Deciding regular grammar logics with converse through first-order logic*

Stéphane Demri[1] and Hans de Nivelle[2]

[1] LSV/CNRS UMR 8643 & INRIA Futurs projet SECSI & ENS Cachan
61, av. Pdt. Wilson, 94235 Cachan Cedex, France
email: demri@lsv.ens-cachan.fr
[2] Max Planck Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken, Germany
email: nivelle@mpi-sb.mpg.de

Abstract. We provide a simple translation of 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.
A consequence of the translation is that the general satisfiability problem for regular grammar logics with converse is in EXPTIME. This extends a previous result of the first author for grammar logics without converse. Using the same method, we show how some other modal logics can be naturally translated into GF^2, including nominal tense logics and intuitionistic logic. In our view, the results in this paper show that the natural first-order fragment corresponding to regular grammar logics is simply GF^2 without extra machinery such as fixed point-operators.

Keywords: modal and temporal logics, intuitionistic logic, relational translation, guarded fragment, 2-variable fragment

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. For *theorem proving* in modal logics, two main approaches can be distinguished. The first approach is to develope a theorem prover directly for the logic under consideration. The second approach is to translate the logic into some general logic, usually first-order logic. The first approach has the advantage that a specialized algorithm can make use of specific properties of the logic under consideration, enabling optimizations that do not work in general. In addition, implementation of a single modal logic is usually easier than implementing full

* Revised version of [DdN03].

first-order logic. But on the other hand, there are many modal logics, and it is simply not feasible to construct optimized theorem provers for all of them. The advantage of the second approach is that only one theorem prover needs to be written which can be reused for all translatable modal logics. In addition, a translation method can be expected to be more robust against small changes in the logic. Therefore translating seems to be the more sensible approach for most modal logics, with the exception of a few main ones.

Translation of modal logics into first-order logic, with the explicit goal to mechanise such logics is an approach that has been introduced in [Mor76]. Morgan distinguishes two types of translations: The *semantical translation*, which is nowadays known as the *relational translation* (see e.g., [Fin75,vB76,Moo77]) and the *syntactic* translation, which consists in reifing modal formulae (i.e., transforming them into first-order terms) and in translating the axioms and inference rules from a Hilbert-style system into classical logic using an additional provability predicate symbol. This is also sometimes called *reflection*. With such a syntactic translation, every propositional normal modal logic with a finite axiomatization can be translated into classical predicate logic. However, using this general translation, decidability of modal logics is lost. Therefore we will study relational translations in this paper, which instead of simply translating a modal formula into full first-order logic, can translate modal formulas into a decidable subset of first-order logic. The fragment that we will be using is GF^2, the intersection of the 2-variable fragment and the guarded fragment We modify the relational translation in such a way that explicit translation of frame properties can be avoided. In this way, many modal logics with frame properties outside GF^2 can be translated into GF^2.

A survey on translation methods for modal logics can be found in [ONdRG01], where more references are provided, for instance about the functional translation (see e.g., [Her89,Ohl93,Non96]), see also in [Orło88,DMP95] for other types of translations.

Guarded fragments. Both the guarded fragment, introduced in [ANvB98] (see also [dN98,GdN99,dNSH00,dNdR03]) and FO^2, the fragment of classical logic with two variables [Gab81,GKV97,dNPH01], have been used for the purpose of 'hosting' translations of modal formulas. The authors of [ANvB98] explicitly mention the goal of identifying 'the modal fragment of first-order logic' as a motivation for introducing the guarded fragment. Apart from having nice logical properties [ANvB98], the guarded fragment GF has an EXPTIME-complete satisfiability problem when the maximal arity of the predicate symbols is fixed in advance [Grä99b]. Hence its worst-case complexity is identical to some simple extensions of modal logic K, as for example K augmented with universal modality [Spa93]. Moreover, mechanisation of the guarded fragment is possible thanks to the design of efficient resolution-based decision procedures [dN98,GdN99]. In [Hla02], a tableau procedure for the guarded fragment with equality based on [HT02] is implemented and tested; see also a prover for FO^2 described in [MMS99].

However, there are some simple modal logics with the satisfiability problem in PSPACE ([Lad77]) that cannot be translated into GF through the relational translation. The reason for this is the fact that the frame condition that characterizes the logic cannot be expressed in GF. The simplest example of such a logic is probably S4 which is characterized by transitivity (many other examples will be given throughout the paper). Adding transitivity axioms to a GF-formula causes undecidability. (See [Grä99a])

Because of the apparent insufficiency of GF to capture basic modal logics, various extensions of GF have been proposed and studied. In [GMV99], it was shown that GF^2 with transitivity axioms is decidable, on the condition that binary predicates occur only in guards. The complexity bound given there is non-elementary, which makes the fragment not useful for practical purposes.

In [ST01], the complexity bound for GF^2 with transitive guards is improved to 2EXPTIME and it was shown 2EXPTIME-hard in [Kie03]. As a consequence, the resulting strategy is not the most efficient strategy to mechanize modal logics with transitive relations (such as S4)

Another fragment was explored in [GW99], see also [Grä99a]. There it was shown that μGF, the guarded fragment extended with a μ-calculus-style fixed point operator is still decidable and in 2EXPTIME. This fragment does contain the simple modal logic S4, but the machinery is much more heavy than the 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. In this paper, we put emphasis on the fact that GF^2 is a sufficiently well-designed fragment of classical logic for dealing with a large variety of modal logics. An approach that seems better suited for theorem proving, and that does more justice to the low complexities of simple modal logics is the approach taken in [dN99,dN01]. There an almost structure-preserving translation from the modal logics S4, S5 and K5 into GF^2 was given. The subformulas of a modal formula are translated in the standard way, except for subformulas which are formed by a universal modality. Universal subformulas are translated into a sequence of formulas, the exact form of which is determined by the frame condition. In [dN99,dN01], the translations and their correctness proofs were ad hoc, and it was not clear upon which principles they are based. In this paper we show that the almost structure preserving translation relies on the fact that the frame conditions for K4, S4 and K5 are *regular* in some sense that will be made precise in Section 2. The simplicity of the almost structure-preserving translation leaves hope that GF^2 may be rich enough after all to naturally capture most of the basic modal logics.

We call the translation method almost structure-preserving because it preserves the structure of the formula almost completely. Only for subformulas of form $[a]\phi$ does the translation differ from the usual relational translation. On these subformulas, a sequence of formulas is generated that simulates an NDFA based on the frame condition of 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. [HSZ96,Gor99,Mas00,dCG02,HS03a]): The tableau rule for subformulas of form $[a]\phi$ can be viewed as simulating an NDFA, in the same way as the almost structure-preserving translation.

Our contribution. We show that the methods of [dN99] can be extended to a very large class of modal logics. Some of the modal logics in this class have frame properties that can be expressed only by recursive conditions, like for example transitivity. By a *recursive condition* we mean a condition that needs to be iterated in order to reach a fixed point. 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 $[a_0]\mathrm{p} \Rightarrow [a_1] \ldots [a_n]\mathrm{p}$ 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 $\mathrm{GF}^2 = \mathrm{FO}^2 \cap \mathrm{GF}$, despite the fact that their frame conditions are not expressible in $\mathrm{FO}^2 \cup \mathrm{GF}$. These logics include the standard modal logics K4, S4, K5, K45, S5, some information logics (see e.g. [Vak87]), nominal tense logics (see e.g. [ABM00]), description logics (see e.g. [Sat96,HS99]), propositional intuitionistic logic (see e.g. [CZ97]) and bimodal logics for intuitionistic modal logics $\mathbf{IntK}_\Box + \Gamma$ as those considered in [WZ97]. Hence the main contribution of the paper is the design of a very simple and generic translation from regular grammar logics with converse into the decidable fragment of classical logic GF^2. The translation is easy to implement and it mimics the behavior of some tableaux-based calculi for modal logics. As a consequence, we are able to show that the source logics that can be translated into GF^2 have a satisfiability problem in EXPTIME. This allows us to establish such an upper bound uniformly for a very large class of modal logics, for instance for intuitionistic modal logics (another approach is followed in [AS03] leading to a bit less tight complexity upper bound). We are considering here the satisfiability problem. However because of the very nature of the regular grammar logics with converse, our results apply also to the global satisfiability problem and to the logical consequence problem. By contrast, this work does not deal with the model-checking problem since this problem amounts to a subproblem of the model-checking problem for classical FOL known to be PSPACE-complete.

We do not claim that for most source logics the existence of a transformation into GF^2 of low complexity is very surprising. In fact it is easy to see that from each simple modal logic for instance in PSPACE there must exist a polynomial transformation into GF^2, because PSPACE is a subclass of EXPTIME. The EXPTIME-completeness of fixed-arity GF implies that there exists a polynomial time transformation from every logic in PSPACE into fixed-arity GF. (It can even be shown that there exists a logarithmic space transformation.) However, the translation that establishes the reduction would normally make use of first principles on Turing machines. Trying to efficiently decide modal logics through such a transformation would amount to finding an optimal implementation in Turing machines, which is no easier than a direct implementation on a standard computer. Although many translated regular grammar logics with converse are EXPTIME-hard, see e.g. [Dem01], our translation is optimal for the logics in PSPACE, and it is part of our future work to look for fragments of GF which are in PSPACE.

Our paper answers a question stated in [Dem01]: Is there a decidable first-order fragment, into which the regular grammar logics can be translated in a natural way? The translation method that we give in this paper suggests that GF^2 is the answer. It is too early to state that the transformation from regular grammar logics with converse into GF^2 defined in the paper can be used to mechanize efficiently such source logics with a prover for GF^2, but we have shown evidence that GF^2 is a most valuable decidable first-order fragment to translate modal logics into, even though their frame conditions are not expressible in GF^2

Related work. Complexity issues for regular grammar logics have been studied in [Dem01,Dem02] (see also [Bal98,BGM98]) whereas grammar logics are introduced in [dCP88]. Frame conditions involving the converse relations are not treated in [Dem01,Dem02]. These are needed for example for S5 modal connectives. The current work can be viewed as a natural continuation of [dN99] and [Dem01]. In this paper, we use a direct translation into decidable first-order fragments instead of a translation into propositional dynamic logic as done in [Dem01] (see details about the latter approach in Section 4).
The frame conditions considered in the present work can be defined by the MSO definable closure operators [GMV99]. However, it is worth noting that by contrast to what is done in [GMV99], we obtain the optimal complexity upper bound for the class of regular grammar logics with converse (EXPTIME) since the first-order fragment we consider is much more restricted than the one in [GMV99]. Moreover, we do not use MSO definable built-in relations, just plain GF^2.

The recent work [HS03b] presents another translation of modal logics into decidable fragments of classical logic by encoding adequately the modal theories (see also [Ohl98]). Although for logics such as S4, the method in [HS03b] is very similar to ours, it is still open how the methods are related in the general case. For instance, no regularity conditions are explicitly involved in [HS03b] whereas this is a central point in our work.

Structure of the paper. Section 2 defines the class of regular grammar logics with converse via semi-Thue systems. It contains standard examples of such logics as well as the statement of a crucial closure property for the class of frames of these logics. Section 3 is the core of the paper and presents the translation into GF^2 as well as obvious extensions. In Section 4, we present alternative logarithmic space transformations that allow to regain the EXPTIME upper bound: one into GF^2 via converse PDL with automata and another one into the multimodal logic with K modalities, converse, and the universal modality. Section 5 illustrates how the transformation in Section 3 can be used to translate intuitionistic propositional logic IPL into GF^2. Section 6 concludes the paper and states open problems.

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 rewrite system. If one views accessibility relations as letters, then conditions on the accessibility relation can be conveniently represented by rewrite rules. For example transitivity $\forall xyz\ \mathbf{R}_a(x,y) \wedge \mathbf{R}_a(y,z) \rightarrow \mathbf{R}_a(x,z)$ can be represented by the rule $a \rightarrow a \cdot a$. Similarly, the inclusion $\forall xyz\ \mathbf{R}_a(x,y) \wedge \mathbf{R}_b(y,z) \rightarrow \mathbf{R}_c(x,z)$ can be represented by the rule $c \rightarrow a \cdot b$. In the next Section 2.1, we recall a few definitions about formal languages, semi-Thue systems, and finite-state automata. After that, in Section 2.2, we introduce modal frames, and explain what it means when a frame satisfies a rewrite rule. Using that, we can define grammar logics. A grammar logics is *regular* if the set of rewrite rules generates, for each letter, a regular language. In Section 2.3 we introduce converse mappings. A converse mapping is a function that assigns to each symbol b of an alphabet a unique symbol $\overline{b}$, *the converse of* $\overline{b}$. After that we define some technical conditions on grammars and modal frames which will ensure that converses do indeed behave like converses. Section 2.3 is concluded with an example and a discussion of the scope of grammar logics with converse.

