HAL open science

Deciding Modal Logics through Relational Translations into GF2

Stéphane Demri, Hans de Nivelle

▶ To cite this version:

Stéphane Demri, Hans de Nivelle. Deciding Modal Logics through Relational Translations into GF2. Proceedings of the 3rd Workshop on Methods for Modalities (M4M-3), Carlos Areces; Patrick Blackburn, Sep 2003, Nancy, France. pp.93-108. hal-03201443

HAL Id: hal-03201443

https://hal.science/hal-03201443

Submitted on 21 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Deciding Modal Logics through Relational Translations into GF^2

Stéphane Demri

LSV/CNRS UMR 8643 & INRIA Futurs projet SECSI & ENS Cachan
email: `demri@lsv.ens-cachan.fr`

Hans de Nivelle

Max Planck Institut für Informatik
email: `nivelle@mpi-sb.mpg.de`

Abstract

We provide a simple translation from the satisfiability problem for regular grammar logics with converse into GF^2, which is the intersection of the guarded fragment and the 2-variable fragment of first-order logic. This translation is theoretically interesting because it translates modal logics with certain frame conditions into first-order logic, without explicitly expressing the frame conditions. Using the same method, one can show that other modal logics can be naturally translated into GF^2, including nominal tense logics and intuitionistic propositional logic. In our view, the results in this paper provide strong evidence that the natural first-order fragment corresponding to modal logics, is GF^2.

1 Introduction

Translating modal logics. Modal logics are used in many areas of Computer Science, as for example knowledge representation, model-checking, and temporal reasoning. In order to reason with knowledge bases, or to check that a certain temporal program meets a certain specification, one needs to be able to do *theorem proving* in modal logics. For theorem proving in modal logics, two main approaches can be distinguished. Either one develops a theorem prover directly for the logic under consideration, or one translates the logic into a general logic, usually first-order logic. Translation of modal logics into first-order logic, with the explicit goal to mechanise such logics is an approach that has been introduced in [31]. Morgan distinguishes two types of translations: The *semantical translation*, which is nowadays known as the *relational translation* (see e.g., [36]) and the *syntactic* translation, which consists in reifing modal formulae by translating the formula, together with the axioms and inference rules from a Hilbert-style system into classical logic using an additional provability predicate. Until recent, it appeared that this method was not suitable for obtaining decision procedures, but in the recent paper [26], this method has been modified. The authors were able to translate a couple of modal logics, including KT, KD, KB, into GF^2 by proving a bound on the size of the formulas that can occur in Hilbert-style proofs. It is surprising that this method, with a philosophy which is completely different from our translation method, arrives for these logics at a very similar translation. This is a point that needs further investigation.

Here we study the semantic method for obtaining decision procedures. Our goal is to translate a modal logic into a decidable fragment of first-order logic, which preferably has the same complexity as the modal logic, and which allows theorem proving techniques that are similar to the ones allowed by the original modal logic. A survey on translation methods for modal logics can be found in [32], where more references are provided.

Guarded fragments. Both the guarded fragment, introduced in [2] (see also [21]) and FO^2, the fragment of classical logic with two variables [17,22], have been used for this purpose. The authors of [2] explicitly mention the goal of identifying 'the modal fragment of first-order logic' as a motivation for introducing the guarded fragment. Apart from satisfying nice logical properties [2], the guarded fragment GF has an **EXPTIME**-complete satisfiability problem, in case the maximal arity of the predicate symbols is fixed in advance [21]. Moreover, efficient mechanisation of the guarded fragment is possible thanks to the design of resolution-based decision procedures [5,8]. In [24], a tableau procedure for the guarded fragment with equality is implemented and tested; A tableau procedure FO^2 is described in [28].

Unfortunately, there are some simple modal logics with the satisfiability problem in **PSPACE**, which can be translated neither into GF, nor into FO^2 by the relational translation. The reason for this is the fact that the frame condition that characterizes the logic cannot be expressed in $GF \cup FO^2$. The simplest examples of such logics are K4 and S4. Transitivity cannot be expressed in $GF \cup FO^2$.

Because of the apparent insufficiency of GF to capture some of the basic modal logics, various extensions of GF have been proposed and studied. For instance, GF^2 with transitive guards is **2EXPTIME**-complete [33,27] whereas μGF, the guarded fragment of first-order logic extended with a μ-calculus-style fixed point operator, is still decidable and in **2EXPTIME**. This fragment does contain the simple modal logics S4, but the resulting decision procedure is much more complicated than a direct decision procedure would be. After all, there exist simple tableaux procedures for S4. In addition, μGF does not have the finite model property, although S4 has.

Almost structure-preserving translations. An approach that seems better suited for theorem proving, and that does more justice to the low complexities of the simple modal logics is the approach taken in [6,7]. There, an almost structure-preserving translation from the modal logics S4, S5 and K5 into GF^2 was given. The correctness proofs there were ad hoc, and at that time it was not clear upon which principles they are based. Here we show that this translation method relies on the fact that the frame conditions for S4 and K5 are *regular* in some sense that will be made precise in Section 2. Some subformulas are translated into a sequence of GF^2 formulas simulating a finite-state automaton, the structure of which depends on the frame condition. The simplicity of the method leaves hope that GF^2 may be rich enough after all to naturally capture most of the usual modal logics.

