Deciding regular grammar logics with converse through first-order logic

Stéphane Demri
LSV/CNRS UMR 8643 & INRIA Futurs projet SECSI & ENS Cachan
61, av. Pdt. Wilson, 94235 Cachan Cedex, France
email: demri@lsv.ens-cachan.fr

Hans De Nivelle
Max Planck Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken, Germany
email: nivelle@mpi-sb.mpg.de

filedate

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. The translation is theoretically interesting because it translates modal logics with certain frame conditions into first-order logic, without explicitly expressing the frame conditions. It is practically relevant because it makes it possible to use a decision procedure for the guarded fragment in order to decide regular grammar logics with converse. The class of regular grammar logics includes numerous logics from various application domains.

A consequence of the translation is that the general satisfiability problem for every regular grammar logics with converse is in EXPTIME. This extends a previous result of the first author for grammar logics without converse. Other logics that can be translated into GF^2 include 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, 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 develop 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 would not work in general. Such optimizations often lead to terminating algorithms. In addition, implementation of a single modal logic is usually easier than implementing full 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 only a few main ones.

Translation of modal logics into first-order logic, with the explicit goal to mechanize such logics is an approach that has been introduced in (Morgan, 1976). Morgan distinguishes two types of translations: the *semantical translation*, which is nowadays known as the *relational translation* (see e.g., (Fine, 1975; van Benthem, 1976; Moore, 1977)) 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 in general, although the work in (Hustadt and Schmidt, 2003) has found a way to avoid this problem for many standard modal logics. 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 (Grädel et al., 1997) and the guarded fragment (Andreka et al., 1998). 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 (Ohlbach et al., 2001), where more references are provided, for instance about the functional translation (see e.g., (Herzig, 1989; Ohlbach, 1993; Nonnengart, 1996)), see also in (Orłowska, 1988; D'Agostino et al., 1995) for other types of translations.

Guarded fragments. Both the guarded fragment, introduced in (Andreka et al., 1998) (see also (de Nivelle, 1998; Grädel, 1999b; Ganzinger and de Nivelle, 1999; de Nivelle et al., 2000; de Nivelle and de Rijke, 2003)) and FO^2, the fragment of classical logic with two variables

(Gabbay, 1981; Grädel et al., 1997; de Nivelle and Pratt-Hartmann, 2001), have been used for the purpose of 'hosting' translations of modal formulas. The authors of (Andreka et al., 1998) 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 (Andreka et al., 1998), the guarded fragment GF has an EXPTIME-complete satisfiability problem when the maximal arity of the predicate symbols is fixed in advance (Grädel, 1999b). Hence its worst-case complexity is identical to some simple extensions of modal logic K, as for example the modal logic K augmented with the universal modality (Spaan, 1993). Moreover, mechanization of the guarded fragment is possible thanks to the design of efficient resolution-based decision procedures (de Nivelle, 1998; Ganzinger and de Nivelle, 1999). In (Hladik, 2002), a tableau procedure for the guarded fragment with equality based on (Hirsch and Tobies, 2002) is implemented and tested; see also a prover for FO^2 described in (Marx et al., 1999).

However, there are some simple modal logics with the satisfiability problem in PSPACE ((Ladner, 1977)) 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 reflexivity and transitivity. Many other examples will be given throughout the paper. Adding transitivity axioms to a GF-formula causes undecidability (see (Grädel, 1999a)).

Because of the apparent insufficiency of GF to capture basic modal logics, various extensions of GF have been proposed and studied. In (Ganzinger et al., 1999), 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 very relevant to deal with logics, say in EXPTIME.

In (Szwast and Tendera, 2001), the complexity bound for GF^2 with transitive guards is improved to 2EXPTIME and it was shown 2EXPTIME-hard in (Kieronski, 2003). 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 (Grädel and Walukiewicz, 1999), see also (Grädel, 1999a). 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 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 than the translation into the rich logic μGF, and that does more justice to the low complexities of simple modal logics is the approach taken in (de Nivelle, 1999; de Nivelle, 2001). There, an almost structure-preserving translation from the modal logics S4, S5 and K5 into GF^2 was given. The subformulae of a modal formula are translated in the standard way, except for subformulae that are $\Box$-formulae. The translation of $\Box$-formulae $\Box\psi$ depends on the frame condition and encodes the propagation of single-steps constraints (as done in (Massacci, 2000)) so that ψ holds true in successor states. In (de Nivelle, 1999; de Nivelle, 2001), 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 2.2. 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 subformulae of the form $[a]\phi$ does the translation differ from the usual relational translation. On these subformulae, the translation 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. (Heuerding et al., 1996; Goré, 1999; Massacci, 2000; del Cerro and Gasquet, 2002; del Cerro and Gasquet, 2004; Horrocks and Sattler, 2004)): the tableau rule for subformulae of form $[a]\phi$ can be viewed as simulating an NDFA, in the same way as the almost structure-preserving translation.

In this paper, we show that the methods of (de Nivelle, 1999) 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 $[a_0]\mathrm{p} \Rightarrow [a_1]\dots[a_n]\mathrm{p}$ 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 2.2.

With our translation, we are able to translate numerous modal logics into $GF^2 = FO^2 \cap GF$, despite the fact that their frame conditions are not expressible in $FO^2 \cup GF$. These logics include the standard modal logics K4, S4, K5, K45, S5, some information logics (see e.g. (Vakarelov, 1987)), nominal tense logics (see e.g. (Areces et al., 2000)), description logics (see e.g. (Sattler, 1996; Horrocks and Sattler, 1999)), propositional intuitionistic logic (see e.g. (Chagrov and Zakharyaschev, 1997)) and bimodal logics for intuitionistic modal logics $\mathbf{IntK}_{\Box} + \Gamma$ as those considered in (Wolter and Zakharyashev, 1997). 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 (Alechina and Shkatov, 2003) leading to less sharp complexity upper bounds). 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.

We do not claim that for most source logics the existence of a transformation into GF^2 of low complexity is surprising at all. 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.

Our paper also answers a question stated in (Demri, 2001): 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 this paper can be used to mechanize efficiently such source logics with a prover for GF^2, but we show evidence that GF^2 is a most valuable decidable first-order fragment to translate modal logics into, even when their frame conditions are not expressible in GF^2.

Structure of the paper. The paper starts by introducing multimodal languages with converses, semi-Thue systems, and some formal language theory. Using this, we can define regular grammar logics in Section 2.2. The same section also contains examples of regular grammar logics, which show that there are some natural modal logics covered by our framework.

Section 2.3 starts by repeating a standard result about derivability in Hilbert-style systems. After that, we prove a new result which characterizes when a grammar rule is a consequence of a set of grammar rules. This characterization will be used in Section 4 for determining which grammars define the same logic. In addition, we prove a closure theorem, which will be used in Section 3.2.

Section 3.1 presents the translation into GF^2. In Section 3.2 it is proven correct. In Section 4, we explore the borders of the translation method, and state some conjectures concerning which classes of logics can be translated. Section 5 contains the comparison of related works with ours. Section 6 concludes the paper and states some open questions, and future directions of research.

2. Multimodal Logic with Converse

We first introduce modal languages, after that we introduce modal frames and models. In standard modal logic, one has two operators $\Box\phi$, and $\Diamond\phi$, which denote that ϕ is true in all successor states, or true in at least one successor state. In multimodal logic, different types of successor relations are distinguished, which are labelled by elements of an alphabet Σ. As usual, 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 from the elements of Σ, and we write ϵ for the empty string. We write $u_1 \cdot u_2$ for the concatenation of u_1 and u_2. 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^*.

Definition 2.1. We assume a countably infinite set PROP of propositional variables. Let Σ be an alphabet. The multimodal language ML^Σ based on Σ is defined by the following schema:

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

where p $\in$ PROP and $a \in \Sigma$ The $\Box$-formulae are the formulae of the form $[a]\psi$. We write $|\phi|$ to denote the *size* of the formula ϕ, that is the number of symbols needed to write ϕ down. A formula ϕ is in *negation normal form* (NNF) if $\neg$ occurs only in front of propositional variables. ∇

Without any loss of generality, we can make use of the NNF when we translate formulae to GF^2. The use is not essential, but it simplifies the presentation.

Definition 2.2. Let Σ be an alphabet. A Σ-*frame* is a pair $\mathcal{F} = \langle W, R\rangle$, such that W is a non-empty set, and R is a mapping from the elements of Σ to binary relations over W. We usually write R_a instead of $R(a)$. A Σ-*model* $\mathcal{M} = \langle W, R, V\rangle$ is obtained by adding a valuation function V with signature PROP $\rightarrow \mathcal{P}(W)$ to the frame. For every p $\in$ PROP, $V(\mathrm{p})$ denotes the set of worlds where p is true.
The satisfaction relation $\models$ is defined in the usual way:

- For every p $\in$ PROP, $\mathcal{M}, x \models \mathrm{p}$ iff $x \in V(\mathrm{p})$.
- For every $a \in \Sigma$, $\mathcal{M}, x \models [a]\phi$ iff for every y such that $R_a(x, y)$, $\mathcal{M}, y \models \phi$.
- For every $a \in \Sigma$, $\mathcal{M}, x \models \langle a\rangle\phi$ iff there is an y such that $R_a(x, y)$ 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 formula ϕ is said to be *satisfiable* if there exist a Σ-model $\mathcal{M} = \langle W, R, V\rangle$ and $w \in W$, such that $\mathcal{M}, w \models \phi$. ∇

In order to be able to cope with properties such as symmetry and euclideanity, one needs to be able to express *converses*. Probably the most natural way to do this, is by extending the modal language with backward modal operators $[a]^{-1}\phi$ and $\langle a\rangle^{-1}\phi$. Unfortunately, this approach does not work for us, because we want to be able to express frame conditions using languages over Σ, and in this way the backward modalities have no counterpart in Σ.

Because of this, we follow another approach and we assume that to each a in the alphabet Σ, a unique converse symbol $\overline{a}$ is associated, which is also in Σ. In this way, one can partition Σ into two parts, the *forward* part and the *backward* part.

Definition 2.3. Let Σ be an alphabet. We call a function $\bar{\cdot}$ on Σ a *converse mapping* if for every $a \in \Sigma$, we have $\overline{a} \neq a$ and $\overline{\overline{a}} = a$. ∇

It is easy to prove the following result.

Lemma 2.1. 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 every $a \in \Sigma^+$, $\overline{a} \in \Sigma^-$, **(2)** for every $a \in \Sigma^-$, $\overline{a} \in \Sigma^+$.

In fact, there exist many partitions $\Sigma = \Sigma^- \cup \Sigma^+$. When we refer to such a partition, we assume that an arbitrary one is chosen. We 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).

Definition 2.4. Let Σ be an alphabet with converse mapping $\bar{\cdot}$. The converse mapping $\bar{\cdot}$ is extended to words over Σ^* as follows:

1. $\overline{\epsilon} \stackrel{\text{def}}{=} \epsilon$,
2. if $u \in \Sigma^*$ and $a \in \Sigma$, then $\overline{u \cdot a} \stackrel{\text{def}}{=} \overline{a} \cdot \overline{u}$.

∇

In order to ensure that converses behave like converses should, we impose the following, obvious constraint on the Σ-frames:

Definition 2.5. Let Σ be an alphabet with converse mapping $\bar{\cdot}$. We call a Σ-frame a $\langle \Sigma, \bar{\cdot} \rangle$-frame if, for every $a \in \Sigma$, $R_{\overline{a}}$ equals $\{\langle y, x \rangle \mid R_a(x, y)\}$. ∇

In the rest of the paper, we adopt the following working definition for a logic.

Definition 2.6. Let Σ be an alphabet with converse mapping $\bar{\cdot}$. A *logic* $\mathcal{L}$ is pair $\langle \mathrm{ML}^\Sigma, \mathcal{C} \rangle$ such that $\mathcal{C}$ is a class of $\langle \Sigma, \bar{\cdot} \rangle$-frames. A formula $\phi \in \mathrm{ML}^\Sigma$ is $\mathcal{L}$-*satisfiable* [resp. $\mathcal{L}$-*valid*] iff there exist a $\langle \Sigma, \bar{\cdot} \rangle$-model $\mathcal{M} = \langle W, R, V \rangle$ and $w \in W$ such that $\mathcal{M}, w \models \phi$ and $\langle W, R \rangle \in \mathcal{C}$ [resp. $\neg\phi$ is not $\mathcal{L}$-satisfiable]. ∇

2.1. Defining Logics by Production Rules

We want to study validity and satisfiability in various modal logics. Modal logics are traditionally defined by subclasses of frames (see Definition 2.6), or by axioms. For example, the logic S4 can be either defined by the modal axioms $[a]\phi \rightarrow [a][a]\phi$ and $[a]\phi \rightarrow \phi$ or by the subclass of frames in which all relations R_a are reflexive and transitive. For many modal logics, the axioms stand in natural correspondence to the condition that has to be imposed on the frames.

