On modal μ-calculus in $S5$ and applications

Giovanna D'Agostino[1] and Giacomo Lenzi[2]

[1] University of Udine, Italy
[2] University of Salerno, Italy gilenzi@unisa.it

Abstract. We show that the vectorial μ-calculus model checking problem over arbitrary graphs reduces to the vectorial, existential μ-calculus model checking problem over $S5$ graphs. We also draw some consequences of this fact. Moreover, we give a proof that satisfiability of μ-calculus in $S5$ is NP-complete, and by using $S5$ graphs we give a new proof that the satisfiability problem of the existential μ-calculus is also NP-complete.

1 Introduction

Model checking is a technique widely used in verification of computer systems, be they hardware or software, see [4]. In model checking, systems are modeled as sets with one or more binary relation (in this paper we focus on systems with one relation, i.e. graphs). The desirable properties a system should have are formalized in some modal-like logic. Actually, modal logic itself is not expressive enough. For this reason, one considers more powerful formalisms. One of them is *modal μ-calculus*, introduced in [15], an extension of modal logic with least and greatest fixpoints of monotonic set-theoretic functions. Intuitively, least fixpoints correspond to inductive definitions, and greatest fixpoints correspond to coinductive definitions. Unlike plain modal logic, the μ-calculus is powerful enough to express global properties of systems, i.e. properties which depend on the whole possible history of the system. For instance, with greatest fixpoints we can capture safety properties such as "the system will never crash", whereas with least fixpoints we can capture termination properties such as "every computation of the system will terminate". More complicate properties, e.g. fairness, can be used by combining least and greatest fixpoints.

The model checking technique raises a natural computational question, which is known as the (μ-calculus) model checking problem. Formally, the μ-calculus model checking problem is: given a μ-calculus formula and a finite graph, check whether the graph satisfies the formula. Because of the importance of model checking in practice, it would be desirable to have an efficient, i.e. polynomial time computable, model checking algorithm for arbitrary (finite) graphs, but this algorithm has not been found. We know that the problem is in the complexity class UP, see [14] (and a $co-UP$ bound follows since the μ-calculus is closed under negation). Recall that the class UP (Unique P) contains the problems solved in polynomial time by nondeterministic Turing machines which have at most one accepting path on each input. So, the class UP lies somewhere between P and NP (in particular, the model checking problem is in NP). We will see that the μ-calculus is tightly related to games, in particular parity games, and in fact a promising approach to the model checking problem is the study of various kinds of games. It must be said, however, that efficient model checking algorithms exist when the number of alternating fixpoints is bounded, and this is often the case in practice.

The other main theme of this paper is given by $S5$ graphs, i.e. graphs whose relation is an equivalence.

The modal logic of $S5$ graphs (also called modal logic $S5$) is important because it is widely recognized as a good epistemic logic, where the box operator $[\,]\phi$ means that some agent knows ϕ. When modal logic is interpreted on Kripke structures, i.e. graphs, the vertices of the structure represent possible situations, and it is reasonable that the knowledge of an agent is represented by an equivalence relation on the vertices, which indicates that certain situations are not distinguishable, in the agent's knowledge.

So, $S5$ is a way of formalizing the ideas of knowledge, and it is used in many applications such as artificial intelligence, etc. Often multimodal versions of $S5$ are considered, where different agents come into play; in this paper, however, we will focus on a single modality, representing a single agent.

We will consider also the class of all transitive graphs, called $K4$ in the modal logic literature. Many interesting relations are transitive: for instance, the relation "the event A is posterior to the event B" defines a transitive relation between events. In this paper $K4$ graphs play only a minor role; papers dedicated to the μ-calculus in $K4$ are [2], [5] and [6].

In this paper we compare the behavior of the μ-calculus on arbitrary graphs and on $S5$ graphs. It is well known that the μ-calculus is expressively equivalent to modal logic over $S5$, but this equivalence does not transfer automatically to an equivalence in complexity, neither for model checking, nor for satisfiability.

From this perspective we first show that the μ-calculus model checking problem for arbitrary graphs is as difficult as the subcase of $S5$ graphs, although the class of $S5$ graphs is significantly simpler than the class of all graphs.

Then we move to the satisfiability problem. Quite generally, recall that the satisfiability problem for a logic L on a class of models C is: given a formula ϕ in L, decide whether there is a model of ϕ which is in C.