The theoretical interest of the translation method lies in the fact the translation method preserves the structure of the formula almost completely. Only for subformulas of form $[a]\phi$ does the translation differ from the relational translation. On these subformulas, a sequence of formulas is generated that simulates an NDFA based on the frame condition for the modal logic. In our view, this translation also provides an explanation why some modal logics like S4, have nice tableau procedures (see e.g. [20,30]): The tableau rule for subformulas of form $[a]\phi$ can be viewed as simulating our translation for a subformula of form $[a]\phi$, which in turn simulates an NDFA corresponding to a closure property of the underlying logic.

Plan of the paper. We show that the methods of [6] can be extended to a large class of modal logics. The class of modal logics that we consider is the class of *regular grammar logics with converse*. The axioms of such modal logics are of form $[a_0]\mathrm{p} \Rightarrow [a_1]\cdots[a_n]\mathrm{p}$, where each $[a_i]$ is either a forward or a backward modality. Another condition called *regularity* is required and will be formally defined in Section 2.

With our translation, we are able to translate numerous modal logics into GF^2 despite the fact that their frame conditions are not expressible in $FO^2 \cup GF$: These logics include the standard modal logics K4, S4, K5, K45, S5, description logics, nominal tense logics (if we allow constants and equality in GF^2), and propositional intuitionistic logic, to quote a few classes of logics.

Related work. Complexity issues for regular grammar logics have been studied in [10,11] (see also [3]), whereas grammar logics are introduced in [15]. Frame conditions involving the converse relations are not treated in [10,11]. The current work can be viewed as a natural continuation of [6] and [10].

The frame conditions considered in the present work can be defined by the MSO definable closure operators [18]. However, in contrast with the method of [18], we obtain the optimal complexity upper bound for the class of regular grammar logics with converse (**EXPTIME**), whereas the method of [18] obtains a non-elementary upper bound.

As already mentiond, the recent paper [26] presents another translation of modal logics into GF^2, by adequately encoding the modal formula and the axioms, and by showing that the set of formulas that can occur in a proof is finite. Although for logics such as S4, the method in [26] results in the same formulas as ours, it is still open how the methods are related in the

general case. For instance, no regularity conditions are explicitly involved in [26] whereas this is a central point in our work.
Finally, it should be noted that this paper is a short version of [12], which contains all proofs that are omitted in this paper.

2 Regular Grammar Logics with Converse

Formal grammars are a convenient way of defining frame properties for modal logics. Many standard modal logics can be nicely defined by a grammar logic. We give examples at the end of this section, in Table 1 and Example 2.9. After Example 2.9 we describe a sequence of logics which are not regular grammar logics. In order to define regular grammar logics, we recall below a few definitions from formal language theory, semi-Thue systems, and finite-state automata.

2.1 Semi-Thue Systems

An *alphabet* Σ is a finite set $\{a_1, \ldots, a_m\}$ of symbols. We write Σ^* to denote the set of finite strings that can be built over elements of Σ, and we write ϵ for the empty string. We write $u_1 \cdot u_2$ for the concatenation of two strings. For a string $u \in \Sigma^*$, we write $|u|$ to denote its length. A *language* over some alphabet Σ is defined as a subset of Σ^*.

A *semi-Thue system* S over Σ is defined as a subset of $\Sigma^* \times \Sigma^*$. The pairs $(u_1, u_2) \in \Sigma^* \times \Sigma^*$ are called *production rules*. We will mostly write $u_1 \rightarrow u_2$ instead of (u_1, u_2) for production rules. The system S will be said to be *context-free* if S is finite and all the production rules are in $\Sigma \times \Sigma^*$. The one-step derivation relation $\Rightarrow_{\mathrm{S}}$ is defined as follows: Put $u \Rightarrow_{\mathrm{S}} v$ iff there exist $u_1, u_2 \in \Sigma^*$, and $u' \rightarrow v' \in \mathrm{S}$, such that $u = u_1 \cdot u' \cdot u_2$, and $v = u_1 \cdot v' \cdot u_2$. The full derivation relation $\Rightarrow_{\mathrm{S}}^*$ is defined as the reflexive and transitive closure of $\Rightarrow_{\mathrm{S}}$. Finally, for every $u \in \Sigma^*$, we write $\mathrm{L}_{\mathrm{S}}(u)$ to denote the language $\{v \in \Sigma^* : u \Rightarrow_{\mathrm{S}}^* v\}$.

A context-free semi-Thue system S, based on Σ is called *regular* if for every $a \in \Sigma$, the language $\mathrm{L}_{\mathrm{S}}(a)$ is regular. In that case, one can associate to each $a \in \Sigma$ an NDFA (*non-deterministic finite automaton*) recognizing the language $\mathrm{L}_{\mathrm{S}}(a)$.

2.2 Grammar Logics with Converse

In grammar logics, modal frame conditions are expressed by the production rules of semi-Thue systems. For example, transitivity on the relation R_a is expressed by the production rule $a \rightarrow a \cdot a$. Similarly, reflexivity can be expressed by $a \rightarrow \epsilon$. We also allow frame conditions that contain the converses of accessibility relations. For this reason, we will associate to every symbol a in the alphabet a unique converse symbol $\overline{a}$. Using converses, for example symmetry on the relation R_a can be represented by the production rule $a \rightarrow \overline{a}$

whereas euclideanity on the relation R_a can be represented by the production rule $a \to \overline{a} \cdot a$.