We will use a language theoretical framework for defining frames conditions. Since the accessibility relations are labelled by letters, paths through a frame can be labelled by words. Using this, certain conditions on the accessibility relation can be represented by production rules. For example, the transitivity rule $\forall xyz\ \mathbf{R}_a(x,y) \wedge \mathbf{R}_a(y,z) \Rightarrow \mathbf{R}_a(x,z)$ can be represented by the rule $a \rightarrow a \cdot a$. Similarly, the implication $\forall xyz\ \mathbf{R}_a(x,y) \wedge \mathbf{R}_b(y,z) \Rightarrow \mathbf{R}_c(x,z)$ can be represented by the rule $c \rightarrow a \cdot b$. In order to formally define how a frame satisfies a production rule, we need the following definition.

Definition 2.7. Let Σ be an alphabet and $\mathcal{F} = \langle W, R \rangle$ be a Σ-frame. The interpretations R_a are recursively extended to words $u \in \Sigma^*$ as follows:

- $R_\epsilon \stackrel{\text{def}}{=} \{\langle x, x \rangle \mid x \in W\}$,
- for all $u \in \Sigma^*$ and $a \in \Sigma$,

$$R_{u \cdot a} \stackrel{\text{def}}{=} \{\langle x, y \rangle \mid \exists z \in W,\ R_u(x,z) \text{ and } R_a(z,y)\}.$$

∇

Definition 2.8. Let Σ be an alphabet with converse mapping $\bar{\cdot}$. A *semi-Thue system* S over Σ is a set of *production rules* of form $u \rightarrow v$ with $u, v \in \Sigma^*$. ∇

A semi-Thue system is similar to a grammar, but it has no start symbol, and there is no distinction between terminal and non-terminal symbols. Using semi-Thue systems, we can define more precisely how semi-Thue systems encode conditions on Σ-frames.

Definition 2.9. Let $u \rightarrow v$ be a production rule over some alphabet Σ with converse mapping $\bar{\cdot}$. We say that a $\langle \Sigma, \bar{\cdot} \rangle$-frame $\mathcal{F} = \langle W, R \rangle$ *satisfies* $u \rightarrow v$ if the inclusion $R_v \subseteq R_u$ holds. A $\langle \Sigma, \bar{\cdot} \rangle$-frame $\mathcal{F}$ satisfies a *semi-Thue system* S if it satisfies each of its rules. We also say that the production rule (or the semi-Thue system) *is true* in $\mathcal{F}$. ∇

Observe that in Definition 2.9, u and v are swapped when passing from the production rule to the relation inclusion.

Definition 2.10. A formula ϕ is said to be S-*satisfiable* iff there is a $\langle \Sigma, \bar{\cdot} \rangle$-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 in all $\langle \Sigma, \bar{\cdot} \rangle$-models $\mathcal{M} = \langle W, R, V \rangle$ that satisfy S, for every $x \in W$, we have $\mathcal{M}, x \models \phi$. ∇

Transitivity on the relation R_a can be expressed by the semi-Thue system $\{a \rightarrow a \cdot a\}$. Similarly, reflexivity can be expressed by the system $\{a \rightarrow \epsilon\}$.

Semi-Thue systems are obviously 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.

Definition 2.11. A multimodal logic $\langle \mathrm{ML}^{\Sigma}, \mathcal{C} \rangle$ with Σ an alphabet with converse mapping $\bar{\cdot}$ is said to be a *grammar logic with converse* if there is a finite semi-Thue system S over Σ such that $\mathcal{C}$ is the set of $\langle \Sigma, \bar{\cdot} \rangle$-frames satisfying S. ∇

We will mostly omit the suffix 'with converse', because we study only grammar logics with converse in this paper. One could give Definition 2.9 without converse, but this will bring no increased generality, because a logic without converse can always be viewed as a sublogic of a logic with converse. Consider a grammar logic $\mathcal{L}$ without converse defined by a semi-Thue system S over alphabet Σ. One can put $\Sigma' = \Sigma \cup \{\overline{a} \mid a \in \Sigma\}$, and for each $a \in \Sigma$, put $\overline{\overline{a}} = a$. Each Σ-frame can now be obviously extended to a $\langle \Sigma', \bar{\cdot} \rangle$-frame.

The modal logic S4 can be defined by the context-free semi-Thue system $\{a \rightarrow \epsilon,\ a \rightarrow aa\}$. The modal logic B can be defined by $\{a \rightarrow \overline{a}\}$. The following correspondence result is standard, see for example (van Benthem, 1984).

Theorem 2.2. Let Σ be an alphabet with converse mapping $\bar{\cdot}$, and S be a semi-Thue system over Σ. The following statements are equivalent:

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

Originally, grammar logics were defined with formal grammars in (del Cerro and Penttonen, 1988) (as in (Baldoni, 1998; Demri, 2001; Demri, 2002)), and they form a subclass of Sahlqvist modal logics (Sahlqvist, 1975) with frame conditions expressible in Π_1 when S is context-free (see e.g. Definition 2.13). 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 (Chagrov and Shehtman, 1994), which is more appropriate.

2.2. Regular grammar logics with converse

In order to define the class of regular grammar logics with converse, we need to recall a few notions from formal language theory.

Definition 2.12. Let S be a semi-Thue system. The one-step derivation relation $\Rightarrow_S$ based on S is defined as follows: $u \Rightarrow_S v$ iff there exist $u_1, u_2 \in \Sigma^*$, and $u' \rightarrow v' \in S$, such that $u = u_1 \cdot u' \cdot u_2$, and $v = u_1 \cdot v' \cdot u_2$. The full derivation relation $\Rightarrow_S^*$ is defined as the reflexive and transitive closure of $\Rightarrow_S$. For every $u \in \Sigma^*$, we write $L_S(u)$ to denote the language $\{v \in \Sigma^* \mid u \Rightarrow_S^* v\}$. ∇

Definition 2.13. The system S is *context-free* if all production rules are of the form $a \rightarrow v$ with $a \in \Sigma$ and $v \in \Sigma^*$. A context-free semi-Thue system S, based on Σ, is called *regular* if for every $a \in \Sigma$, the language $L_S(a)$ is regular. In that case, we assume there is a function that associates to each $a \in \Sigma$, an automaton $\mathcal{A}_a$ that accepts $L_S(a)$.

The *converse closure* $\overline{S}$ of a system S over an alphabet Σ with converse mapping $\bar{\cdot}$ is the semi-Thue system $\{\overline{u} \rightarrow \overline{v} : u \rightarrow v \in S\}$. A system S is said to be *closed under converse* if $S = \overline{S}$. ∇

Regular languages can be recognized by finite-state automata. We recall the definition of finite-state automaton, so that we can refer to it later.

Definition 2.14. A *non-deterministic finite automaton* (NDFA) $\mathcal{A}$ is defined by a tuple (Q, s, F, δ). Here Q is the finite, non-empty set of states. $s \in Q$ is the *initial* state. $F \subseteq Q$ is the set of *accepting* states. $\delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times Q$ is the *transition relation*.
The *extension* δ^* of δ is recursively defined as follows:

– For every state $q \in Q$, $\langle q, \epsilon, q \rangle \in \delta^*$.

– For all strings u and states q, q', q'', if $\langle q, u, q' \rangle \in \delta^*$ and $\langle q', \epsilon, q'' \rangle \in \delta$, then $\langle q, u, q'' \rangle \in \delta^*$.

– For all strings u, letters a, and states q, q', q'', if $\langle q, u, q' \rangle \in \delta^*$ and $\langle q', a, q'' \rangle \in \delta$, then $\langle q, u \cdot a, q'' \rangle \in \delta^*$.

$\mathcal{A}$ *accepts* a word u if there is a state $q \in F$, such that $\langle s, u, q \rangle \in \delta^*$. We write $\mathrm{L}(\mathcal{A})$ to denote the set of finite words accepted by $\mathcal{A}$. A language L is *regular* if there exists an NDFA $\mathcal{A}$, such that $\mathrm{L} = \{u \mid \mathcal{A} \text{ accepts } u\}$. ∇

For more details on NDFA's, we refer to (Hopcroft and Ullman, 1979). In Definition 2.13, we do not specify which automaton $\mathcal{A}_a$ is associated to a.

Definition 2.15. A multimodal logic $\langle \mathrm{ML}^{\Sigma}, \mathcal{C} \rangle$ with Σ an alphabet with converse mapping $\bar{\cdot}$ is said to be a *regular grammar logic with converse* if there is a finite regular semi-Thue system S over Σ closed under converse such that $\mathcal{C}$ is the set of $\langle \Sigma, \bar{\cdot} \rangle$-frames satisfying S. ∇

Example 2.1. 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 I, we specify the semi-Thue systems through regular expressions for the languages $\mathrm{L}_{\mathrm{S}}(a)$.
$\triangleleft$

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:

– description logics (with role hierarchy, transitive roles), see e.g. (Horrocks and Sattler, 1999).
– knowledge logics, see e.g. $\mathrm{S5}_m(\mathrm{DE})$ in (Fagin et al., 1995).
– bimodal logics for intuitionistic modal logics of the form $\mathbf{IntK}_{\Box} + \Gamma$ (Wolter and Zakharyashev, 1997). 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 $a \notin \Sigma'$ or $\overline{a} \notin \Sigma'$. Then, the semi-Thue system $\mathrm{S} \cup \{b \rightarrow bab, \overline{b} \rightarrow \overline{b}\overline{a}\overline{b} \mid a \in \Sigma'\}$ over $\Sigma \cup \{b, \overline{b}\}$ is also regular, assuming $b, \overline{b} \notin \Sigma$. By taking advantage of (Ganzinger et al., 1999), in (Alechina and Shkatov, 2003) decidability of intuitionistic modal logics is also shown in a uniform manner.
– fragments of logics designed for the access control in distributed systems (Abadi et al., 1993; Massacci, 1997).

Table I. Regular languages for standard modal logics

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.

- extensions with the universal modality (Goranko and Passy, 1992). 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 (Goranko and Passy, 1992) (add a new letter U such that $[U]$ is an S5 modality and $[U]\mathrm{p} \Rightarrow [a]\mathrm{p}$ is a modal axiom for every letter a). Hence, satisfiability, global satisfiability and logical consequence can be handled uniformly with no increase of worst-case complexity.
- information logics, see e.g. (Vakarelov, 1987). For instance, the Nondeterministic Information Logic NIL introduced in (Vakarelov, 1987; Demri, 2000) can be shown to be a fragment of a regular grammar logic with converse with $\Sigma^+ = \{\mathrm{fin}, \mathrm{sim}\}$ and the production rules below (augmented with the converse closure):
  - $\mathrm{fin} \rightarrow \mathrm{fin} \cdot \mathrm{fin}$, $\mathrm{fin} \rightarrow \epsilon$,
  - $\mathrm{sim} \rightarrow \overline{\mathrm{sim}}$, $\mathrm{sim} \rightarrow \epsilon$,
  - $\mathrm{sim} \rightarrow \overline{\mathrm{fin}} \cdot \mathrm{sim} \cdot \mathrm{fin}$.

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

Assuming that $\mathcal{L} = \langle \mathrm{ML}^{\Sigma}, \mathcal{C}_{\mathrm{S}} \rangle$ is a grammar logic with converse, checking whether $\mathcal{L}$ is regular is not an easy task. 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. (Mateescu and Salomaa, 1997, page 31)). However, if S is closed under converse and all the production rules in S are either right-linear or left-linear, then $\mathcal{L}$ is regular. We recall that S is right-linear if there is a partition $\{V, T\}$ of Σ such that the production rules in S are in $V \rightarrow T^* \cdot (V \cup \{\epsilon\})$. Similarly, S is left-linear if there is a partition $\{V, T\}$ of Σ such that the production rules are in $V \rightarrow (V \cup \{\epsilon\}) \cdot T^*$. Also, regularity is guaranteed if one can show that for every $a \in \Sigma$, the language $\mathrm{L}_{\mathrm{S}}(a)$ is regular. All the modal logics cited above fall in this category. However, there is a remaining possible situation which is quite interesting. It might be the case that for some $a \in \Sigma$, the language $\mathrm{L}_{\mathrm{S}}(a)$ is not regular but that there is another semi-Thue system S′ s.t. $\langle \mathrm{ML}^{\Sigma}, \mathcal{C}_{\mathrm{S}'} \rangle$ defines the same logic as $\langle \mathrm{ML}^{\Sigma}, \mathcal{C}_{\mathrm{S}} \rangle$ and all $\mathrm{L}_{\mathrm{S}}(a)$ are regular. This topic will be discussed in Section 4. In full generality, one should not expect to find a way to compute effectively a regular system S′ but this shows the large scope of our translation.

2.3. Characterizations of Consequences

In this section we study the following two questions. Let Σ be an alphabet and S be a finite context-free semi-Thue system over Σ.