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 $\mathrm{L_S}(u)$ to denote the language $\{v \in \Sigma^* : u \Rightarrow^*_{\mathrm{S}} v\}$.

The behaviour of a context-free semi-Thue system is fully characterized by the sets $\mathrm{L_S}(a)$, for each $a \in \Sigma$. A context-free semi-Thue system S, based on Σ is called *regular* if for every $a \in \Sigma$, the language $\mathrm{L_S}(a)$ is regular. In that case, one can associate to each $a \in \Sigma$ an NDFA (*non-deterministic finite automaton*, see [HU79]) recognizing the language $\mathrm{L_S}(a)$. We assume that there is some function $\mathcal{A}_a$, that associates such an NDFA to each $a \in \Sigma$. We do not specify which automaton $\mathcal{A}_a$ is associated to a. It would be possible to make $\mathcal{A}_a$ canonic, for example by putting $\mathcal{A}_a$ equal to the smallest DFA recognizing $\mathrm{L_S}(a)$, which is unique, but there seems to be no advantage in doing so. In many cases, an NDFA can have much less states than a corresponding DFA.

Observe that it is undecidable to check whether a context-free semi-Thue system is regular since it is undecidable whether the language generated by a linear grammar is regular (see e.g. [MS97, page 31]).

Semi-Thue systems are closely related to formal grammars, but in a semi-Thue system, the production rules are used for defining a relation between words, rather than for defining a subset of words. The former is precisely what we need to define grammar logics.

2.2 Grammar Logics

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 $a \to a \cdot a$. Similarly, reflexivity can be expressed by $a \to \epsilon$.

We first introduce modal languages, then we introduce modal frames and models. 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 write $|\phi|$ to denote the *length* of the formula ϕ, that is the number of symbols needed to write ϕ down. 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 standard 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 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 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 $\{a \to \epsilon,\ a \to aa\}$.

2.3 Converse Mappings

In order to be able to cope with properties such as symmetry and euclideanity, one also needs *converses.* For this reason, we associate to every symbol a in the alphabet a unique converse symbol $\overline{a}$. Using the converse symbols, for example symmetry on the relation R_a can be represented by the production rule $a \to \overline{a}$ whereas euclideanity on the relation R_a can be represented by the production rule $a \to \overline{a} \cdot a$.

Definition 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 1. *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 refer to such a partition, we assume that an arbitrary one is chosen. We call the modal operators indexed by letters in Σ^+ *forward modalities* (conditions on successor states) whereas the modal operators indexed by letters in Σ^- are called *backward modalities* (conditions on predecessor states).
In order to make sure that the converses really behave like converses, we need to put the following, obvious constraint on the Σ-frames.

Definition 3. *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 $a \in \Sigma$, $R_{\overline{a}}$ equals $\{\langle y, x \rangle : R_a(x, y)\}$.*

Definition 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 $\overline{u \cdot a} = \overline{a} \cdot \overline{u}$.*

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

Lemma 2. *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 $u \in \Sigma^*$, $R_{\overline{u}} = \{\langle y, x \rangle : R_u(x, y)\}$.*

An obvious consequence of Lemma 2 is that a $\langle \Sigma, \bar{\cdot} \rangle$-frame $\langle W, R \rangle$ satisfies some production rule $u \to v$ if and only if $\langle W, R \rangle$ satisfies the converse $\overline{u} \to \overline{v}$. Therefore, one can add the closures of the rules to a semi-Thue system without changing the $\langle \Sigma, \bar{\cdot} \rangle$-frames that satisfy it.

Definition 5. *Given a semi-Thue system* S *over some alphabet Σ with converse mapping $\bar{\cdot}$, we call* S closed under 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' *closed under converse, for which $S \subseteq \mathrm{S}'$.*

It is easily seen that the converse closure is always well-defined. Because of the remark before Lemma 5, every set of $\langle \Sigma, \bar{\cdot} \rangle$-frames that can be characterized by an arbitrary semi-Thue system, can also be characterized by a semi-Thue system which is closed under converse.

We call the logics that correspond to a set of $\langle \Sigma, \bar{\cdot} \rangle$-frames which can be characterized by a semi-Thue system *grammar logics with converse.* Those logics that correspond to $\langle \Sigma, \bar{\cdot} \rangle$-frames which can be characterized by a regular semi-Thue system, are called *regular grammar logics with converse.*
We give some remarks about the class of regular grammar logics with converse.

1. Because of the remarks before Definition 5, allowing or disallowing semi-Thue systems which are not closed under converse does not have consequences for the logics one can define.
2. It can be easily checked that every regular grammar logic (without converse) can be viewed as a fragment of a regular grammar logic with converse. This is due to the fact that one can add a set of converses $\overline{\Sigma}$ to the alphabet Σ and extend the Σ-frame into a $\langle \Sigma \cup \overline{\Sigma}, \bar{\cdot} \rangle$-frame.
3. Originally, grammar logics were defined with formal grammars in [dCP88] (as in [Bal98,Dem01,Dem02]), and they form a subclass of Sahlqvist modal logics [Sah75] with frame conditions expressible in Π_1 when S is context-free. Π_1 is the class of first-order formulae of the form $\forall\, x_1\, \forall\, x_2\, \ldots \forall\, x_n\, \phi$ where ϕ is quantifier-free. In the present paper, we adopt a lighter presentation based on semi-Thue systems as done in [CS94], which is more appropriate.

Example 1. 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 $\mathrm{L}_{\mathrm{S}}(a)$.

Numerous other logics for specific application domains are in fact regular grammar logics with converse, or logics that can be reduced to such logics. We list below some examples:

logic	$\mathrm{L_S}(a)$	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

- description logics (with role hierarchy, transitive roles), see e.g. [HS99];
- knowledge logics, see e.g. $\mathrm{S5}_m(\mathrm{DE})$ in [FHMV95];
- bimodal logics for intuitionistic modal logics of the form $\mathbf{IntK}_\Box + \Gamma$ [WZ97]. Indeed, let S be a regular semi-Thue system (over Σ) closed under converse and let $\Sigma' \subset \Sigma$ be such that for every $a \in \Sigma$, either $a \not\in \Sigma'$ or $\overline{a} \not\in \Sigma'$. Then, the semi-Thue system $\mathrm{S} \cup \{b \to bab, \overline{b} \to \overline{b}\overline{a}\overline{b} : a \in \Sigma'\}$ over $\Sigma \cup \{b, \overline{b}\}$ is also regular, assuming $b, \overline{b} \not\in \Sigma$. By taking advantage of [GMV99], in [AS03] decidability of intuitionistic modal logics is also shown in a uniform manner.
- fragments of logics designed for the access control in distributed systems [ABLP93,Mas97].
- extensions with the universal modality [GP92]. 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 [GP92] (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. [Vak87]. For instance, the Nondeterministic Information Logic NIL introduced in [Vak87] (see also [Dem00]) 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:
  - $fin \to fin \cdot fin$; $fin \to \epsilon$;
  - $sim \to \overline{sim}$; $sim \to \epsilon$;
  - $sim \to \overline{fin} \cdot sim \cdot fin$.

  For instance $\mathrm{L_S}(sim) = \{\overline{fin}\}^* \cdot \{sim, \overline{sim}, \epsilon\} \cdot \{fin\}^*$.

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 corresponding to this inclusion is $\mathrm{S}_n = \{a \to \overline{a}^n a, \ \overline{a} \to \overline{a} a^n\}$. The case $n = 1$ corresponds to euclideanity. Although we have seen that for $n = 1$, the language $\mathrm{L}_{\mathrm{S}_1}(a)$ is regular, one can establish that in general, for $n > 1$, the language $\mathrm{L}_{\mathrm{S}_n}(a)$

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. [Gab75,HS03b]. To see why the languages $\mathrm{L}_{\mathrm{S}_n}(a)$ 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

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

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

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

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

1. A string of form $\sigma_n(0,j)$ with $j > 0$ has no predecessor.
2. A string of form $\sigma_n(i+1,j)$ has only one predecessor, namely $\overline{\sigma}_n(i,j)$.
3. A string of form $\overline{\sigma}_n(i,0)$ has no predecessor.
4. A string of form $\overline{\sigma}_n(i,j+1)$ has only one predecessor, namely $\sigma_n(i,j)$.

To have a predecessor, 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.
We have

$$\mathrm{L}_{\mathrm{S}_n}(a) \cap \{\sigma_n(i,j) : i \geq 0, \ j \geq 0\} = \{\sigma_n(i,i) : i \geq 0\}.$$

$\{\sigma_n(i,i) : i \geq 0\}$ is clearly not regular (we assume $n > 1$) and $\{\sigma_n(i,j) : i \geq 0, \ j \geq 0\}$ is clearly regular. Since the regular languages are closed under intersection, $\mathrm{L}_{\mathrm{S}_n}(a)$ cannot be regular for $n > 1$. 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 [HKT00]).

2.4 Characterizations of $\langle \Sigma, \bar{\cdot} \rangle$-Frames

Theorem 1 below states the usual relations between derivations, validity and frame conditions.

Theorem 1. *Let Σ be an alphabet, $u, v \in \Sigma^*$, and S be a context-free semi-Thue system over alphabet Σ, which is closed under converse. Consider the statements below:*

(I) $u \Rightarrow^*_{\mathrm{S}} v$.

(II) *In every $\langle \Sigma, \bar{\cdot} \rangle$-frame $\mathcal{F}$ satisfying* S, *for arbitrary $p \in$* PROP,
$[u]\mathrm{p} \Rightarrow [v]\mathrm{p}$ *is valid.*
(For a word $u = (u_1, \ldots, u_m)$, $[u]\mathrm{p}$ is an abbreviation for $[u_1] \cdots [u_m]\mathrm{p}$.)
(III) *$R_v \subseteq R_u$ in every $\langle \Sigma, \bar{\cdot} \rangle$-frame $\mathcal{F}$ satisfying* S.
(This is the same as saying that $\mathcal{F}$ makes $u \to v$ true.)

Then, (II) is equivalent to (III), (I) implies (II), but (II) does not necessarily imply (I).

The equivalence between (II) and (III) is a classical correspondence result in modal logic theory (see e.g., [vB84]). The proof does not make use of the fact that the frame $\mathcal{F}$ is a $\langle \Sigma, \bar{\cdot} \rangle$-frame. (I) implies (III) holds for every Σ-frame, and therefore also for every $\langle \Sigma, \bar{\cdot} \rangle$-frame. It can be proved by induction on the length of the derivation of $u \Rightarrow^*_{\mathrm{S}} v$. More precisely, when (I) holds, then there is an i, such that $u \Rightarrow^i_{\mathrm{S}} v$. Then (III) is proven by induction on i.

In order to show that (II) does not necessarily imply (I), consider the semi-Thue system $\mathrm{S} = \{a \to \overline{a},\ \overline{a} \to a,\ b \to a^3,\ \overline{b} \to \overline{a}^3\}$. In this system, $b \not\Rightarrow^*_{\mathrm{S}} a$. However, $a \to \overline{a}$ expresses symmetry, which in a $\langle \Sigma, \bar{\cdot} \rangle$-frame satisfying S implies that whenever $\langle x, y \rangle \in R_a$, then also $\langle x, y \rangle \in R_{a^3}$. Therefore, $R_a \subseteq R_b$.

It is worth observing that in the absence of converse, (II) implies (I) by the proof of [CS94, Theorem 3] (see also the tableaux-based proof in [Bal98]). This is based on the fact that every ordered monoid is embeddable into some ordered monoid of binary relations (see more details in [CS94]). In the presence of converse, this property does not hold since it is not true that every tense ordered monoid is embeddable into some tense ordered monoid of binary relations.

In Section 3.2, we need only the implication (I) $\Rightarrow$ (III) for proving the first direction of Theorem 3.

If some Σ-frame $\mathcal{F} = \langle W, R \rangle$ does satisfy a regular semi-Thue system, then one can add missing edges to R until the resulting Σ-frame does satisfy the semi-Thue system. In this way, one obtains a closure function that assigns to each Σ-frame the smallest frame that satisfies the semi-Thue system.

