formula F' of the form (5) that is conservative over F^n. The formula F' has the size $O(|F^n|) = O(|F|)$ and the translation can be computed in polynomial time. □

2.4 The tree-model property for $\mathcal{GF}^2\mathcal{N}$

As Vardi (1996) has argued, the tree-model property is the main reason why modal logics are decidable. The existence of tree-models for satisfiable formulas is a basis of all tableau and automata-based decision procedures for modal and description logics. In many cases the tree-model property allows to establish a finite model property and even to extract bounds on the sizes of models for satisfiable formulas, which gives rise to model-enumeration techniques.

Definition 1. *An interpretation $\mathcal{M} = (D, \cdot^{\mathcal{I}})$ has a tree width k if k is the minimal natural number such that there exists a tree $T = (V, E)$ (a connected acyclic graph) and a function $\pi : V \rightarrow 2^D$ with $|\pi(v)| \leq k+1$ for every $v \in V$, such that the following conditions hold:*

(i) *$\mathcal{M} \vDash a(d_1,...,d_n)$ implies $\{d_1,...,d_n\} \subseteq \pi(v)$ for some $v \in V$; and*

(ii) *The set $\mathcal{O}(d) := \{v \in V \mid d \in \pi(v)\}$ induces a connected subtree in T.* ◊

Grädel (1999*a*) has shown that every satisfiable guarded formula with width k has a model of the tree width $k-1$. In particular, every satisfiable formula from $\mathcal{GF}^2$ has a tree-model (a model with the tree width = 1). We are going to extend this results to $\mathcal{GF}^2\mathcal{N}$ by taking into account number restrictions. For convenience, we use the following (equivalent) definition of a tree model:

Definition 2. *An interpretation $\mathcal{T} = (V, \cdot^{\mathcal{T}})$ is called a* tree *if there is a tree $T = (V, E)$, such that $\mathcal{T} \vDash a(v_1,...,v_n)$ implies $\{v_1,\ldots,v_n\} \subseteq \{v_1', v_2'\}$ for some edge $e = (v_1', v_2') \in E$.* ◊

A correspondence between these two definitions can be established by assigning $\pi(v)$ to be the set $\{v, v_{-1}\} \subseteq V$, where v_{-1} is a parent of v in T (we assume that the root of T is a parent of itself). For proving the tree-model property for $\mathcal{GF}^2\mathcal{N}$ we show how a tree-model can be extracted from any model of a $\mathcal{GF}^2\mathcal{N}$-formula. Surprisingly, a tree-model satisfying number restrictions can be extracted from interpretations that satisfy weaker conditions than number restrictions themselves.

Definition 3. *A* counting pattern *for a model $\mathcal{M} = (D, \cdot^{\mathcal{M}})$ and number restrictions $\mathcal{N}(x)$ is a pair $\mathcal{P} = (R, w)$, where $R \subseteq D \times D$ is a* reachability relation *and $w : R \to 2^R$ is a* witnessing function *such that: **(i)** $(d_r, d_r) \in R$ for some $d_r \in D$ and **(ii)** for any $(d_{-1}, d_0) \in R$, $w[(d_{-1}, d_0)] = \{(d_0, d_1), (d_0, d_2), \ldots, (d_0, d_b)\}$ for some $d_1, \ldots, d_b \in D \setminus \{d_{-1}, d_0\}$ such that $\mathcal{M}|_{\{d_{-1}, d_0, \ldots, d_b\}} \vDash \mathcal{N}(d_0)$.* ♢

Example 1. The fragment $\mathcal{GF}^2\mathcal{N}$ does not have the finite model property. To demonstrate this, consider the formula *BinaryTree* $\equiv \forall x.\mathcal{N}_{bt}(x)$, where $\mathcal{N}_{bt}(x) \equiv [\exists^{\geq 2} y.s(x, y) \wedge \exists^{<2} y.s(y, x)]$. It is easy to see that the formula *BinaryTree* is satisfiable in a tree-model $\mathcal{T}$ (see Fig 3), but has no finite models. However, there is a finite interpretation $\mathcal{M} = (D = \{a, b, c, d, e\}, \cdot^{\mathcal{M}})$ given on Fig. 3 that has a counting pattern for $\mathcal{N}_{bt}(x)$. The counting pattern $\mathcal{P}_{bt} = (R, w)$ for $\mathcal{N}_{bt}(x)$ in $\mathcal{M}$ can be defined by taking $R = D \times D$ and setting the witnessing function $w[(d_{-1}, d_0)]$ to return the remaining edges incident to the node d_0. For instance, $w[(e, a)] := \{(a, b), (a, c)\}$, $w[(d, b)] := \{(b, c)\}$. ♢