The satisfiability problem of the μ-calculus on arbitrary graphs is settled, in the sense that it is $EXPTIME$-complete: $EXPTIME$-hardness of the problem follows from [12], and membership to $EXPTIME$ is proved in [8]. We note that $S5$ has also an application to the satisfiability problem of fragments of the μ-calculus: the satisfiability problem of the so-called existential (or box-free) μ-calculus on arbitrary graphs is as difficult as the same problem on $S5$ graphs. By using this observation we give an alternative proof of a result of [13] to the effect that the satisfiability problem for the existential μ-calculus is NP-complete. We also give a proof that satisfiability of μ-calculus in $S5$ is NP-complete, so in this respect we have a better complexity than the $EXPTIME$ complexity the full μ-calculus. Both results depend on a linear size model property for μ-calculus formulas in $S5$.

1.1 Related work

Given the relevance of $S5$ as epistemic logic, many papers in the modal logic literature are dedicated to it, and in particular on its proof theory. Finding a good axiom system for $S5$ is a longstanding open problem, see [19]. The situation is even more difficult for the modal μ-calculus in $S5$, where a recent contribution is [1].

2 Syntax

2.1 Scalar modal μ-calculus

We present here the usual modal μ-calculus, and we call it scalar because, as we will see, there is also a vectorial version of the μ-calculus. We follow the standard presentation of the formulas of modal μ-calculus:

$$\phi = A \mid \neg A \mid X \mid \phi \vee \phi \mid \phi \wedge \phi \mid \langle\ \rangle\phi \mid [\ ]\phi \mid \mu X.\phi \mid \nu X.\phi,$$

where A ranges over a set At of atoms and X ranges over a set Var of fixpoint variables. $\langle\ \rangle$ and $[\ ]$ denote the modal operators: the diamond, or the existential operator, and the box, or the universal operator.

Intuitively, $\mu X.\phi(X)$ denotes the least fixpoint of the function $\phi(X)$, and $\nu X.\phi(X)$ denotes the greatest fixpoint of this function.

A μ-calculus formula ϕ is called guarded if for every fixpoint subformula of ϕ, say $\nu X.\psi(X)$ or $\mu X.\psi(X)$, every occurrence of X in ψ is in the scope of a modal operator.

Free and bound variables are defined in analogy with first order logic, because fixpoints μX and νX are syntactically analogous to quantifiers $\exists x$ and $\forall x$ (note however that semantically, fixpoint variables correspond to monadic second order variables, i.e. variables ranging over sets, rarther than first order variables ranging over individuals).

A μ-calculus formula is called a sentence if it has no free variables. Although formulas are not closed under negation, a negation of sentences is available: the negation of a sentence is obtained by exchanging A and $\neg A$, $\wedge$ and $\vee$, $\langle\ \rangle$ and $[\ ]$, and μ with ν.

Given a formula ϕ, we denote by $|\phi|$ the size of ϕ.

2.2 Functional μ-calculus

We can generalize modal μ-calculus to functional μ-calculus, following [3]. Functional μ-calculus has n-ary function symbols, to be interpreted by monotonic functions on powersets (or more generally, on complete lattices). The syntax is

$$X|\phi \wedge \psi|\phi \vee \psi|f(\phi_1, \ldots, \phi_n)|\mu X.\phi|\nu X.\phi.$$

2.3 Vectorial μ-calculus

The most standard presentation of modal μ-calculus is in the scalar syntax of the previous section. In this section we generalize the syntax by allowing systems of equations: although this extension does not affect the expressiveness of the logic, it may increase succinctness.

We essentially follow the presentation of [3]. We restrict to powersets rather than arbitrary complete lattices. So we can consider a set V and n monotonic functions $f_1, \ldots, f_n$ from $P(V)^{n+m}$ to $P(V)$. A μ-system is a system S of n equations

$$S : \begin{cases} x_1 =_{\theta_1} f_1(x_1, \ldots, x_n, y_1, \ldots, y_m) \\ \qquad\qquad \ldots \\ x_n =_{\theta_n} f_n(x_1, \ldots, x_n, y_1, \ldots, y_m) \end{cases}$$

where $\theta_1, \ldots, \theta_n \in \{\mu, \nu\}$.

The μ-system S is by definition equivalent to a n tuple of scalar μ-calculus formulas, called the solution of S, computed inductively as follows.

If $n = 1$ then the solution is $\theta x_1.f_1(x_1, y_1, \ldots, y_m)$.

If $n > 1$, let $g_1(x_2, \ldots, x_n, y_1, \ldots, y_m) = \theta_1 x_1.f_1(x_1, \ldots, x_n, y_1, \ldots, y_m)$. The solution of S is $(g_1(h_2, \ldots, h_n, y_1, \ldots, y_m), h_2, \ldots, h_n)$, where $(h_2, \ldots, h_n)$ is the solution of the system

$$S_1 : \begin{cases} x_2 =_{\theta_2} f_2(g_1(x_2, \ldots, x_n, y_1, \ldots, y_m), \ldots, x_n, y_1, \ldots, y_m) \\ \qquad\qquad \ldots \\ x_n =_{\theta_n} f_n(g_1(x_2, \ldots, x_n, y_1, \ldots, y_m), \ldots, x_n, y_1, \ldots, y_m) \end{cases}$$

We denote by $sol_i(S)$ the i-th component of the solution of S.