Given an alphabet Σ, we define the multimodal language $\mathcal{L}^{\Sigma}$ based on Σ. In order to do this, we assume a countably infinite set $\mathrm{PROP} = \{\mathrm{p}_0, \mathrm{p}_1, \ldots\}$ of propositional variables. Then $\mathcal{L}^{\Sigma}$ is recursively defined as follows:

$$\phi, \psi ::= \mathrm{p} \mid \bot \mid \top \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid [a]\phi \mid \langle a \rangle \phi$$

for $\mathrm{p} \in \mathrm{PROP}$ and $a \in \Sigma$. We define the *negation normal form* (NNF) as usual: $\neg$ is applied only on members of PROP. We will make use of the NNF when we translate formulas to GF^2.

Let Σ be an alphabet. A Σ-*frame* is a pair $\mathcal{F} = \langle W, R \rangle$, such that W is non-empty, and R is a mapping from the elements of Σ to binary relations over W. So, for each $a \in \Sigma$, $R_a \subseteq W \times W$. A Σ-*model* $\mathcal{M} = \langle W, R, V \rangle$ is obtained by adding a valuation function V with signature $\mathrm{PROP} \to \mathcal{P}(W)$ to the frame. The satisfaction relation $\models$ is defined in the usual way.

- For $\mathrm{p} \in \mathrm{PROP}$, $\mathcal{M}, x \models \mathrm{p}$ iff $x \in V(\mathrm{p})$.
- For $a \in \Sigma$, $\mathcal{M}, x \models [a]\phi$ iff for all y, s.t. $R_a(x, y)$, $\mathcal{M}, y \models \phi$.
- For $a \in \Sigma$, $\mathcal{M}, x \models \langle a \rangle \phi$ iff there is an y, s.t. $R_a(x, y)$ and $\mathcal{M}, y \models \phi$.
- $\mathcal{M}, x \models \phi \wedge \psi$ iff $\mathcal{M}, x \models \phi$ and $\mathcal{M}, x \models \psi$.
- $\mathcal{M}, x \models \phi \vee \psi$ iff $\mathcal{M}, x \models \phi$ or $\mathcal{M}, x \models \psi$.
- $\mathcal{M}, x \models \neg\phi$ iff it is not the case that $\mathcal{M}, x \models \phi$.

A formula ϕ is said to be *true* in the Σ-model $\mathcal{M}$ (written $\mathcal{M} \models \phi$) iff for every $x \in W$, $\mathcal{M}, x \models \phi$. A Σ-frame maps the symbols in Σ to binary relations on W. This mapping can be extended to the full language Σ^* as follows:

- R_ϵ equals $\{\langle x, x \rangle : x \in W\}$,
- For $u \in \Sigma^*$, $a \in \Sigma$, $R_{u \cdot a}$ equals $\{\langle x, y \rangle : \exists z \in W,\ R_u(x, z) \text{ and } R_a(z, y)\}$.

Now we define how semi-Thue systems encode conditions on Σ-frames.

Definition 2.1 Let $u \to v$ be a production rule over some alphabet Σ. We say that the Σ-frame $\mathcal{F} = \langle W, R \rangle$ *satisfies* $u \to v$ if the inclusion $R_v \subseteq R_u$ holds. $\mathcal{F}$ satisfies a *semi-Thue system* S if it satisfies each of its rules. We also say that S *is true* in $\mathcal{F}$. A formula ϕ is said to be S-*satisfiable* iff there is a Σ-model $\mathcal{M} = \langle W, R, V \rangle$ which satisfies S, and which has an $x \in W$ such that $\mathcal{M}, x \models \phi$. Similarly, a formula ϕ is said to be S-*valid* iff for all Σ-models $\mathcal{M} = \langle W, R, V \rangle$ that satisfy S, for all $x \in W$, we have $\mathcal{M}, x \models \phi$.

We assume that a *logic* is characterized by its set of satisfiable formulas, or equivalently, by its set of universally valid formulas. We call logics that can be characterized by a semi-Thue system using Definition 2.1 *grammar logics*. Those logics that can be characterized by regular semi-Thue systems are called *regular grammar logics*. For instance, the modal logic S4 is the regular modal logic defined by the regular semi-Thue system $\{a \to \epsilon,\ a \to aa\}$. In order to be able to cope with properties such as symmetry and euclideanity, we now introduce converses:

Definition 2.2 Let Σ be an alphabet. We call a function $\bar{\cdot}$ on Σ a *converse mapping* if for all $a \in \Sigma$, we have $\overline{a} \neq a$ and $\overline{\overline{a}} = a$.