It will be shown below, that every model of number restrictions has a counting pattern. However, the converse does not hold: the interpretation $\mathcal{M}$ from Fig. 3 does not satisfy $\forall x.\mathcal{N}_{bt}(x)$, but has a counting pattern for $\mathcal{N}_{bt}(x)$. Yet, we show that a tree-model satisfying number restrictions $\mathcal{N}(x)$ can be *extracted* from any model having a counting pattern for $\mathcal{N}(x)$.

Lemma 3. *Let $\mathcal{N}(x)$ be number restrictions of the form (3), and $\mathcal{M} = (D, \cdot^{\mathcal{M}})$ be a model of $\forall x.\mathcal{N}(x)$. Then $\mathcal{M}$ has a counting pattern for $\mathcal{N}(x)$.*

Proof. The intended counting pattern $\mathcal{P} = (R, w)$ can be defined as follows. Let $R = D \times D$. This guarantees that condition ***(i)*** in Definition 3 holds since D is a non-empty set. Note that for every vector $(d_{-1}, d_0) \in R$ one can find a finite set of elements $d_1, d_2, \ldots, d_b$ such that $\mathcal{M}|_{\{d_{-1}, d_0, d_1, \ldots, d_b\}} \vDash \mathcal{N}(d_0)$, since $\mathcal{M} \vDash \mathcal{N}(d_0)$ and number restrictions require only finitely many witnesses. This suggests us to define a witnessing function by $w[(d_{-1}, d_0)] := \{(d_0, d_1), (d_0, d_2), \ldots, (d_0, d_b)\}$. This definition guarantees the remaining property ***(ii)*** of Definition 3. □

Lemma 4. *Let $F \equiv F_g \wedge \forall x.\mathcal{N}(x)$ be a $\mathcal{GF}^2\mathcal{N}$-formula in a normal form (5). Let $\mathcal{M}$ be a model of F_g that has a counting pattern for the number restrictions $\mathcal{N}(x)$. Then F has a tree-model.*

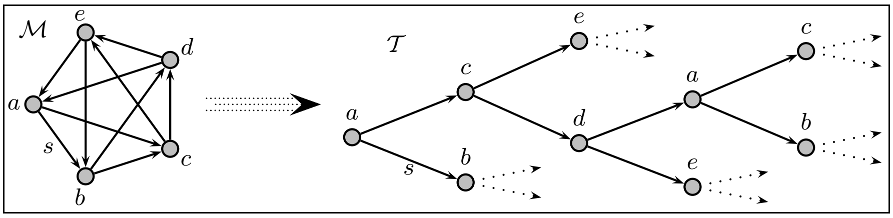

Fig. 3. Tree decomposition of an interpretation having a counting pattern

Proof. Given a model $\mathcal{M}$ for F_g and a counting pattern $\mathcal{P} = (R, w)$ for $\mathcal{N}(x)$ in $\mathcal{M}$, we construct a tree-model $\mathcal{T} = (V, \cdot^{\mathcal{T}})$ for F based on a tree $T = (V, E)$. The model $\mathcal{T}$ is constructed inductively together with a function $\pi : V \to D$ that represents a *bisimulation relation* between $\mathcal{M}$ and $\mathcal{T}$, such that for every node $v_0 \in V$ and its parent v_{-1}, the substructures $\mathcal{M}|_{\{v_{-1},v_0\}}$ and $\mathcal{T}|_{\{\pi(v_{-1}),\pi(v_0)\}}$ are isomorphic and $(\pi(v_{-1}), \pi(v_0)) \in R$.

