On one decidable generalized quantifier logic corresponding to a decidable fragment of first-order logic

Natasha Alechina

November 10, 1994

Abstract

Van Lambalgen (1990) proposed a translation from a language containing a generalized quantifier Q into a first-order language enriched with a family of predicates R_i, for every arity i (or an infinitary predicate R) which takes $Qx\phi(x, y_1, \ldots, y_n)$ to $\forall x(R(x, y_1, \ldots, y_n) \rightarrow \phi(x, y_1, \ldots, y_n))$ ($y_1, \ldots, y_n$ are precisely the free variables of $Qx\phi$). The logic of Q (without ordinary quantifiers) corresponds therefore to the fragment of first-order logic which contains only specially restricted quantification. We prove that it is decidable using the method of semantic tableaux. Similar results were obtained by Andreka and Nemeti (1994) using the methods of algebraic logic.

1 Introduction

Roughly speaking, in the history of modal logic modalities and their axioms came first and relational semantics giving them their precise meaning came second, bringing with it among other things the standard translation of modal logic into classical first-order logic. Although the translation from a generalized quantifier language into a first-order language considered in this paper reminds of the one for modal logic, the rest of the story is completely different. For generalized quantifiers, the meaning (defined in set theory) comes first, the axioms come (if at all) second, and the "standard translation" appears as a technical trick used to devise a Gentzen-style proof theory for some generalized quantifiers (see van Lambalgen (1991)). However as argued in van Benthem and Alechina (1993), the standard translation may have independent interest. We now explain these points in more detail.

A. Mostowski defined a generalized quantifier $\mathbf{Q}$ as a class of subsets of the domain, so that a model M satisfies $Qx\varphi(x, \bar{d})$ if the set of elements $\{e : M \models \varphi[e, \bar{d}]\}$ is in $\mathbf{Q}$. For example, $\mathbf{Q}$ may be the set of all uncountable subsets of the domain. Generalized quantifiers were studied for the most part from a model-theoretic point of view.

It is clear that the language containing generalized quantifiers is much more expressive than the ordinary first-order language. However, in some sense it can be reduced to a fragment of first-order language. Van Lambalgen (1991) proposed natural deduction systems for several generalized quantifiers (the filter quantifier and some of its extensions), where introduction and elimination rules for the quantifier have side conditions involving a dependency relation (between free variables). This dependency relation R comes from the following satisfiability preserving standard translation from a language containing a generalized quantifier Q into a first-order language containing R (I shall denote this language as $L(R)$):

$(P(x_1, \ldots, x_n))^* = P(x_1, \ldots, x_n);$

$(\neg\varphi)^* = \neg\varphi^*$;

$(\varphi \wedge \psi)^* = \varphi^* \wedge \psi^*$;

$(\forall x\varphi)^* = \forall x\varphi^*$;

$(Qx\varphi(x,\bar{y}))^* = \forall x(R(x,\bar{y}) \rightarrow \varphi^*(x,\bar{y}))$, where $\bar{y}$ are all the free variables of $Qx\varphi$.

To determine the side conditions, van Lambalgen shows that some quantifier axioms correspond to first-order conditions on R. This correspondence is studied systematically in van Benthem & Alechina (1993) and Alechina & van Lambalgen (1994). The standard translation seems to be a useful technical tool for studying generalized quantifiers.

Van Benthem proposed to look at the standard translation above as corresponding to an alternative, "modal" semantics for generalized quantifiers. This semantics is described in detail in the following section. The idea behind it (and it is this which makes the corresponding restricted fragment of the first order logic interesting) is as follows. For ordinary quantifiers $\forall$ and $\exists$ the domain of quantification does not depend on an assignment to individual variables or on the variables which are free in the quantifier formula. Classical quantifiers have what Blackburn & Seligman in this workshop call an "external" view of the domain, i.e. they always can "see" all objects which live there. On the contrary, "local" quantification depends on the view which one can have from a certain point inside the model. A good example of such local quantification is a modal operator: it quantifies over the worlds which are seen from a given world. To make quantifiers behave in the same way, i.e. to make first-order quantification local, one can define an accessibility relation between assignments (as is done in van Benthem (1994)) or between finite sequences of elements and elements (as the standard translation suggests). In the latter case, the domain of quantification is determined by the parameters of the formula: $\Diamond_x\varphi(x,\bar{d})$ is true if there is an object d accessible from $\bar{d}$ such that $\varphi[d,\bar{d}]$ is true. The theory of such quantification has a lot in common with modal logic: for example, we shall see that it is decidable. That is why the present paper gives one more answer to the question treated in Nemeti (1993): how to weaken standard predicate logic to make it decidable?