It is easy to prove the following:

Lemma 2.3 *Let Σ be an alphabet with converse mapping $\bar{\cdot}$. Then $\bar{\cdot}$ is a bijection on Σ. In addition, Σ can be partitioned into two disjoint sets Σ^+ and Σ^-, such that* **(1)** *for all $a \in \Sigma^+$, $\overline{a} \in \Sigma^-$,* **(2)** *for all $a \in \Sigma^-$, $\overline{a} \in \Sigma^+$.*

In fact, there exist many partitions $\Sigma = \Sigma^- \cup \Sigma^+$. When we refer to such a partition, we assume that an arbitrary one is chosen. We will assume that the modal operators indexed by letters in Σ^+ are forward modalities (conditions on successor states) whereas the modal operators indexed by letters in Σ^- are backward modalities (conditions on predecessor states).

Definition 2.4 A converse mapping $\bar{\cdot}$ can be extended to words over Σ^* as follows: **(1)** $\overline{\epsilon} = \epsilon$, **(2)** if $u \in \Sigma^*$ and $a \in \Sigma$, then $\overline{u \cdot a} = \overline{a} \cdot \overline{u}$.

Definition 2.5 Given a semi-Thue system S over some alphabet Σ with converse mapping $\bar{\cdot}$, we call S a semi-Thue system *with converse* if $u \to v \in \mathrm{S}$ implies $\overline{u} \to \overline{v} \in \mathrm{S}$. The *converse closure* of a semi-Thue system S is the $\subseteq$-smallest semi-Thue system S$'$ with converse for which $S \subseteq \mathrm{S}'$.

It is easily checked that the converse closure is always well-defined.

Definition 2.6 Let Σ be an alphabet with converse mapping $\bar{\cdot}$. A $\langle \Sigma, \bar{\cdot} \rangle$-frame is a Σ-frame for which in addition, for each $a \in \Sigma$,

$$R_{\overline{a}} \text{ equals } \{\langle y, x\rangle : R_a(x, y)\}.$$

The following property of $\langle \Sigma, \bar{\cdot} \rangle$-frames is easily checked:

Lemma 2.7 *Let Σ be an alphabet with converse mapping $\bar{\cdot}$. Let $\mathcal{F} = \langle W, R\rangle$ be a $\langle \Sigma, \bar{\cdot} \rangle$-frame. Then for each $u \in \Sigma^*$, $R_{\overline{u}} = \{\langle y, x\rangle : R_u(x, y)\}$.*

As a consequence, semi-Thue systems not satisfying Definition 2.5 cannot characterize more $\langle \Sigma, \bar{\cdot} \rangle$-frames than semi-Thue systems that do satisfy Definition 2.5:

Lemma 2.8 *Let Σ be an alphabet with converse mapping $\bar{\cdot}$. Let $\mathcal{F}$ be a $\langle \Sigma, \bar{\cdot} \rangle$-frame. Let* S *be a semi-Thue system over* Σ, *let* S$'$ *be its converse closure. Then* $\mathcal{F}$ *satisfies* S *iff* $\mathcal{F}$ *satisfies* S$'$.

We call the logics that can be characterized by a regular semi-Thue system with converse using Definition 2.6 and Definition 2.1 *regular grammar logics with converse*. The models of such logics are based on $\langle \Sigma, \bar{\cdot} \rangle$-frames which guarantees that converses are taken into account.

Originally, grammar logics were defined with formal grammars in [15] (as in [3,10,11]), and they form a subclass of Sahlqvist modal logics with frame

logic	$L_S(a)$	frame condition
K	$\{a\}$	(none)
KT	$\{a, \epsilon\}$	reflexivity
KB	$\{a, \overline{a}\}$	symmetry
KTB	$\{a, \overline{a}, \epsilon\}$	refl. and sym.
K4	$\{a\} \cdot \{a\}^*$	transitivity
KT4 = S4	$\{a\}^*$	refl. and trans.
KB4	$\{a, \overline{a}\} \cdot \{a, \overline{a}\}^*$	sym. and trans.
K5	$(\{\overline{a}\} \cdot \{a, \overline{a}\}^* \cdot \{a\}) \cup \{a\}$	euclideanity
KT5 = S5	$\{a, \overline{a}\}^*$	equivalence rel.
K45	$(\{\overline{a}\}^* \cdot \{a\})^*$	trans. and eucl.

Table 1
Regular languages for standard modal logics

conditions expressible in Π_1 when S is context-free. In fact, numerous logics are (fragments of) regular grammar logics with converse, or logics that can be reduced to such logics.

Example 2.9 The standard modal logics K, T, B, S4, K5, K45, and S5 can be defined as regular grammar logics over the singleton alphabet $\Sigma = \{a\}$. In Table 1, we specify the semi-Thue systems through regular expressions for the languages $L_S(a)$.

We list some of other logics for specific application domains, which can also be seen as regular grammar logics with converse.

- description logics (with role hierarchy, transitive roles), see e.g. [25];
- knowledge logics, see e.g. $S5_m$(DE) in [14];
- bimodal logics for intuitionistic modal logics of the form $\mathbf{IntK}_\Box + \Gamma$ [37];
- fragments of logics designed for the access control in distributed systems [1,29].
- extensions with the universal modality [19]. Indeed, for every regular grammar logic with converse, its extension with a universal modal operator is also a regular grammar logic with converse by using simple arguments from [19] (add an S5 modal connective stronger than any other modal connective). Hence, satisfiability, global satisfiability and logical consequence can be handled uniformly with no increase of worst-case complexity;
- information logics, see e.g. [35]. For instance, the Nondeterministic Information Logic NIL introduced in [35] (see also [9]) can be shown to be a

fragment of a regular grammar logic with converse with $\Sigma^+ = \{fin, sim\}$ and the converse closure of the production rules below:
· $fin \to \overline{fin} \cdot fin$; $fin \to \epsilon$;
· $sim \to \overline{sim}$; $sim \to \epsilon$;
· $sim \to \overline{fin} \cdot sim \cdot fin$.
For instance $L_S(sim) = \{\overline{fin}\}^* \cdot \{sim, \overline{sim}, \epsilon\} \cdot \{fin\}^*$.