A μ-system of equations is called a modal μ-system if all functions f_i are combinations of variables, atoms, negated atoms, conjunctions, disjunctions, diamonds, and boxes.

The modal μ-calculus vectorial model checking problem is: given a finite graph G and a modal μ-system S, decide whether G satisfies $sol_1(S)$.

2.4 The *LEFT* relation

Given a μ-system S, we define a relation $LEFT$ between the variables of S as follows.

Let y, z two variables of S. We say that y is at left of z, written $y \ LEFT \ z$, if there is an equation of S where y is the variable at the left hand side of the equality and z occurs at the right hand side.

A modal μ-system is called a modal system if the $LEFT$ relation on variables is acyclic. Every modal system is equivalent to a formula of modal logic.

2.5 Composition

Let $\phi(X)$ be a formula containing a free variable X and let ψ be a sentence. Then the composition $\phi[X/\psi]$ is the formula obtained by replacing X everywhere with ψ in ϕ. Note that ψ is a sentence, hence there is no variable capturing.

The usual notion of composition of formulas can be extended to μ-systems as follows.

Let S be a μ-system. The scope of a left variable y in S is the set of all variables z such that there is a $LEFT$ path from y to z.

Let S, T be two systems where the variables at left of S and T are disjoint. Let A be an atom of S. Then the composition of S and T is the system obtained by concatenating the equations of S and of T and by replacing A with the left variable of the first equation of T. Composition is possible only without capture, i.e. T must not have free variables y such that some occurrence of A is in the scope of y in S.

2.6 Vectorial alternation depth hierarchy

We define the vectorial hierarchies $VEC-\Sigma_n$, $VEC-\Pi_n$, $VEC-\Delta_n$ as follows. $VEC-\Sigma_0 = VEC-\Pi_0$ are the modal systems. $VEC-\Sigma_{n+1}$ is the closure of $VEC-\Pi_n$ under composition and adding a μ equation as a first equation of the system. $VEC-\Pi_{n+1}$ is the closure of $VEC-\Sigma_n$ under composition and adding a ν equation as a first equation of the system. $VEC-\Delta_n = VEC-\Sigma_n \cap VEC-\Pi_n$. The alternation depth of a system S is the least n such that S is in $VEC-\Delta_{n+1}$.

3 Semantics and related concepts

3.1 Graphs and models

Like it is usually done for modal logic, we give Kripke semantics to the μ-calculus by using the notion of model.

A graph (also called frame) is a pair $G = (V, E)$, where V is a set of vertices and E is a binary edge relation on V.

A graph $G = (V, E)$ is called total if $E = V^2$, i.e., all possible edges are present.

A path in a graph G from a vertex x to a vertex y is a finite sequence of vertices $z_1, \ldots, z_n$ such that $z_1 = x$, $z_n = y$ and $z_i E z_{i+1}$ for every $i < n$.

A point y is reachable from a point x if there is a path from x to y.

A model is a pair (G, Col), where G is a graph and Col is a coloring function from some domain D to the powerset of V.

3.2 Bisimulation

Intuitively, bisimulation between models indicates that the two models have the same observable behavior. Formally, we can define a bisimulation between two models (G, Col) and (G', Col') as a relation $B \subseteq V(G) \times V(G')$ such that, whenever (xBx'):

- $x \in Col(d)$ if and only if $x' \in Col'(d)$ for every $d \in D$;
- if xEy, then there is y' such that $x'E'y'$ and yBy';
- if $x'E'y'$, then there is y such that xEy and yBy'.

We say that two pointed colored graphs (G, Col, x) and (G', Col', x') are bisimilar if and only if there is a bisimulation B such that xBx'.

3.3 Special classes of graphs

We consider a few subclasses of graphs.

A graph (V, E) is called total if $E = V^2$, that is, all possible edges are present.

The class $K4$ is the class of all (vertex colored) graphs whose relation is transitive. The class $S5$ is the class of all (vertex colored) graphs whose relation is an equivalence relation. Since equivalence relations are reflexive, symmetric and transitive, $S5$ is included in $K4$ (as a class of graphs). The names $K4$ and $S5$ come from the modal logic literature.

We also speak of total models, $K4$ models and $S5$ models in the obvious sense.