1. Which formulas are true in all $\langle \Sigma, \bar{\cdot} \rangle$-frames that satisfy S?
2. Which production rules $u \rightarrow v$ are true in all $\langle \Sigma, \bar{\cdot} \rangle$-frames that satisfy S?

The first question can be answered in the standard way by Hilbert-style deduction systems.

Definition 2.16. Let Σ be an alphabet with converse mapping $\bar{\cdot}$. Let S be semi-Thue system over Σ. The set of *derivable formulas* $\mathcal{H}$ is recursively defined as follows:

1. If ϕ is a propositional tautology, then $\phi \in \mathcal{H}$.
2. For all formulae ϕ and letters $a \in \Sigma$, $\quad [a]\phi \leftrightarrow \neg\langle a \rangle \neg\phi \in \mathcal{H}$.
3. If $\phi \in \mathcal{H}$, and $a \in \Sigma$, then also $[a]\phi \in \mathcal{H}$.
4. For all formulae ϕ, ψ and letters $a \in \Sigma$, $[a]\phi \wedge [a](\phi \Rightarrow \psi) \Rightarrow [a]\psi \in \mathcal{H}$,
5. For all formulas ϕ and letters $a \in \Sigma$, $\langle a \rangle [\overline{a}]\phi \rightarrow \phi \in \mathcal{H}$.
6. For every rule $u \rightarrow v \in \mathrm{S}$, for every formula ϕ,

$$[u]\phi \rightarrow [v]\phi \in \mathcal{H} \text{ and } [\overline{u}]\phi \rightarrow [\overline{v}]\phi \in \mathcal{H}.$$

∇

The following follows from (Sahlqvist, 1975).

Theorem 2.3. Let Σ be an alphabet with converse mapping $\bar{\cdot}$. Let S be a semi-Thue system over Σ. Then $\phi \in \mathcal{H}$ if and only if ϕ is true in all $\langle \Sigma, \bar{\cdot} \rangle$-frames satisfying S.

Regarding the second question, it is quite easy to see that $u \Rightarrow_{\mathrm{S}}^{*} v$ implies that $u \rightarrow v$ is true in every $\langle \Sigma, \bar{\cdot} \rangle$-frame that satisfies S. The converse does not always hold as shown in Example 2.2 below. When there is no converse mapping, it is indeed the case that all rules of form $u \rightarrow v$, which are true in all $\langle \Sigma, \bar{\cdot} \rangle$-frames, are derivable as $u \Rightarrow_{\mathrm{S}'}^{*} v$. This follows from (Chagrov and Shehtman, 1994, Theorem 3) (see also the tableaux-based proof in (Baldoni, 1998)) and it is related to the fact that every ordered monoid is embeddable into some ordered monoid of binary relations (see more details in (Chagrov and Shehtman, 1994)).

Example 2.2. Consider the semi-Thue system $\mathrm{S} = \{a \rightarrow \overline{a},\ b \rightarrow a^3\ \}$. In this system, $b \not\Rightarrow_{\mathrm{S}}^{*} a$. However, the rule $a \rightarrow \overline{a}$ expresses symmetry, which means that in a $\langle \Sigma, \bar{\cdot} \rangle$-frame satisfying S, whenever $\langle x, y \rangle \in R_a$, then also $\langle x, y \rangle \in R_{a^3}$. Therefore, $R_a \subseteq R_b$. One may think that the situation improves when the converse rules are added to S, but if one puts $\mathrm{S}' = \{a \rightarrow \overline{a},\ \ \overline{a} \rightarrow a,\ \ b \rightarrow a^3,\ \ \overline{b} \rightarrow \overline{a}^3\}$, then still $b \not\Rightarrow_{\mathrm{S}'}^{*} a$.

The production rule $b \Rightarrow_{S}^{*} a$ can be derived as follows: whenever $\langle x, y \rangle \in R_a$, then $\langle y, x \rangle \in R_{\overline{a}}$. As a consequence, $\langle x, y \rangle \in R_{a\overline{a}a}$. This means that the (non context-free) production rule $a\overline{a}a \rightarrow a$ is true in every frame. By combining $b \rightarrow aaa$, $a \rightarrow \overline{a}$, and $a\overline{a}a \rightarrow a$, we can derive $b \rightarrow a$. $\lhd$

In the sequel, we provide a complete characterization of the production rules that follow from a semi-Thue system S inspired from the (non context-free) rules added in Example 2.2.

Definition 2.17. Let Σ be an alphabet with converse mapping $\bar{\cdot}$. The *expansion system* E_{Σ} of Σ is the semi-Thue system

$$\{\ u \cdot \overline{u} \cdot u \rightarrow u \mid u \in \Sigma^{*} \backslash \{\epsilon\}\ \ \}.$$

∇

Lemma 2.4. Let Σ be an alphabet with converse mapping $\bar{\cdot}$.

(I) Every production rule in E_{Σ} is true in all $\langle \Sigma, \bar{\cdot} \rangle$-frames.

(II) For all $\langle\Sigma,\bar{\cdot}\rangle$-frames $\mathcal{F}$ and production rules $u \rightarrow v$, $\mathcal{F}$ satisfies $u \rightarrow v$ iff $\mathcal{F}$ satisfies $\overline{u} \rightarrow \overline{v}$.

(III) Let S be a semi-Thue system over Σ, u, v be two strings verifying $u \Rightarrow^*_{\mathrm{S}} v$, and $\mathcal{F}$ be a $\langle\Sigma,\bar{\cdot}\rangle$-frame satisfying S. Then $\mathcal{F}$ satisfies $u \rightarrow v$.

Proof. (I), (II) and (III) are by an easy verification. For instance, (III) can be proven by induction on the i for which $u \Rightarrow^i_{\mathrm{S}} v$. □

Theorem 2.5. Let Σ be an alphabet with converse mapping $\bar{\cdot}$ and S be a context-free semi-Thue system over Σ. Then, for all strings $u, v \in \Sigma^*$, the following are equivalent:

1. In every $\langle\Sigma,\bar{\cdot}\,\rangle$-frame $\mathcal{F} = \langle W, R\rangle$ that satisfies S, we have $R_v \subseteq R_u$.
2. There exists a string $z \in \Sigma^*$ such that $u \Rightarrow^*_{\mathrm{S}\cup\overline{\mathrm{S}}} z$ and $z \Rightarrow^*_{E_\Sigma} v$.
3. $u \Rightarrow^*_{\mathrm{S}\cup\overline{\mathrm{S}}\cup E_\Sigma} v$.

It is easy to see that **(2)** implies **(3)**. It follows from Lemma 2.4 that **(3)** implies **(1)**. We will use the rest of this section to show that **(1)** implies **(2)**. In order to do this, it is convenient to use the notion of *closure*. The closure will be also used in Section 3.2. If some $\langle\Sigma,\bar{\cdot}\rangle$-frame $\mathcal{F} = \langle W, R\rangle$ does not satisfy some context-free semi-Thue system, then one can add the missing edges to R and obtain a $\langle\Sigma,\bar{\cdot}\rangle$-frame that does satisfy the semi-Thue system. For context-free semi-Thue systems, one can define a function that assigns to each $\langle\Sigma,\bar{\cdot}\rangle$-frame the smallest frame that satisfies the semi-Thue system.

Definition 2.18. We first define an inclusion relation on $\langle\Sigma,\bar{\cdot}\rangle$-frames. Let Σ be an alphabet with converse mapping $\bar{\cdot}$. Let $\mathcal{F}_1 = \langle W, R_1\rangle$ and $\mathcal{F}_2 = \langle W, R_2\rangle$ be two $\langle\Sigma,\bar{\cdot}\rangle$-frames sharing 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 the inclusion relation, we define the *closure operator* C_{S} as follows: for a context-free semi-Thue system S over alphabet Σ with converse mapping $\bar{\cdot}$, for a $\langle\Sigma,\bar{\cdot}\rangle$-frame $\mathcal{F}$, the *closure* of $\mathcal{F}$ under S is defined as the smallest $\langle\Sigma,\bar{\cdot}\rangle$-frame that satisfies S, and which has $\mathcal{F}$ as a subframe. We write C_{S} for the closure operator. ∇

The closure always exists, and is unique, due to the Knaster-Tarski fixed point theorem. It can also be proven from Theorem 2.6, which states a crucial property of C_{S}, namely that every edge added by C_{S} can be justified by $\Rightarrow^*_{\mathrm{S}\cup\overline{\mathrm{S}}}$.

When S is regular, the map C_{S} is a monadic second-order definable graph transduction in the sense of (Courcelle, 1994) and it is precisely the inverse substitution h^{-1} in the sense of (Caucal, 2003) (see also (Caucal, 1996)) when the extended substitution h is defined by $a \in \Sigma \mapsto \mathrm{L}_{\mathrm{S}}(a)$.

Theorem 2.6. Let S be a context-free semi-Thue system over alphabet Σ with converse mapping $\bar{\cdot}$. Let $\mathcal{F} = \langle W, R \rangle$ be a $\langle \Sigma, \bar{\cdot} \rangle$-frame. For every letter $a \in \Sigma$, the relations R'_a of the $\langle \Sigma, \bar{\cdot} \rangle$-frame $\mathcal{F}' = C_{\mathrm{S}}(\mathcal{F}) = \langle W, R' \rangle$ are defined as follows:

$$R'_a = \{\langle x, y \rangle \mid \exists u \in \Sigma^* \text{such that } a \Rightarrow^*_{\mathrm{S} \cup \overline{\mathrm{S}}} u \text{ and } \langle x, y \rangle \in R_u\}.$$

Then $\mathcal{F}'$ is the closure of $\mathcal{F}$.

Proof. We have to show that

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

In order to show **(1)**, we show that for every rule $a \rightarrow u$ 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 need 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 $u_1 \Rightarrow^*_{\mathrm{S} \cup \overline{\mathrm{S}}} v_1, \quad \ldots, \quad u_n \Rightarrow^*_{\mathrm{S} \cup \overline{\mathrm{S}}} v_n$, 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, $\langle x, y \rangle \in R_{v_1 \cdot \ldots \cdot v_n}$. Because $a \Rightarrow_{\mathrm{S} \cup \overline{\mathrm{S}}} u, \ \ u = (u_1, \ldots, u_n)$, and each $u_i \Rightarrow^*_{\mathrm{S} \cup \overline{\mathrm{S}}} v_i$, we also have $a \Rightarrow^*_{\mathrm{S} \cup \overline{\mathrm{S}}} v_1 \cdot \ldots \cdot v_n$. It follows that $\langle x, y \rangle \in R'_a$, from the way R'_a was constructed.
Next we show **(2)**. As a preparation, it can be shown by induction that $a \Rightarrow^*_{\mathrm{S} \cup \overline{\mathrm{S}}} u$ iff $\overline{a} \Rightarrow^*_{\mathrm{S} \cup \overline{\mathrm{S}}} \overline{u}$. We need to show that for every $a \in \Sigma$,

$$\langle x, y \rangle \in R'_a \text{ iff } \langle y, x \rangle \in R'_{\overline{a}}.$$

$$\langle x, y \rangle \in R'_a \text{ iff}$$

$$\text{there exists a word } u \in \Sigma^*, \text{ for which } a \Rightarrow^*_{\mathrm{S} \cup \overline{\mathrm{S}}} u, \text{ and } \langle x, y \rangle \in R_u \text{ iff}$$

$$\text{there exists a word } \overline{u} \in \Sigma^*, \text{ for which } \overline{a} \Rightarrow^*_{\mathrm{S} \cup \overline{\mathrm{S}}} \overline{u} \text{ and } \langle y, x \rangle \in R_{\overline{u}} \text{ iff}$$

$$\langle y, x \rangle \in R_{\overline{a}}.$$

Finally we show **(3)**. Let $\langle W, R'' \rangle$ be a $\langle \Sigma, \bar{\cdot} \rangle$-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 every $a \in \Sigma$, $\quad R'_a \subseteq R''_a$.

Assume that $\langle x, y \rangle \in R'_a$. This means that there exists an $u \in \Sigma^*$, for which $\langle x, y \rangle \in R_u$ and $a \Rightarrow^*_{\mathrm{S} \cup \overline{\mathrm{S}}} u$. From Lemma 2.4(II,III), we know that $a \rightarrow u$ is true in $\langle W, R \rangle$. Therefore, we have $\langle x, y \rangle \in R_a$. Because $R_a \subseteq R''_a$, we also have $\langle x, y \rangle \in R''_a$.
□

Actually, only Part 1 and Part 2 of Theorem 2.6 are needed in the proof of Theorem 2.5 and in Section 3.2.

We can now give the proof of Theorem 2.5 "(1) implies (2)". Let S be a context-free semi-Thue system. Let u, v be two words, such that in every $\langle \Sigma, \bar{\cdot} \rangle$-frame $\mathcal{F} = \langle W, R \rangle$ satisfying S, we have $R_v \subseteq R_u$.