First, we create a root $v_r \in V$ of a tree T and set $\pi(v_r) := d_r$, where $d_r \in D$ is given in Definition 3, so $(\pi(v_r), \pi(v_r)) \in R$. For every leaf v_0 of a tree T constructed so far and its parent v_{-1}, consider $(d_{-1}, d_0) = (\pi(v_{-1}), \pi(v_0)) \in R$ (by induction hypothesis), and let $d_1, \ldots, d_b \in D$ be such that $R(d_{-1}, d_0) = \{(d_0, d_1), (d_0, d_2), \ldots, (d_0, d_b)\}$. For every d_i with $1 \le i \le b$ we create a child v_i of v_0 in T, set $\pi(v_i) := d_i$, and extend the interpretation $\mathcal{T}$ in such a way that the substructure $\mathcal{T}|_{\{v_0,v_i\}}$ is isomorphic to the substructure $\mathcal{M}|_{\{d_0,d_i\}}$ for $1 \le i \le b$. This can be always done in a consistent way. Note that $\mathcal{T} \models \mathcal{N}(v_0)$ because $\mathcal{M}|_{\{d_{-1},d_0,d_1,\ldots,d_b\}} \models \mathcal{N}(d_0)$ and π is a bijection between the sets $\{v_{-1}, v_0, \ldots, v_b\}$ and $\{d_{-1}, d_0, \ldots, d_b\}$. The constructed interpretation $\mathcal{T}$ is a tree. It is a model of F_g since every edge of a tree is isomorphic to a substructure of the model $\mathcal{M}$ and since F_g is a guarded formula. Therefore $\mathcal{T} \models F$. □

The process described in the proof of Lemma 4 is known as a *tree decomposition* of a structure (Grädel 1999*a*). Figure 3 demonstrates the construction given in the proof of Lemma 4 for the interpretation $\mathcal{M}$ from Example 1. Now the tree-model property for $\mathcal{GF}^2\mathcal{N}$ is an easy consequence of Lemma 4:

Theorem 1. *Every satisfiable formula $F \in \mathcal{GF}^2\mathcal{N}$ has a tree-model.*

Proof. By Lemma 2, we may assume that F is of the form (5). Let $\mathcal{M}$ be a model for $F = F_g \wedge \forall x.\mathcal{N}(x)$. By Lemma 3, $\mathcal{M}$ has a counting pattern for $\mathcal{N}(x)$. Therefore, by Lemma 4, F has a tree-model. □

3 The Translation

In this section we give a polynomial-time translation mapping any formula $F \in \mathcal{GF}^2\mathcal{N}$ of the form (5) to a formula $F' \in \mathcal{GF}^3$ such that ***(i)*** every tree-model of F can be expanded to a model of F', ***(ii)*** for every model of F' one can construct a tree-model of F and ***(iii)*** $|F'| = O(|F|)$. Note that it is not possible to give a conservative translation from $\mathcal{GF}^2\mathcal{N}$ to $\mathcal{GF}^3$, since $\mathcal{GF}^2\mathcal{N}$ does not have a finite model property in contrast to $\mathcal{GF}^3$ (see Example 1).

We describe the translation in the following way. Given a formula $F = F_g \wedge \forall x.\mathcal{N}(x) \in \mathcal{GF}^2\mathcal{N}$ in the normal form (5) and a tree-model $\mathcal{T}$ for F, we expand the model $\mathcal{T}$ to by defining additional predicates to encode a counting pattern for $\mathcal{N}(x)$. Then we construct a formula $F' = F_g \wedge F'' \in \mathcal{GF}^3$ that describes the expanded model. Finally we show that every model of F' has a counting pattern for $\mathcal{N}(x)$. This will prove that F is satisfiable whenever F' is.

Let $\mathcal{T} = (V, \cdot^{\mathcal{T}})$ be a tree-model for a formula $F \equiv F_g \wedge \forall x.\mathcal{N}(x) \in \mathcal{GF}^2\mathcal{N}$ of the form (5). We introduce a new binary predicate $\mathtt{R}$ to encode the reachability relation of a counting pattern $\mathcal{P} = (R, w)$. $\mathtt{R}$ is interpreted in $\mathcal{T}$ by setting $\mathcal{T} \models \mathtt{R}(v_1, v_2)$ *iff* either $v_1 = v_2$ or v_2 is a child of the node v_1.

To encode the number restrictions, for every node $v \in V$ consider an ordered set $\mathcal{O}(v) = [(v_{-1},)\, v_0, v_1, ..., v_{b_v}]$ of *neighbors* of v in $\mathcal{T}$: The first element in $\mathcal{O}(v)$ should be the parent v_{-1} of v, which is followed by the node v itself: $v_0 := v$ (we assume that $v_{-1} = v_0$ for the root node). After that, all children of v are listed in some order $v_1, ..., v_{b_v}$: see Fig. 4. The order on the set of neighbors is used to count the number of edges satisfying the counting relations. We encode this order using a special ternary predicate symbol $\mathtt{Next}$ (this will be the only ternary relation that is used in our construction). The intended interpretation of $\mathtt{Next}$ is given by $\mathcal{T} \models \mathtt{Next}(v, v', v'')$ *iff* $v' = v_i$ and $v'' = v_{i+1}$ for some $v_i, v_{i+1} \in \mathcal{O}(v)$ with $0 \leq i \leq b_v$. In other words, $\mathtt{Next}(x, y, z)$ holds if and only if z is the child of x that comes directly after y.