Note that every total graph belongs to $S5$. Moreover, for every $S5$ model M and every vertex x of M, there is a total model M' containing x such that (M, x) and (M', x) are bisimilar: in fact M' is the submodel of M given by all points of M reachable from x.

Moreover, every $S5$ model M is bisimilar to a $S5$ graph M', colored in the same way, where every two different points have different colors: in fact, M' is M modulo the equivalence relation of having the same color, and the bisimulation is the projection function.

3.4 Semantics

The semantics of μ-calculus extends the usual Kripke semantics for modal logic. So, to give semantics to the μ-calculus, we must consider models of the form $M = (G, Val)$ where $G = (V, E)$ is a graph and Val is a valuation function from $At \cup Var$ to the powerset of V. To each model M and each formula ϕ we can associate a subset $||\phi||M$ of V, defined in this way:

- $||A||M = Val(A)$ and $||\neg A||M = V \setminus Val(A)$ if A is an atom;
- $||X||M = Val(X)$;
- $||\phi \vee \psi||M = ||\phi||M \cup ||\psi||M$;
- $||\phi \wedge \psi||M = ||\phi||M \cap ||\psi||M$;
- $||\langle\ \rangle\phi||M$ is the set of all elements of V having some successor in $||\phi||M$;
- $||[\ ]\phi||M$ is the set of all elements of V having every successor in $||\phi||M$;
- $||\mu X.\phi(X)||M$ is the smallest set $S \subseteq V$ such that $S = ||\phi||M[X := S]$, where $M[X := S]$ is obtained from M by letting $Val(X) = S$;
- $||\nu X.\phi(X)||M$ is the greatest set $S \subseteq V$ such that $S = ||\phi||M[X := S]$.

The last two items are well defined since the map sending S to $||\phi||M[X := S]$ is a monotonic function on the powerset of V and so, by the Knaster-Tarski Theorem, this map has both a least and a greatest fixpoint.

We also say that a vertex v of a model M verifies a formula ϕ, written $M, v \models \phi$, if $v \in ||\phi||M$.

4 Parity games

4.1 Definition

It is notoriously difficult to understand μ-calculus formulas, especially when there are many alternating fixpoints. A means to understand the μ-calculus is given by parity games. We will see that the semantics of μ-calculus formulas can be given in terms of parity games.

Intuitively, a parity game is a game where two players, called c and d, move in a graph (the notation, due to Arnold, suggests that c means conjunctive and d means disjunctive). The vertexes of the graph are labeled with finitely many positive integers. d wants to have many high even numbers along the play, whereas c wants to have many odd numbers.

Let us define parity games more formally. A parity game is a structure $\Gamma = (V_c, V_d, E, v_0, \Omega)$, where V_c and V_d are disjoint sets, E is a binary relation on $V_c \cup V_d$, $v_0 \in V_c \cup V_d$ is the initial vertex, and $\Omega : V_c \cup V_d \to \{1, \ldots, n\}$ is the priority function; the number n is called the index of the game.

A play of Γ is a sequence of vertices, starting from v_0, where the successor of the current vertex must be an E-successor of that vertex, and this successor is chosen by the player d, if the vertex is in V_d, and by c if the vertex is in V_c.

If the play reaches a position where either player has no moves, the other wins. If this never happens, then the play is infinite, and d wins if the greatest priority occurring infinitely often is even, and c wins otherwise.

A strategy of a player p is a function which, given an initial segment of a play ending with a p- position, determines the next move of p. A strategy is positional if the move depends only on the last vertex of the segment.

A strategy Σ of a player p is winning if every play where p moves according to Σ is won by p.

Note that parity games are Borel games, so by Borel determinacy, see [17], they enjoy determinacy: there is always one of the two players who has a winning strategy.

A well known property of parity games is positional determinacy:

Theorem 1. *(See [9], Theorem 4.4) If either player has a winning strategy in a parity game, then it has a positional winning strategy.*

However in this paper we will use a slightly stronger form of determinacy. Let us call a strategy *strongly positional* if the move on a position depends only on the successors of the position (not on the position itself). Clearly, a strongly positional strategy is positional, but the converse does not hold: if two different nodes have the same successors, a stronlgy positional strategy gives the same answer on the two nodes, whereas a positional one need not to.

Now a careful analysis of [9] gives the following strenghtening of the previous theorem:

Corollary 1. *If either player has a winning strategy in a parity game, then it has a strongly positional winning strategy.*

Proof. Note that players c and d are called AND and OR in [9]. The successor of an OR position in the positional strategy of Theorem 4.4 of [9] is chosen in a way which does not quite depend on the OR position, but depends only on the set of positions reachable in one step from that OR position. So, the resulting strategy is actually strongly positional. □

4.2 From the μ-calculus to parity games

The semantics of the μ-calculus can be given in terms of a parity game. More precisely, given a sentence ϕ and a model $M = (V, E, Val)$, with a distinguished vertex v_0 of V, we can define an evaluation game $\Gamma(\phi, v_0, M)$ as follows (we consider sentences rather than arbitrary formulas for simplicity).