A frame condition outside our current framework.
The euclideanity condition can be slightly generalized by considering frame conditions of the form
$(R_a^{-1})^n; R_a \subseteq R_a$ for some $n \geq 1$. The context-free semi-Thue system with converse corresponding to this inclusion is $S_n = \{a \to \overline{a}^n a,\ \overline{a} \to \overline{a}a^n\}$. The case $n = 1$ corresponds to euclideanity. Although we have seen that for $n = 1$, the language $L_{S_1}(a)$ is regular, one can establish that in general, for $n > 1$, the language $L_{S_n}(a)$ is not regular. This is particularly interesting since S_n-satisfiability restricted to formulae with only the modal operator $[a]$ is decidable, see e.g. [16,26]. To see why the systems $L_{S_n}(a)$ are not regular, consider strings of the following form:

$$\sigma_n(i_1, i_2) = (\overline{a}a^{n-1})^{i_1}\ a\ (\overline{a}^{n-1}a)^{i_2}.$$
$$\overline{\sigma}_n(i_1, i_2) = (\overline{a}a^{n-1})^{i_1}\ \overline{a}\ (\overline{a}^{n-1}a)^{i_2}.$$

We show that

$$(\ a \Rightarrow^*_{S_n} \sigma_n(i_1, i_2) \text{ and } a \Rightarrow^*_{S_n} \overline{\sigma}_n(i_1, i_2 + 1)\ ) \text{ iff } i_1 = i_2.$$

In order to check that the equivalence holds from right to left, observe that $a = \sigma_n(0, 0)$, and

$$\sigma_n(0,0) \Rightarrow_{S_n} \overline{\sigma}_n(0,1) \Rightarrow_{S_n} \sigma_n(1,1) \Rightarrow_{S_n} \cdots$$
$$\Rightarrow_{S_n} \sigma_n(i,i) \Rightarrow_{S_n} \overline{\sigma}_n(i,i+1) \Rightarrow_{S_n} \sigma_n(i+1,i+1) \Rightarrow_{S_n} \cdots$$

We now prove the equivalence from left to right. Let us say that u is *an ancestor of* v if $u \Rightarrow_{S_n} v$. Then it is sufficient to observe the following:

(i) A string of form $\sigma_n(0, j)$ has no ancestor.
(ii) A string of form $\sigma_n(i + 1, j)$ has only one ancestor, namely $\overline{\sigma}_n(i, j)$.
(iii) A string of form $\overline{\sigma}_n(i, 0)$ has no ancestor.
(iv) A string of form $\overline{\sigma}_n(i, j + 1)$ has only one ancestor, namely $\sigma_n(i, j)$.

To have an ancestor, a string must have a sequence of at least n consecutive a's or $\overline{a}$'s. The strings of form 1 or 3 have no such sequence. The strings of form 2 or 4 have exactly one such sequence.
Since the regular languages are closed under intersection, the languages $\{\sigma_n(i, j) : i \geq 0,\ j \geq 0\}$ are regular, while the languages $\{\sigma_n(i, i) : i \geq 0\}$ are clearly not regular when $n > 1$, the languages $L_{S_n}(a)$ cannot be regular when $n > 1$. Hence, this will leave open the extension of our translation method to the case of context-free semi-Thue systems with converse when decidability holds (see e.g. decidable extensions of PDL with certain context-free programs in [23]).

3 The Translation into GF^2

We consider our translation *almost structure preserving*. This is because the transformation is defined by a simple recursion, which preserves the structure of the formula for all cases, except for the case where the subformula has form $[a]\phi$. In case the subformula does have form $[a]\phi$, the translation constructs a sequence of formulas that simulates the regular automaton recognizing $\mathrm{L_S}(a)$. Some specific features of our translation are the following:

- The translation introduces new predicate symbols, which it needs for simulating automata. Because of this, the translated formula is not logically equivalent to the original formula. The transformation is only satisfiability preserving. Our translation is a reduction, as understood in complexity theory, from the satisfiability problem for regular grammar logics with converse into the satisfiability problem for GF^2.
- Although our transformation is a reduction in the complexity theoretic sense, it is a relatively mild one. It does not make use of encodings of Turing machines, or another computation mechanism. It uses only a mutual recursion between the encoding of the frame conditions and the translation of logical operators.