Write $v = (v_1, \ldots, v_n)$ with $n \geq 0$. Let the frame $\mathcal{F}_v = \langle W, R \rangle$ be defined as follows:

- $W = \{w_1, \ldots, w_n, w_{n+1}\}$,
- If (and only if) $v_i = a$, then $\langle w_i, w_{i+1} \rangle \in R_a$, for $1 \leq i \leq n$ and $a \in \Sigma$.

Intuitively, the frame $\mathcal{F}_v$ consists of a single path, which is labelled with the word v. Let $\mathcal{F}'_v = \langle W, R' \rangle$ be obtained from $\mathcal{F}_v$ by the construction of Theorem 2.6, i.e. $\mathcal{F}'_v = C_{\mathrm{S}}(\mathcal{F}_v)$. Since by Theorem 2.6, $\mathcal{F}'_v$ is a frame satisfying S and by hypothesis $R'_v \subseteq R'_u$, we have $\langle w_1, w_{n+1} \rangle \in R'_u$.
If one writes $u = (u_1, \ldots, u_m)$, then there must exist $w'_1, \ldots, w'_{m+1} \in W$, such that

$$\langle w'_1, w'_2 \rangle \in R'_{u_1}, \quad \langle w'_2, w'_3 \rangle \in R'_{u_2}, \ldots, \langle w'_m, w'_{m+1} \rangle \in R'_{u_m},$$

with $w'_1 = w_1$ and $w'_{m+1} = w_{n+1}$.
By construction of $\mathcal{F}'_v$, there exist words $z_1, \ldots, z_m \in \Sigma^*$, such that

$$\langle w'_1, w'_2 \rangle \in R_{z_1}, \quad \langle w'_2, w'_3 \rangle \in R_{z_2}, \ldots, \langle w'_m, w'_{m+1} \rangle \in R_{z_m},$$

and

$$u_1 \Rightarrow^*_{\mathrm{S} \cup \overline{\mathrm{S}}} z_1, \quad u_2 \Rightarrow^*_{\mathrm{S} \cup \overline{\mathrm{S}}} z_2, \ldots, u_m \Rightarrow^*_{\mathrm{S} \cup \overline{\mathrm{S}}} z_m.$$

Therefore, for the word $z = z_1 \cdot z_2 \cdot \ldots \cdot z_m$, it follows that $\langle w_1, w_{n+1} \rangle \in R_z$ and $u \Rightarrow^*_{\mathrm{S} \cup \overline{\mathrm{S}}} z$. We will show that also $z \Rightarrow^*_{E_\Sigma} v$, from which then follows that $u \Rightarrow^*_{\mathrm{S} \cup \overline{\mathrm{S}} \cup E_\Sigma} v$.

Lemma 2.7. Let $\mathcal{F}_v = \langle W, R \rangle$ be the frame defined above from v. Let $z \in \Sigma^*$ be a string such that $\langle w_1, w_{n+1} \rangle \in R_z$. Then $z \Rightarrow^*_{E_\Sigma} v$.

The word z corresponds to a walk from w_1 to w_{n+1} in the frame $\mathcal{F}_v$. The frame $\mathcal{F}_v$ consists of a single path, which is labelled by the word v.

The word z is obtained by a walk on this path, which possibly changes direction a few times.

We call a maximal subpath that does not change direction *a segment*. A segment is either forward directed, or backward directed. So a segment s is of the form either

$$s = (w_i, w_{i+1}, \ldots, w_{j-1}, w_j)$$

or

$$s = (w_i, w_{i-1}, \ldots, w_{j+1}, w_j)$$

with $1 \leq i, j \leq n+1$ and $i \neq j$.
Using segments, the path can be written in the form

$$s_1 = (x_1, \ldots, y_1), \ldots, s_k = (x_k, \ldots, y_k), \tag{1}$$

where

- all the states in the segments are in $\{w_1, \ldots, w_{n+1}\}$,
- k is odd,
- for every i, $x_i \neq y_i$ and $x_{i+1} = y_i$ (assuming $i + 1 \leq k$).

Given two states $w, w' \in \{w_1, \ldots, w_{n+1}\}$, we write $w < w'$ whenever there are $1 \leq j < j' \leq n+1$ such that $w_j = w$ and $w_{j'} = w'$. Observe that if i is odd, then $x_i < y_i$. If i is even, then $x_i > y_i$. For every i, there is a unique string $v_i \in ((\Sigma^+)^* \cup (\Sigma^-)^*) \setminus \{\epsilon\}$ such that $\langle x_i, y_i \rangle \in R_{v_i}$. If i is odd, then v_i is a substring of v. If i is even, then v_i is a substring of $\overline{v}$. We call v_i *the associated string* of the segment $s_i = (x_i, \ldots, y_i)$. We have $z = v_1 \cdot \ldots \cdot v_k$.

If $k = 1$, then $z = v_1 = v$, and we are done because $z \Rightarrow^0_{E_\Sigma} v$. Otherwise, let i with $1 < i < k$ be chosen in such a way that $(x_i, \ldots, y_i)$ is a segment with minimal length. Such an i must exist, because $k \geq 3$. The segment $(x_{i-1}, \ldots, y_{i-1})$ before $(x_i, \ldots, y_i)$ cannot be strictly shorter than $(x_i, \ldots, y_i)$. Suppose that it were. Then, if $i > 2$, the position $i - 1$ would have been chosen instead of i. If $i = 2$, then $(x_2, \ldots, y_2)$ is a segment that walks backwards in the direction of w_1. The first segment $(x_1, \ldots, y_1)$ starts with $x_1 = w_1$ and must be at least as long, because otherwise $(x_2, \ldots, y_2)$ would walk back through w_1, which is not possible because $\mathcal{F}_v$ is a chain starting in w_1.

For the same reason, the segment $(x_{i+1}, \ldots, y_{i+1})$ cannot be strictly shorter than $(x_i, \ldots, y_i)$. As a consequence, the neighbours of the i-th segment s_i are of the form

$$s_{i-1} = (x_{i-1}, \ldots, y_{i-1}) = (x_{i-1}, \ldots, y_i, \ldots, x_i),$$

and

$$s_{i+1} = (x_{i+1}, \ldots, y_{i+1}) = (y_i, \ldots, x_i, \ldots, y_{i+1}).$$

Then the associated strings are of the form

$$v_{i-1} = \alpha \cdot \overline{v_i} \text{ and } v_{i+1} = \overline{v_i} \cdot \beta, \text{ for some } \alpha, \beta \in \Sigma^*.$$

The complete walk can be written as

$$s_1, \ldots, (x_{i-1}, \ldots, y_i, \ldots, x_i), (x_i, \ldots, y_i), (y_i, \ldots, x_i, \ldots, y_{i+1}), \ldots s_k. \quad (2)$$

The complete string z is of the form

$$z = v_1 \cdot \ldots \cdot \alpha \cdot \overline{v_i} \cdot v_i \cdot \overline{v_i} \cdot \beta \cdot \ldots \cdot v_k,$$

which can be rewritten by the following rule, which is in E_Σ :

$$\overline{v_i} \cdot v_i \cdot \overline{v_i} \to \overline{v_i}.$$

The result is the string

$$v_1 \cdot \ldots \cdot \alpha \cdot \overline{v_i} \cdot \beta \cdot \ldots \cdot v_k. \quad (3)$$

If one replaces walk 2 by

$$s_1, \ldots, (x_{i-1}, \ldots, y_i, \ldots, x_i, \ldots, y_{i+1}), \ldots, s_k,$$

then the result is a walk from w_1 to w_{n+1} with associated string 3. Since the walk consists of $k-2$ segments, it follows by induction that

$$v_1 \cdot \ldots \cdot \alpha \cdot \overline{v_i} \cdot \beta \cdot \ldots \cdot v_k \Rightarrow^*_{E_\Sigma} v,$$

from which follows that

$$v_1 \cdot \ldots \cdot \alpha \cdot \overline{v_i} \cdot v_i \cdot \overline{v_i} \cdot \beta \cdot \ldots \cdot v_k \Rightarrow^*_{E_\Sigma} v.$$

Example 2.3. Assume that $\Sigma = \{a, \overline{a}\}$, and that $\mathrm{S} = \{a \to \overline{a}\overline{a}a\}$. In every $\langle \Sigma, \bar{\cdot} \rangle$-frame in which S is true, also the production rule $a \to \overline{a}aa$ is true. This can be seen as follows. Let $\mathcal{F} = \langle W, R \rangle$ be a $\langle \Sigma, \bar{\cdot} \rangle$-frame for which $\langle x, y \rangle \in R_{\overline{a}aa}$. Then there are $w_1, w_2 \in W$, such that

$$\langle x, w_1 \rangle \in R_{\overline{a}}, \quad \langle w_1, w_2 \rangle \in R_a, \quad \langle w_2, y \rangle \in R_a.$$

Since $\mathcal{F}$ is a $\langle \Sigma, \bar{\cdot} \rangle$-frame, we have

$$\langle y, w_2 \rangle \in R_{\overline{a}}, \quad \langle w_2, w_1 \rangle \in R_{\overline{a}}, \quad \langle w_1, x \rangle \in R_a.$$

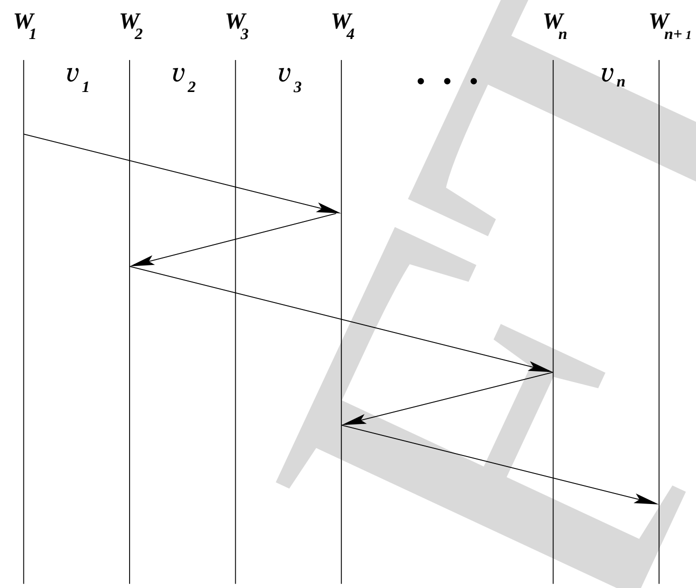


Figure 1. A walk from w_1 to w_{n+1} with 5 segments. In the corresponding string, it is possible to replace the substring $(v_2, v_3, \overline{v}_3, \overline{v}_2, v_2, v_3)$ by (v_2, v_3).

Because S is true in $\mathcal{F}$, also

$$\langle y, x\rangle \in R_a.$$

Now we have

$$\langle x, y\rangle \in R_{\overline{a}}, \quad \langle y, w_2\rangle \in R_{\overline{a}}, \quad \langle w_2, y\rangle \in R_a,$$

which implies $\langle x, y\rangle \in R_a$. Clearly not $a \Rightarrow^*_{\mathrm{S}} \overline{a}aa$, and also not $a \Rightarrow^*_{\mathrm{S}\cup\overline{\mathrm{S}}} \overline{a}aa$. However, $a \Rightarrow_{\mathrm{S}} (\overline{\overline{a}a}a) \Rightarrow_{\overline{\mathrm{S}}} (\overline{a}aa)\overline{a}a \Rightarrow_{E_\Sigma} \overline{a}aa$. $\triangleleft$

3. The Translation into GF^2

In this section, we define the transformation from regular grammar logics with converse into GF^2. The transformation can be carried out in logarithmic space. It behaves the same as the standard relational translation on all subformulae, with the exception of $\Box$-subformulae. On a $\Box$-subformula, it simulates the behaviour of an NDFA in order to determine to which worlds the $\Box$-formula applies. The translation generalises the results in (de Nivelle, 1999; de Nivelle, 2001) for the logics S4 and K5, which were at an ad hoc basis.

Unless otherwise stated, in the rest of this section, we assume that $\mathcal{L} = \langle \mathrm{ML}^{\Sigma}, \mathcal{C} \rangle$ is a regular grammar logic with converse such that $\mathcal{C}$ is the class of $\langle \Sigma, \bar{\cdot} \rangle$-frames satisfying S, a finite regular semi-Thue system closed under converse. For every $a \in \Sigma$, the automaton $\mathcal{A}_a$ is an NDFA recognizing the language $\mathrm{L_S}(a)$. It would be possible to make $\mathcal{A}_a$ canonic, for example by defining $\mathcal{A}_a$ to be the minimal DFA accepting $\mathrm{L_S}(a)$ -which is unique up to isomorphism- but there is no advantage in this. In contrast, as we shall see, this could even blow up the translation because an NDFA can have exponentially less states than a DFA accepting the same language (see e.g. (Hopcroft and Ullman, 1979)). It is in principle possible that $\mathcal{A}_a$ and $\mathcal{A}_{\overline{a}}$ are different 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 $\mathcal{A}_a = (Q_a, s_a, F_a, \delta_a)$. When all rules in S are either right-linear or left-linear, then each automaton $\mathcal{A}_a$ can be effectively built in logarithmic space in $|\mathrm{S}|$, the size of S with some reasonably succinct encoding.