Definition 6. *We first define an inclusion relation on frames. Let Σ be an alphabet. Let $\mathcal{F}_1 = \langle W, R_1 \rangle$ and $\mathcal{F}_2 = \langle W, R_2 \rangle$ be two Σ-frames with the same set of worlds W. We say that $\mathcal{F}_1$ is a* subframe *of $\mathcal{F}_2$ if for every $a \in \Sigma$, $R_{1,a} \subseteq R_{2,a}$.*

Using this, we define the closure operator C_{S} *as follows: For every context-free semi-Thue system* S *over alphabet Σ, for every Σ-frame $\mathcal{F}$, the* closure *of $\mathcal{F}$ under* S *is defined as the smallest Σ-frame (under the subframe relation) that satisfies* S, *and which has $\mathcal{F}$ as a subframe. We write $C_{\mathrm{S}}(\mathcal{F})$ for the closure of $\mathcal{F}$.*

The closure always exists, and is unique, because of the Knaster-Tarski fixed point theorem. It can also be proven from the forthcoming Theorem 2. Definition 6 does not mention converse mappings. We will later (Lemma 3) show that C_{S} transforms $\langle \Sigma, \bar{\cdot} \rangle$-frames into $\langle \Sigma, \bar{\cdot} \rangle$-frames, even if S is not closed under converse.

We now prove a crucial property of C_{S}, namely that every edge added by C_{S} can be justified in terms of an $\mathrm{L}_{\mathrm{S}}(a)$.

Theorem 2. *Let* S *be a context-free semi-Thue system, which is closed under converse. Let* $\mathcal{F} = \langle W, R \rangle$ *be a* Σ*-frame. Let the* Σ*-frame* $\mathcal{F}' = \langle W, R' \rangle$ *be defined from*

$$R'_a = \bigcup_{u \in \mathrm{L}_{\mathrm{S}}(a)} R_u, \text{ for } a \in \Sigma.$$

Then $\mathcal{F}' = C_{\mathrm{S}}(\mathcal{F})$.

Proof. We have to show that

1. $\langle W, R' \rangle$ satisfies S, and
2. among the frames that satisfy S, and that have $\mathcal{F}$ as subframe, $\langle W, R' \rangle$ is a minimal such frame.

In order to show **(1)**, we show that for every rule $a \to u$ in S, the inclusion $R'_u \subseteq R'_a$ holds. Write $u = (u_1, \ldots, u_n)$, with $n \geq 0$, and each $u_i \in \Sigma$. Let $\langle x, y \rangle \in R'_u$. We intend to show that $\langle x, y \rangle \in R'_a$.
By definition, there are $z_1, \ldots, z_{n-1} \in W$, such that

$$\langle x, z_1 \rangle \in R'_{u_1}, \quad \langle z_1, z_2 \rangle \in R'_{u_2}, \ldots, \quad \langle z_{n-1}, y \rangle \in R'_{u_n}.$$

By construction of R', there are words $v_1, \ldots, v_n \in \Sigma^*$, such that $u_1 \Rightarrow^*_{\mathrm{S}} v_1, \quad \ldots, \quad u_n \Rightarrow^*_{\mathrm{S}} v_n$, and

$$\langle x, z_1 \rangle \in R_{v_1}, \quad \langle z_1, z_2 \rangle \in R_{v_2}, \ldots, \quad \langle z_{n-1}, y \rangle \in R_{v_n}.$$

As a consequence, $\langle x, y \rangle \in R_{v_1 \cdot \ldots \cdot v_n}$ Now because $a \Rightarrow_{\mathrm{S}} u, \quad u = (u_1, \ldots, u_n)$, and each $u_i \Rightarrow^*_{\mathrm{S}} v_i$, we also have $a \Rightarrow^*_{\mathrm{S}} v_1 \cdot \ldots \cdot v_n$. It follows that $\langle x, y \rangle \in R'_a$, by the way R' is constructed.
Next we show **(2)**. Let $\langle W, R'' \rangle$ be a Σ-frame, such that $\langle W, R \rangle$ is a subframe of $\langle W, R'' \rangle$ and $\langle W, R'' \rangle$ satisfies S. We want to show that for each $a \in \Sigma, \quad R'_a \subseteq R''_a$.

If $\langle x, y \rangle \in R'_a$, this means that $\langle x, y \rangle \in R_u$ for a u with $a \Rightarrow^*_{\mathrm{S}} u$. Because $R_u \subseteq R''_u$, we also have $\langle x, y \rangle \in R''_u$. Because $\langle W, R'' \rangle$ is an Σ-frame satisfying S, it follows from Theorem 1 ((I) $\Rightarrow$ (III)) that $R''_u \subseteq R''_a$, and we have $\langle x, y \rangle \in R''_a$.

Lemma 3. *Let* Σ *be an alphabet with converse mapping* $\bar{\cdot}$ *and let* $\mathcal{F} = \langle W, R \rangle$ *be a* $\langle \Sigma, \bar{\cdot} \rangle$*-frame. Then for every context-free semi-Thue system* S *over* Σ*, the closure* $C_{\mathrm{S}}(\mathcal{F})$ *is also a* $\langle \Sigma, \bar{\cdot} \rangle$*-frame.*

Proof. Write $C_{\mathrm{S}}(\mathcal{F}) = \langle W, R' \rangle$. We need to show that for each $a \in \Sigma$,

$$R'_a = \{ \langle x, y \rangle \mid \langle y, x \rangle \in R'_{\bar{a}} \}.$$

In case that this not hold, one can define R'' from

$$R''_a = \{ \langle x, y \rangle \mid \langle x, y \rangle \in R'_a \text{ and } \langle y, x \rangle \in R'_{\bar{a}} \}, \text{ for } a \in \Sigma.$$

and $\langle W, R'' \rangle$ is a strict subframe of $\langle W, R' \rangle$ satisfying S. Because also $\langle W, R \rangle$ is a subframe of $\langle W, R'' \rangle$, we obtained a contradiction with the minimality of $\langle W, R \rangle$.

When S is regular, the map C_{S} is a monadic second-order definable graph transduction in the sense of [Cou94] and it is precisely the inverse substitution h^{-1} in the sense of [Cau03] (see also [Cau96]) when the extended substitution h is defined by $a \in \Sigma \mapsto \mathrm{L}_{\mathrm{S}}(a)$.

3 The Translation into GF^2

In this chapter, we define the transformation from regular grammar logics with converse into GF^2. The transformation can be carried out in logarithmic space. When translating a $\Box$-subformula, the translation simulates the behaviour of an NDFA in order to determine to which worlds the $\Box$-formula applies. This generalises the results in [dN99,dN01] for the logics S4 and K5, which were at an ad hoc basis. Here we show that it is the regularity of the frame condition that makes the translation method work. The translation allows us to provide an EXPTIME upper bound for the satisfiability problem for regular grammar logics with converse. Other specific features of our translation are the following ones:

- This is not an exact translation of the Kripke semantics of modal logics, which makes it different from the relational translation. Indeed, we rather define a transformation from the satisfiability problem for a regular grammar logic with converse into the satisfiability problem for GF^2. Hence, our translation is a reduction as understood in complexity theory, see e.g. [Pap94].
- The translation is based on 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 closed under converse over an alphabet Σ with converse mapping $\bar{\cdot}$ (and partition $\{\Sigma^+, \Sigma^-\}$). For every $a \in \Sigma$, the automaton $\mathcal{A}_a$ is an NDFA (possibly with ϵ-transitions) recognizing the language $\mathrm{L_S}(a)$. It is in principle allowed that $\mathcal{A}_a$ and $\mathcal{A}_{\overline{a}}$ are unrelated automata, although they have to accept isomorphic languages (because $u \in \mathrm{L_S}(a)$ iff $\overline{u} \in \mathrm{L_S}(\overline{a})$ for every $u \in \Sigma^*$). 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, $F_a \subseteq Q_a$ is the set of accepting states, and δ_a is the transition function, which is possibly non-deterministic. When all rules in S are either right-linear[1] or left-linear[2], 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.

In the sequel we assume that the two variables in GF^2 are $\{x_0, x_1\}$. α and β are used as distinct meta-variables in $\{x_0, x_1\}$. Observe that in Definition 9 the quantification alternates over α and β.

Definition 7. *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.*

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

[1] i.e., there is a partition $\{V, T\}$ of Σ such that the production rules are in $V \rightarrow T^* \cdot (V \cup \{\epsilon\})$.

[2] i.e., there is a partition $\{V, T\}$ of Σ such that the production rules are in $V \rightarrow (V \cup \{\epsilon\}) \cdot T^*$.

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

We could have separated the symmetry from the function t itself, in the same way as we did with frames in Definition 2.2 and Definition 3. Then we would have first defined t_a for each $a \in \Sigma$ without any conditions, and later defined that some t *respects* $\cdot$ if always t_a is the converse of $t_{\overline{a}}$. This would in principle be elegant and allow us to specify more clearly which property of the translation depends on which part of the definition of the translation. However it would make the formulations of the properties more tedious, so we decided not to do this.

We now define the main part of the translation. It takes two parameters, a one-place first-order 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 8. *Let $\mathcal{A} = \langle Q, s, F, \delta \rangle$ 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.

The function $t_{\mathcal{A}}(\alpha, \psi)$ is applied on formulas ψ that are subformulas of an initial formula ϕ. The definition requires that in each application of $t_{\mathcal{A}}$, distinct predicate symbols of form $\mathbf{q}$ for $q \in Q$ are introduced. This can be done either occurrence-wise, or subformula-wise. Occurrence-wise means that, if some subformula ψ of ϕ occurs more than once, then different fresh predicate symbols have to be introduced for each occurrence. Subformula-wise means that the different occurrences can share the fresh predicates. In the sequel, we will assume the subformula-wise approach.

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 process 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.

Now we can give the translation itself. It behaves like 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 it, but it would have more cases.

Definition 9. *Let $\phi \in \mathcal{L}^{\Sigma}$ be a modal formula in NNF. Let* S *be a regular semi-Thue system closed under converse over alphabet Σ with converse mapping $\bar{\cdot}$. Assume that for each $a \in \Sigma$ an automaton $\mathcal{A}_a$ recognizing $\mathrm{L}_{\mathrm{S}}(a)$ is given. We define the translation $T_{\mathrm{S}}(\phi)$ as $t(\phi, x_0, x_1)$ from the following function $t(\psi, \alpha, \beta)$, which is defined by recursion on the subformulas ψ of ϕ :*

- *$t(\mathrm{p}, \alpha, \beta)$ equals $\mathbf{p}(\alpha)$, where* **p** *is a unary predicate symbol uniquely associated to the propositional variable* p.
- *$t(\neg\mathrm{p}, \alpha, \beta)$ 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 8 is used. The only difference with the standard relational translation is the translation of $[a]$-formulae.

Lemma 4. *For the translation $T_{\mathrm{S}}(\phi)$, the following holds:*

(I) *The only variables occurring in $T_{\mathrm{S}}(\phi)$ are in $\{x_0, x_1\}$ and α is the only free variable in $t(\psi, \alpha, \beta)$.*
(II) *$T_{\mathrm{S}}(\phi)$ is in the guarded fragment.*
(III) *The size of $T_{\mathrm{S}}(\phi)$ is in $\mathcal{O}(|\phi| \times m)$.*
(IV) *$T_{\mathrm{S}}(\phi)$ can be computed in logarithmic space in $|\phi| + m$.*

Here m is the size of the largest $\mathcal{A}_a$, i.e. $m = \max\{\ |\mathcal{A}_a|\ \mid\ a \in \Sigma\ \}$.

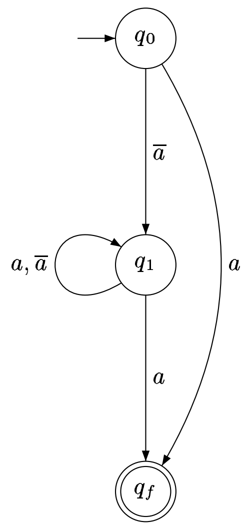
When S is formed of production rules of a formal grammar that is either right-linear or left-linear, then m is in $\mathcal{O}(|\mathrm{S}|)$. For a given semi-Thue system S, the number m is fixed. As a consequence, $T_{\mathrm{S}}(\phi)$ has size linear in $|\phi|$ for a given logic.

Unlike the standard relational translation from modal logic into classical predicate logic (see e.g., [Fin75,vB76,Mor76,Moo77]), the subformulae in $T_{\mathrm{S}}(\phi)$ mix the frame conditions and the interpretation of the logical connectives. Besides $T_{\mathrm{S}}(\phi)$ can be viewed as the logical counterpart of the propagation rules defined in [CdCGH97] (see also [Gor99,Mas00,dCG02]).