2 Language, Models, Semantic Tableaus

$L_\Diamond$ is a language which contains predicate symbols, individual variables and a generalized quantifier $\Diamond$ (no ordinary quantifiers and no equality are present). If φ is a w.f.f, so is $\Diamond_x\varphi$. $\Box_x$ (the universal-type generalized quantifier denoted by Q above) is defined as a dual of $\Diamond_x$.

A *model* for $L_\Diamond$ is a structure of the form $M =< D, R, V >$, where D is a domain, R is a relation of indefinite arity (or a family of relations R_i for every arity i) over D, and V a valuation (a function assigning n-ary predicate symbols subsets of D^n). The truth conditions for atomic formulas, negation and conjunction are standard.

We assume that the individual variables are ordered; $\varphi(y_1,\ldots,y_n)$ denotes a formula with precisely n different free variables ordered as displayed. The truth definition for $\Diamond$ reads as follows:

$$M \models^\alpha \Diamond_x\varphi(x,y_1,\ldots,y_n) \Leftrightarrow \exists\alpha' =_x \alpha(R(\alpha'(x),\alpha'(y_1),\ldots,\alpha'(y_n))\wedge M \models^{\alpha'} \varphi(x,y_1,\ldots,y_n)).$$

A formula is satisfiable, if there is a model and a variable assignment under which it is true. A formula is valid if its negation is not satisfiable.

Note that a formula of $L_\diamond$, φ, is valid (satisfiable) if its translation into first-order logic φ^* is valid (satisfiable).

Let us call the logic resulting from this notion of validity L^-_{min}. (The minus sign stands for the absence of ordinary quantifiers).

To check effectively whether a formula of L^-_{min} is satisfiable, we introduce *semantic tableaux* for L^-_{min}. A very clear description of the method can be found in Smullyan (1968). The calculus described here is slightly different in form but the same in all essential aspects.

Every formula is *signed* by T or F (intuitively standing for truth and falsity). For every connective, there is a pair of rules for decomposing a signed formula with this connective as a principal connective: one rule for a formula signed with T and one rule for a formula signed with F. They correspond to the truth conditions for this connective. For example, the rules for $\neg$ are as follows (Δ is a set of signed formulas):

$T_\neg$

$$\frac{\Delta, T\neg\varphi}{\Delta, F\varphi}$$

$F_\neg$

$$\frac{\Delta, F\neg\varphi}{\Delta, T\varphi}$$

The T-rule for conjunction $\varphi \wedge \psi$ is as it has to be, the F-rule requires the tableau to split in two columns: one containing Δ and $F\varphi$, the other Δ and $F\psi$.

To give rules for $\diamond$ we need to introduce, in addition to signed formulas, atomic formulas keeping track of the dependency relation, of the form $R(x, \bar{y})$. Now we assume that Δ (also in the rules above) is a union of a set of signed formulas Σ and a set of R-formulas Γ.

$T_\diamond$

$$\frac{\Delta, T\diamond_x\varphi(x, \bar{y})}{\Delta, R(z, \bar{y}), T\varphi(z, \bar{y}),}$$

where z is a new variable (not occurring in Δ);

$F_\diamond$

$$\frac{\Delta, F\diamond_x\varphi(x, \bar{y})}{\Delta, F\varphi(z, \bar{y}), (F\diamond_x\varphi(x, \bar{y})),}$$

for every variable z such that $R(z, \bar{y}) \in \Delta$.

This means that the rule can be used only if previously some variable z with $R(z, \bar{y})$ was introduced by $T_\diamond$. $F\diamond_x\varphi(x, \bar{y})$ in brackets in the conclusion of the rule means that the application of the rule should be repeated if a new variable u with $R(u, \bar{y})$ is introduced.

A column is called *closed* if the same formula occurs both under T and under F. A column is open if this is not the case and no rule can be applied any more. A tableau is closed if all its columns are closed. φ is derivable from Σ if a tableau for $T\Sigma, F\varphi$ is closed.

Theorem 1 *The tableau calculus described above is sound and complete for L^-_{min}.*

Proof. The proof is analogous to the proofs in Smullyan (1968). To demonstrate consistency, we must show that for every rule, if a set in the premise is satisfiable (i.e. the formulas signed with T are true and the formulas signed with F are false in some model M under some assignment α), then the set in the conclusion is satisfiable. For the $T_\diamond$ rule, notice that if $\Delta \cup \{\diamond_x \varphi(x, \bar{y})\}$ is satisfiable in M, then there is an element a with $R(a, \alpha(\bar{y}))$ such that for $\alpha' = \alpha[x/a]$ $M \models^{\alpha'} \varphi(x, \bar{y})$. Therefore there is a model and an assignment α' such that the conclusion of the rule is true.