The positions of $\Gamma(\phi, v_0, M)$ are the pairs (ψ, v), where $v \in V$ and ψ is a subformula of ϕ. The d positions are the pairs of the form $(\psi \vee \chi, v)$ or $(\langle\ \rangle\psi, v)$; all other positions are c positions. (ϕ, v_0) is the initial position.

There are edges from $(\psi \vee \chi, v)$ or $(\psi \wedge \chi, v)$ to (ψ, v) and (χ, v); from $(\langle\ \rangle\psi, v)$ or $([\ ]\psi, v)$ to (ψ, w) for every successor w of v in V; from $(\mu X.\psi, v)$ and $(\nu X.\psi, v)$ to (ψ, v); and if a variable occurrence X appears in a subformula $\mu X.\psi$ or $\nu X.\psi$, there is an edge from (X, v) to $(\mu X.\psi, v)$ or $(\nu X.\psi, v)$ respectively. Finally, we put also an edge from (A, v) or $(\neg A, v)$ to itself for every atom A, and from (Y, v) to itself for every variable Y free in ϕ.

To define the Ω function we proceed as follows. First we assign priorities to fixpoint subformulas of ϕ: we assign to each greatest fixpoint subformula $\nu X.\chi$ in ϕ a priority $2|\chi|$, and we assign to $\mu X.\chi$ a priority $2|\chi| + 1$. This ensures that:

- least fixpoints have odd priority;
- greatest fixpoints have even priority;
- the priority of larger subformulas is larger.

Now we let $\Omega(\psi, v)$ be the priority of ψ, if ψ is a fixpoint formula; $\Omega(A, v) = 2$ if $M, v \models A$ and $\Omega(A, v) = 1$ otherwise, if A is an atom, a negated atom or a free variable of ϕ; and $\Omega(\psi, v) = 1$ otherwise.

It results that $M, v_0 \models \phi$ if and only if player d has a winning strategy in $\Gamma(\phi, v_0, M)$.

4.3 From parity games to μ-calculus

We have seen that in a sense, μ-calculus reduces to parity games. However, as is well known, also the other way round is true: if we consider parity game of index

n as a graph vertex-colored by $c, d, 1, \ldots, n$, then there is a μ-calculus formula W_n, due to Walukiewicz, such that an arena for parity games (G, v_0) verifies W_n if and only if player d has a winning strategy in the parity game associated to (G, v_0). This formula is

$$W_n = \mu X_1 \nu X_2 \ldots \theta X_n.(d \to \langle\ \rangle \bigwedge_i (i \to X_i) \wedge (c \to [\ ] \bigwedge_i (i \to X_i).$$

5 The reduction to $S5$

In [13] there is a reduction of μ-calculus model checking to box free μ-calculus model checking. Here we modify the result by specializing to $S5$ and by referring to the vectorial model checking rather than the scalar one:

Theorem 2. *Given a finite model M and a modal μ-system S, there is a finite $S5$ model M' and a box free modal μ-system S', such that M' and S' are built in time polynomial in the size of M plus the size of S, and such that M verifies $sol_1(S)$ if and only if M' verifies $sol_1(S')$.*

Proof. Let $M = (V, E, Val)$ be a model. Let S be a modal μ-system. Up to perform a polynomial time rewriting of S, we can suppose that the equations of S have one of the following forms: $X = A$, $X = \neg A$ where A is an atom, $X = Y \vee Z$, $X = Y \wedge Z$, $X = \langle\ \rangle Y$, $X = [\ ]Y$, $X = Y$.

Now M' is obtained as follows. The vertices of M' are the vertices of M. Let us enumerate these vertices as $v_1, \ldots, v_n$. Let A_i be an atom which is true in G' only in the point v_i. The relation E' of M' holds for every pair of vertices of M', so M' is an $S5$ model.

Moreover S' is obtained by replacing every equation of S of the form

$$X = \langle\ \rangle Y$$

with

$$X = \bigvee \{A_i \wedge \langle\ \rangle (A_j \wedge Y) | v_i R v_j\},$$

and every equation of S of the form

$$X = [\ ]Y$$

with

$$X = \bigwedge \{A_i \to \langle\ \rangle (A_j \wedge Y) | v_i R v_j\}.$$

Note that one could expect that the right hand side of $X = [\ ]Y$ is replaced by the De Morgan dual of $X = \langle\ \rangle Y$, so to have $\bigwedge\{A_i \to [\ ](A_j \to Y) | v_i R v_j\}$. However, the atoms A_i are interpreted as singletons, so

$$[\ ](A_j \to Y)$$

is in fact equivalent to its De Morgan dual

$$\langle\ \rangle(A_j \wedge Y),$$

and this allows us to replace the box with the diamond.