Example 2. Let $\phi = \Diamond\mathrm{p} \wedge \Diamond\Box\neg\mathrm{p}$ be the negation normal form of the formula $\neg(\Diamond\mathrm{p} \Rightarrow \Box\Diamond\mathrm{p})$. We consider K5, and assume one modality a, so $\Box$ is an abbreviation for $[a]$, and $\Diamond$ is an abbreviation for $\langle a \rangle$. Table 2 contains to the left an automaton $\mathcal{A}_a$ recognizing the language defined in Table 1 for K5. To the right is the translation $t_{\mathcal{A}_a}(\alpha, \varphi(\alpha))$ for some first-order formula $\varphi(\alpha)$. The translation $T_{\mathrm{S}}(\phi)$ of ϕ is equal to

$$\exists\beta\ [\ \mathbf{R}(\alpha,\beta) \wedge \mathbf{p}(\beta)\ ] \wedge \exists\beta\ [\ \mathbf{R}(\alpha,\beta) \wedge t_{\mathcal{A}_a}(\beta, \mathbf{p}(\beta)\ )\ ].$$

$\mathcal{A}_a$



$t_{\mathcal{A}_a}(\alpha, \varphi(\alpha))$

$$\mathbf{q}_{0,\varphi}(\alpha)$$

$$\forall\alpha\beta\ [\ \mathbf{R}(\alpha,\beta) \rightarrow \mathbf{q}_{0,\varphi}(\alpha) \rightarrow \mathbf{q}_{f,\varphi}(\beta)\ ]$$
$$\forall\alpha\beta\ [\ \mathbf{R}(\beta,\alpha) \rightarrow \mathbf{q}_{0,\varphi}(\alpha) \rightarrow \mathbf{q}_{1,\varphi}(\beta)\ ]$$

$$\forall\alpha\beta\ [\ \mathbf{R}(\alpha,\beta) \rightarrow \mathbf{q}_{1,\varphi}(\alpha) \rightarrow \mathbf{q}_{1,\varphi}(\beta)\ ]$$
$$\forall\alpha\beta\ [\ \mathbf{R}(\beta,\alpha) \rightarrow \mathbf{q}_{1,\varphi}(\alpha) \rightarrow \mathbf{q}_{1,\varphi}(\beta)\ ]$$

$$\forall\alpha\beta\ [\ \mathbf{R}(\alpha,\beta) \rightarrow \mathbf{q}_{1,\varphi}(\alpha) \rightarrow \mathbf{q}_{f,\varphi}(\beta)\ ]$$

$$\forall\alpha\ [\ \mathbf{q}_{f,\varphi}(\alpha) \rightarrow \varphi(\alpha)\ ]$$

Table 2. The K5 automaton and $t_{\mathcal{A}_a}(\alpha, \varphi(\alpha))$ for arbitrary $\varphi(\alpha)$

Since we perform 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. (We will use this in Section 4.2). 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.

Extension with nominals. The map T_{S} can be obviously extended to admit nominals in the language of the regular grammar logics with converse. The treatment of nominals can be done in the usual way by extending the definition of t as follows: $t(\mathbf{i}, \alpha, \beta) \stackrel{\mathrm{def}}{=} \mathbf{c}_{\mathbf{i}} = \alpha$ where $\mathbf{c}_{\mathbf{i}}$ is a constant associated with the nominal $\mathbf{i}$. The target first-order fragment is GF^2 with constants and identity. For instance, nominal tense logics with transitive frames (see e.g., [ABM00]), and description logics with transitive roles and converse (see e.g., [Sat96]), can be translated into $\mathrm{GF}^2[=]$ with constants in such a way. Additionnally, by using [BM02, Sect. 4] regular grammar logics with converse augmented with Gregory's "actually" operator [Gre01] can be translated into such nominal tense logics.

Relationships with first-order logic over finite words. The method of translating finite automata into first-order formulas by introducing unary predicate symbols for the states, is reminiscent to the characterization of regular languages in terms of Monadic Second-Order Logic over finite words, namely SOM[+1], see e.g. [Str94]. Similarly, the class of languages with a *finite* syntactic monoid is precisely the class of regular languages. Our encoding into GF^2 is quite specific since

- we translate into an EXPTIME fragment of FOL, namely GF^2, neither into full FOL nor into a logic over finite words;
- we do not encode regular languages into GF^2 but rather modal logics whose frame conditions satisfy some regularity conditions, expressible in GF with built-in relations [GMV99];
- not every regularity condition can be encoded by our method since we require a closure condition.

Hence, the similarity between the encoding of regular languages into SOM[+1] and our translation is quite superficial. The following argument provides some more evidence that the similarity exists only at the syntactic level. The class of regular languages definable with the first-order theory of SOM[+1] is known as the class of star-free languages (their syntactic monoids are finite and aperiodic), see e.g. [Per90]. However, the regular language $\mathrm{L}_{\mathrm{S}}(a) = (b{\cdot}b)^*(a{\cup}\epsilon)$ obtained with the regular semi-Thue system $\mathrm{S} = \{a \rightarrow bba, a \rightarrow \epsilon\}$ produces a regular grammar logic with converse that can be translated into GF^2 by our method. Observe that the language $(b \cdot b)^*(a \cup \epsilon)$ is not star-free, see e.g. [Pin94]. By contrast, $(a \cdot b)^*$ is star-free but it is not difficult to show that there is no context-free semi-Thue system S such that $\mathrm{L}_{\mathrm{S}}(a) = (a \cdot b)^*$ since a is not in $(a \cdot b)^*$. As a conclusion, our translation into GF^2 is based on principles different from those between star-free regular languages and first-order logic on finite words. Other problems on (tree) automata translatable into classical logic can be found in [Ver03].

3.2 Satisfiability Preservation

We show that map T_{S} preserves satisfiability. First, we introduce some notation. A *first-order model* is denoted by $\langle W, V \rangle$ where W is a non-empty set and V maps unary [resp. binary] predicate symbols into subsets of W [resp. $W \times W$]. Given a first-order model $\langle W, V \rangle$, we define $V(a)$ for every $a \in \Sigma$ as follows:
$$V(a) = \begin{cases} V(\mathbf{R}_a) \text{ if } a \in \Sigma^+ \\ V(\mathbf{R}_a)^{-1} \text{ if } a \in \Sigma^- \end{cases}$$
The following, rather technical, lemma states roughly the following: Suppose we

have a first-order model $\langle W, V\rangle$ containing some point $w \in W$, such that in every point v, reachable from w through a path that is accepted by the automaton $\mathcal{A}$ the formula $\varphi(\alpha)$ is true, then we can extend V in such a way, that the new model $\langle W, V'\rangle$ will satisfy the translation $t_{\mathcal{A}}(\alpha, \varphi)$ in w.

Lemma 5. *Let $\mathcal{A}$ be an NDFA and $\varphi(\alpha)$ be a first-order formula with one free variable α. Let $\mathcal{M} = \langle W, V\rangle$ be a first-order structure not interpreting any of the fresh symbols introduced by $t_{\mathcal{A}}(\alpha, \varphi)$ (those of the form $\mathbf{q}_\varphi$ for each state q of $\mathcal{A}$). Then there is an extension $\mathcal{M}' = \langle W, V'\rangle$ of $\mathcal{M}$, such that for every $w \in W$, satisfying (I) below, w also satisfies*

$$\mathcal{M}',\ v[\alpha \leftarrow w] \models t_{\mathcal{A}}(\alpha, \varphi).$$

(I) *For every word $b_1 \cdots b_n \in \Sigma^*$ that is accepted by $\mathcal{A}$, for every sequence $w_1, \ldots, w_n$ of elements of W such that*

$$\langle w, w_1\rangle \in V(b_1),\ \ \langle w_1, w_2\rangle \in V(b_2),\ \ldots,\ \ \langle w_{n-1}, w_n\rangle \in V(b_n),$$

we have $\mathcal{M}, v[\alpha \leftarrow w_n] \models \varphi(\alpha)$.

Proof. We want to extend V in such a way that the added interpretations for the symbols $\mathbf{q}_\varphi$ simulate $\mathcal{A}$. This can be obtained by the following extension: For all $w \in W$ and all states q of $\mathcal{A}$, we set $w \in V'(\mathbf{q}_\varphi)$ iff for every word $b_1 \cdots b_n \in \Sigma^*$ such that there is an accepting state q_f of $\mathcal{A}$, such that $q_f \in \delta^*(q, b_1 \cdots b_n)$, and for every sequence $w_1, \ldots, w_n$ of elements of W, s.t.

$$\langle w, w_1\rangle \in V(b_1),\ \ \langle w_1, w_2\rangle \in V(b_2),\ \ldots,\ \ \langle w_{n-1}, w_n\rangle \in V(b_n),$$

we have

$$\mathcal{M}, v[\alpha \leftarrow w_n] \models \varphi(\alpha).$$

Here δ^* is the natural extension of δ to words over Σ^*.

Write $\mathcal{A} = \langle Q, s, F, \delta\rangle$. It is easy to check (but tedious to write out because of the size of the statements involved) that for every $w \in W$ satisfying (I) in the statement of the lemma, we have

- for the initial state s,
$$\mathcal{M}', v[\alpha \leftarrow w] \models \mathbf{s}_\varphi(\alpha).$$
- for each $q \in Q$, for each $a \in \Sigma$, for each $r \in \delta(q, a)$,
$$\mathcal{M}' \models \forall\alpha\beta\ [\ t_a(\alpha, \beta) \to \mathbf{q}_\varphi(\alpha) \to \mathbf{r}_\varphi(\beta)\ ].$$
- for each $q, r \in Q$, such that $r \in \delta(q, \epsilon)$,
$$\mathcal{M}' \models \forall\alpha\ [\ \mathbf{q}_\varphi(\alpha) \to \mathbf{r}_\varphi(\alpha)\ ].$$
- for each final state $q \in F$,
$$\mathcal{M}' \models \forall\alpha\ [\ \mathbf{q}_\varphi(\alpha) \to \varphi(\alpha)\ ].$$

$\mathcal{M}'$ and $\mathcal{M}$ agree on all formulas that do not contain any symbols introduced by $t_{\mathcal{A}}(\alpha, \varphi)$.

Next follows the main theorem about satisfiability preservation.

Theorem 3. *Let Σ be an alphabet with converse mapping $\bar{\cdot}$, let* S *be a regular semi-Thue system closed under converse over Σ, and let $\phi \in \mathcal{L}^{\Sigma}$ be a modal formula. Then,*

(I) *ϕ is S-satisfiable iff*
(II) *$T_{\mathrm{S}}(\phi)$ is satisfiable in FOL.*

The proof relies on the regularity of the languages $\mathrm{L}_{\mathrm{S}}(a)$ and on Theorem 2.

Proof. We first prove **(I)** $\rightarrow$ **(II)**. Assume that ϕ is S-satisfiable. This means that there exists a $\langle \Sigma, \bar{\cdot} \rangle$-model $\mathcal{M} = \langle W, R, V \rangle$ with a $w \in W$ such that $\mathcal{M}, w \models \phi$ and $\langle W, R \rangle$ satisfies S. We need to construct a model $\mathcal{M}'$ (noted $\mathcal{M}_n$ in the sequel) of $T_{\mathrm{S}}(\phi)$. In order to do this, we first construct an incomplete interpretation $\mathcal{M}_0 = \langle W, V_0 \rangle$, and after that we will complete it through successive applications of Lemma 5.

- for every $a \in \Sigma^+$, $V_0(\mathbf{R}_a) \stackrel{\text{def}}{=} R_a$;
- for every propositional variable p, we set $V_0(\mathbf{p}) \stackrel{\text{def}}{=} V(\mathrm{p})$.

We now have a model interpreting the symbols introduced by $t(\psi, \alpha, \beta)$, but not the symbols introduced by $t_{\mathcal{A}}(\alpha, \psi)$. In order to complete the model construction, we order the box-subformulas of ϕ in a sequence $[a_1]\psi_1, \ldots, [a_n]\psi_n$ such that every box-subformula is preceeded by all its box-subformulae. Hence, $i < j$ implies that $[a_j]\psi_j$ is not a subformula of $[a_i]\psi_i$. Then we iterate the following construction ($1 \leq i \leq n$):

- $\mathcal{M}_i = \langle W, V_i \rangle$ is obtained from $\mathcal{M}_{i-1} = \langle W, V_{i-1} \rangle$ by applying the construction of Lemma 5 on $\mathcal{A}_{a_i}$ and $t(\psi_i, \alpha, \beta)$.

Then $\mathcal{M}_n = \langle W, V_n \rangle$ is our final model. Roughly speaking, V_i is equal to V_{i-1} extended to the unary predicate symbols of the form $\mathbf{q}_{\psi_i}$ with q a state of $\mathcal{A}_{a_i}$. The values of the other predicate symbols remain constant, that is:

- for every $a \in \Sigma^+$, $V_0(\mathbf{R}_a) = \cdots = V_n(\mathbf{R}_a)$;
- for every propositional variable p in ϕ, $V_0(\mathbf{p}) = \cdots = V_n(\mathbf{p})$.

Additionally, for every $j \in \{1, \ldots, n\}$, for every state q of $\mathcal{A}_{a_j}$, $V_j(\mathbf{q}_{\psi_j}) = V_{j+1}(\mathbf{q}_{\psi_j}) = \ldots = V_n(\mathbf{q}_{\psi_j})$.