3.1. The Transformation

In the sequel we assume that the two variables in GF^2 are $\{x_0, x_1\}$. The symbols α and β are used as distinct meta-variables in $\{x_0, x_1\}$. Observe that in Definition 3.2 the quantification alternates over α and β.

In the translation, we use atomic formulae of the form $\mathbf{R}_a(\alpha, \beta)$ for every $a \in \Sigma$. Because of the conditions between R_a and $R_{\overline{a}}$ in $\langle \Sigma, \bar{\cdot} \rangle$-frames, we add the axioms of the form $\forall \alpha\beta\, \mathbf{R}_a(\alpha, \beta) \Leftrightarrow \mathbf{R}_{\overline{a}}(\beta, \alpha)$ which belong to GF^2. Although this treatment of converse relations allows us to avoid some case distinctions in the proofs, in practice we might adopt an alternative treatment with a smaller signature. Indeed, one can replace syntactically in the translation process $\mathbf{R}_{\overline{a}}(\alpha, \beta)$ by $\mathbf{R}_a(\beta, \alpha)$ for every $a \in \Sigma^+$.

Definition 3.1. We assume that to each letter $a \in \Sigma$, a unique binary predicate symbol $\mathbf{R}_a$ is associated. The formula CONV_Σ defined below deals with converses:

$$\mathrm{CONV}_\Sigma \stackrel{\mathrm{def}}{=} \bigwedge_{a \in \Sigma^+} \forall x_0\ x_1\ \mathbf{R}_a(x_0, x_1) \Leftrightarrow \mathbf{R}_{\overline{a}}(x_1, x_0).$$

∇

The formula CONV_Σ is in GF^2. When a subformula $[a]\phi$ is translated, it is replaced by formulae stating

> At every point that is reachable through via a sequence of transitions labelled by a word in $\mathrm{L_S}(a)$ (i.e. accepted by the automaton $\mathcal{A}_a$), the translation of ϕ holds.

We define a function that 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 labelled by a word that accepted by the automaton, the original one-place formula holds.

Definition 3.2. Let $\mathcal{A} = \langle Q, s, F, \delta \rangle$ be an NDFA and $\varphi(\alpha)$ be a *first-order* formula with one free variable α. Assume that for every 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 every transition $\langle q, a, r \rangle \in \delta$, the formula

$$\forall \alpha \beta \, [\, \mathbf{R}_a(\alpha, \beta) \Rightarrow (\mathbf{q}(\alpha) \Rightarrow \mathbf{r}(\beta)) \,]$$

  is included in the conjunction.
- For every ϵ-transition, $\langle q, \epsilon, r \rangle \in \delta$, the formula

$$\forall \alpha \, [\, (\mathbf{q}(\alpha) \Rightarrow \mathbf{r}(\alpha)) \,]$$

  is included in the conjunction.
- For each accepting state $q \in F$, 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 subformulae of an initial formula ϕ. Definition 3.2 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 adopt the subformula-wise approach. For every state $q \in Q$, we should write $\mathbf{q}_\varphi$ instead of $\mathbf{q}$ in the translation of $t_\mathcal{A}(\alpha, \varphi)$. We sometimes omit the subscript when it is not confusing.

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 subformulae, except for those of the form $[a]\psi$, on which $t_{\mathcal{A}_a}$ will be used. In order to easily recognize the $\Box$-subformulae, 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 3.3. Let $\phi \in \mathrm{ML}^\Sigma$ be a modal formula in NNF. 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 subformulae ψ of ϕ :

- $t(\mathrm{p}, \alpha, \beta)$ equals $\mathbf{p}(\alpha)$, where $\mathbf{p}$ 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 every $a \in \Sigma$, $t(\langle a\rangle\psi,\ \alpha, \beta)$ equals $\exists\beta\ [\ \mathbf{R}_a(\alpha, \beta) \wedge t(\psi, \beta, \alpha)\ ]$,
- for every $a \in \Sigma$, $t([a]\psi,\ \alpha, \beta)$ equals $t_{\mathcal{A}_a}(\alpha, t(\psi, \alpha, \beta)\ )$.

∇

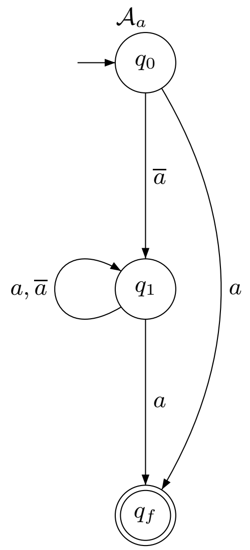
Hence, the first-order vocabulary used in $T_\mathrm{S}(\phi)$ includes

- unary predicate symbols $\mathbf{p}$ for every propositional variable p occurring in ϕ,
- binary predicate symbols $\mathbf{R}_a$ for every letter $a \in \Sigma$,
- unary predicate symbols $\mathbf{q}_{[a]\psi}$ for every $\Box$-subformula $[a]\psi$ of ϕ and $q \in Q_a$.

Lemma 3.1. Let $\phi \in \mathrm{ML}^\Sigma$ be a modal formula in NNF such that $m = \max\{|\mathcal{A}_a| \mid a \in \Sigma\}$.

1. The only variables occurring in $T_\mathrm{S}(\phi)$ are in $\{x_0, x_1\}$ and α is the only free variable in $t(\psi, \alpha, \beta)$.

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



$t_{\mathcal{A}_a}(\alpha, \varphi(\alpha))$
$\mathbf{q}_{0,\varphi}(\alpha)$
$\forall\alpha\beta\ [\ \mathbf{R}_a(\alpha,\beta) \Rightarrow (\mathbf{q}_{0,\varphi}(\alpha) \Rightarrow \mathbf{q}_{f,\varphi}(\beta))\ ]$
$\forall\alpha\beta\ [\ \mathbf{R}_{\overline{a}}(\alpha,\beta) \Rightarrow (\mathbf{q}_{0,\varphi}(\alpha) \Rightarrow \mathbf{q}_{1,\varphi}(\beta))\ ]$
$\forall\alpha\beta\ [\ \mathbf{R}_a(\alpha,\beta) \Rightarrow (\mathbf{q}_{1,\varphi}(\alpha) \Rightarrow \mathbf{q}_{1,\varphi}(\beta))\ ]$
$\forall\alpha\beta\ [\ \mathbf{R}_{\overline{a}}(\alpha,\beta) \Rightarrow (\mathbf{q}_{1,\varphi}(\alpha) \Rightarrow \mathbf{q}_{1,\varphi}(\beta))\ ]$
$\forall\alpha\beta\ [\ \mathbf{R}_a(\alpha,\beta) \Rightarrow \mathbf{q}_{1,\varphi}(\alpha) \Rightarrow \mathbf{q}_{f,\varphi}(\beta)\ ]$
$\forall\alpha\ [\ \mathbf{q}_{f,\varphi}(\alpha) \Rightarrow \varphi(\alpha)\ ]$

2. $T_{\mathrm{S}}(\phi)$ is in the guarded fragment.
3. The size of $T_{\mathrm{S}}(\phi)$ is in $\mathcal{O}(|\phi| \times m)$.
4. $T_{\mathrm{S}}(\phi)$ can be computed in logarithmic space in $|\phi| + m$.

When S is formed from production rules of a semi-Thue system 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. Observe also that $|\mathrm{CONV}_\Sigma|$ is a constant of the logic.

Example 3.1. 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 II contains to the left an automaton $\mathcal{A}_a$ recognizing the language defined in Table I for K5 (page 13). 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

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

$\triangleleft$

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. At the position where $t_{\mathcal{A}_a}(\alpha, t(\psi, \alpha, \beta))$ is translated, only $\mathbf{q}_{\mathbf{0},\psi}(\alpha)$ needs to be inserted where q_0 is the initial state of $\mathcal{A}_a$. The rest of the (translation of the) automaton yields an independent conjunct of translation.

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: $t(\mathbf{i}, \alpha, \beta) \stackrel{\text{def}}{=} \mathbf{c_i} = \alpha$ where $\mathbf{c_i}$ 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., (Areces et al., 2000)), and description logics with transitive roles and converse (see e.g., (Sattler, 1996)), can be translated into $\mathrm{GF}^2[=]$ with constants in such a way. Additionnally, by using (Blackburn and Marx, 2002, Sect. 4) regular grammar logics with converse augmented with Gregory's "actually" operator (Gregory, 2001) can be translated into such nominal tense logics.

3.2. Satisfiability Preservation

We show that the 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 variable valuation $v : \{x_0, x_1\} \rightarrow W$ and a first-order formula ψ using at most the individual variables $\{x_0, x_1\}$, we write $\mathcal{M}, v \models \psi$ to denote that ψ holds true in $\mathcal{M}$ under the valuation v (we use here the standard definition). We write $v' = v[\alpha \leftarrow w]$ to denote the valuation obtained from v by putting $v'(\beta) = v(\beta)$ and $v'(\alpha) = w$.

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 that is reachable from w through a path labelled by a finite word 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 3.2. 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 $\mathbf{q}$ for every state q of $\mathcal{A}$). Then there is an extension $\mathcal{M}' = \langle W, V' \rangle$ of $\mathcal{M}$ such that, for every $w \in W$, $(\star)$ below is satisfied:

($\star$) 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

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

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

we have

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

Proof. Write $\mathcal{A} = \langle Q, s, F, \delta \rangle$. We extend V to also interpret the symbols $\mathbf{q}$, and also we do this in a way that is consistent with the runs of $\mathcal{A}$. For all $w \in W$ and states q of $\mathcal{A}$, we define $w \in V'(\mathbf{q}) \stackrel{\text{def}}{\Leftrightarrow}$

– for every word $b_1 \cdots b_n \in \mathrm{L}(\mathcal{A})$ and for every sequence $w_1, \ldots, w_n$ of elements of W, such that

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

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

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 the condition ($\star$), we have

– for the initial state s,

$$\mathcal{M}', v[\alpha \leftarrow w] \models \mathbf{s}(\alpha).$$

– for every transition $\langle q, a, r \rangle \in \delta$,

$$\mathcal{M}' \models \forall \alpha \beta\ [\ \mathbf{R}_a(\alpha, \beta) \Rightarrow (\mathbf{q}(\alpha) \Rightarrow \mathbf{r}(\beta))\ ].$$

– for every ϵ-transition $\langle q, \epsilon, r \rangle \in \delta$,

$$\mathcal{M}' \models \forall \alpha\ [\ \mathbf{q}(\alpha) \rightarrow \mathbf{r}(\alpha)\ ].$$

– for each accepting state $q \in F$,

$$\mathcal{M}' \models \forall \alpha\ [\ \mathbf{q}(\alpha) \rightarrow \varphi(\alpha)\ ].$$

$\mathcal{M}'$ and $\mathcal{M}$ agree on all formulas that do not contain any symbols introduced by $t_{\mathcal{A}}(\alpha, \varphi)$, i.e. those of the form $\mathbf{q}$ for some state q of $\mathcal{A}$. □

Next follows the main theorem about satisfiability preservation.

Theorem 3.3. Let $\phi \in \mathrm{ML}^{\Sigma}$ be a modal formula in NNF. Then the following are equivalent:

1. There exist a $\langle \Sigma, \bar{\cdot} \rangle$-model $\mathcal{M} = \langle W, R, V \rangle$ and a $w \in W$ such that $\mathcal{M}$ satisfies S and $\mathcal{M}, w \models \phi$.
2. $T_{\mathrm{S}}(\phi) \wedge \mathrm{CONV}_{\Sigma}$ is satisfiable in FOL.

Proof. We first prove that **(1)** implies **(2)**. Assume that there exists a $\langle \Sigma, \bar{\ } \rangle$-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 $\mathcal{M}'$ of $T_{\mathrm{S}}(\phi) \wedge \mathrm{CONV}_{\Sigma}$.

In order to do this, we first construct an incomplete interpretation $\mathcal{M}_0 = \langle W, V_0 \rangle$, which will be completed through successive applications of Lemma 3.2. V_0 is obtained as follows:

– For every $a \in \Sigma$, $V_0(\mathbf{R}_a) \stackrel{\mathrm{def}}{=} R_a$,

– For every propositional variable p occurring in ϕ, we set $V_0(\mathbf{p}) \stackrel{\mathrm{def}}{=} V(\mathrm{p})$.

We now have a model interpreting the symbols introduced by $t(\psi, \alpha, \beta)$, but not the symbols introduced by $t_{\mathcal{A}}(\alpha, \psi)$. It is easily checked that CONV_{Σ} holds true in $\langle W, V_0 \rangle$. In order to complete the model construction, we order the $\Box$-subformulae 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 3.2 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 with 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 unchanged. We have

– for every $a \in \Sigma$, $V_0(\mathbf{R}_a) = \cdots = V_n(\mathbf{R}_a)$,

– for every propositional variable p of 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 $\mathcal{M}_n, v[\alpha \leftarrow x] \models t(\psi, \alpha, \beta)$. We treat only the modal cases, because the propositional cases are trivial.