3.1 The Transformation

We assume that S is a regular semi-Thue system with converse over alphabet Σ with converse mapping $\bar{\cdot}$ (and partition $\{\Sigma^+, \Sigma^-\}$). For every $a \in \Sigma$, the automaton $\mathcal{A}_a$ is an NDFA (with ϵ-transitions) recognizing the language $\mathrm{L_S}(a)$. We write $\mathcal{A}_a = (Q_a, s_a, F_a, \delta_a)$. Here Q_a is the finite set of states, s_a is the starting state, $F_a \subseteq Q_a$ are the accepting states, and δ_a is the transition function, which is possibly non-deterministic. When all rules in S are either right-linear or left-linear, then each automaton $\mathcal{A}_a$ can be effectively built in logarithmic space in $|\mathrm{S}|$, the size of S with some reasonably succinct encoding.

Definition 3.1 Assume that for each letter $a \in \Sigma^+$, a unique binary predicate symbol $\mathbf{R}_a$ is given. We define a translation function t_a, mapping letters in Σ to binary predicates (α and β are variables).

- For each letter $a \in \Sigma^+$, we define $t_a(\alpha, \beta) = \mathbf{R}_a(\alpha, \beta)$,
- For each letter $a \in \Sigma^-$, we define $t_a(\alpha, \beta) = \mathbf{R}_a(\beta, \alpha)$.

We now define the part of the translation that takes into account the frame conditions. It takes two parameters, a one-place formula and an NDFA. The result of the translation is a first-order formula (one-place again) that has the following meaning:

> In every point that is reachable by a sequence of transitions that are accepted by the automaton, the original one-place formula holds.

Definition 3.2 Let $\mathcal{A} = (Q, s, F, \delta)$ be an NDFA. Let $\varphi(\alpha)$ be a *first-order* formula with one free variable α. Assume that for each state $q \in Q$, a fresh unary predicate symbol $\mathbf{q}$ is given. We define $t_{\mathcal{A}}(\alpha, \varphi)$ as the conjunction of the following formulas (the purpose of the first argument is to remember that α is the free variable of φ):

- For the initial state s, the formula $\mathbf{s}(\alpha)$ is included in the conjunction.
- For each $q \in Q$, for each $a \in \Sigma$, for each $r \in \delta(q, a)$, the formula $\forall \alpha\beta \; [\; t_a(\alpha, \beta) \rightarrow \mathbf{q}(\alpha) \rightarrow \mathbf{r}(\beta) \;]$ is included in the conjunction.
- For each $q \in Q$, for each $r \in \delta(q, \epsilon)$, the formula $\forall \alpha \; [\; \mathbf{q}(\alpha) \rightarrow \mathbf{r}(\alpha) \;]$ is included in the conjunction.
- For each $q \in F$, the formula $\forall \alpha \; [\; \mathbf{q}(\alpha) \rightarrow \varphi(\alpha) \;]$ is included in the conjunction.

It is assumed that in each application of $t_{\mathcal{A}_a}$ distinct predicate symbols are used. One can introduce the new predicate symbols of the form $\mathbf{q}$ for $q \in Q$ either occurrence-wise, or subformula-wise. In the sequel, we adopt a subformula-wise approach and we fix that in Definition 3.2, the fresh unary predicate symbol for $q \in Q$ will be always $\mathbf{q}_\varphi$, where φ is the formula being translated. As a consequence, two applications of $t_{\mathcal{A}}(\alpha, \varphi)$ reuse the same predicate symbols associated to the states of Q.

If the automaton $\mathcal{A}$ has more than one accepting state, then $\varphi(\alpha)$ occurs more than once in the translation $t_{\mathcal{A}}(\alpha, \varphi)$. This may cause an exponential blow-up in the translation but this problem can be easily solved by adding a new accepting state to the automaton, and adding ϵ-translations from the old accepting states into the new accepting state.

Using Definition 3.2, we can now give the translation itself. It is a standard relational translation on all subformulas, except for those of the form $[a]\psi$, on which $t_{\mathcal{A}_a}$ will be used. In order to easily recognize the $\Box$-subformulas, we require the formula ϕ to be in negation normal form. One could define the translation without NNF, but it would be more complicated, because we would have to add cases for $\leftarrow, \leftrightarrow$, and we would have to take the polaraties into account while translating.

Definition 3.3 Let $\phi \in \mathcal{L}^\Sigma$ be a modal formula in NNF. Let S be a regular semi-Thue system with converse over alphabet Σ with converse mapping $\bar{\cdot}$. Assume that for each $a \in \Sigma$ an automaton $\mathcal{A}_a$ recognizing $\mathrm{L_S}(a)$ is given. We define the translation $T_{\mathrm{S}}(\phi)$ as $t(\phi, \alpha, \beta)$ from the following function $t(\psi, \alpha, \beta)$, which is defined by recursion on the subformulas ψ of ϕ :

- For a propositional symbol p, the translation $t(\mathrm{p}, \alpha, \beta)$ equals $\mathbf{p}(\alpha)$, where $\mathbf{p}$ is a unary predicate symbol uniquely associated to the propositional variable p. The translation $t(\neg\mathrm{p}, \alpha, \beta)$ of $\neg$p equals $\neg\mathbf{p}(\alpha)$,
- $t(\psi \wedge \psi', \alpha, \beta)$ equals $t(\psi, \alpha, \beta) \wedge t(\psi', \alpha, \beta)$,
- $t(\psi \vee \psi', \alpha, \beta)$ equals $t(\psi, \alpha, \beta) \vee t(\psi', \alpha, \beta)$,
- for $a \in \Sigma$, $\;t(\; \langle a \rangle\psi, \; \alpha, \beta)$ equals $\exists \beta \; [\; t_a(\alpha, \beta) \wedge t(\psi, \beta, \alpha) \;]$,
- for $a \in \Sigma$, $\;t(\; [a]\psi, \; \alpha, \beta)$ equals $t_{\mathcal{A}_a}(\alpha, t(\psi, \alpha, \beta) \;)$.