We show by induction that for every subformula ψ of ϕ, for every $x \in W$, for every valuation v, $\mathcal{M}, x \models \psi$ implies $\mathcal{M}_n, v[\alpha \leftarrow x] \models t(\psi, \alpha, \beta)$. Here $v[\alpha \leftarrow x]$ denotes the valuation v' obtained from v by putting $v'(\beta) \stackrel{\text{def}}{=} v(\beta)$ and $v'(\alpha) = x$. We treat only the modal cases, because the propositional cases are trivial.

- If ψ has form $[a]\psi'$ with $a \in \Sigma$, then $t([a]\psi', \alpha, \beta) = t_{\mathcal{A}_a}(\alpha, t(\psi', \alpha, \beta))$. We have that for every word $b_1 \cdots b_l$ accepted by $\mathcal{A}_a$, for every sequence $w_1, \ldots, w_l \in W_n$ s.t.

$$\langle x, w_1 \rangle \in V_n(b_1), \ \ \langle w_1, w_2 \rangle \in V_n(b_2), \ \ldots \ , \langle w_{l-1}, w_l \rangle \in V_n(b_l),$$

also

$$\langle x, w_1 \rangle \in R_{b_1}, \ \langle w_1, w_2 \rangle \in R_{b_2}, \ \cdots \ , \langle w_{l-1}, w_l \rangle \in R_{b_l},$$

by construction of $V_0, V_1, \cdots, V_n$. Because $\mathcal{M}$ satisfies S, $\langle x, w_l \rangle \in R_a$ (by Theorem 1((I) $\rightarrow$ (III))), which again implies $\langle x, w_l \rangle \in V_n(a)$, by construction of the V_i. Therefore, we have $\mathcal{M}, w_l \models \psi'$. By the induction hypothesis, we have $\mathcal{M}_n, v[\beta \leftarrow w_l] \models t(\psi', \beta, \alpha)$. Let n' be the position of ψ' in the enumeration of box-subformulae $[a_1]\psi_1, \ldots, [a_n]\psi_n$. It is easily checked that

$$\mathcal{M}_{n'}, v[\beta \leftarrow w_l] \models t(\psi', \beta, \alpha).$$

Now we have all ingredients of Lemma 5 complete, and it follows that

$$\mathcal{M}_{n'}, v[\alpha \leftarrow x] \models t_{\mathcal{A}_a}(\alpha, t(\psi', \alpha, \beta)).$$

Since $\mathcal{M}_n$ is a conservative extension $\mathcal{M}_{n'}$, we also get

$$\mathcal{M}_n, v[\alpha \leftarrow x] \models t_{\mathcal{A}_a}(\alpha, t(\psi', \alpha, \beta)).$$

- If ψ has form $\langle a \rangle \psi'$, then there is a y such that $\langle x, y \rangle \in R_a$ and $\mathcal{M}, y \models \psi'$. By definition of V_0, we have $\langle x, y \rangle \in V_0(a)$ and therefore $\langle x, y \rangle \in V_n(a)$. By the induction hypothesis, $\mathcal{M}_n, v[\beta \leftarrow y] \models t(\psi', \beta, \alpha)$. Hence,

$$\mathcal{M}_n, v[\alpha \leftarrow x] \models \exists \beta \ [\ t_a(\alpha, \beta) \wedge t(\psi', \beta, \alpha)].$$

(II) $\rightarrow$ (I) Suppose that $T_{\mathrm{S}}(\phi)$ is FOL-satisfiable. This means that there exist a FOL model $\mathcal{M} = \langle W, V \rangle$ and a valuation v such that $\mathcal{M}, v \models T_{\mathrm{S}}(\phi)$. We construct a model $\mathcal{M}'$ of ϕ in two stages: First we construct $\mathcal{M}'' = \langle W'', R'', V'' \rangle$ as follows:

- $W'' = W$.
- For every $a \in \Sigma$, $R''_a = V(a)$.
- For every propositional variable p, $V''(\mathrm{p}) = V(\mathbf{p})$.

Then define $\mathcal{M}' = \langle W', R', V' \rangle$ where R' is defined from $\langle W', R' \rangle = C_S(\langle W'', R'' \rangle)$ and $V' = V''$. Here C_S is the closure operator, defined in Definition 6. Intuitively, we construct $\mathcal{M}'$ by copying W and the interpretation of the accessibility relations from $\mathcal{M}$, and applying C_{S} on it. The constructions imply that $W' = W$. By definition of C_{S}, $\mathcal{M}'$ is an S-model, and by Lemma 3, $\langle W'', R'' \rangle$ is a $\langle \Sigma, \bar{\cdot} \rangle$-frame. We now show by induction that for every subformula ψ of ϕ, $\mathcal{M}, v \models t(\psi, \alpha, \beta)$ implies $\mathcal{M}', v(\alpha) \models \psi$.

- If ψ has form $\langle a\rangle\psi'$, then $\mathcal{M}, v \models t(\langle a\rangle\psi', \alpha, \beta)$, that is $\mathcal{M}, v \models \exists\beta\, [\, t_a(\alpha,\beta) \wedge t(\psi', \beta, \alpha)\,]$.
  This means there is a $y \in W$, such that $\mathcal{M}, v[\beta \leftarrow y] \models t_a(\alpha, \beta)$, and
  $$\mathcal{M}, v[\beta \leftarrow y] \models t(\psi', \beta, \alpha).$$
  By the induction hypothesis, $\mathcal{M}', y \models \psi'$. It follows from the definition of R', using the fact that C_S is increasing (by its definition), that $\langle x, y\rangle \in R'_a$, so we have $\mathcal{M}', x \models \langle a\rangle\psi$.
- If ψ has form $[a]\psi'$, then suppose $\mathcal{M}, v \models t_{\mathcal{A}_a}(\alpha, t(\psi', \alpha, \beta))$. One can establish that for every word $b_1 \cdots b_l$ accepted by $\mathcal{A}_a$, for every sequence $w_1, \ldots, w_l$ of elements of W, for which it is the case that
  $$\langle v(\alpha), w_1\rangle \in V(b_1),\ \langle w_1, w_2\rangle \in V(b_2),\ \ldots, \langle w_{l-1}, w_l\rangle \in V(b_l),$$
  the following holds
  $$\mathcal{M}, v[\alpha \leftarrow w_l] \models t(\psi', \alpha, \beta).$$
  Indeed, $\mathcal{M}, v \models \mathbf{s}(\alpha)$, for the initial state s of $\mathcal{A}_a$. It is easy to show by induction that the following holds: Let $b_1 \cdots b_l$ be some word over Σ^*. Let δ^* be the natural extension of δ to words. Let q be a state of $\mathcal{A}_a$ such that $q \in \delta^*(s, b_1 \cdots b_l)$, for the initial state $s \in Q$. Then for every sequence $w_1, \ldots, w_l$ of elements of W such that
  $$\langle v(\alpha), w_1\rangle \in V(b_1), \quad \langle w_1, w_2\rangle \in V(b_2), \quad \ldots, \quad \langle w_{l-1}, w_l\rangle \in V(b_l),$$
  it must be the case that $\mathcal{M}, v[\alpha \leftarrow w_l] \models \mathbf{q}(\alpha)$. Then the result follows from the fact that $\mathcal{M}, v[\alpha \leftarrow w_l] \models \mathbf{q}(\alpha) \rightarrow \varphi(\alpha)$, for each accepting state q of $\mathcal{A}_a$.

  Now assume that in $\mathcal{M}'$, we have a world y for which $R'_a(x, y)$. Then, using Theorem 2, there is a word w that is accepted by $\mathcal{A}_a$, such that $R''_w(x, y)$. By the above property, we have $\mathcal{M}, v[\alpha \leftarrow y] \models t(\psi', \alpha, \beta)$. By the induction hypothesis, we have $\mathcal{M}', y \models \psi'$.

The uniformity of the translation allows us to establish forthcoming Theorem 4. We first define the general satisfiability problem for regular grammar logic with converse, denoted by GSP(REG^c), as follows:

input: A semi-Thue system S with converse, in which either all rewrite rules are left-linear, or all rewrite rules are right-linear, and an $\mathcal{L}^{\Sigma}$-formula ϕ;
question: is ϕ S-satisfiable?

We need to restrict the form of the semi-Thue system to a form from which the automata $\mathcal{A}_a$ can be computed. Even if one knows that some language L is regular, then there is no effective way of obtaining an NDFA for L. This is a consequence of Theorem 2.12 (iii) in [RS94].

Theorem 4.

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

Theorem 4(II) is stronger than Theorem 4(I) because GSP(REG^c) covers the satisfiability problems for all regular grammar logics with converse. Theorem 4(I) is a corollary of Theorem 3. The lower bound in Theorem 4(II) is easily obtained by observing that there exist known regular grammar logics (even without converse) that are already EXPTIME-complete, e.g. K with the universal modality. The upper bound in Theorem 4(II) is a consequence of the facts that $T_{\mathrm{S}}(\phi)$ can be computed in logarithmic space in $|\phi| + |\mathrm{S}|$ and the guarded fragment has an EXPTIME-complete satisfiability problem when the arity of the predicate symbols is bounded by some fixed $k \geq 2$ [Grä99b]. We use here the fact that one needs only logarithmic space to build a finite automaton recognizing the language of a right-linear [resp. left-linear] grammar.

Extensions to context-free grammar logics with converse. When S is a context-free semi-Thue system with converse, S-satisfiability can be encoded as for the case of regular semi-Thue systems with converse by adding an argument to the predicate symbols of the form $\mathbf{q}_{\psi}$. The details are omitted here but we provide the basic intuition. Each language $\mathrm{L_S}(a)$ is context-free and therefore there is a pushdown automaton (PDA) $\mathcal{A}$ recognizing it. The extra argument for the $\mathbf{q}_{\psi}$s represents the content of the stack and the map $t_{\mathcal{A}}(\alpha, \varphi)$ can be easily extended in the presence of stacks. For instance, the stack content aab can be represented by the first-order term $a(a(b(\epsilon)))$ with the adequate arity for the function symbols a, b, and ϵ. Suppose we have the following transition rule: if the PDA is in state q, the current input symbol is a, and the top symbol of the stack is b_0, then the new state is q' and b_0 is replaced by $b_1 \cdots b_n$ on the top of the stack. This rule is encoded in FOL as follows:

$$\forall\ \alpha, \beta, \gamma,\ (t_a(\alpha, \beta) \Rightarrow (\mathbf{q}(\alpha, b_0(\gamma)) \Rightarrow \mathbf{q}'(\beta, b_1(\ldots b_n(\gamma)\ldots)))).$$

The translation T_{S} is then defined with the context-free version of $t_{\mathcal{A}}(\alpha, \varphi)$. Satisfiability preservation is also guaranteed but the first-order fragment in which the translation is performed (beyond GF) is not anymore decidable. Hence, although this provides a new translation of context-free grammar logics with converse, from the point of view of effectivity, this is not better than the relational translation which is also known to be possible when S is a context-free semi-Thue system with converse.

4 Alternative Proofs of the EXPTIME Upper Bound

In this section, we provide two alternative ways to show that GSP(REG^c) is in EXPTIME. We believe that not only this sheds some new light to the proof of Section 3 but also it emphasizes the peculiarities of the class of regular grammar logics with converse. Observe that filtration-like techniques might also establish decidability of logics from GSP(REG^c), if not to the whole fragment. However, with such a technique the size of the built models is usually at least exponential in the size of the formulae, so we might get at best an NEXPTIME upper bound. That is why we did not develope further here this kind of proof.

4.1 Converse PDL with Automata

In [Dem01], it is shown how to translate the general satisfiability problem for regular grammar logics without converse into satisfiability for PDL but this map was not logarithmic space because given a regular grammar, equivalent regular expressions can be of exponential size (see e.g. [HU79]). That is why PDL with automata (APDL), see e.g. [HKT00], has been considered in [Dem01] in order to obtain a logarithmic space transformation into an EXPTIME logic. Similarly, it is possible to define a logarithmic space transformation from GSP(REG^c) into GF^2 by first translating GSP(REG^c) into ACPDL (converse PDL with automata) and then by translating a fragment of ACPDL into GF^2. ACPDL is an extension of APDL where the set of atomic programs is a countably infinite alphabet with converse mapping of the form $\{a_i : i \in \mathbb{N}^*\} \cup \{\overline{a_i} : i \in \mathbb{N}^*\}$ In the ACPDL models, we have $R_{\overline{a}} = R_a^{-1}$. The translation from GSP(REG^c) into ACPDL is mainly based on the step translating $[a]\psi$ into $[\mathcal{A}_a]t(\psi)$ where $\mathcal{A}_a$ is a ACPDL-automaton accepting the regular language $\mathrm{L_S}(a)$. When S is either right-linear or left-linear, $\mathcal{A}_a$ can be computed in polynomial-time in $|\mathrm{S}|$. Observe that ACPDL is more expressive than the class of regular grammar logics with converse since there is no context-free semi-Thue system S such that $\mathrm{L_S}(a_1) = (a_1 \cdot a_2)^*$.