To prove completeness, define a Hintikka set for (a set of variables) D and accessibility relation R between the variables as a set of signed formulas Σ such that

for no formula φ both $T\varphi$ and $F\varphi$ are in Σ;

if $T\neg\varphi \in \Sigma$, then $F\varphi \in \Sigma$, dually for $F\neg\varphi$;

if $T\varphi \wedge \psi \in \Sigma$, then $T\varphi, T\psi \in \Sigma$; if $F\varphi \wedge \psi \in \Sigma$, then either $F\varphi \in \Sigma$ or $F\psi \in \Sigma$;

if $T\diamond_x\varphi(x, \bar{y}) \in \Sigma$, then $T\varphi(z, \bar{y}) \in \Sigma$ for at least one $z \in D$ with $R(z, \bar{y})$;

if $F\diamond_x\varphi(x, \bar{y}) \in \Sigma$, then $F\varphi(z, \bar{y}) \in \Sigma$ for all z such that $R(z, \bar{y})$ (may be none such z).

It is easy to check that every Hintikka set for D and R is satisfiable. A satisfying model has domain D, accessibility relation R, the assignment is defined by $\alpha(x) = x$ and the valuation is any valuation satifying

$$TP(x_1, \ldots, x_n) \in \Sigma \Rightarrow < x_1, \ldots, x_n > \in V(P)$$

and

$$FP(x_1, \ldots, x_n) \in \Sigma \Rightarrow < x_1, \ldots, x_n > \notin V(P).$$

Now notice that any open column of a tableau forms a Hintikka set. (Actually, as it is shown in Smullyan (1968), in case when a column is infinite one has to do a bit more work. But Lemma 6 below shows that a tableau construction always stops after a finite number of steps). $\Box$

3 Decidability

We show that a tableau construction for formulas in *normal form* always stops. Then we prove that every formula has an equivalent in normal form.

Let us say that a subformula φ is *immediately in the scope of* a quantifier $\diamond_x$ if it is in the scope of $\diamond_x$ and there is no quantifier $\diamond_y$ "in between", that is, such that $\diamond_y$ is in the scope of $\diamond_x$ and φ is in the scope of $\diamond_y$.

Definition 1 *A formula χ is in* normal form *if in χ every subformula immediately in the scope of a quantifier contains the quantified variable of this quantifier free.*

For example, $\diamond_y\diamond_x(P(x, y) \wedge \diamond_z S(x, z))$ is in normal form, and $\diamond_x(P(x) \wedge \diamond_z S(z))$ is not (since $\diamond_z S(z)$ is immediately in the scope of $\diamond_x$ and does not have x free).

Lemma 1 *If a formula is in normal form, then each of its subformulas is.*

Proof. Obvious. $\square$

We are going to show that for a formula in normal form a tableau is always finite. This is not true for an arbitrary formula. Consider a tableau for

$$T(\Diamond_x Q(x) \wedge \neg\Diamond_x(P(x) \vee \neg\Diamond_z S(z)))$$

$T_{\wedge,\neg}$ $T\Diamond_x Q(x), F\Diamond_x(P(x) \vee \neg\Diamond_z S(z))$

$T_\Diamond$ $R(d), TQ(d), F\Diamond_x(P(x) \vee \neg\Diamond_z S(z))$

$F_\Diamond$ $R(d), TQ(d), F(P(d) \vee \neg\Diamond_z S(z)), (F\Diamond_x(P(x) \vee \neg\Diamond_z S(z)))$

$F_{\vee\neg}$ $R(d), TQ(d), FP(d), T\Diamond_z S(z), (F\Diamond_x(P(x) \vee \neg\Diamond_z S(z)))$

$T_\Diamond$ $R(d), TQ(d), FP(d), R(e), TS(e)(F\Diamond_x(P(x) \vee \neg\Diamond_z S(z)))$

$F_\Diamond$ $\ldots R(e), F(P(e) \vee \neg\Diamond_z S(z))(F\Diamond_x(P(x) \vee \neg\Diamond_z S(z)))$

$F_{\vee\neg}$ $\ldots T\Diamond_z S(z)(F\Diamond_x(P(x) \vee \neg\Diamond_z S(z)))$

$T_\Diamond$ $R(e_1), TS(e_1), (F\Diamond_x(P(x) \vee \neg\Diamond_z S(z)))$

$F_\Diamond$ $\ldots$

Here the tableau construction starts to loop. Note that $F_\Diamond$ is the rule responsible for that: it does not decrease the complexity of the formula.

To prove that a tableau construction always stops for formulas in normal form, we need to define a notion of *dependency between variables.* This notion appears in Fine [1985]. In the present context it has the following meaning:

Definition 2 *A variable x* depends *on a variable y in a given tableau, if in this tableau either $R(x,\ldots y\ldots)$ holds or (b) there is a variable z such that $R(x,\ldots z\ldots)$ holds and z depends on y.*

For example, in the following tableau:

$$T\Diamond_x(S(x,y) \wedge \Diamond_z P(x,z))$$

$$T_\Diamond, T_\wedge \quad R(x,y), TS(x,y), T\Diamond_z P(x,z)$$

$$T_\Diamond \quad R(x,y), R(z,x), TS(x,y), TP(x,z)$$

x depends on y and z depends on x and y.