□

We do not know whether Theorem 2 can be specialized to the scalar μ-calculus. In fact, the problem is that the translation from a modal μ-system to a single formula of the modal μ-calculus (i.e., the algorithm which builds the solution of a modal μ-system) takes exponential time in general.

6 Corollaries

It is well known that there is a translation of vectorial μ-calculus in $S5$ to vectorial modal logic in $S5$. In fact, given a modal μ-system in $S5$, we can first consider its solution and translate it into modal logic. From the previous theorem we obtain:

Corollary 2. *If there is a* polynomial time computable *translation from box-free vectorial μ-calculus in $S5$ to vectorial modal logic in $S5$, then the vectorial μ-calculus model checking problem is in P.*

Proof. By [16], model checking for vectorial modal logic (over arbitrary graphs) reduces in polynomial time to the problem of solving Boolean equation systems with only one type of fixpoint, and this problem is in P by [3]. By the previous theorem, vectorial model checking reduces to vectorial model checking over $S5$ in polynomial time; so if the translation in the statement exists, then by a chain of reductions, the vectorial μ-calculus model checking problem is in P. □

Considerations analogous to $S5$ hold in the larger class of graphs $K4$. In fact, from [2], [6] and [5] it follows that there is a translation from vectorial μ-calculus in $K4$ to $VEC - \Pi_2$ in $K4$. From the previous theorem we obtain:

Corollary 3. *If there is a* polynomial time computable *translation from vectorial μ-calculus in $K4$ to $VEC - \Pi_2$ in $K4$, then the μ-calculus model checking problem is in P.*

Proof. Every $S5$ graph is also a $K4$ graph, so by the previous theorem, there is a polynomial time reduction from vectorial model checking over arbitrary graphs to vectorial model checking in $K4$. Moreover, by [16], model checking for $VEC - \Pi_2$ (over arbitrary graphs) reduces in polynomial time to the problem of solving Boolean equation systems of class Π_2, and this problem is in P by [3]. So if the translation in the statement exists, then by a chain of reductions, the vectorial μ-calculus model checking problem is in P.

□

7 Satisfiability in $S5$

In this section we investigate the μ-calculus satisfiability problem for $S5$. We begin with establishing a linear size model property.

Lemma 1. *If a formula ϕ has a $S5$ model, then it has a $S5$ model of size linear in ϕ.*

Proof. Let (M, x_0) be a $S5$ model of ϕ. Up to bisimulation we can suppose that M is total. Then player d has a winning strategy in the game $\Gamma(\phi, x_0, M)$. By Corollary 1, d has a strongly positional winning strategy in the game, call it Σ. Consider a diamond position $(\langle\ \rangle\psi, y)$ of the game. Since M is total, the set of successors of $(\langle\ \rangle\psi, y)$ does not depend on y, so the choice of Σ also does not depend on y, but only on ψ. Let us denote by (ψ, x_ψ) the successor position of $(\langle\ \rangle\psi, y)$ chosen by Σ. Let N be the submodel of M given by x_0 plus all points x_ψ.

First, N has size linear in ϕ because its size is at most the number of diamond subformulas of ϕ (plus one). Moreover, we note that if player c in the game always chooses elements of N in box positions, then the game remains in N forever, and is won by d since Σ is winning on M.

□

As a consequence we have:

Theorem 3. *The satisfiability problem for the μ-calculus in $S5$ is NP-complete.*

Proof. NP-hardness holds because the μ-calculus contains propositional logic.

To show that the problem is in NP, suppose that models and formulas are encoded as strings in a convenient finite alphabet (e.g. ASCII code). We prove that there is a problem S in $PTIME$ and a polynomial p such that ϕ is satisfiable in $S5$ iff there exists a witness z such that $(\phi, z) \in S$ and $\mathit{length}(z) \leq p(\mathit{length}(\phi))$.

Since μ-calculus model checking is in NP (see [7]), we know that there exists a problem S' in $PTIME$ and a polynomial q such that, for any finite model M, M satisfies ϕ iff there exists a y with $(M, \phi, y) \in S'$ and $\mathit{length}(y) \leq q(\mathit{length}(M) + \mathit{length}(\phi))$. Moreover, by Lemma 1, ϕ is satisfiable in $S5$ iff ϕ is satisfiable in a model M of size linear in ϕ, which we can code with a length at most $r(\mathit{length}(\phi))$ for some polynomial r.