By mimicking the translation into GF^2 from Section 3.1, one cannot translate full ACPDL into GF^2. Indeed, a syntactic restriction on ACPDL formulae similar to the existence of a closure operator on Σ-frames can be defined so that such a ACPDL fragment can be translated into GF^2. Such a fragment contains the translated formulae from GSP(REG^c) However, we omit here the definition of the translation since then the translation is not so much more informative that the one from Section 3.1.

The ACPDL fragment in question contains the formulae with ACPDL automata $\mathcal{A}_1, \overline{\mathcal{A}_1}, \ldots, \mathcal{A}_n, \overline{\mathcal{A}_n}$ built over the atomic programs $\{a_1, \ldots, a_n\} \cup \{\overline{a_1}, \ldots, \overline{a_n}\}$ satisfying the properties below:

- the words of $\mathrm{L}(\mathcal{A}_i)$ are the reverse words of $\mathrm{L}(\overline{\mathcal{A}_i})$.
- for every $i \in \{1, \ldots, n\}$, $a_i \in \mathrm{L}(\mathcal{A}_i)$.
- for every word $u \in \mathrm{L}(\mathcal{A}_i)$ [resp. $u \in \mathrm{L}(\overline{\mathcal{A}_i})$],
  1. for every occurrence a_j, the word obtained from u by replacing that occurrence of a_j by any word in $\mathrm{L}(\mathcal{A}_j)$ is also in $\mathrm{L}(\mathcal{A}_i)$ [resp. $\mathrm{L}(\overline{\mathcal{A}_i})$];
  2. for every occurrence $\overline{a_j}$, the word obtained from u by replacing that occurrence of $\overline{a_j}$ by any word in $\mathrm{L}(\overline{\mathcal{A}_j})$ is also in $\mathrm{L}(\mathcal{A}_i)$ [resp. $\mathrm{L}(\overline{\mathcal{A}_i})$].

4.2 Multimodal $\mathrm{K}_t + [U]$

We have defined in Section 3.1 an almost structure-preserving map from regular grammar logics with converse into GF^2. Below, we provide hints to understand how this map can be turned into a map into the multimodal logic K_t with forward and backward modalities $\{[i], [i]^{-1}, \langle i \rangle, \langle i \rangle^{-1} : i \geq 1\}$[3] augmented with the universal modality $[U]$. More interestingly, this logic can be then viewed as an EXPTIME-complete pivot logic between regular grammar logics with converse and GF^2 and many decision procedures exist for it (see e.g., [DM00,HS00,BT01]).

Let S be a regular semi-Thue system with converse and let ϕ be an $\mathcal{L}^{\Sigma}$-formula in negation normal form (NNF). Let us define $t(\phi)$ by induction on the subformulae of ϕ.

- $t(l) \stackrel{\text{def}}{=} l$ for every literal l;
- $t(\psi \wedge \psi') \stackrel{\text{def}}{=} t(\psi) \wedge t(\psi')$ (similar for $\vee$);
- $t(\langle a \rangle \psi) \stackrel{\text{def}}{=} \langle i_a \rangle t(\psi)$ where i_a is a modal index associated with $a \in \Sigma^+$;
- $t(\langle \overline{a} \rangle \psi) \stackrel{\text{def}}{=} \langle i_a \rangle^{-1} t(\psi)$ with $a \in \Sigma^+$;
- $t([a]\psi) \stackrel{\text{def}}{=} \mathrm{p}_{s,\psi}$ where $\mathrm{p}_{s,\psi}$ is a propositional variable associated with the initial state s of $\mathcal{A}_a$, $a \in \Sigma$, and $[a]\psi \in sub(\phi)$.

More generally, for every $[a]\psi \in sub(\phi)$ and for every $q \in \mathcal{A}_a$, we shall introduce a new propositional variable $\mathrm{p}_{q,\psi}$.

As done in Section 3, for every $a \in \Sigma$, for every ψ such that $[a]\psi$ occurs in ϕ, we construct formulas $t(\mathcal{A}_a, \psi)$. The conjunctions contain the following formulas:

- For each $b \in \Sigma^+$, $q, r \in Q_a$, if $r \in \delta_a(q, b)$, then the formula $[U](\mathrm{p}_{q,\psi} \Rightarrow [i_b]\mathrm{p}_{r,\psi})$ is present.
- For each $b \in \Sigma^-$, $q, r \in Q_a$, if $r \in \delta_a(q, b)$, then the formula $[U](\mathrm{p}_{q,\psi} \Rightarrow [i_b]^{-1}\mathrm{p}_{r,\psi})$ is present.
- For $q, r \in Q_a$, if $r \in \delta_a(q, \epsilon)$, then the formula $[U](\mathrm{p}_{q,\psi} \Rightarrow \mathrm{p}_{r,\psi})$ is present.
- For each $q \in F_a$, the formula $[U](\mathrm{p}_{q,\psi} \Rightarrow t(\psi))$ is present.

The translation $T_{\mathrm{S}}(\phi)$ is defined as $t(\phi) \wedge \bigwedge_{[a]\psi \in sub(\phi)} t(\mathcal{A}_a, \psi)$. The size of $T_{\mathrm{S}}(\phi)$ is in $\mathcal{O}(|\phi| + |\mathrm{S}|)$ and $T_{\mathrm{S}}(\phi)$ can be computed in logarithmic space in $|\phi| + |\mathrm{S}|$.

Theorem 5. *Let Σ be an alphabet with converse mapping $\bar{\cdot}$, S be a regular semi-Thue system with converse over Σ, and $\phi \in \mathcal{L}^{\Sigma}$. Then, ϕ is S-satisfiable iff $T_{\mathrm{S}}(\phi)$ is $K_t + [U]$ satisfiable.*

The proof is by an easy verification by observing that the first-order formula obtained with the relational translation from the $\mathrm{K}_t + [U]$ formula $T_{\mathrm{S}}(\phi)$ is almost syntactically equal to the formula $T_{\mathrm{S}}(\phi)$ from Section 3 when the last three clauses of Definition 8 are extracted from $t_{\mathcal{A}}(\alpha, \varphi)$ which is correct when the

[3] $\langle i \rangle^{-1}$ is also noted P_i (existential past-time operator) and $[i]^{-1}$ is also noted H_i (universal past-time operator).

translation is done subformula-wise. Theorem 3 concludes the proof. In general, the first-order formulae obtained by relational translation from $\mathrm{K}_t + [U]$ formulae are not in the guarded fragment. However, the relational translation of a formula $T_{\mathrm{S}}(\phi)$ is always in GF^2. Extensions with nominals in $\mathcal{L}^{\Sigma}$ and in $\mathrm{K}_t + [U]$ is obvious by adding the clause $t(\mathbf{i}) = \mathbf{i}$ for each nominal $\mathbf{i}$.

Let us consider CPDL, the version of PDL with ; (composition), $\cup$ (nondeterministic choice), * (iteration), $^{-1}$ (converse). The set of program constants is $\{\mathrm{c}_1, \mathrm{c}_2, \ldots\}$. For additional material on CPDL we refer to [HKT00]. A formula ϕ of $\mathrm{K}_t + [U]$ with modal indices in $\{1, \ldots, n\}$ can be translated to CPDL by replacing every occurrence of $[i]$ by $[\mathrm{c}_i]$, every occurrence of $[i]^{-1}$ by $[\mathrm{c}_i^{-1}]$, and every occurrence of $[U]$ by $[(\mathrm{c}_1 \cup \ldots \cup \mathrm{c}_{n+1} \cup \mathrm{c}_1^{-1} \cup \ldots \cup \mathrm{c}_{n+1}^{-1})^*]$. One can show that ϕ is $\mathrm{K}_t + [U]$ satisfiable iff the translated formula is CPDL satisfiable. Elimination of the universal modality does not make any problem because we are dealing with connected models. The proof is standard (see e.g. [FL79,Tuo90,GP92]). Hence, by combining this result with Theorem 5, we obtain a logarithmic space transformation from $\mathrm{GSP}(\mathrm{REG}^c)$ into CPDL.

5 Translating Intuitionistic Propositional Logic into GF^2

We define a new translation from intuitionistic propositional logic IPL with connectives $\rightarrow, \vee, \wedge$ and $\perp$ (see e.g. details in [CZ97]) into GF^2.

The translation method we present is technically not difficult since the translation is obtained by composing Gödel's translation into S4 with our translation from S4 into the guarded fragment, see e.g. [TS96] for the Gödel translation. This provides another logarithmic space embedding of IPL into a decidable fragment of classical logic (see e.g. [KK97]. Our translation could be used as a method for theorem proving in intuitionistic logic, but it still has to be determined whether our translation results in an efficient procedure. For more direct methods, we refer to the contraction-free calculus of Dyckhoff [Dyc92], or a resolution calculus, see [Min90], which is implemented in [Tam].

Let ϕ be an intuitionistic formula. We cannot base the translation on the negation normal form of ϕ, as we did in Section 3, because intuitionistic logic does not admit an equivalent negation normal form. (For example $\neg\phi \vee \psi$ is not equivalent to $\phi \rightarrow \psi$) Therefore, we explicitly add the polarity to the translation function. A similar technique was used in for example [DG00].

Before defining the map, we repeat the translation from IPL into S4, as given in [TS96].

Definition 10. *Function t_{S4} is defined as follows by recursion on the subformulas of ϕ.*

- $t_{S4}(\perp)$ *equals* $\perp$,
- *for a propositional symbol* p, $t_{S4}(\mathrm{p})$ *equals* $\Box\mathrm{p}$,
- $t_{S4}(\psi \wedge \psi')$ *equals* $t_{S4}(\psi) \wedge t_{S4}(\psi')$,
- $t_{S4}(\psi \vee \psi')$ *equals* $t_{S4}(\psi) \vee t_{S4}(\psi')$,
- $t_{S4}(\psi \rightarrow \psi')$ *equals* $\Box(t_{S4}(\psi) \rightarrow t_{S4}(\psi'))$.

Translation t_{S4} takes an intuitionistic formula ϕ and returns an S4-formula, not necessarilly in NNF. Translation t_{S4} preserves *provability*, so formula ϕ is provable in IPL iff $t_{S4}(\phi)$ is S4-valid. A formula $t_{S4}(\phi)$ is provable iff its negation $\neg t_{S4}(\phi)$ is unsatisfiable. In order to use the methods of Section 3, the translation function t_{S4} has to be modified in such a way that it (1) directly constructs the negated formula, and (2) it constructs a modal formula in negation normal form. The result is the following modified transformation t_{S4}. It takes an intuitionistic formula ϕ and the polarity $\pi \in \{0,1\}$, which is determined by the context of the result. In order to construct the translation of a formula ϕ, one needs to construct $t_{S4}(\phi,0)$. Then ϕ is provable in IPL iff $t_{S4}(\phi,0)$ is unsatisfiable. The new translation function t_{S4} is defined by recursion:

Definition 11.

- $t_{S4}(\bot,1)$ *equals* $\bot$,
- $t_{S4}(\bot,0)$ *equals* $\top$,
- *for a propositional symbol* p, $t_{S4}(\mathrm{p},1)$ *equals* $\Box\mathrm{p}$,
- *for a propositional symbol* p, $t_{S4}(\mathrm{p},0)$ *equals* $\Diamond\neg\mathrm{p}$,
- $t_{S4}(\psi \wedge \psi',1)$ *equals* $t_{S4}(\psi,1) \wedge t_{S4}(\psi',1)$,
- $t_{S4}(\psi \wedge \psi',0)$ *equals* $t_{S4}(\psi,0) \vee t_{S4}(\psi',0)$,
- $t_{S4}(\psi \vee \psi',1)$ *equals* $t_{S4}(\psi,1) \vee t_{S4}(\psi',1)$,
- $t_{S4}(\psi \vee \psi',0)$ *equals* $t_{S4}(\psi,0) \wedge t_{S4}(\psi',0)$,
- $t_{S4}(\psi \to \psi',1)$ *equals* $\Box(\ t_{S4}(\psi,0) \vee t_{S4}(\psi',1)\ )$,
- $t_{S4}(\psi \to \psi',0)$ *equals* $\Diamond(\ t_{S4}(\psi,1) \wedge t_{S4}(\psi',0)\ )$.