Dependency in a tableau has some obvious properties. Transitivity follows from the definition.

Lemma 2 *If x depends on y, then y appeared free in the tableau before x was introduced.*

Proof. Assume that x depends on y. If $R(x,\ldots y\ldots)$ holds, then x was introduced by $T_\Diamond$ rule with a premise which had y free. By definition, x is a new variable, therefore there was a stage in the tableau construction (immediately before the rule was applied) when y occurred free in the tableau and x did not.

Assume that x depends on y because there are $z_1, \ldots, z_n$ such that $R(x, \ldots z_1 \ldots), \ldots, R(z_n, \ldots y \ldots)$. As before, this means that z_n appeared after y, ..., z_1 after z_2, and x after z_1. Thus x appeared after y. $\square$

Lemma 2 implies that dependency is asymmetric: if x depends on y, then y does not depend on x. It also implies that x cannot depend on a variable which was introduced later in the tableau.

If a formula is in normal form, its nested quantifiers are "hooked" into one another: if $\Diamond_z \varphi$ is immediately in the scope of $\Diamond_y$, then y is free in φ. This means that $\Diamond_z$ will be instantiated on a variable dependent on y. More generally,

Lemma 3 *Any result of decomposing in accordance with the tableau rules of a formula $\Diamond_x \chi$ in normal form will contain a variable dependent on the variable used to instantiate $\Diamond_x$, or this variable itself.*

Proof. Assume that φ is a subformula of $\Diamond_x \chi$ in normal form which is in the scope of quantifiers $\Diamond_x, \Diamond_{x_1}, \ldots, \Diamond_{x_n}$ ($\Diamond_x$ being the outermost). If $n = 0$ (φ is immediately in the scope of $\Diamond_x$), then φ contains x free (since χ is in normal form) and therefore the variable used to instantiate $\Diamond_x$ will be free in φ.

Let $n \geq 1$, and $T(F)\varphi(d_1, \ldots, d_m)$ be the result of decomposing χ according to the tableau rules. We show that at least one of the d_i depends on the variable used to instantiate $\Diamond_x$.

For the sake of simplicity, we assume that $\Diamond_{x_i}$ is instantiated by x_i. By ψ_i we denote the biggest subformula of χ in the scope of $\Diamond_{x_i}$. Note that φ is a subformula of ψ_n, and x_n is free in φ (since χ is in normal form). Since $\Diamond_{x_1}\psi_1$ is immediately in the scope of $\Diamond_x$, x is free in ψ_1, etc. Therefore for the variables used to instantiate the quantifiers holds

$$R(x_1, \ldots x \ldots), R(x_2, \ldots x_1 \ldots), R(x_3, \ldots x_2 \ldots), \ldots, R(x_n, \ldots x_{n-1} \ldots)$$

and by transitivity x_n depends on x. But x_n is free in φ. Thus, $\varphi(d_1, \ldots, d_m)$ contains at least one variable dependent on x. $\square$

Lemma 4 *Assume that a tableau construction for a formula χ in normal form has reached a stage when the only applicable rule is $T_\Diamond$. Let $z_1, \ldots, z_n$ be the list of all variables which occur free in the tableau at this stage. Any formula which appears later in the tableau will contain at least one free variable which is not among $z_1, \ldots, z_n$.*

Proof. If the only applicable rule is $T_\Diamond$, then every formula in the tableau at this stage is either atomic, or begins with $T\Diamond_x$ or $F\Diamond_x$. Since χ is in normal form, every formula in the tableau is (Lemma 1). Any result of decomposing a formula in normal form beginning with $\Diamond_x$ will contain free a variable which depends on the variable used to instantiate $\Diamond_x$, or this variable itself (Lemma 3). This variable (used to instantiate $\Diamond_x$) is not among $z_1, \ldots, z_n$ since it will be introduced by $T_\Diamond$ rule at a later stage (we have assumed that the $F_\Diamond$ rule was no more applicable, therefore even applications of the $F_\Diamond$ rule to the formulas beginning with $F\Diamond_x$ will use a variable not among $z_1, \ldots z_n$). Since neither of z_i can depend on a variable introduced later in the tableau (Lemma 2), every result of decomposition will contain a variable which is not among $z_1, \ldots, z_n$. $\square$