For every index $c \in L \cup M$, $v \in V$ and $v_i \in \mathcal{O}(v)$, $1 \leq i \leq b_v$, we define the number $n_{c,i} := \#\{v_j \in \mathcal{O}(v) \mid j \leq i \;\&\; \mathcal{T} \models e_c(v, v_j)\}$, that is the number of times the relation e_c has been realized up to v_i. Note that for every $l \in L$ there exists i with $0 \leq i \leq b_v$ such that $n_{l,i} \geq n_l$, and for every $m \in M$ and every j with $0 \leq j \leq b_v$, $n_{m,j} < n_m$ since $\mathcal{T} \models \mathcal{N}(v)$. Moreover, the following property can be assumed for the set $\mathcal{O}(v)$. We say that a child v_i with $1 \leq i \leq b_v$ of the node v is *essential*, if whenever $n_{l,i} \leq n_l$ for some $l \in L$, then there exists an $l' \in L$ such that $\mathcal{T} \models e_{l'}(v_0, v_i)$ and $n_{l',i} \leq n_{l'}$. In other words, if some "at-least" number restrictions are not yet realized up to v_i, then the node v_i should contribute in one of them. By reordering the children of v in $\mathcal{O}(v)$, if needed, one can always fulfill this condition.

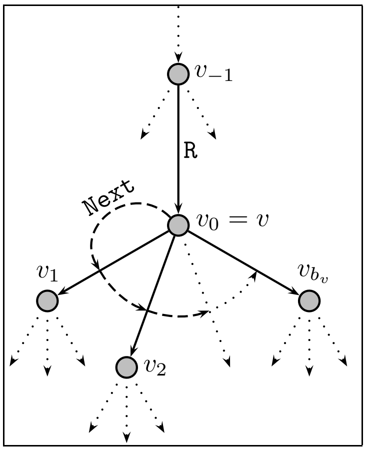

Fig. 4. The counting order of the neighbors of a node

For encoding the numbers $n_{c,i}$ for an index $c \in L \cup M$, we introduce additional binary predicate symbols $e_c^0, e_c^1, \ldots, e_c^{k_c}$, where $k_c := [\log n_c]$. In fact, we will encode not the numbers $n_{c,i}$ themselves, but $n'_{c,i} := min(2^{k_c} - 1, n_{c,i})$, since we do not need to count beyond n_c to check number restrictions. Every number $n'_{c,i}$ in binary coding is a *bit vector* of the size k_c. We represent this bit vector using the values of the predicates $e_c^0, e_c^1, \ldots, e_c^{k_c}$ on the edge (v, v_i), in such a way that $\mathcal{T} \models$ "$\overline{e}_c(v, v_i) = n'_{c,i}$". Formally, for every $c \in L \cup M$ and $v_1, v_2 \in V$ let

$$\overline{e}_c^{\mathcal{I}}(v_1, v_2) := \sum_{i=1}^{k_c} e_c^{i\,\mathcal{I}}(v_1, v_2) \cdot 2^i, \text{ where } e^{\mathcal{I}}(v_1, v_2) := \begin{cases} 1 & \text{if} \quad \mathcal{T} \models e(v_1, v_2), \\ 0 & \text{if} \quad \mathcal{T} \models \neg e(v_1, v_2). \end{cases}$$

Arithmetical expressions involving $\overline{e}_c(x, y)$ and $e_c(x, y)$ are evaluated in interpretations as usual. For instance, given an interpretation $\mathcal{I} = (D, \cdot^{\mathcal{I}})$ and a valuation of variables δ, the proposition "$\overline{e}_c(x, y) \geq e_c(x, y) + 1$" is evaluated to $\mathtt{true}$ *iff* "$\overline{e}_c^{\mathcal{I}}(\delta(x), \delta(y)) \geq e_c^{\mathcal{I}}(\delta(x), \delta(y)) + 1$".