– If ψ has form $[a]\psi'$ with $a \in \Sigma$, then $t([a]\psi', \alpha, \beta) = t_{\mathcal{A}_a}(\alpha, t(\psi', \alpha, \beta))$.

  For every word $b_1 \cdots b_k$ accepted by $\mathcal{A}_a$, for every sequence $w_1, \ldots, w_k \in W_n$ such that

  $$\langle x, w_1 \rangle \in V_n(\mathbf{R}_{b_1}),\ \langle w_1, w_2 \rangle \in V_n(\mathbf{R}_{b_2}),\ \ldots\ , \langle w_{k-1}, w_k \rangle \in V_n(\mathbf{R}_{b_k}),$$

  also

  $$\langle x, w_1 \rangle \in R_{b_1},\ \langle w_1, w_2 \rangle \in R_{b_2},\ \cdots\ , \langle w_{k-1}, w_k \rangle \in R_{b_k},$$

  by construction of $V_0, V_1, \cdots, V_n$. Because $\mathcal{M}$ satisfies S, by Lemma 2.4, we also have $\langle x, w_k \rangle \in R_a$, which in turn implies $\langle x, w_k \rangle \in V_n(\mathbf{R}_a)$, by construction of the V_i. Therefore, we have $\mathcal{M}, w_k \models \psi'$. By the induction hypothesis, we have $\mathcal{M}_n, v[\beta \leftarrow w_k] \models t(\psi', \beta, \alpha)$. 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

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

  Now we have all ingredients of Lemma 3.2 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 $\langle x, y \rangle \in V_0(\mathbf{R}_a)$ and therefore also $\langle x, y \rangle \in V_n(\mathbf{R}_a)$. By the induction hypothesis, $\mathcal{M}_n, v[\beta \leftarrow y] \models t(\psi', \beta, \alpha)$. Hence,

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

Next we show that **(2)** implies **(1)**. Assume that $T_{\mathrm{S}}(\phi) \wedge \mathrm{CONV}_\Sigma$ is FOL-satisfiable. This means that there exist a FOL model $\mathcal{M} = \langle W, V \rangle$ and a valuation v such that $\mathcal{M}, v \models T_{\mathrm{S}}(\phi) \wedge \mathrm{CONV}_\Sigma$. We construct a model $\mathcal{M}'$ of ϕ in two stages. First we construct $\mathcal{M}'' = \langle W'', R'', V'' \rangle$ as follows.

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

Then define $\mathcal{M}' = \langle W', R', V' \rangle$ where R' is defined from $\langle W', R' \rangle = C_{\mathrm{S}}(\langle W'', R'' \rangle)$ and $V' = V''$. Here C_{S} is the closure operator, defined in Definition 2.18. 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 $W' = W$. Because $\mathcal{M} \models \mathrm{CONV}_{\Sigma}$, the frame $\mathcal{M}''$ is a $\langle \Sigma, \bar{\cdot} \rangle$-frame. By definition of C_{S}, the structure $\mathcal{M}'$ is an S-model, and also 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 $\mathcal{M}, v \models \exists \beta \ [\ \mathbf{R}_a(\alpha, \beta) \wedge t(\psi', \beta, \alpha)\ ]$.

  This means there is a $y \in W$, such that $\langle v(\alpha), y \rangle \in V(\mathbf{R}_a)$ 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 assume that $\mathcal{M}, v \models t_{\mathcal{A}_a}(\alpha, t(\psi', \alpha, \beta))$. First, we show that for every word $b_1 \cdots b_k$ accepted by $\mathcal{A}_a$, for every sequence $w_1, \ldots, w_k$ of elements of W, for which it is the case that
  $$\langle v(\alpha), w_1 \rangle \in V(\mathbf{R}_{b_1}),\ \langle w_1, w_2 \rangle \in V(\mathbf{R}_{b_2}),\ \ldots, \langle w_{k-1}, w_k \rangle \in V(\mathbf{R}_{b_k}),$$
  the following holds
  $$\mathcal{M}, v[\alpha \leftarrow w_k] \models t(\psi', \alpha, \beta).$$
  Indeed, $\mathcal{M}, v \models \mathbf{s}(\alpha)$, for the initial state s of $\mathcal{A}_a$. It is easy to show by induction on k that the following holds: Let $b_1 \cdots b_k$ be some word over Σ^k. Let q be a state of $\mathcal{A}_a$ such that $\langle s, b_1 \cdot \ldots \cdot b_k, q \rangle \in \delta^*$, for the initial state $s \in Q$. Then for every sequence $w_1, \ldots, w_k$ of elements of W such that
  $$\langle v(\alpha), w_1 \rangle \in V(\mathbf{R}_{b_1}),\ \ \langle w_1, w_2 \rangle \in V(\mathbf{R}_{b_2}),\ \ \ldots,\ \ \langle w_{k-1}, w_k \rangle \in V(\mathbf{R}_{b_k}),$$
  it must be the case that $\mathcal{M}, v[\alpha \leftarrow w_k] \models \mathbf{q}(\alpha)$. Then the result follows from the fact that $\mathcal{M}, v[\alpha \leftarrow w_k] \models \mathbf{q}(\alpha) \Rightarrow \psi'(\alpha)$, for every 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, by Theorem 2.6, there is a word $u \in \mathrm{L}(\mathcal{A}_a)$ for which $a \Rightarrow^*_{\mathrm{S} \cup \overline{\mathrm{S}}} u$ and $R''_u(x, y)$. By the above property, we have
  $$\mathcal{M}, v[\alpha \leftarrow y] \models t(\psi', \alpha, \beta).$$
  By the induction hypothesis, we obtain $\mathcal{M}', y \models \psi'$.

□

The uniformity of the translation allows us to establish forthcoming Theorem 3.4. We first define the general satisfiability problem for regular grammar logic with converse, denoted by $\mathrm{GSP}(\mathrm{REG}^c)$, as follows:

input: A finite semi-Thue system S closed under converse, in which either all production rules are left-linear, or all production rules are right-linear, and an ML^{Σ}-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 (Rozenberg and Salomaa, 1994).

Theorem 3.4.

(I) The S-satisfiability problem is in EXPTIME for every regular semi-Thue system closed under converse.

(II) $\mathrm{GSP}(\mathrm{REG}^c)$ is EXPTIME-complete.

Theorem 3.4(I) is a corollary of Theorem 3.3. The lower bound in Theorem 3.4(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 3.4(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 $k \geq 2$ (Grädel, 1999b). 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 $\mathrm{L}_{\mathrm{S}}(a)$ 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 $T_{\rm S}$ 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 no longer 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. The Borders of the Translation Method

In this section we try to answer the following question: given a finite context-free semi-Thue system S closed under converse, how to find out whether the modal logic based on S can be translated by the method of Section 3.1? This is a natural question, because modal logics are usually presented by modal axioms, and in most cases the semi-Thue system naturally corresponds to the modal axioms.

As stated in Sect. 2.2, a logic can be translated if for every letter $a \in \Sigma$, the language $\mathrm{L}_{\rm S}(a) = \{u \in \Sigma^* \mid a \Rightarrow^*_{\rm S} u\}$ are regular. This question is in general undecidable, because it is already undecidable whether the language generated by a linear grammar is regular (see e.g. (Mateescu and Salomaa, 1997, page 31)).

However, regularity of the languages $\mathrm{L}_{\rm S}(a)$ is not a necessary condition for existence of a translation. The reason for this is that different semi-Thue systems may characterize the same logic. Exactly which context-free semi-Thue systems characterize the same logic is determined by Theorem 4.1 below.

Theorem 4.1. Let Σ be an alphabet with converse mapping $\bar{\cdot}$ and $\mathrm{S}_1, \mathrm{S}_2$ be context-free semi-Thue systems closed under converse. If for all $(i_1, i_2) \in \{(1,2),(2,1)\}$ and rules $u \rightarrow v \in \mathrm{S}_{i_1}$, there is a string $z \in \Sigma^*$, such that

$$u \Rightarrow^*_{\mathrm{S}_{i_2}} z \text{ and } z \Rightarrow^*_{E_\Sigma} v,$$

then S_1 and S_2 define the same set of $\langle \Sigma, \bar{\cdot} \rangle$-frames.

Proof. It follows from Theorem 2.5 that a $\langle \Sigma, \bar{\cdot} \rangle$-frame satisfies S_1 if and only if it satisfies S_2. □

Theorem 4.1 determines when two context-free semi-Thue systems define the same modal logic. Using this equivalence, the criterion above can be refined to: a modal logic $\mathcal{L}$ based on finite context-free semi-Thue system S closed under converse can be translated by the method of Section 3.1 if (and approximately only-if) there is a finite regular semi-Thue system S′ closed under converse which is equivalent to S, for which the languages $L_{S'}(a)$ are regular. We do not have a general method for answering this question, and we don't know whether the problem is decidabile.

There exist pairs of context-free semi-Thue systems S_1 and S_2 defining the same logic, where S_1 is non-regular while S_2 is regular. An example of such a pair is given in the following examples:

Example 4.1. The euclideanity condition can be 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 $S_n = \{a \rightarrow \overline{a}^n a,\ \overline{a} \rightarrow \overline{a} a^n\}$. The case $n = 1$ corresponds to euclideanity, which is regular, see Example 2.1 and Example 3.1. We will show that in general, for $n > 1$, the language $L_{S_n}(a)$ is not regular. Nevertheless, S_n-satisfiability restricted to formulae with only the modal operator $[a]$ is known to be decidable, see e.g. (Gabbay, 1975; Hustadt and Schmidt, 2003). To see why the languages $L_{S_n}(a)$ are not regular, consider strings of the following form:

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

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

We show that

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

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

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

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

We now prove the equivalence from left to right. Let us say that u is *a predecessor of* v if $u \Rightarrow_{S_n} v$. 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 $\overline{\sigma}_n(i, 0)$ with $i \geq 0$ 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 the form either 1 or 3 have no such sequence. The strings of the form either 2 or 4 have exactly one such sequence.
We have

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

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

We will now show that, although

$$\mathrm{S}_2 = \{\ a \rightarrow \overline{a}\overline{a}a, \quad \overline{a} \rightarrow \overline{a}aa\ \}$$

is not regular, the logic defined by it, is regular. The reason for this is the fact that S_2 defines the same logic as

$$\mathrm{S}_2' = \{\ a \rightarrow \overline{a}\overline{a}a, \quad a \rightarrow \overline{a}aa, \quad \overline{a} \rightarrow \overline{a}\overline{a}a, \quad \overline{a} \rightarrow \overline{a}aa\ \},$$

which is regular. In order to show that S_2 and S_2' define the same logic, it suffices to observe that $\mathrm{S}_2 \subseteq \mathrm{S}_2'$, and that

$$a \Rightarrow_{\mathrm{S}_2} \overline{a}\overline{a}a \Rightarrow_{\mathrm{S}_2} (\overline{a}aa)\overline{a}a \Rightarrow_{E_{\{a,\overline{a}\}}} \overline{a}a(a), \text{ and}$$

$$\overline{a} \Rightarrow_{\mathrm{S}_2} \overline{a}aa \Rightarrow_{\mathrm{S}_2} \overline{a}a(\overline{a}\overline{a}a) \Rightarrow_{E_{\{a,\overline{a}\}}} (\overline{a})\overline{a}a.$$

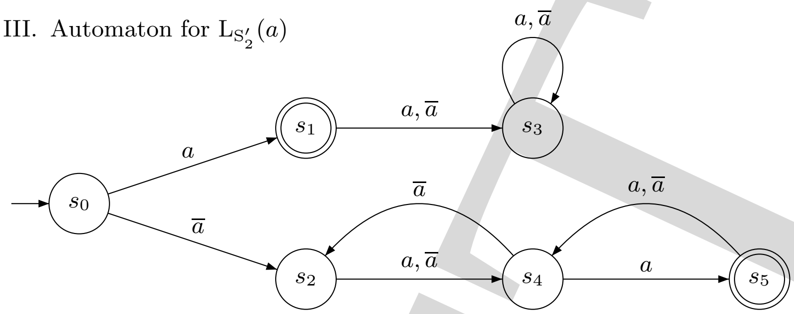
In order to show that S_2' is regular, we show that $\mathrm{L}_{\mathrm{S}_2'}(a)$ is recognized by the automaton of Table III.

Lemma 4.2. The automaton in Table III recognizes the language $\mathrm{L}_{\mathrm{S}_2'}(a)$.