Let S be the set of tuples (ϕ, M, y) such that:

- M is an $S5$ model (i.e. the accessibility relation is an equivalence);
- $(M, \phi, y) \in S'$;
- $\mathit{length}(M) \leq r(\mathit{length}(\phi))$;
- $\mathit{length}(y) \leq q(\mathit{length}(M) + \mathit{length}(\phi))$.

So, ϕ is satisfiable in $S5$ if and only if there exists a witness $z = (M, y)$ such that $(\phi, z) \in S$. Note that S is in $PTIME$. Moreover, $(\phi, z) \in S$ implies $\mathit{length}(z) \leq p(\mathit{length}(\phi))$, where $p(x) = r(x) + q(r(x) + x)$. So, ϕ is satisfiable in $S5$ if and only if there exists a witness $z = (M, y)$ such that $(\phi, z) \in S$ and $\mathit{length}(z) \leq p(\mathit{length}(\phi))$. So S and p satisfy the desired properties.

□

Note that the restriction of the previous theorem to modal logic was already known, see [11].

8 On existential μ-calculus

A formula of the μ-calculus is called existential, or box-free, if it contains no box operators $[\ ]\phi$. Intuitively, existential μ-calculus is considerably simpler than general μ-calculus. In fact, the satisfiability problem for the μ-calculus is $EXPTIME$-complete, whereas, as shown in [13], the same problem for the existential μ-calculus is NP-complete. Note that this last result can be obtained in a way different from [13] as follows.

First we observe:

Lemma 2. *The satisfiability problem for existential μ-calculus is polynomial time equivalent to the same problem on $S5$.*

Proof. If an existential formula ϕ has a model M, then ϕ is also true on the reflexive, symmetric, transitive closure of M, which is an $S5$ graph of the same size as M.

□

Moreover, as a corollary of the previous section we have:

Corollary 4. *The satisfiability problem for existential μ-calculus in $S5$ is NP-complete.*

Proof. The satisfiability problem for existential μ-calculus in $S5$ is NP because, by Theorem 3, it is a particular case of an NP problem. Moreover, the problem is NP-hard because existential μ-calculus contains propositional logic. □

From the previous lemma and the previous corollary it follows:

Corollary 5. *The satisfiability problem for existential μ-calculus is NP-complete.*

9 On μ-calculus and modal logic in $S5$

It is known that μ-calculus in $S5$ is as expressive as modal logic, so in $S5$ there is no fixpoint alternation hierarchy. In this section we describe two translations from μ-calculus to modal logic; the first is due to Alberucci and Facchini, whereas the second is based on a bisimulation argument.

9.1 On the complexity of translations from the μ-calculus to modal logic over $S5$

In [2] a recursive translation of μ-calculus into modal logic in $S5$ is given. The construction is performed by induction on ordinals (rather than ordinary induction on numbers). In order to set up the construction, a notion of ordinal rank of μ-calculus formulas is introduced, with the following properties:

- $rank(A) = rank(\neg A) = 1$;
- $rank(\langle\ \rangle\phi) = rank([\ ]\phi) = rank(\phi) + 1$;
- $rank(\phi \wedge \psi) = rank(\phi \vee \psi) = max\{rank(\phi), rank(\psi)\} + 1$;
- $rank(\mu X.\phi(X)) = rank(\nu X.\phi(X)) = sup\{rank(\phi^n(X)) + 1; n \in \mathbb{N}\}$.

A μ-calculus sentence ϕ is *well named* if it is guarded and, for any variable X, no two distinct occurrences of fixpoint operators in ϕ bind X, and the atom X occurs only once in ϕ.

Using the semantical laws $\mu X.\psi(X, X) = \mu X.\mu Y.\psi(X, Y)$, $\nu X.\psi(X, X) = \nu X.\nu Y.\psi(X, Y)$ (see [3]), and renaming of bounded variables, we see that any μ-calculus formula is equivalent to a well named formula of a size which is linear in the size of ϕ. For instance, the formula $\mu X([\ ]X \wedge \langle\ \rangle X) \wedge \nu X[\ ]X$ is equivalent to the well named formula $\mu X \mu Y([\ ]X \wedge \langle\ \rangle Y) \wedge \nu Z[\ ]Z$.