Lemma 5. *For every index $c \in L \cup M$ there exist quantifier-free formulas of size linear in k_c that express the following relations:*

$$\begin{array}{lcl}
\mathit{Init}_c(x,y) & \equiv & \text{“}\overline{e}_c(x,y) = e_c(x,y)\text{”}, \\
\mathit{Full}_c(x,y) & \equiv & \text{“}\overline{e}_c(x,y) = 2^{k_c} - 1\text{”}, \\
\mathit{Copy}_c(x,y,z) & \equiv & \text{“}\overline{e}_c(x,z) = \overline{e}_c(x,y)\text{”}, \\
\mathit{Incr}_c(x,y,z) & \equiv & \text{“}\overline{e}_c(x,z) = \overline{e}_c(x,y) + 1 \ \mathit{mod}\ 2^{k_c}\text{”}, \\
\mathit{Less}_c(x,y) & \equiv & \text{“}\overline{e}_c(x,y) < n_c\text{”}.
\end{array}$$

$$\begin{array}{rcccccccccc}
 & & k_c & & & & i & & & & 0 \\
n & : & * & * & \cdots & * & 0 & 1 & 1 & \cdots & 1 \\
 & & \| & \| & & \| & & & & & \\
n+1 & : & * & * & \cdots & * & 1 & 0 & 0 & \cdots & 0 \\
 & & k_c & & & & j & & & & 0 \\
n & : & * & * & \cdots & * & 0 & * & * & \cdots & * \\
\wedge & & \| & \| & & \| & & & & & \\
m & : & * & * & \cdots & * & 1 & * & * & \cdots & *
\end{array}$$

Fig. 5. The arithmetical properties of bit vectors

Proof. The first three formulas can be defined in the straightforward way:

$$\mathit{Init}_c(x,y) \equiv [e^0_c(x,y) \leftrightarrow e_c(x,y)] \wedge \bigwedge_{1 \le i \le k_c} \neg e^i_c(x,y),$$

$$\mathit{Full}_c(x,y) \equiv \bigwedge_{i \le k_c} e^i_c(x,y),$$

$$\mathit{Copy}_c(x,y,z) \equiv \bigwedge_{i \le k_c} [e^i_c(x,y) \leftrightarrow e^i_c(x,z)]$$

The relation “$m = n + 1$ mod 2^k” holds *iff* ***(i)*** the lowest bits of n and m are different and ***(ii)*** the correspondent bits of any other position of n and m are different *iff* the preceding bits of n and m are 1 and 0 respectively (see Fig. 5):

$$\begin{array}{rcl}
\mathit{Incr}_c(x,y,z) & \equiv & [e^0_c(x,y) \leftrightarrow \neg e^0_c(x,z)] \wedge \\
 & & \displaystyle\bigwedge_{0 < i \le k_c} ([e^i_c(x,y) \leftrightarrow \neg e^i_c(x,z)] \leftrightarrow [e^{i-1}_c(x,y) \wedge \neg e^{i-1}_c(x,z)]).
\end{array}$$

The relation “$m^k ... m^1 m^0 > n^k ... n^1 n^0$” between binary numbers holds if either “$m^k > n^k$”, or, “$m^k = n^k$” and “$m^{k-1} ... m^1 m^0 > n^{k-1} ... n^1 n^0$”:

$$\begin{array}{rcl}
\mathit{Less}_c(x,y) & \equiv & \mathit{Less}_{c,k_c}(x,y), \ \ \mathit{where} \\
\mathit{Less}_{c,i}(x,y) & \equiv & [\text{“}n^i_c = 1\text{”} \wedge \neg e^i_c(x,y)] \vee \\
 & & ([\text{“}n^i_c = 1\text{”} \leftrightarrow e^i_c(x,y)] \wedge \mathit{Less}_{c,i-1}(x,y)), \quad 0 < i < k_c, \\
\mathit{Less}_{c,0}(x,y) & \equiv & [\text{“}n^0_c = 1\text{”} \wedge \neg e^0_c(x,y)].
\end{array}$$

where the expressions “$n^i_c = 1$” stand for the respective boolean constants. It is easy to see that the length of every formula above is linear in k_c. More precisely, every predicate e^i_c, $1 \le i \le k_c$ is used in every definition at most 4 times. □