Lemma 5 *Assume that a tableau construction for a formula χ in normal form has reached a stage where the only applicable rule is $T_\Diamond$. Let $F\Diamond_x\psi_1, \ldots, F\Diamond_x\psi_n$ be the list of all formulas in the tableau beginning with $F\Diamond$. For every $F\Diamond_x\psi_i$, the $F_\Diamond$ rule with this formula as a premise will be applied only finitely many times.*

Proof. Let $\bar{z}$ be the free variables of $\Diamond_x\psi_i$. The $F_\Diamond$ rule with $F\Diamond_x\psi_i(x,\bar{z})$ as a premise can be repeated only if a new variable d with $R(d,\bar{z})$ is introduced by the $T_\Diamond$ rule with a formula of the form $T\Diamond_x\alpha(x,\bar{z})$ as a premise. Assume that there are m such formulas present in the tableau at this stage. No formula with $\bar{z}$ as its only free variables will appear later in the tableau by Lemma 4. Therefore the $F_\Diamond$ rule with $F\Diamond_x\psi_i(x,\bar{z})$ as a premise will be applied precisely m more times. □

Lemma 6 *If χ is in normal form, the tableau construction for χ always stops.*

Proof. Take an arbitrary column of a tableau. Perform all rules for the Booleans and all $F_\Diamond$ rules with respect to previously introduced variables. One can show that at any stage of constructing a tableau this process stops after a finite number of steps. At some stage we cannot procede any further without applying a $T_\Diamond$ rule. Assume that at this stage the formulas in the brackets (which give rise to repetitions of $F_\Diamond$ rules) are

$$F\Diamond_{x_1}\theta_1, \ldots, F\Diamond_{x_k}\theta_k$$

Due to Lemma 5, for every $\Diamond_{x_i}\theta_i$ the rule $F_\Diamond$ with this formula as a premise will be repeated only finitely many times.

In a finite number of steps the tableau construction reaches the stage when $F_\Diamond$ will not be repeated with $F\Diamond_x\theta_i$ as a premise, and again only the $T_\Diamond$ rule is applicable. Assume that at this point the tableau contains atomic formulas, formulas beginning with $T\Diamond_x$, new formulas beginning with $F\Diamond$, $F\Diamond_x\psi_1, \ldots, F\Diamond_x\psi_m$, and $\{F\Diamond_x\theta_i : 1 \leq i \leq k\}$. The complexity of all formulas in the tableau except for $\{F\Diamond_x\theta_i : 1 \leq i \leq k\}$ is decreased. For each of the $F\Diamond_x\psi_j$ the $F_\Diamond$ rule will be applied finitely many times. After that the complexity of all formulas in the tableau except for the ones which won't be used any more is again decreased. An easy induction shows that the tableau construction will stop after a finite number of steps. □

Consequence 1 *The satisfiability problem for the formulas in normal form is decidable.*

Lemma 7 *Every formula has an equivalent in normal form.*

Proof. Take an arbitrary formula χ. Let us call the quantifiers which have in their immediate scope subformulas not containing the quantified variable free *quasivacuous.* Our aim is to eliminate all quasivacuous quantifiers. The proof that this is always possible goes by induction on the maximal quantifier depth a quasivacuous quantifier. (The quantifier depth of a *subformula* is the number of quantifiers in whose scope this subformula is situated. By the quantifier depth of a *quantifier* I mean the quantifier depth of the subformula which begins with this quantifier.)

For example, in $\Diamond_x(P(x) \wedge \Diamond_z S(z))$ $\Diamond_x$ is quasivacuous, because $\Diamond_z S(z)$ is immediately in its scope, and it does not contain x free. The quantifier depth of $\Diamond_x$ in this formula equals 0. In $\Diamond_x(P(x) \wedge \Diamond_y(Q(x,y) \wedge \Diamond_z S(z)))$ $\Diamond_x$ is not quasivacuous, since $\Diamond_z S(z)$ is not immediately in the scope of $\Diamond_x$: there is $\Diamond_y$ in between. $\Diamond_y$ is quasivacuous and its quantifier depth is 1.

Basis. If the quantifier depth of all quasivacuous quantifiers in χ equals 0, then the procedure described below (taking the subformulas, not having the quantified variable free, out of the scope of the quantifier) brings χ in normal form.

Inductive step. We show that the maximal quantifier depth of a quasivacuous quantifier in χ can be reduced by one.

Fix a quasivacuous quantifier $\Diamond_x$ with maximal depth (not having in its scope other quasivacuous quantifiers). Denote the biggest subformula of χ in the scope of $\Diamond_x$ as φ. We give an algorithm for finding an equivalent of $\Diamond_x\varphi$, to be called φ', with the the same free variables, such that in φ' there are no more subformulas immediately in the scope of $\Diamond_x$ which do not have x free, and the quantifiers in the scope of $\Diamond_x$ are still not quasivacuous. Note that $\chi \equiv \chi[\Diamond_x\varphi/\varphi']$.