Proof. It is clear that state s_1 accepts only the word a. Every run ending in s_5 consists of three parts:

1. The path (s_0, s_2, s_4), accepting either $\overline{a}a$ or $\overline{a}\overline{a}$.

Table III. Automaton for $L_{S'_2}(a)$



2. A number of cycles of form (s_4, s_5, s_4), or (s_4, s_2, s_4) accepting aa, $a\overline{a}$, $\overline{a}a$, or $\overline{a}\overline{a}$. These are all possible words of length two.
3. The path (s_4, s_5), accepting the word a.

As a consequence, a word u is accepted by the automaton iff it has one of the following three forms:

$$a, \quad \overline{a}a\Sigma^{2i}a, \quad \text{or } \overline{a}\overline{a}\Sigma^{2i}a, \tag{4}$$

where Σ^{2i} is an arbitrary word of length $2i$ on the alphabet $\Sigma = \{a, \overline{a}\}$.

We first show that every string in $L_{S'_2}(a)$ has the form 4. The language $L_{S'_2}(a)$ is inductively defined as the smallest set containing a and closed under rewriting by rules of S'_2. Therefore, it is sufficient to show that a has form 4 and that form 4 is preserved by rewriting under rules in S'_2.

If u is of form $\overline{a}a\Sigma^{2i}a$, rewriting at $\overline{a}$ results in $(\overline{a}\overline{a}a)a\Sigma^{2i}a$ or $(\overline{a}aa)a\Sigma^{2i}a$, which is of the form $\overline{a}\overline{a}\Sigma^{2i+1}a$ or $\overline{a}a\Sigma^{2i+1}a$. Rewriting at the first a results either in $\overline{a}(\overline{a}\overline{a}a)\Sigma^{2i}a$ or $\overline{a}(\overline{a}aa)\Sigma^{2i}a$. Both can be written as $\overline{a}\overline{a}\Sigma^{2i+1}a$. Rewriting at the last a results either in $\overline{a}a\Sigma^{2i}\overline{a}\overline{a}a$ or $\overline{a}a\Sigma^{2i}\overline{a}aa$, which both can be written as $\overline{a}a\Sigma^{2i+1}a$. Rewriting in Σ^{2i} results also in a string of one of the three forms. The 7 possible rewrites in $\overline{a}\overline{a}\Sigma^{2i}a$ can be analyzed analogously.
Next we show by induction on i that every word of the form 4 is in $L_{S'_2}(a)$. For the base case $i = 0$, it is immediate that a, $\overline{a}aa$, $\overline{a}\overline{a}a \in L_{S'_2}(a)$. Now assume that, for some i, every word of the form $\overline{a}a\Sigma^{2i}a$ or $\overline{a}\overline{a}\Sigma^{2i}a$ belongs to $L_{S'_2}(a)$. We show that every word of the form $\overline{a}a\Sigma^{2(i+1)}a$ or $\overline{a}\overline{a}\Sigma^{2(i+1)}a$ also belongs to $L_{S'_2}(a)$. In each case, we find a string of form 4 (but with parameter i) from which the current string can be obtained using a single rewrite step by a rule in S'_2. If one

expands the first two letters of Σ^{2i+2}, one obtains the following forms:

$$\overline{a}a(aa)\Sigma^{2i}a,\ \ \overline{a}a(a\overline{a})\Sigma^{2i}a,\ \ \overline{a}a(\overline{a}a)\Sigma^{2i}a,\ \ \overline{a}a(\overline{a}\overline{a})\Sigma^{2i}a,\ \text{and}$$

$$\overline{a}\overline{a}(aa)\Sigma^{2i}a,\ \ \overline{a}\overline{a}(a\overline{a})\Sigma^{2i}a,\ \ \overline{a}\overline{a}(\overline{a}a)\Sigma^{2i}a,\ \ \overline{a}\overline{a}(\overline{a}\overline{a})\Sigma^{2i}a.$$

Case 1.
By induction hypothesis, $\overline{a}a\Sigma^{2i}a \in \mathrm{L}_{\mathrm{S}_2'}(a)$ and

$$\overline{a}a\Sigma^{2i}a \Rightarrow_{\{\overline{a}\rightarrow\overline{a}aa\}} \overline{a}a(aa)\Sigma^{2i}a$$

by rewriting at the first position. Hence, $\overline{a}a(aa)\Sigma^{2i}a \in \mathrm{L}_{\mathrm{S}_2'}(a)$.
Case 2.
By induction hypothesis, $\overline{a}\overline{a}\Sigma^{2i}a \in \mathrm{L}_{\mathrm{S}_2'}(a)$ and

$$\overline{a}\overline{a}\Sigma^{2i}a \Rightarrow_{\{\overline{a}\rightarrow\overline{a}aa\}} \overline{a}a(a\overline{a})\Sigma^{2i}a$$

by rewriting at the first position. Hence, $\overline{a}a(a\overline{a})\Sigma^{2i}a \in \mathrm{L}_{\mathrm{S}_2'}(a)$.
Cases 5 and 6. Similar to the cases 5 and 6 using the rule $\overline{a} \rightarrow \overline{a}\overline{a}a$.
Case 7.
By induction hypothesis, $\overline{a}\overline{a}\Sigma^{2i}a \in \mathrm{L}_{\mathrm{S}_2'}(a)$ and

$$\overline{a}\overline{a}\Sigma^{2i}a \Rightarrow_{\{\overline{a}\rightarrow\overline{a}\overline{a}a\}} \overline{a}\overline{a}(\overline{a}a)\Sigma^{2i}a$$

by rewriting at the second position. Hence, $\overline{a}\overline{a}(\overline{a}a)\Sigma^{2i}a \in \mathrm{L}_{\mathrm{S}_2'}(a)$.
Cases 4 and 8.
In order to treat both cases, we pose u to denote a string in $\{\overline{a}a, \overline{a}\overline{a}\}$.
By induction hypothesis, $u\Sigma^{2i}a \in \mathrm{L}_{\mathrm{S}_2'}(a)$ and

$$u\Sigma^{2i}a \in \mathrm{L}_{\mathrm{S}_2'}(a) \Rightarrow_{\{a\rightarrow\overline{a}\overline{a}a\}} u(\overline{a}\overline{a})\Sigma^{2i}a$$

by rewriting at the first occurrence of a in $\Sigma^{2i}a$. Hence, $u(\overline{a}\overline{a})\Sigma^{2i}a \in \mathrm{L}_{\mathrm{S}_2'}(a)$.
Case 3.
By induction hypothesis, $\overline{a}a\Sigma^{2i}a \in \mathrm{L}_{\mathrm{S}_2'}(a)$. Write $\Sigma^{2i} = (\overline{a}a)^{i_1}\Sigma^{2i_2}$, where $i = i_1 + i_2$, and i_1 is maximal.
Case 3.1.: $\Sigma^{2i_2}a$ starts by a.
We have

$$\overline{a}a\Sigma^{2i}a \Rightarrow_{\{a\rightarrow\overline{a}aa\}} \overline{a}a(\overline{a}a)\Sigma^{2i}a$$

by rewriting at the first occurrence of a in $\Sigma^{2i_2}a$.
Case 3.2: Σ^{2i_2} starts with $\overline{a}\overline{a}$.
$\Sigma^{2i_2}a$ is of the form $\overline{a}^k au$ for some $k \geq 2$ and string u. Since $\overline{a}^{k-2}au \Rightarrow_{\{a\rightarrow\overline{a}\overline{a}a\}}$ $\overline{a}^k au$ and $(\overline{a}a)^{i_1}\overline{a}^{k-2}au \in \mathrm{L}_{\mathrm{S}_2'}(a)$ by induction hypothesis, we have that $\overline{a}a\overline{a}a\Sigma^{2i}a \in \mathrm{L}_{\mathrm{S}_2'}(a)$.

□

The regularity of S_2' and its equivalence to S_2 make it possible to deduce the following upper bound.

Theorem 4.3. The bimodal logic whose set of frames is the set of frames satisfying S_2 is a regular grammar logic with converse. Hence, its satisfiability problem can be solved in EXPTIME.

The automaton in Table III was discovered by a computer program. Given a language L over an alphabet Σ, one can define the following equivalence relation $\equiv_L$ on strings over Σ : For two words $u_1, u_2 \in \Sigma^*$, $u_1 \equiv_L u_2$ iff for all $v \in \Sigma^*$, $u_1 \cdot v \in L \Leftrightarrow u_2 \cdot v \in L$. By the Myhill-Nerode theorem, if $\equiv_L$ partitions Σ^* into a finite set of equivalence-classes, then L is regular, and the equivalence classes define the states of the minimal DFA recognizing L. If one tries sufficiently many v's, one has a high chance of finding the right equivalence classes. Of course one cannot be certain in general that the automata returned by the program are correct, but in most cases verifying an automaton is easier than finding one. The same computer program has also proposed an automaton for the modal logic defined by $S_3 = \{a \rightarrow \overline{a}^3 a, \quad \overline{a} \rightarrow \overline{a} a^3\}$. This automaton has 19 states. Based on the fact that the grammar logics defined by S_1, S_2 and S_3 can be characterized by a regular language, we conjecture that all of the grammar logics defined by a grammar of form S_i are regular. At the same time, it appears that the modal logics defined by the context-free semi-Thue systems $\hat{S}_i = \{a \rightarrow a^i \overline{a}, \quad \overline{a} \rightarrow a \overline{a}^i\}$ with $i > 1$, are non-regular, based on the output of the same computer program. We conjecture that the output of the program is correct and that the grammar logics defined by $\hat{S}_i,\ i > 1$ are indeed non-regular.

5. Related work

In this section, we compare our contribution to translations of modal logics similar to ours, to tableaux-based proof systems with single steps and to the characterization of star-free languages with first-order logic over finite words. Before doing so, let us mention some other relevant works.

Complexity issues for regular grammar logics have been studied in (Demri, 2001; Demri, 2002) (see also (Baldoni, 1998; Baldoni et al., 1998)) whereas grammar logics are introduced in (del Cerro and Penttonen, 1988). Frame conditions involving the converse relations are not treated in (Demri, 2001; Demri, 2002). These are needed for example for S5 modal connectives. The current work can be viewed as a natural continuation of (de Nivelle, 1999) and (Demri, 2001). Translation of regular grammar logics into converse PDL can be found in the preprint (Demri and de Nivelle, 2004, Sect. 4) extending (Demri, 2001).

The frame conditions considered in the present work can be defined by the MSO definable closure operators (Ganzinger et al., 1999). However, it is worth noting that by contrast to what is done in (Ganzinger et al., 1999), 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 (Ganzinger et al., 1999). Moreover, we do not use MSO definable built-in relations, just plain GF^2.

5.1. Incorporating a theory in the translation

Unlike the standard relational translation from modal logic into classical predicate logic (see e.g., (Fine, 1975; van Benthem, 1976; Morgan, 1976; Moore, 1977)), the subformulae in $T_{\mathrm{S}}(\phi)$ mix the frame conditions and the interpretation of the logical connectives. Frame conditions are incorporated in our translation as done also in (Schmidt and Hustadt, 2004). Such a feature is shared by many other translations dealing for modal logics, see e.g. (Balbiani and Herzig, 1994; Demri and Goré, 2002). However, the work (Schmidt and Hustadt, 2004) is closely related to ours. Probably the main similarities are the following ones.

- Both translations are from a large class of modal logics into GF^2.
- Translations of the modal logics K,T,K4 and S4 in (Schmidt and Hustadt, 2004) and in this paper are essentially the same, once minor differences are disgarded (NNF is our work and renaming in (Schmidt and Hustadt, 2004)). For example, the clause schema for the K4 axiom $[a]\mathrm{p} \Rightarrow [a][a]\mathrm{p}$ in (Schmidt and Hustadt, 2004) is the following:

  $$\forall\ x\ Q_{\Box \mathrm{p}}(x) \Rightarrow (\forall\ y\ \mathbf{R}_a(x,y) \Rightarrow Q_{\Box \mathrm{p}}(y)).$$

  This clause schema is obtained from $\Box \mathrm{p} \Rightarrow \Box\Box \mathrm{p}$ by performing a partial translation that stops before the innermost modalities are eliminated and by renaming subformulae. In this paper, the letter a from a K4 modal operator $[a]$ is related to the regular language a^+ that can be recognized by the finite-state automaton $\mathcal{A}_0 = \langle \{q_0, q_1\}, q_0, q_1, \{q_0 \stackrel{a}{\rightarrow} q_1, q_1 \stackrel{a}{\rightarrow} q_1\}\rangle$. The formula $t_{\mathcal{A}_0}(\alpha, \varphi(\alpha))$ introduced in Definition 3.2 contains a conjunct of the form

  $$\forall\ x\ \mathbf{q_1}(x) \Rightarrow (\forall\ y\ \mathbf{R}_a(x,y) \Rightarrow \mathbf{q_1}(y)),$$

  where $\mathbf{q_1}$ is also a monadic predicate symbol depending on $\varphi(\alpha)$ which is after all nothing else than a predicate symbol of the above form $Q_{\Box \mathrm{p}}$ depending on p.
- Both methods require some preliminary knowledge. In our case, the grammar logic at hand needs to be shown regular whereas

in (Schmidt and Hustadt, 2004) one needs to determine how many finite instances of the clause schemata obtained from axioms are sufficient for completeness. Both problems are difficult in general but for many known logics the problem can be solved.