When translating a subformula of form $[a]\psi$, the translation function $t_{\mathcal{A}_a}$ of Definition 3.2 is used. The only difference with the standard relational translation is the translation of $[a]$-formulae.

Lemma 3.4 *$T_{\mathrm{S}}(\phi)$ belongs to* GF^2 *and $T_{\mathrm{S}}(\phi)$ can be computed in logspace in $|\phi| + m$ with $m = \max\{\ |\mathcal{A}_a|\ \mid\ a \in \Sigma\ \}$.*

When S is formed of production rules of a formal grammar that is either right-linear or left-linear, then m is in $\mathcal{O}(|\mathrm{S}|)$. For a given semi-Thue system S, the number m is fixed. Moreover, $T_{\mathrm{S}}(\phi)$ has size linear in $|\phi|$ when the logic is fixed.

Because we do the introduction of new symbols subformula-wise, it is possible to put the translation of the automaton outside of the translation of the modal formula. At the position where $t_{\mathcal{A}_a}(\alpha, t(\psi, \alpha, \beta))$ is translated, only $\mathbf{q_{0,\psi}}(\alpha)$ needs to be inserted (where q_0 is the initial state of $\mathcal{A}_a$). The rest of the (translation of the) automaton can be put elsewhere.

Here is the main theorem about satisfiability preservation.

Theorem 3.5 *Let Σ be an alphabet with converse mapping $\bar{\cdot}$, let* S *be a regular semi-Thue system with converse over Σ, and let $\phi \in \mathcal{L}^{\Sigma}$ be a modal formula. Then, ϕ is S-satisfiable iff $T_{\mathrm{S}}(\phi)$ is satisfiable in first-order logic.*

The proof relies on the regularity of the languages $\mathrm{L}_{\mathrm{S}}(a)$ and on the existence of a closure operator on frames. For details of the proof, we refer to [13] or [12]. The uniformity of the translation allows us to establish the forthcoming Theorem 3.6. The general satisfiability problem for regular grammar logic with converse, denoted by $\mathrm{GSP}(\mathrm{REG}^c)$, is defined as follows:

input: a semi-Thue system with converse S represented either as a right-linear grammar or as a left-linear grammar, and an $\mathcal{L}^{\Sigma}$-formula ϕ;
question: is ϕ S-satisfiable?

Theorem 3.6 *The* S*-satisfiability problem is in* **EXPTIME** *for every regular semi-Thue system with converse and GSP(REG^c) is* **EXPTIME***-complete.*

If we would allow arbitrary semi-Thue systems in the definition of $\mathrm{GSP}(\mathrm{REG}^c)$, then Theorem 3.6 would not hold, because it is in general undecidable to determine whether a context-free semi-Thue system is regular.

4 Concluding Remarks

We have defined almost-structure preserving logspace transformations from a large class of modal logics into GF^2. We think that this provides evidence that the first-order fragment corresponding to the class of regular grammar logics with converse is simply GF^2, in particular no fixed point operators or transitive gaurds are needed. In addition, we characterized the complexity of the satisfiability problem for regular grammar logics and our method can be extended to other non-classical logics including nominal tense logics and intuitionistic logic (see details in [12]). The translation for intuitionistic propositional logic can be obtained by combining our translation, applied on modal logic S4, with Gödel's translation from intuitionistic logic into S4. (See [34] for the Gödel translation)

The encoding we used is reminiscent to the propagation of formulas in tableaux calculi (see e.g. [20,30,4]) and the study of this relationship may be worth being pursued. We also plan to implement our method, in order to obtain insight into the practical usefulness of our translation method.

References

[1] Abadi, M., M. Burrows, B. Lampson and G. Plotkin, *A calculus for access control in distributed systems*, ACM Transactions on Prog. Languages and Systems **15** (1993), pp. 706–734.

[2] Andreka, H., I. Nemeti and J. van Benthem, *Modal languages and bounded fragments of predicate logic*, JPL **27** (1998), pp. 217–274.

[3] Baldoni, M., L. Giordano and A. Martelli, *A tableau calculus for multimodal logics and some (un)decidability results*, in: H. de Swart, editor, *TABLEAUX-8*, LNAI **1397** (1998), pp. 44–59.

[4] Castilho, M., L. Fariñas del Cerro, O. Gasquet and A. Herzig, *Modal tableaux with propagation rules and structural rules*, FI **32** (1997), pp. 281–297.

[5] de Nivelle, H., *A resolution decision procedure for the guarded fragment*, in: C. Kirchner and H. Kirchner, editors, *CADE-15*, LNAI **1421** (1998), pp. 191–204.

[6] de Nivelle, H., *Translation of S4 and K5 into GF and 2VAR* (1999), manuscript.