If x is not free in φ, apply the following derivable equivalence:

$$\Diamond_x\varphi(\bar{z}) \equiv \Diamond_x\top(x,\bar{z}) \wedge \varphi(\bar{z})$$

(where $\bar{z}$ are all the free variables of φ, and $x \notin \bar{z}$).

Otherwise write φ as a disjunction of conjunctions, where each conjunct is either an atomic formula or its negation, or a formula beginning with a quantifier or its negation. Denote this disjunction as $\theta_1 \vee \ldots \vee \theta_n$. Obviously,

$$\Diamond_x\varphi \equiv \Diamond_x(\bigvee_i \theta_i),$$

$1 \leq i \leq n$. Note that the θ's may have different free variables, therefore the distributivity property

$$\Diamond_x(\bigvee_i \theta_i) \equiv \bigvee_i \Diamond_x\theta_i$$

does not necessarily hold. But if the free variables of φ are $x, \bar{y}$, then the following holds:

$$\Diamond_x(\bigvee_i \theta_i) \equiv \Diamond_x(\bigvee_i (\theta_i \wedge \top(x,\bar{y}))) \equiv \bigvee_i \Diamond_x(\theta_i \wedge \top(x,\bar{y}))$$

In each disjunct θ_i, denote the conjunction of subformulas having x free (including $\top(x,\bar{y})$) as $\psi_{i1}(x)$ and the conjunction of formulas not having x free as ψ_{i2}. Our aim is to find an equivalent of

$$\bigvee_i \Diamond_x(\psi_{i1}(x) \wedge \psi_{i2})$$

where no ψ_{i2} is in the scope of $\Diamond_x$. To do that, apply

$$\Diamond_x(\varphi(x,\bar{y}) \wedge \psi(\bar{z})) \equiv \Diamond_x(\varphi(x,\bar{y}) \wedge \top(x,\bar{z})) \wedge \psi(\bar{z}),$$

given that x is not free in ψ.

The result of applying this derivability is the desired formula φ'. Observe that since we have not changed the subformulas under the quantifiers in the scope of $\Diamond_x$, there are still no quasivacuous quantifiers in the scope of $\Diamond_x$. The result of substituting φ' in χ instead of $\Diamond_x\varphi$ gives an equivalent formula with maximal depth of a quasivacuous quantifier reduced by 1.

Note that this procedure can create new quasivacuous quantifiers, but with a smaller depth. Consider the example above:

$$\Diamond_x(P(x) \wedge \Diamond_y(Q(x,y) \wedge \Diamond_z S(z))).$$

In this formula $\Diamond_y$ is a quasivacuous quantifier with depth 1. After applying the algorithm to $\Diamond_y$ this formula becomes

$$\Diamond_x(P(x) \wedge \Diamond_y Q(x,y) \wedge \Diamond_z S(z))$$

and $\Diamond_x$ is now a quasivacuous quantifier; but it has depth 0. After repeating the procedure for $\Diamond_x$, we obtain

$$\Diamond_x(P(x) \wedge \Diamond_y Q(x,y)) \wedge \Diamond_z S(z)$$

which is in normal form. $\square$

Consequence 2 L^-_{min} *has finite model property.*

Proof. Every formula φ of L^-_{min} has an equivalent φ' in normal form, and if φ is satisfiable, then there is a *finite* open column of a tableau for $F\varphi'$. But then a satisfying model for φ' (and therefore for φ) as constructed in the proof of the Theorem 1 is finite.

Theorem 2 L^-_{min} *is decidable.*

Proof. The theorem follows from Lemmata 6 and 7 and Theorem 1. $\square$

Consequence 3 *The fragment of first-order logic which is the image of* $L_\Diamond$ *under the standard translation is decidable.*

4 Extensions

4.1 Adding axioms: two cases

In this section we look at some extensions of the minimal logic to check if decidability is preserved, or rather, if the present decidability proof is still applicable.

Negative case. As it was mentioned before, in the minimal logic only restricted monotonicity holds, namely, if $\vdash \varphi \rightarrow \psi$ *and* $\Diamond_x\varphi$ and $\Diamond_x\psi$ have the same free variables, then

$$\vdash \Diamond_x\varphi \rightarrow \Diamond_x\psi.$$

The present decidability proof is not applicable if we weaken the restriction to $FV(\Diamond_x\psi) \subseteq FV(\Diamond_x\varphi)$, which is equivalent to accepting an axiom

$$\Diamond_x(\varphi \wedge \psi) \rightarrow \Diamond_x\psi.$$

The logic which is obtained from the minimal logic by adding this axiom is complete with respect to the class of models where R has the following property:

$$R(x, \bar{y}\bar{z}) \rightarrow R(x, \bar{y})$$

(see, for example, Alechina and van Lambalgen (1994)). The corresponding adaptation of the tableau rule $T_\Diamond$ would be to add all formulas of the form $R(x, \bar{y})$, $\bar{y} \subseteq \bar{z}$, when introducing a new variable x with $R(x, \bar{z})$. But then cycles in the tableaux become possible again, even for formulas in normal form. To be more precise, Lemma 5 does not hold any more.