5.2. 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. (Straubing, 1994). 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 (Ganzinger et al., 1999);
- 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. (Perrin, 1990). 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. (Pin, 1994). 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 (Verma, 2003).

6. Concluding Remarks

The two main contributions of the paper are the following:

- for every regular grammar logic with converse the design of a logspace translation into GF^2,
- to characterize when two regular grammar logics define the same set of satisfiable formulae.

As a by-product, our work allows us to answer positively to some questions left open in (Demri, 2001). Typically, we provide evidence that the first-order fragment to translate the regular grammar logics with converse into is simply GF^2: there is no need for first-order fragment augmented with fixed-point operators, as far as regular grammar logics are concerned.

In our view, Theorem 2.5 can be interpreted as confirming that the use of grammars for defining modal logics, is natural. Theorem 2.5 completely determines the behaviour of grammar rules on frames in terms of the behaviour of grammar rules on words. We end by listing a few open problems that we believe are worth investigating.

1. The study of the computational behaviour of the translation to mechanize modal logics using for instance (de Nivelle and Pratt-Hartmann, 2001) should be further investigated.
2. Although regular grammar logics (with converse) can be viewed as fragments of propositional dynamic logic (see e.g. (Demri and de Nivelle, 2004, Sect. 4)), 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.
3. Is there a PSPACE fragment of GF^2 in which the following modal logics can be naturally embedded: S4, $S4_t$ (S4 with past-time operators), Grz, and G? (to quote a few modal logics in PSPACE, see e.g. (Chagrov and Zakharyaschev, 1997)).
4. Can our translation method be extended to a reasonable fragment of first-order modal logics?
5. Combining the translation from S4 into GF^2 with Gödel's translation from intuitionistic logic into S4 (see (Troelstra and Schwichtenberg, 1996)), one obtains a translation from intuitionistic logic into GF^2, see e.g. the preprint (Demri and de Nivelle, 2004, Sect. 5). Can it be extended to a reasonable fragment of first-order intuitionistic logic?

6. Are the conjectures at the end of Section 4 true?

Acknowledgements

The authors would like to thank the referees for many valuable remarks and suggestions greatly improving the readability of this document.

References

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

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

Andreka, H., I. Nemeti, and J. van Benthem: 1998, 'Modal languages and Bounded Fragments of Predicate Logic'. *Journal of Philosophical Logic* **27**(3), 217–274.

Areces, C., P. Blackburn, and M. Marx: 2000, 'Complexity of Hybrid Temporal Logics'. *Logic Journal of the IGPL* **8**(5), 653–679.

Balbiani, P. and A. Herzig: 1994, 'A translation from the modal logic of provability into K4'. *Journal of Applied Non-Classical Logics* **4**, 73–77.

Baldoni, M.: 1998, 'Normal Multimodal Logics: Automated Deduction and Logic Programming'. Ph.D. thesis, Università degli Studi di Torino.

Baldoni, M., L. Giordano, and A. Martelli: 1998, 'A Tableau Calculus for Multimodal Logics and Some (Un)Decidability results'. In: H. de Swart (ed.): *TABLEAUX-8*, Vol. 1397 of *Lecture Notes in Artificial Intelligence*. pp. 44–59, Springer.

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

Caucal, D.: 1996, 'On infinite transition graphs having a decidable monadic theory'. In: *ICALP'96*, Vol. 1099 of *Lecture Notes in Computer Science*. pp. 194–205, Springer, Berlin.

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

Chagrov, A. and V. Shehtman: 1994, 'Algorithmic aspects of propositional tense logics'. In: L. Pacholski and J. Tiuryn (eds.): *CSL-8, Kazimierz, Poland*, Vol. 933 of *Lecture Notes in Computer Science*. pp. 442–455, Springer, Berlin.

Chagrov, A. and M. Zakharyaschev: 1997, *Modal Logic*. Clarendon Press, Oxford.

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

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

de Nivelle, H.: 1998, 'A Resolution Decision Procedure for the Guarded Fragment'. In: C. Kirchner and H. Kirchner (eds.): *CADE-15, Lindau, Germany*, Vol. 1421 of *Lecture Notes in Artificial Intelligence*. pp. 191–204, Springer.

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

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

de Nivelle, H. and M. de Rijke: 2003, 'Deciding the Guarded Fragments with Resolution'. *Journal of Symbolic Computation* **35**(1), 21–58.

de Nivelle, H. and I. Pratt-Hartmann: 2001, 'A Resolution-Based Decision Procedure for the Two-Variable fragment With Equality'. In: R. Goré, A. Leitsch, and T. Nipkow (eds.): *IJCAR'01*, Vol. 2083 of *Lecture Notes in Computer Science*. pp. 211–225, Springer.

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

del Cerro, L. F. and O. Gasquet: 2002, 'A General Framework for Pattern-Driven Modal Tableaux'. *Logic Journal of the IGPL* **10**(1), 51–83.

del Cerro, L. F. and O. Gasquet: 2004, 'Modal tableaux for reasoning about diagrams'. In: A. Pietruszczak and J. Malinowski (eds.): *Essays in Logic and Ontology (Dedicated to Jerzy Perzanowski)*, Poznan Studies in the Philosophy and the Humanities. Amsterdam, The Netherlands: Rodopi, p. To appear.

del Cerro, L. F. and M. Penttonen: 1988, 'Grammar logics'. *Logique et Analyse* **121–122**, 123–134.

Demri, S.: 2000, 'The Nondeterministic Information Logic NIL is PSPACE-complete'. *Fundamenta Informaticae* **42**, 211–234.

Demri, S.: 2001, 'The Complexity of Regularity in Grammar Logics and Related Modal Logics'. *Journal of Logic and Computation* **11**(6), 933–960.

Demri, S.: 2002, 'Modal Logics with Weak Forms of Recursion: PSPACE specimens'. In: M. de Rijke, H. Wansing, F. Wolter, and M. Zakharyaschev (eds.): *Advances in Modal Logics, selected papers from 3rd Workshop on Advances in Modal Logics (AIML'2000), Leipzig, Germany, Oct. 2000*. pp. 113–138, World Scientific.

Demri, S. and H. de Nivelle: 2004, 'Deciding regular grammar logics with converse through first-order logic'. arXiv:cs.LO/0306117.

Demri, S. and R. Goré: 2002, 'Theoremhood Preserving Maps Characterising Cut Elimination for Modal Provability Logics'. *Journal of Logic and Computation* **12**(5), 861–884.

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

Fine, K.: 1975, 'Some connections between elementary and modal logic'. In: S. Kanger (ed.): *3rd Scandinavian Logic Symposium*. pp. 15–31, North Holland.

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

Gabbay, D.: 1981, 'Expressive Functional Completeness in Tense Logic'. In: U. Mönnich (ed.): *Aspects of Philosophical Logic*. Reidel, pp. 91–117.

Ganzinger, H. and H. de Nivelle: 1999, 'A Superposition Decision Procedure for the Guarded Fragment with Equality'. In: *LICS'99*. pp. 295–305.

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

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

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

Grädel, E.: 1999a, 'Decision procedures for guarded logics'. In: H. Ganzinger (ed.): *CADE'99*, Vol. 1632 of *Lecture Notes in Artificial Intelligence*. pp. 31–51, Springer.

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

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

Grädel, E. and I. Walukiewicz: 1999, 'Guarded Fixed Point Logic'. In: *LICS'99*. pp. 45–54.

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

Herzig, A.: 1989, 'Raisonnement automatique en logique modale et algorithmes d'unification'. Ph.D. thesis, Université Paul Sabatier, Toulouse.

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

Hirsch, C. and S. Tobies: 2002, 'A Tableau Algorithm for the Clique Guarded Fragment'. In: M. de Rijke, H. Wansing, F. Wolter, and M. Zakharyaschev (eds.): *Advances in Modal Logics, selected papers from 3rd Workshop on Advances in Modal Logics (AIML'2000), Leipzig, Germany, Oct. 2000*. pp. 257–278, World Scientific.

Hladik, J.: 2002, 'Implementation and Optimisation of a Tableau Algorithm for the Guarded Fragment'. In: U. Egly and C. Fermüller (eds.): *Automated Reasoning with Analytic Tableaux and Related Methods*, Vol. 2381 of *Lecture Notes in Computer Science*. pp. 145–159, Springer Verlag.

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

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

Horrocks, I. and U. Sattler: 2004, 'Decidability of SHIQ with complex role inclusion axioms'. *Artificial Intelligence* **160**(1–2), 79–104.

Hustadt, U. and R. Schmidt: 2003, 'A Principle for Incorporating Axioms into the First-Order Translation of Modal Formulae.'. In: *CADE'03*, Vol. 2741 of *Lecture Notes in Artificial Intelligence*. pp. 412–426, Springer, Berlin. Long version in (Schmidt and Hustadt, 2004).

Kieronski, E.: 2003, 'The Two-Variable Guarded Fragment with Transitive Guards is 2EXPTIME-hard'. In: A. Gordon (ed.): *6th Int. Conf. on Foundations of Software Science and Computational Structures (FOSSACS'03), Warsaw, Poland*, Vol. 2620 of *Lecture Notes in Computer Science*. pp. 299–312, Springer, Berlin.

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

Marx, M., S. Mikulas, and S. Schlobach: 1999, 'Tableau Calculus for Local Cubic Modal Logic and its implementation'. *Logic Journal of the IGPL* **7**(6), 755–778.

Massacci, F.: 1997, 'Tableaux methods for Access Control in Distributed Systems'. In: D. Galmiche (ed.): *TABLEAUX'97*, Vol. 1227 of *Lecture Notes in Artificial Intelligence*. pp. 246–260, Springer, Berlin.

Massacci, F.: 2000, 'Single Steps Tableaux for Modal Logics'. *Journal of Automated Reasoning* **24**(3), 319–364.

Mateescu, A. and A. Salomaa: 1997, 'Formal languages: an introduction and a synopsis'. In: G. Rozenberg and A. Salomaa (eds.): *Handbook of Formal Languages - Volume 1: Word, Language and Grammar*. pp. 1–40, Springer.

Moore, R.: 1977, 'Reasoning about knowledge and action'. In: *IJCAI-5*. pp. 223–227.

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

Nonnengart, A.: 1996, 'Resolution-based calculi for modal and temporal logics'. In: M. McRobbie and J. Slaney (eds.): *13th Conference on Automated Deduction*, Vol. 1104 of *Lecture Notes in Artificial Intelligence*. pp. 599–612, Springer, Berlin.

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

Ohlbach, H., A. Nonnengart, M. de Rijke, and D. Gabbay: 2001, 'Encoding two-valued non-classical logics in classical logic'. In: A. Robinson and A. Voronkov (eds.): *Handbook of Automated Reasoning*. pp. 1403–1486, Elsevier Science Publishers B.V.

Orłowska, E.: 1988, 'Relational interpretation of modal logics'. In: H. Andréka, D. Monk, and I. Németi (eds.): *Algebraic logic. Colloquia Mathematica Societatis Janos Bolyai 54*. Amsterdam, pp. 443–471, North Holland.

Perrin, D.: 1990, 'Finite Automata'. In: J. V. Leeuwen (ed.): *Handbook of Theoretical Computer Science, Volume B, Formal models and semantics*. pp. 1–57, Elsevier.

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

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

Sahlqvist, H.: 1975, 'Completeness and correspondence in the first and second order semantics for modal logics'. In: S. Kanger (ed.): *3rd Scandinavian Logic Symposium, Uppsala, Sweden, 1973*. pp. 110–143, North Holland.

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

Schmidt, R. and U. Hustadt: 2004, 'A Principle for Incorporating Axioms into First-Order Translation of modal formulae'. Technical Report CSPP-22, The University of Manchester.

Spaan, E.: 1993, 'Complexity of Modal Logics'. Ph.D. thesis, ILLC, Amsterdam University.

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

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

Troelstra, A. and H. Schwichtenberg: 1996, *Basic Proof Theory*. Cambridge University Press.

Vakarelov, D.: 1987, 'Abstract characterization of some knowledge representation systems and the logic NIL of nondeterministic information'. In: P. Jorrand and V. Sgurev (eds.): *Artificial Intelligence: Methodology, Systems, Applications*. pp. 255–260, North-Holland, Amsterdam.

van Benthem, J.: 1976, 'Correspondence Theory'. Ph.D. thesis, Mathematical Institute, University of Amsterdam.

van Benthem, J.: 1984, 'Correspondence Theory'. In: D. Gabbay and F. Günthner (eds.): *Handbook of Philosophical Logic, Volume II*. pp. 167–247, Reidel, Dordrecht.

Verma, K.: 2003, 'Automates d'arbres bidirectionnels modulo théories équationnelles'. Ph.D. thesis, ENS de Cachan.

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