It is easily checked that $t_{S4}(\phi,\pi) \Leftrightarrow \neg t_{S4}(\phi,1-\pi)$ is a theorem of modal logic K (and of modal logic S4 *a fortiori*). Using this, it is easily checked that $t_{S4}(\phi,0)$ equals the negation normal form of $\neg t_{S4}(\phi)$.

Logic S4 has the frame condition that the accessibility relation should be reflexive and transitive. This can be expressed by a semi-Thue system S by taking a singleton alphabet $\Sigma = \{a\}$, and $\mathrm{L_S}(a) = a^*$. This language can be easily recognized by a one-state regular automaton $\mathcal{A} = (\{q\}, q, \{q\}, \{(q,a,q)\})$. A translation $t_{\mathcal{A}}(\alpha,\varphi)$ (Definition 8) will introduce one unary predicate $\mathbf{q}_\varphi$. We are now ready to define the translation. It introduces the following symbols:

- One binary relation $\mathbf{R}$ (interpreted as the S4 reflexive and transitive relation);
- A unary predicate symbol $\mathbf{p}$, for every propositional variable p in ϕ. This symbol serves two purposes at the same time: It is the unary predicate symbol representing the unique state of $\mathcal{A}$ in $t_{\mathcal{A}}(\alpha, t(\mathrm{p},\alpha,\beta,0))$, and also the translation of p itself.
- A unary predicate symbol $\mathbf{P}_{\psi\to\psi'}$, for every subformula of ϕ that occurs negatively and that has form $\psi \to \psi'$. This is the unary predicate needed for representing the state of $\mathcal{A}$ in the translation $t_{\mathcal{A}}(\alpha, t(\ \Box(\ t_{S4}(\psi,0) \vee t_{S4}(\psi',1)\ )))$.

Definition 12. *The translation function* $t(\phi,\alpha,\beta,\pi)$ *is defined by recursion on the subformulae of* ϕ*, for* $\pi \in \{0,1\}$.

- $t(\bot, \alpha, \beta, 1)$ *equals* $\bot$,
- $t(\bot, \alpha, \beta, 0)$ *equals* $\top$,
- $t(\psi \wedge \psi', \alpha, \beta, 1)$ *equals* $t(\psi, \alpha, \beta, 1) \wedge t(\psi', \alpha, \beta, 1)$,
- $t(\psi \wedge \psi', \alpha, \beta, 0)$ *equals* $t(\psi, \alpha, \beta, 0) \vee t(\psi', \alpha, \beta, 0)$,
- $t(\psi \vee \psi', \alpha, \beta, 1)$ *equals* $t(\psi, \alpha, \beta, 1) \vee t(\psi', \alpha, \beta, 1)$,
- $t(\psi \vee \psi', \alpha, \beta, 0)$ *equals* $t(\psi, \alpha, \beta, 0) \wedge t(\psi', \alpha, \beta, 0)$,
- $t(\psi \rightarrow \psi', \alpha, \beta, 1)$ *equals the conjunction*

$$\mathbf{P}_{\psi \rightarrow \psi'}(\alpha) \wedge \forall \alpha\beta \; [\; \mathbf{R}(\alpha, \beta) \rightarrow \mathbf{P}_{\psi \rightarrow \psi'}(\alpha) \rightarrow \mathbf{P}_{\psi \rightarrow \psi'}(\beta) \;] \; \wedge$$

$$\forall \alpha \; [\; \mathbf{P}_{\psi \rightarrow \psi'}(\alpha) \rightarrow t(\psi, \alpha, \beta, 0) \vee t(\psi', \alpha, \beta, 1) \;].$$

- $t(\psi \rightarrow \psi', \alpha, \beta, 0)$ *equals* $\exists \beta \; [\; \mathbf{R}(\alpha, \beta) \wedge t(\psi, \beta, \alpha, 1) \wedge t(\psi', \beta, \alpha, 0) \;]$,
- $t(\mathrm{p}, \alpha, \beta, 1)$ *equals*

$$\mathbf{p}(\alpha) \wedge \forall \alpha\beta \; [\; \mathbf{R}(\alpha, \beta) \rightarrow \mathbf{p}(\alpha) \rightarrow \mathbf{p}(\beta) \;],$$

- $t(\mathrm{p}, \alpha, \beta, 0)$ *equals* $\exists \beta \; [\; \mathbf{R}(\alpha, \beta) \wedge \neg \mathbf{p}(\beta) \;]$.

We write $T(\phi)$ to denote $t(\phi, \alpha, \beta, 0)$. Using Theorem 3 for S4 and Gödel's translation from IPL into S4, one can easily show the following:

Theorem 6. *ϕ is intuitionistically valid iff $T(\phi)$ is* GF^2 *unsatisfiable.*

Proof. Indeed, one can easily show that $T(\phi)$ is satisfiable iff $T_{\mathrm{S}}(t_{S4}(\phi, 0))$ is satisfiable where T_{S} is defined for the semi-Thue system for S4. By Theorem 3, $t_{S4}(\phi, 0)$ is S4-unsatisfiable iff $T(\phi)$ is GF^2 unsatisfiable. However, $t_{S4}(\phi, 0)$ is equivalent to $\neg t_{S4}(\phi, 1)$ and $t_{S4}(\phi, 1)$ is S4 valid iff ϕ is intuitionistically valid. Hence, ϕ is intuitionistically valid iff $T(\phi)$ is GF^2 unsatisfiable.

The translation from IPL into GF^2 can be extended in a similar way to various intuitionistic modal logics from [WZ97]. This allows us to get a uniform decidability proof for such logics even though in the case of plain IPL the translation is not the tightest one since IPL provability is PSPACE-complete whereas GF^2 is EXPTIME-complete.

6 Concluding Remarks

Fig. 1 contains logarithmic space transformations from $\mathrm{GSP}(\mathrm{REG}^c)$ into logics such as GF^2_2 (GF^2 restricted to predicate symbols of arity at most 2), CPDL, and $\mathrm{K}_t + [U]$ (possibly augmented with nominals, extensions noted with "+ N"). In a sense, although the main contribution of the paper consists in designing a simple logarithmic space transformation from $\mathrm{GSP}(\mathrm{REG}^c)$ into GF^2_2 by simulating the behaviour of finite automata, Fig. 1 shows other encodings of $\mathrm{GSP}(\mathrm{REG}^c)$ which could be effectively used to mechanize modal logics captured by $\mathrm{GSP}(\mathrm{REG}^c)$ (see examples of such logics throughout the paper).

The encoding we used is reminiscent to the propagation of formula in tableaux calculi (see e.g. [Gor99,Mas00,CdCGH97,dCG02]) and the study of such a relationship may be worth being pursued. The study of the computational behaviour

$$
\begin{tikzcd}
\mathrm{FO}^2[=] + \text{const.} & & & \\
\mathrm{GF}^2_2[=] + \text{const.} \arrow[u, dotted] & \mathrm{GSP}(\mathrm{REG}^c) + \mathrm{N} \arrow[l, "\text{Sect. 3}"'] \arrow[r, "\text{Sect. 4.2}"] & \mathrm{K}_t + [U] + \mathrm{N} \arrow[r, "\text{Sect. 4.2}"] \arrow[llu, bend right, "\text{relational translation}"'] & \mathrm{CPDL} + \mathrm{N} \\
\mathrm{GF}^2_2 \arrow[u, dotted] & \mathrm{GSP}(\mathrm{REG}^c) \arrow[l, "\text{Sect. 3}"'] \arrow[r, "\text{Sect. 4.2}"] \arrow[u, dotted] & \mathrm{K}_t + [U] \arrow[r, "\text{Sect. 4.2}"] \arrow[u, dotted] & \mathrm{CPDL} \arrow[u, dotted]
\end{tikzcd}
$$

Fig. 1. Logarithmic space transformations

of the translation to mechanize modal logics using for instance [dNPH01] is also an interesting direction for future work.

Additionally, our work allows us to answer positively to some questions left open in [Dem01]. Typically, we provide evidence that the first-order fragment to translate into the regular grammar logics with converse is simply GF^2: no need for first-order fragment augmented with fixed-point operators. Moreover, we characterize the complexity of such logics and we illustrate how the map can be extended for other non-classical logics including nominal tense logics and intuitionistic logic.

We list a few open problems that we believe are worth investigating.

1. Although regular grammar logics (with converse) can be viewed as fragments of propositional dynamic logic, it remains open whether the full PDL can be translated into GF^2 with a similar, almost-structure preserving transformation. We know that there exists a logarithmic space transformation, but we do not want to use first principles on Turing machines.
2. How to design a PSPACE fragment of GF^2 in which the following modal logics can be naturally embedded: S4, $\mathrm{S4}_t$ (S4 with past-time operators), Grz, and G? (to quote a few modal logics in PSPACE, see e.g. [CZ97]).
3. Can our translation method be extended to first-order modal logics?
4. Can it be extended to first-order intuitionistic logic?
5. Finally, a further comparison of the recent work [HS03b] with ours should be carried out.

References

[ABLP93] M. Abadi, M. Burrows, B. Lampson, and G. Plotkin. A calculus for access control in distributed systems. *ACM Transactions on Prog. Languages and Systems*, 15(4):706–734, 1993.

[ABM00] C. Areces, P. Blackburn, and M. Marx. Complexity of hybrid temporal logics. *Logic Journal of the IGPL*, 8(5):653–679, 2000.

[ANvB98] H. Andreka, I. Nemeti, and J. van Benthem. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.

[AS03] N. Alechina and D. Shkatov. On decidability of intuitionistic modal logics. In *Third Workshop on Methods for Modalities, Nancy*, 2003.

[Bal98] M. Baldoni. *Normal Multimodal Logics: Automated Deduction and Logic Programming*. PhD thesis, Università degli Studi di Torino, 1998.

[BGM98] M. Baldoni, L. Giordano, and A. Martelli. A tableau calculus for multimodal logics and some (un)decidability results. In H. de Swart, editor, *TABLEAUX-8*, volume 1397 of *Lecture Notes in Artificial Intelligence*, pages 44–59. Springer, 1998.

[BM02] P. Blackburn and M. Marx. Remarks in Gregory's "actually" operator. *Journal of Philosophical Logic*, 31(1):281–288, 2002.

[BT01] F. Baader and S. Tobies. The inverse method implements the automata approach for modal satisfiability. In A. Leitsch, R. Goré, and T. Nipkow, editors, *IJCAR'01*, volume 2083 of *Lecture Notes in Artificial Intelligence*, pages 92–106. Springer, 2001.

[Cau96] D. Caucal. On infinite transition graphs having a decidable monadic theory. In *ICALP'96*, volume 1099 of *Lecture Notes in Computer Science*, pages 194–205. Springer, Berlin, 1996.

[Cau03] D. Caucal. On infinite transition graphs having a decidable monadic theory. *Theoretical Computer Science*, 290:79–115, 2003.

[CdCGH97] M. Castilho, L. Fariñas del Cerro, O. Gasquet, and A. Herzig. Modal tableaux with propagation rules and structural rules. *Fundamenta Informaticae*, 32(3–4):281–297, 1997.

[Cou94] B. Courcelle. Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science*, 126:53–75, 1994.

[CS94] A. Chagrov and V. Shehtman. Algorithmic aspects of propositional tense logics. In L. Pacholski and J. Tiuryn, editors, *CSL-8, Kazimierz, Poland*, volume 933 of *Lecture Notes in Computer Science*, pages 442–455. Springer, Berlin, 1994.

[CZ97] A. Chagrov and M. Zakharyaschev. *Modal Logic*. Clarendon Press, Oxford, 1997.

[dCG02] L. Fariñas del Cerro and O. Gasquet. A general framework for pattern-driven modal tableaux. *Logic Journal of the IGPL*, 10(1):51–83, 2002.

[dCP88] L. Fariñas del Cerro and M. Penttonen. Grammar logics. *Logique et Analyse*, 121–122:123–134, 1988.

[DdN03] S. Demri and H. de Nivelle. Deciding regular grammar logics with converse through first-order logic. arXiv:cs.LO/0306117, June 2003.

[Dem00] S. Demri. The nondeterministic information logic NIL is PSPACE-complete. *Fundamenta Informaticae*, 42:211–234, 2000.

[Dem01] S. Demri. The complexity of regularity in grammar logics and related modal logics. *Journal of Logic and Computation*, 11(6):933–960, 2001.

[Dem02] S. Demri. Modal logics with weak forms of recursion: PSPACE specimens. In M. de Rijke, H. Wansing, F. Wolter, and M. Zakharyaschev, editors, *Advances in Modal Logics, selected papers from 3rd Workshop on Advances in Modal Logics (AIML'2000), Leipzig, Germany, Oct. 2000*, pages 113–138. World Scientific, 2002.