The same holds for adding unrestricted monotonicity: the present decidability proof does not work. It is plausible that the minimal logic extended with unrestricted monotonicity is undecidable, but the question is still open.

Positive case. Another extension is decidable, which can be shown by a slight modification of the present proof. Adding $\Diamond_x\top(x, \bar{y})$ as an axiom corresponds to the condition that R is non-empty: $\exists x R(x, \bar{y})$. This can be easily shown to be equivalent to the following modification of the $F_\Diamond$ rule:

$F_\diamond$

$$\frac{\Delta, F\diamond_x \varphi(x,\bar{y})}{\Delta, F\varphi(z,\bar{y}), (F\diamond_x\varphi(x,\bar{y})),}$$

for every variable z such that $R(z,\bar{y}) \in \Delta$; *if there is no such variable, add a new variable u with $R(u,\bar{y})$ and $F\varphi(u,\bar{y})$.*

The decidability proof is precisely the same, only in the proof of Lemma 5 the last sentence has to be changed from "Therefore the $F_\diamond$ rule with $F\diamond_x\psi_i(x,\bar{z})$ as a premise will be applied precisely m more times" to "Therefore the $F_\diamond$ rule with $F\diamond_x\psi_i(x,\bar{z})$ as a premise will be applied at most $m+1$ times".

However, although a tableau for a formula in normal form is always finite, a model which is to be built given an open column of this tableau is infinite. The construction goes as follows: all free variables which are present in this column become elements of the domain. R should obey the restrictions which are imposed by the R-formulas in the tableau, plus $\forall\bar{y}\exists x R(x,\bar{y})$. Therefore, to complete the model, for every sequence of elements $\bar{z}$ in the tableau for which no element u with $R(u,\bar{z})$ is present, we should introduce such an element. The modified $F_\diamond$ rule takes care that there is no formula beginning with $F\diamond_x$ such that its free variables are $\bar{z}$, so this will not start an application of the rule. Now the domain has more elements, and there are more possible sequences of elements. For every such sequence introduce a new element in the relation R to it, etc. Every step in this construction is justified by the fact that Γ has an open tableau if and only if $\Gamma \cup \{T\diamond_x\top(x,\bar{u})\}$ does. To be more precise, the construction goes as follows:

$\Gamma_0 = \Gamma$, where Γ is a set of all formulas in an open column of a tableau with no more rules applicable;

$\vdots$

Γ_n is the result of applying $T_\diamond$ rules to $\Gamma_{n-1} \cup \{T\diamond_x\top(x,\bar{z})$: $\bar{z}$ occurs in Γ_{n-1} but no formula of the form $R(u,\bar{z})$ does $\}$. As we argued above, the $F_\diamond$ rule does not become applicable;

$\vdots$

The union of these sets is still a Hintikka set containing infinitely many variables. The resulting model is infinite. (Note that in principle there can be finite models for $L^-_{min} + \diamond_x\top$: for example, a one-element models with $D = \{a\}$ and $R(a,\bar{a})$ for every tuple $\bar{a}$; but on the basis of a tableau we can construct only an infinite one, since by the $T_\diamond$ rule we may add only *new* elements. The situation is similar to the one in first-order logic.)

4.2 The minimal logic with equality

A natural question to ask is if the minimal logic with equality is decidable. I don't have the answer yet. The present decidability proof does not work; the logic can as well be decidable. Adding one more axiom, namely $\neg\diamond_x(x = y \wedge \top(\bar{z}))$ (which corresponds to $\neg R(x, \ldots x \ldots)$), yields a decidable system.

There are several ways to introduce equality in semantic tableaux. Beth (1962) adds the following axioms, with all variables universally quantified:

$x = x$;

$x = y \to y = x$;

$x = y \wedge y = z \to x = z$;

$x = y \to (\varphi(x) \to \varphi(y))$, where φ may contain more free variables.

It is not difficult to show that the minimal logic plus these axioms (for all x, y and z) is equivalent to the system obtained by adding the following rule to the minimal logic (given that x is free for y and y is free for x in all formulas, which can be achieved by using different symbols for bound and free variables):

$T_=$

$$\frac{\Gamma, Tx = y}{\Gamma[x/y], \Gamma[y/x]}$$

where $\Gamma[x/y]$ denotes the result of substituting x for y everywhere in Γ, analogously for $\Gamma[y/x]$,

plus the condition that if for some x $Fx = x$ occurs in a column, then this column is closed. (I really don't know where I have seen this formulation, but it's unlikely that I have made it up all myself).