Using the formulas defined in Lemma 5, we introduce additional quantifier-free formulas, that are linear in the size of the input formula F:

$$\begin{array}{rcl}
\mathit{Init}(x,y) & \equiv & \displaystyle\bigwedge_{l \in L} \mathit{Init}_l(x,y) \wedge \bigwedge_{m \in M} \mathit{Init}_m(x,y), \\
\mathit{Count}_c(x,y,z) & \equiv & [(\neg \mathit{Full}(x,y) \wedge e_c(x,z)) \rightarrow \mathit{Incr}_c(x,y,z)] \wedge \\
 & & [(\mathit{Full}(x,y) \vee \neg e_c(x,y)) \rightarrow \mathit{Copy}_c(x,y,z)], \\
\mathit{Count}(x,y,z) & \equiv & \displaystyle\bigwedge_{l \in L} \mathit{Count}_l(x,y,z) \wedge \bigwedge_{m \in M} \mathit{Count}_m(x,y,z),
\end{array}$$

$$\begin{aligned}
\mathit{AtMost}(x,y) &\equiv \bigwedge_{m\in M} \mathit{Less}_m(x,y),\\
\mathit{Require}(x,y) &\equiv [\bigvee_{l\in L} \mathit{Less}_l(x,y)],\\
\mathit{Child}(x,y) &\equiv \mathtt{R}(x,y)\wedge\mathtt{R}(y,y)\wedge \mathit{Init}(y,x)\wedge \mathit{Count}(y,x,y),\\
\mathit{Fair}(x,y,z) &\equiv \bigvee_{l\in L}[\mathit{Less}_l(x,y)\wedge e_l(x,z)],
\end{aligned}$$

The formula $\mathit{Init}(x,y)$ is used in our encoding of number restrictions for initializing the counters on the edge (x,y). The formula $\mathit{Count}(x,y,z)$ increments every counter between the edges (x,y) and (x,z), if the correspondent binary relation shows up on the edge (x,z) and if the limit of the counter is not yet reached. Otherwise, the current value of the counter on (x,y) is copied to (x,z). The formula $\mathit{AtMost}(x,y)$ ensures that the values of the counters corresponding "at-most" restrictions do not go beyond the maximal allowed limits. Dually, the formula $\mathit{Require}(x,y)$ expresses that the "at-least" restrictions are not yet fulfilled. This should require a `Next` child of x to be created. The child of a node x is created using the formula $\mathit{Child}(x,y)$. It expresses initialization operations for the node y: the predicate `R` should be defined; the counters for the node y should be initialized on the edge (y,x) and computed for the edge (y,y). The formula $\mathit{Fair}(x,y,z)$ is responsible for termination of the process of creating the required children of a node. It says, essentially, that at least one of the relations required on (x,y) should be realized on the next edge (x,z). This guaranties that all "at-least" restrictions will be eventually fulfilled for a node if the process of creating new children can be continued consistently.

The result of the translation is define by $F'\equiv F_g\wedge\Phi_1\wedge\Phi_2\wedge\Phi_3\in\mathcal{GF}^3$, where

$$\begin{aligned}
&\Phi_1\equiv\exists x.[\mathtt{R}(x,x)\wedge \mathit{Init}(x,x)],\\
&\Phi_2\equiv\forall xy.[\mathtt{R}(x,y)\rightarrow(\mathit{AtMost}(x,y)\wedge[\mathit{Require}(x,y)\rightarrow\exists z.\mathtt{Next}(\mathtt{x},\mathtt{y},\mathtt{z})])]^3, \qquad (6)\\
&\Phi_3\equiv\forall xyz.(\mathtt{Next}(x,y,z)\rightarrow[\mathit{Child}(x,z)\wedge \mathit{Fair}(x,y,z)\wedge \mathit{Count}(x,y,z)]).
\end{aligned}$$

The following lemma is immediate from our construction:

Lemma 6. *Every tree-model of F can be expanded to a model of F'.*

Proof. Given a tree-model $\mathcal{T}=(V,\cdot^{\mathcal{T}})$ for F, we interpret the new predicates `R`, `Next` and e^i_c for $c\in L\cup M$ as given in the construction above. It is a routine to check that all formulas Φ_1, Φ_2 and Φ_3 are true in $\mathcal{T}$. For example, the implication $\forall xyz.[\mathtt{Next}(x,y,z)\rightarrow \mathit{Fair}(x,y,z)]$ holds in $\mathcal{T}$ since every child node of every $v\in V$ is essential in $\mathcal{O}(v)$. □

Now we show that a witness for number restrictions $\mathcal{N}(x)$ can be extracted from a model of F':