[DG00] S. Demri and R. Goré. An $O((n.log\ n)^3)$-time transformation from Grz into decidable fragments of classical first-order logic. In R. Caferra and

G. Salzer, editors, *Automated Deduction in Classical and Non Classical Logics*, volume 1761 of *Lecture Notes in Artificial Intelligence*, pages 153–167. Springer, 2000.

[DM00] F. Donini and F. Massacci. EXPTIME tableaux for ALC. *Artificial Intelligence*, 124(1):87–138, 2000.

[DMP95] G. D'Agostino, A. Montanari, and A. Policriti. A set-theoretical translation method for polymodal logics. *Journal of Automated Reasoning*, 15:317–337, 1995.

[dN98] H. de Nivelle. A resolution decision procedure for the guarded fragment. In C. Kirchner and H. Kirchner, editors, *CADE-15, Lindau, Germany*, volume 1421 of *Lecture Notes in Artificial Intelligence*, pages 191–204. Springer, 1998. Available via `http://www.mpi-sb.mpg.de/~nivelle/` on WWW.

[dN99] H. de Nivelle. Translation of S4 and K5 into GF and 2VAR, 1999. Manuscript, available from `http://www.mpi-sb.mpg.de/~nivelle`.

[dN01] H. de Nivelle. Translation of S4 and K5 into GF and 2VAR, April 2001. Slides available from `http://www.mpi-sb.mpg.de/~nivelle/` on WWW.

[dNdR03] H. de Nivelle and M. de Rijke. Deciding the guarded fragments with resolution. *Journal of Symbolic Computation*, 35(1):21–58, 2003.

[dNPH01] H. de Nivelle and I. Pratt-Hartmann. A resolution-based decision procedure for the two-variable fragment with equality. In R. Goré, A. Leitsch, and T. Nipkow, editors, *IJCAR'01*, volume 2083 of *Lecture Notes in Computer Science*, pages 211–225. Springer, 2001.

[dNSH00] H. de Nivelle, R. Schmidt, and U. Hustadt. Resolution-based methods for modal logics. *Logic Journal of the IGPL*, 8(3):265–292, 2000.

[Dyc92] R. Dyckhoff. Contraction-free sequent calculi for intuitionistic logic. *The Journal of Symbolic Logic*, 57:795–807, 1992.

[FHMV95] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. The MIT Press, 1995.

[Fin75] K. Fine. Some connections between elementary and modal logic. In S. Kanger, editor, *3rd Scandinavian Logic Symposium*, pages 15–31. North Holland, 1975.

[FL79] M. Fischer and R. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18:194–211, 1979.

[Gab75] D. Gabbay. Decidability results in non-classical logics. *Annals of Mathematical Logic*, 8:237–295, 1975.

[Gab81] D. Gabbay. Expressive functional completeness in tense logic. In U. Mönnich, editor, *Aspects of Philosophical Logic*, pages 91–117. Reidel, 1981.

[GdN99] H. Ganzinger and H. de Nivelle. A superposition decision procedure for the guarded fragment with equality. In *LICS'99*, pages 295–305, 1999.

[GKV97] E. Grädel, Ph. Kolaitis, and M. Vardi. On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic*, 3(1):53–69, 1997.

[GMV99] H. Ganzinger, C. Meyer, and M. Veanes. The two-variable guarded fragment with transitive relations (extended abstract). In *LICS'99*, pages 24–34. IEEE Computer Society Press, 1999.

[Gor99] R. Goré. Tableaux methods for modal and temporal logics. In M. d'Agostino, D. Gabbay, R. Hähnle, and J. Posegga, editors, *Handbook of Tableaux Methods*, pages 297–396. Kluwer, 1999.

[GP92] V. Goranko and S. Passy. Using the universal modality: gains and questions. *Journal of Logic and Computation*, 2(1):5–30, 1992.

[Grä99a] E. Grädel. Decision procedures for guarded logics. In H. Ganzinger, editor, *CADE'99*, volume 1632 of *Lecture Notes in Artificial Intelligence*, pages 31–51. Springer, 1999.

[Grä99b] E. Grädel. On the restraining power of guards. *The Journal of Symbolic Logic*, 64(4):1719–1742, 1999.

[Gre01] D. Gregory. Completeness and decidability results for some propositional modal logics containing "actually" operators. *Journal of Philosophical Logic*, 30(1):57–78, 2001.

[GW99] E. Grädel and I. Walukiewicz. Guarded fixed point logic. In *LICS'99*, pages 45–54, 1999.

[Her89] A. Herzig. *Raisonnement automatique en logique modale et algorithmes d'unification*. PhD thesis, Université Paul Sabatier, Toulouse, 1989.

[HKT00] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, 2000.

[Hla02] J. Hladik. Implementation and optimisation of a tableau algorithm for the guarded fragment. In U. Egly and C. Fermüller, editors, *Automated Reasoning with Analytic Tableaux and Related Methods*, volume 2381 of *Lecture Notes in Computer Science*, pages 145–159. Springer Verlag, 2002.

[HS99] I. Horrocks and U. Sattler. A description logic with transitive and inverse roles and role hierarchies. *Journal of Logic and Computation*, 9(3):385–410, 1999.

[HS00] U. Hustadt and R. Schmidt. MSPASS: Modal reasoning by translation and first-order resolution. In R. Dyckhoff, editor, *Automated Reasoning with Analytic Tableaux and Related Methods, St Andrews, Scotland, UK*, volume 1847 of *Lecture Notes in Artificial Intelligence*, pages 67–71. Springer, Berlin, 2000.

[HS03a] I. Horrocks and U. Sattler. Decidability of SHIQ with complex role inclusion axioms. In *IJCAI'03*, 2003. to appear.

[HS03b] U. Hustadt and R. Schmidt. A principle for incorporating axioms into the first-order translation of modal formulae. In *CADE'03*, volume 2741 of *Lecture Notes in Artificial Intelligence*, pages 412–426. Springer, Berlin, 2003.

[HSZ96] A. Heuerding, M. Seyfried, and H. Zimmermann. Efficient loop-check for backward proof search in some non-classical propositional logic. In P. Miglioli, U. Moscato, D. Mundici, and M. Ornaghi, editors, *Theorem Proving with Analytic Tableaux and Related Methods, 5th International Workshop (TABLEAUX'96)*, volume 1071 of *Lecture Notes in Computer Science*, pages 210–225. Springer, Berlin, 1996.

[HT02] C. Hirsch and S. Tobies. A tableau algorithm for the clique guarded fragment. In M. de Rijke, H. Wansing, F. Wolter, and M. Zakharyaschev, editors, *Advances in Modal Logics, selected papers from 3rd Workshop on Advances in Modal Logics (AIML'2000), Leipzig, Germany, Oct. 2000*, pages 257–278. World Scientific, 2002.

[HU79] J. Hopcroft and J. Ullman. *Introduction to automata theory, languages, and computation*. Addison-Wesley, 1979.

[Kie03] E. Kieronski. The two-variable guarded fragment with transitive guards is 2EXPTIME-hard. In A. Gordon, editor, *6th Int. Conf. on Foundations of Software Science and Computational Structures (FOSSACS'03), Warsaw, Poland*, volume 2620 of *Lecture Notes in Computer Science*, pages 299–312. Springer, Berlin, 2003.

[KK97] D. Korn and C. Kreitz. Deciding intuitionistic propositional logic via translation into classical logic. In W. McCune, editor, *14th International Conference on Automated Deduction*, volume 1249 of *Lecture Notes in Artificial Intelligence*, pages 131–145. Springer Verlag, 1997.

[Lad77] R. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal of Computing*, 6(3):467–480, September 1977.

[Mas97] F. Massacci. Tableaux methods for access control in distributed systems. In D. Galmiche, editor, *TABLEAUX'97*, volume 1227 of *Lecture Notes in Artificial Intelligence*, pages 246–260. Springer, Berlin, 1997.

[Mas00] F. Massacci. Single steps tableaux for modal logics. *Journal of Automated Reasoning*, 24(3):319–364, 2000.

[Min90] G. Mints. Gentzen-type systems and resolution rules. In G. Mints and P. Martin-Löf, editors, *COLOG-88*, volume 417 of *Lecture Notes in Mathematics*, pages 198–231. Springer, 1990.

[MMS99] M. Marx, Sz. Mikulas, and S. Schlobach. Tableau calculus for local cubic modal logic and its implementation. *Logic Journal of the IGPL*, 7(6):755–778, 1999.

[Moo77] R. Moore. Reasoning about knowledge and action. In *IJCAI-5*, pages 223–227, 1977.

[Mor76] Ch. Morgan. Methods for automated theorem proving in non classical logics. *IEEE Transactions on Computers*, 25(8):852–862, 1976.

[MS97] A. Mateescu and A. Salomaa. Formal languages: an introduction and a synopsis. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages - Volume 1: Word, Language and Grammar*, pages 1–40. Springer, 1997.

[Non96] A. Nonnengart. Resolution-based calculi for modal and temporal logics. In M. McRobbie and J. Slaney, editors, *13th Conference on Automated Deduction*, volume 1104 of *Lecture Notes in Artificial Intelligence*, pages 599–612. Springer, Berlin, 1996.

[Ohl93] H.J. Ohlbach. Translation methods for non-classical logics: an overview. *Bulletin of the Interest Group in Propositional and Predicate Logics*, 1(1):69–90, 1993.

[Ohl98] H.J. Ohlbach. Combining Hilbert style and semantic reasoning in a resolution framework. In C. Kirchner and H. Kirchner, editors, *CADE-15, Lindau, Germany*, volume 1421 of *Lecture Notes in Artificial Intelligence*, pages 205–219. Springer, Berlin, 1998.

[ONdRG01] H.J. Ohlbach, 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*, pages 1403–1486. Elsevier Science Publishers B.V., 2001.

[Orło88] E. Orłowska. Relational interpretation of modal logics. In H. Andréka, D. Monk, and I. Németi, editors, *Algebraic logic. Colloquia Mathematica Societatis Janos Bolyai 54*, pages 443–471, Amsterdam, 1988. North Holland.

[Pap94] Ch. Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing Company, 1994.

[Per90] D. Perrin. Finite automata. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B, Formal models and semantics*, pages 1–57. Elsevier, 1990.

[Pin94] J.E. Pin. Logic on words. *Bulletin of the European Association of Theoretical Computer Science*, 54:145–165, 1994.

[RS94] G. Rozenberg and A. Salomaa. *Cornerstones of Undecidability.* International Series in Computer Science. Prentice Hall, 1994.

[Sah75] H. Sahlqvist. Completeness and correspondence in the first and second order semantics for modal logics. In S. Kanger, editor, *3rd Scandinavian Logic Symposium, Uppsala, Sweden, 1973*, pages 110–143. North Holland, 1975.

[Sat96] U. Sattler. A concept language extended with different kinds of transitive roles. In *20. Deutsche Jahrestagung für Künstliche Intelligenz.* LNM 1137, Springer, 1996.

[Spa93] E. Spaan. *Complexity of Modal Logics.* PhD thesis, ILLC, Amsterdam University, 1993.

[ST01] W. Szwast and L. Tendera. On the decision problem for the guarded fragment with transitivity. In *LICS'01*, pages 147–156, 2001.

[Str94] H. Straubing. *Finite Automata, Formal Logic, and Circuit Complexity.* Progress in Theoretical Computer Science. Birkhäuser, 1994.

[Tam] T. Tammet. The gandalf family of automated provers. `http://www.cs.chalmers.se/~tammet/gandalf/` .

[TS96] A.S. Troelstra and H. Schwichtenberg. *Basic Proof Theory.* Cambridge University Press, 1996.

[Tuo90] H. Tuominen. Dynamic logic as a uniform framework for theorem proving in intensional logic. In M. Stickel, editor, *10th Conference on Automated Deduction*, volume 449 of *Lecture Notes in Computer Science*, pages 514–527. Springer, 1990.

[Vak87] D. Vakarelov. Abstract characterization of some knowledge representation systems and the logic NIL of nondeterministic information. In Ph. Jorrand and V. Sgurev, editors, *Artificial Intelligence: Methodology, Systems, Applications*, pages 255–260. North-Holland, Amsterdam, 1987.

[vB76] J. van Benthem. *Correspondence Theory.* PhD thesis, Mathematical Institute, University of Amsterdam, 1976.

[vB84] J. van Benthem. Correspondence Theory. In D. Gabbay and F. Günthner, editors, *Handbook of Philosophical Logic, Volume II*, pages 167–247. Reidel, Dordrecht, 1984.

[Ver03] K. Verma. *Automates d'arbres bidirectionnels modulo théories équationnelles.* PhD thesis, ENS de Cachan, Verma 2003.

[WZ97] F. Wolter and M. Zakharyashev. On the relation between intuitionistic and classical modal logics. *Algebra and Logic*, 36:121–155, 1997.