Lemma 2 does not hold for this system. Consider a tableau for $T\Diamond_x(x = y)$. As a result of $T_\Diamond$ we get $R(u, y)$, $Tu = y$, and as a result of $T_=$ we get $R(y, y)$, i.e. according to the lemma y must have appeared free in the tableau before y was introduced.

Even if a formula is in normal form, it can now have an infinite tableau. Consider a tableau for

$$F(\neg\Diamond_x \top(x, y) \vee \Diamond_x(x \neq y \vee \neg\Diamond_z P(x, y, z))).$$

At some point we come to

$$R(u, y),\ Tu = y,\ T\Diamond_z P(u, y, z)\ (F\Diamond_x(x \neq y \vee \neg\Diamond_z P(x, y, z))),$$

and by $T_\Diamond$ to

$$R(u, y),\ Tu = y,\ R(v, yu),\ TP(u, y, v)\ (F\Diamond_x(x \neq y \vee \neg\Diamond_z P(x, y, z))).$$

After applying the $T_=$ rule we get along with other things $R(y, y)$, and the $F_\Diamond$ rule can be activated again. The result is

$$Ty = y,\ T\Diamond_z P(y, y, z)\ (F\Diamond_x(x \neq y \vee \neg\Diamond_z P(x, y, z)))$$

and by the $T_\Diamond$ rule applied to $T\Diamond_z P(y, y, z)$,

$$R(u_1, y),\ P(y, y, u_1).$$

Now the cycle can start again with u_1 in place of u.

However, the following holds:

Theorem 3 *L^-_{min} with equality and additional axiom $\neg\Diamond_x(x = y \wedge \top(\bar{z}))$ is decidable.*

Proof. It is easy to check that the axiom implies that every formula φ is equivalent to a formula which does not contain positive occurrences of equality in the scope of $\Diamond_x$. (Namely, if the positive occurrence is of the form $x = y$, with x the quantifier variable, this follows from the axiom; if it is of the form $y = z$, this follows from the normal form lemma.) Consider a tableau for such a formula. Since there are no positive occurrences of equality under $\Diamond_x$, for no new variable z introduced by the $T_\Diamond$ rule $Tz = y$ occurs in the tableau. Note that we have a formulation of our system where the $T_=$ rule is the only rule for equality.

This allows us to divide the construction of the tableau in two parts, the first one containing no applications of the quantifier rules and all applications of the $T_=$ rule, and the second one containing no applications of the $T_=$ rule. The construction terminates by Lemma 6. $\square$

Andreka and Nemeti (1994) gave a suggestion of how the mosaics method applied in Nemeti (1993) can be used to prove decidability of L^-_{min} with equality. This means that although the tableaux fail to provide a decidability proof, the minimal logic with equality must be decidable.

5 Conclusion

The image of $L_\Diamond$ under the standard translation can be viewed as a "modal fragment" of first-order logic (by the way, it includes the standard translations of ordinary modal formulas, as it is shown in van Benthem and Alechina (1993)). The interest of such fragments can be seen in their departure from the standard quantification (local vs global, generalized dependencies between variables vs ordinary Scolem functions) which can shed light on quantification theory in general.

6 References

N. Alechina & M. van Lambalgen (1994). Correspondence and completeness for generalized quantifiers. *ILLC Technical Note X-94-03,* Univ. of Amsterdam.

H. Andreka & I. Nemeti (1994). Private communication.

J.F.A.K. van Benthem & N. Alechina (1993). Modal quantification over structured domains. *ILLC Report ML-93-02,* Univ. of Amsterdam. (To appear in M. de Rijke, ed., *Second Yearbook of Modal Logic.)*

J.F.A.K. van Benthem (1994). Modal state semantics. *Manuscript,* ILLC, Univ. of Amsterdam.

E. Beth (1962). Formal methods. An introduction to symbolic logic and to the study of effective operations in arithmetic and logic. D. Reidel Publishing Company, Dordrecht.

K. Fine (1985). Natural deduction and arbitrary objects. *J. Phil. Logic,* **14**(1), 57 – 107.

M. van Lambalgen (1990). The axiomatization of randomness. *J. Symb. Logic,* **55**, 1143–1167.

M. van Lambalgen (1991). Natural deduction for generalized quantifiers, in J. van der Does & J. van Eijck (eds.), *Generalized Quantifier Theory and Applications,* Dutch Network for Language, Logic and Information, Amsterdam, 143-154. (To appear as CSLI Lecture Notes.)

I. Nemeti (1993) Decidability of weakened versions of first order logic. *Workshop Logic at Work, Amsterdam.*

R.M.Smullyan (1968) First-order logic. Springer-Verlag, Berlin etc.