[7] de Nivelle, H., *Translation of S4 and K5 into GF and 2VAR* (2001), slides available via `http://www.mpi-sb.mpg.de/~nivelle/` on WWW.

[8] de Nivelle, H. and M. de Rijke, *Deciding the guarded fragments with resolution*, JSC **35** (2003), pp. 21–58.

[9] Demri, S., *The nondeterministic information logic NIL is PSPACE-complete*, FI **42** (2000), pp. 211–234.

[10] Demri, S., *The complexity of regularity in grammar logics and related modal logics*, JLC **11** (2001), pp. 933–960.

[11] Demri, S., *Modal logics with weak forms of recursion: PSPACE specimens*, in: M. de Rijke, H. Wansing, F. Wolter and M. Zakharyaschev, editors, *AIML'2000* (2002), pp. 113–138.

[12] Demri, S. and H. de Nivelle, *Deciding regular grammar logics with converse through first-order logic*, arXiv:cs.LO/0306117 (2003), 32 pages, submitted to TOCL.

[13] Demri, S. and H. de Nivelle, *Deciding regular grammar logics with converse through first-order logic*, Technical Report LSV-03-4, LSV, ENS de Cachan (2003).

[14] Fagin, R., J. Halpern, Y. Moses and M. Vardi, "Reasoning about Knowledge," The MIT Press, 1995.

[15] Fariñas del Cerro, L. and M. Penttonen, *Grammar logics*, Logique et Analyse **121–122** (1988), pp. 123–134.

[16] Gabbay, D., *Decidability results in non-classical logics*, Annals of Mathematical Logic **8** (1975), pp. 237–295.

[17] Gabbay, D., *Expressive functional completeness in tense logic*, in: U. Mönnich, editor, *Aspects of Philosophical Logic*, Reidel, 1981 pp. 91–117.

[18] Ganzinger, H., C. Meyer and M. Veanes, *The two-variable guarded fragment with transitive relations (extended abstract)*, in: *LICS'99* (1999), pp. 24–34.

[19] Goranko, V. and S. Passy, *Using the universal modality: gains and questions*, JLC **2** (1992), pp. 5–30.

[20] Goré, R., *Tableaux methods for modal and temporal logics*, in: M. d'Agostino, D. Gabbay, R. Hähnle and J. Posegga, editors, *Handbook of Tableaux Methods* (1999), pp. 297–396.

[21] Grädel, E., *On the restraining power of guards*, JSL **64** (1999), pp. 1719–1742.

[22] Grädel, E., P. Kolaitis and M. Vardi, *On the decision problem for two-variable first-order logic*, Bulletin of Symbolic Logic **3** (1997), pp. 53–69.

[23] Harel, D., D. Kozen and J. Tiuryn, "Dynamic Logic," MIT Press, 2000.

[24] Hladik, J., *Implementation and optimisation of a tableau algorithm for the guarded fragment*, in: U. Egly and C. Fermüller, editors, *TABLEAUX-12*, LNCS **2381** (2002), pp. 145–159.

[25] Horrocks, I. and U. Sattler, *A description logic with transitive and inverse roles and role hierarchies*, JLC **9** (1999), pp. 385–410.

[26] Hustadt, U. and R. Schmidt, *A principle for incorporating axioms into the first-order translation of modal formulae*, in: F. Baader, editor, *CADE-19*, LNAI **2741** (2003), pp. 412–426.

[27] Kieronski, E., *The two-variable guarded fragment with transitive guards is 2EXPTIME-hard*, in: A. Gordon, editor, *FOSSACS'03*, LNCS **2620** (2003), pp. 299–312.

[28] Marx, M., S. Mikulas and S. Schlobach, *Tableau calculus for local cubic modal logic and its implementation*, Logic Journal of the IGPL **7** (1999), pp. 755–778.

[29] Massacci, F., *Tableaux methods for access control in distributed systems*, in: D. Galmiche, editor, *TABLEAUX'97*, LNAI **1227** (1997), pp. 246–260.

[30] Massacci, F., *Single steps tableaux for modal logics*, JAR **24** (2000), pp. 319–364.

[31] Morgan, C., *Methods for automated theorem proving in non classical logics*, IEEE Transactions on Computers **25** (1976), pp. 852–862.

[32] Ohlbach, H., A. Nonnengart, M. de Rijke and D. Gabbay, *Encoding two-valued non-classical logics in classical logic*, in: A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning* (2001), pp. 1403–1486.

[33] Szwast, W. and L. Tendera, *On the decision problem for the guarded fragment with transitivity*, in: *LICS'01*, 2001, pp. 147–156.

[34] Troelstra, A. and H. Schwichtenberg, "Basic Proof Theory," Cambridge Tracts in Computer Science **43**, Cambridge University Press, 1996.

[35] Vakarelov, D., *Abstract characterization of some knowledge representation systems and the logic NIL of nondeterministic information*, in: P. Jorrand and V. Sgurev, editors, *Artificial Intelligence: Methodology, Systems, Applications* (1987), pp. 255–260.

[36] van Benthem, J., "Correspondence Theory," Ph.D. thesis, Mathematical Institute, University of Amsterdam (1976).

[37] Wolter, F. and M. Zakharyashev, *On the relation between intuitionistic and classical modal logics*, Algebra and Logic **36** (1997), pp. 121–155.