Lemma 7. *Every model of F' has a counting pattern for $\mathcal{N}(x)$.*

Proof. Given a model $\mathcal{M}' = (D', \cdot^{\mathcal{M}'})$ of F', we construct a counting pattern $\mathcal{P} = (R, w)$ in the following way. We define the reachability relation by $R := \{(d_1, d_2) \in D' \times D' \mid \mathcal{M}' \models \mathtt{R}(d_1, d_2)\}$. For every pair $(d_{-1}, d_0) \in R$, let $w[(d_{-1}, d_0)] := \{(d_0, d_1), \dots, (d_0, d_b)\}$ be such that ***(i)*** $\mathcal{M}' \models \mathtt{Next}(d_0, d_i, d_{i+1})$ for $0 \leq i < b$ and ***(ii)*** $\mathcal{M}' \models \mathtt{Next}(d_0, d_b, d)$ for no $d \in D'$. Such a (finite) set always can be found since for any $d_0, d_i, d_{i+1} \in D'$ with $\mathcal{M}' \models \mathtt{Next}(d_0, d_i, d_{i+1})$, we have $\mathcal{M}' \models$ "$\sum_{l \in L} \overline{e}_l(d_0, d_i) \leq \sum_{l \in L} \overline{e}_l(d_0, d_{i+1})$" because $\mathcal{M}' \models \mathit{Fair}(d_0, d_i, d_{i+1}) \wedge \mathit{Count}(d_0, d_i, d_{i+1})$. Therefore, there cannot be infinitely many d_i connected in a $\mathtt{Next}$-chain. So the witnessing function w is well-defined.

Now we show that $\mathcal{P}$ is a counting pattern for the number restrictions $\mathcal{N}(x)$. The condition ***(i)*** from Definition 3 holds since $\mathcal{M}' \models \Phi_1$, so there exists $d \in D'$ such that $\mathcal{M}' \models \mathtt{R}(d, d)$. To show that the condition ***(ii)*** also holds, consider the elements $d_{-1}, d_0, d_1, \dots, d_b \in D'$ such that $w[(d_{-1}, d_0)] = \{(d_0, d_1), \dots, (d_0, d_b)\}$. Let $n_{c,i} := \#\{d_j \mid j \leq i \ \&\ \mathcal{M}' \models e_c(d_0, d_j)\}$ and $n'_{c,i} := \mathit{min}(2^{k_c} - 1, n_{c,i})$ for $c \in L \cup M$ and $-1 \leq i \leq b$, be similar counters as were introduced for a tree-model. By induction on i with $-1 \leq i \leq b$ it is possible to show that $\mathcal{M}' \models$ "$\overline{e}_c(d_0, d_i) = n'_{c,i}$" for every $c \in L \cup M$, because $\mathcal{M}' \models \mathit{Child}(d_{-1}, d_0)$ and $\mathcal{M}' \models \mathit{Count}(d_0, d_i, d_{i+1})$ for i with $0 \leq i < d$. Moreover, $\mathcal{M}' \models$ "$\overline{e}_l(d_0, d_b) \geq n_l$" and $\mathcal{M}' \models$ "$\overline{e}_m(d_0, d_b) < n_m$" for every $l \in L$ and $m \in M$, since $\mathcal{M}' \models \mathit{AtMost}(d_0, d_b) \wedge \neg \mathit{Require}(d_0, d_b)$ (there is no $\mathtt{Next}$-successor of d_b). As a conclusion, we have $\mathcal{M}'|_{\{d_{-1}, d_0, \dots, d_b\}} \models \mathcal{N}(b_0)$, which implies the condition ***(ii)*** for a counting pattern. So, $\mathcal{P}$ is indeed a counting pattern for the number restrictions $\mathcal{N}(x)$. □

Corollary 2. *F' is satisfiable iff F is satisfiable.*

Proof. The *"if"* part follows directly from Theorem 1 and Lemma 6. To prove the converse, assume that $\mathcal{M}'$ is a model of F'. In particular, $\mathcal{M}'$ is a model of F_g. By Lemma 7, $\mathcal{M}'$ has a counting pattern for $\mathcal{N}(x)$. Therefore, by Lemma 4 there is a tree-model $\mathcal{T}$ for $F \equiv F_g \wedge \forall x.\mathcal{N}(x)$. □

The results obtained in this section can be summarized in our main theorem:

Theorem 2. *For any formula $F \in \mathcal{GF}^2\mathcal{N}$ there exists a formula $F' \in \mathcal{GF}^3$ such that **(i)** F is satisfiable iff F' is satisfiable, **(ii)** $|F'| = O(|F|)$ and F' can be computed in polynomial time from F.*

Proof. Let $F \in \mathcal{GF}^2\mathcal{N}$. By Lemma 2, one can find $F^n \equiv F_g \wedge \forall x.\mathcal{N}(x)$ in the normal form (5) that is equisatisfiable with F. Taking $F' \equiv F_g \wedge \Phi_1 \wedge \Phi_2 \wedge \Phi_3$ as defined in (6), by Corollary 2, F' is equisatisfiable with F^n, and thus with F. F' has a linear size and can be computed in polynomial time in $|F|$. □

Corollary 3. *There is a decision procedure for $\mathcal{GF}^2\mathcal{N}$ that can be implemented in time $2^{O|F|}$, where $|F|$ is the size of a formula $F \in \mathcal{GF}^2\mathcal{N}$.*

Proof. A decision procedure for $F \in \mathcal{GF}^3$ can be implemented in time $2^{O|F|}$. (see Grädel 1999*b*, Hladik 2002, Ganzinger & de Nivelle 1999)[4]. □

[4] Most decision procedures are given here for the full guarded fragment, but is easy to see that their specializations for the bounded-variable case run in EXPTIME.

4 Conclusions and the Future Work

We have described a procedure, that allows one to translate every formula from $\mathcal{GF}^2\mathcal{N}$ to an equisatisfiable formula from $\mathcal{GF}^3$. The procedure is quite intriguing, since it runs in polynomial time and has only linear overhead in the size of the produced formula. However, only experimental evaluation can judge the practical usefulness of the procedure. For the future work we try to find a translation from $\mathcal{GF}^2\mathcal{N}$ to smaller fragments, in particular to $\mathcal{GF}^2$, or to the description logic $\mathcal{ALCI}$, so that existing systems for description logics can be employed. This idea is related to the work of Hladik & Sattler (2003). We believe that a translation to simpler formalisms can be found by exploiting the automata translation proposed by Tobies (2001).

References

Andréka, H., van Benthem, J. & Németi, I. (1998), 'Modal languages and bounded fragments of predicate logic', *Journal of Philosophical Logic* **27**, 217–274.

Ganzinger, H. & de Nivelle, H. (1999), A superposition decision procedure for the guarded fragment with equality, *in* 'Proc. 14th IEEE Symposium on Logic in Computer Science', IEEE Computer Society Press, pp. 295–305.

Grädel, E. (1999*a*), Decision procedures for guarded logics, *in* 'Automated Deduction - CADE16. Proceedings of 16th International Conference on Automated Deduction, Trento, 1999', Vol. 1632 of *LNCS*, Springer-Verlag.

Grädel, E. (1999*b*), 'On the restraining power of guards', *Journal of Symbolic Logic* **64(4)**, 1719–1742.

Haarslev, V. & Möller, R. (2001), Optimizing reasoning in description logics with qualified number restrictions, *in* 'Proceedings of the International Workshop on Description Logics (DL-2001)', Stanford, USA, pp. 142–151.

Haarslev, V., Timmann, M. & Möller, R. (2001), Combining tableau and algebraic methods for reasoning with qualified number restrictions in description logics, *in* 'Proceedings of the International Workshop on Methods for Modalities 2 (M4M-2)', Amsterdam, Netherlands.

Hladik, J. (2002), Implementation and optimisation of a tableau algorithm for the guarded fragment, *in* U. Egly & C. G. Fermüller, eds, 'Proceedings of the International Conference on Automated Reasoning with Tableaux and Related Methods (Tableaux 2002)', Vol. 2381 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag.

Hladik, J. & Sattler, U. (2003), A translation of looping alternating automata to description logics, *in* 'Proc. of the 19th Conference on Automated Deduction (CADE-19)', Vol. 2741 of *Lecture Notes in Artificial Intelligence*, Springer Verlag.

Tobies, S. (2001), Complexity Results and Practical Algorithms for Logics in Knowledge Representation, PhD thesis, RWTH Aachen, Germany.

Vardi, M. (1996), Why is modal logic so robustly decidable?, *in* N. Immerman & P. G. Kolaitis, eds, 'Descriptive Complexity and Finite Models', Vol. 31 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, American Mathematical Society, Princeton University, pp. 149–184.