The following translation t from well named μ-formulas to modal logic in $S5$ can be defined:

- $t(A) = A, t(\neg A) = \neg A$;
- $t(true) = true$, $t(false) = false$;
- $t(\langle\ \rangle\phi) = \langle\ \rangle t(\phi)$;
- $t([\ ]\phi) = [\ ]t(\phi)$;
- $t(\phi \wedge \psi) = t(\phi) \wedge t(\psi)$;
- $t(\phi \vee \psi) = t(\phi) \vee t(\psi)$;
- $t(\mu X.\phi(X)) = t((\phi(\phi(false))^*)$;
- $t(\nu X.\phi(X)) = t((\phi(\phi(true))^*)$,

where $(\phi(\phi(false))^*$, $(\phi(\phi(true))^*$ denote the well named formulas obtained from $\phi(\phi(false)), \phi(\phi(true))$ by renaming repeated bound variables. The translation t is given by induction on the rank, so it is well defined. Moreover:

Lemma 3. *If ϕ is a well named formula, then the length of $t(\phi)$ is at most $2^{|\phi|}$.*

Proof. We need a preliminary composition lemma:

Lemma 4. $t(\phi[X/\psi]) = t(\phi)[X/t(\psi)]$.

Proof. By induction on ϕ. □

Now the bound as in the lemma can be proved by induction on the rank of ϕ. The most delicate case is $\mu X.\psi(X)$ and $\nu X.\psi(X)$. Now, by Lemma 4, we have $t(\mu X.\psi(X)) = t(\psi(\psi(false)) = t(\psi)[X/t(\psi(false))]$, and since X occurs only once in ψ, we obtain the bound $|t(\mu X.\psi)| = |t(\psi)[X/t(\psi(false))]| \leq 2|t(\psi(false))| \leq 2 \times 2^{|\psi(false)|} = 2^{|\psi(false)|+1} \leq 2^{|\mu X.\psi(X)|}$. The case of ν is analogous. □

So, the translation t is at most exponential. We also can show that the exponential upper bound for the translation t is tight. In fact, consider the well named formulas:

$$\phi_n = \mu X_1 \ldots \mu X_n.X_1 \vee (X_2 \vee \ldots \vee X_n).$$

We can show by induction that $t(\phi_n)$ has size at least 2^n. In fact, the base case $n = 1$ is true; for the inductive case, we begin with a lemma:

Lemma 5. *The following equivalences hold:*

- $|t(\phi(false \vee \alpha))| > |t(\phi(\alpha)|$*;*
- *if* X *is a variable free in* ϕ*, then* $|t(\phi(X \vee \alpha))| > |t(\phi(\alpha)|$*.*

Proof. By induction on ϕ. □

Now consider $t(\phi_n)$ with $n > 1$. By definition of t we have

$$t(\phi_n) = t(\mu X_2 \dots \mu X_n.(\mu X_2 \dots \mu X_n.false \vee X_2 \vee \dots \vee X_n) \vee X_2 \dots \vee X_n)$$

and by Lemma 4 we obtain

$$t(\phi_n) = t(\mu X_2 \dots \mu X_n.Y \vee X_2 \dots \vee X_n)[Y/t(\mu X_2 \dots \mu X_n.false \vee X_2 \vee \dots \vee X_n).$$

By evaluating the sizes we see that

$$|t(\phi_n)| \geq |t(\mu X_2 \dots \mu X_n.Y \vee X_2 \dots \vee X_n)| + |t(\mu X_2 \dots \mu X_n.false \vee X_2 \vee \dots \vee X_n)]| - 1;$$

by Lemma 5 we obtain

$$|t(\phi_n)| \geq 2|t(\mu X_2 \dots \mu X_n.X_2 \vee \dots X_n)|,$$

and, by renaming the variables,

$$|t(\phi_n)| \geq 2|t(\phi_{n-1})|.$$

Finally, the inductive hypothesis gives $|t(\phi_{n-1})| \geq 2^{n-1}$, so $|t(\phi_n)| \geq 2^n$.

9.2 An alternative translation

A translation from μ-calculus to modal logic different from [2] is obtained as follows. Let ϕ be a μ-calculus formula containing a set At of atoms. Then, up to bisimulation, there are finitely many $S5$ models colored with At, more precisely exponentially many of them. Each bisimulation class is described by a characteristic formula, that is, the conjunction of $\langle\ \rangle\gamma$ where γ is a conjunction of atoms and negated atoms present in the model of the class, and $\neg\langle\ \rangle\gamma$ where γ is a conjunction not present in the class. So, ϕ is equivalent to the disjunction of the characteristic formulas of the bisimulation classes of the models of ϕ. Note that this alternative translation is also (at most) exponential.

9.3 A corollary

Both translations of the previous subsections give an exponential blow up in the size of the formula. It is then natural to ask whether there exists a polynomial time computable translation from the μ-calculus to modal logic over $S5$. This question is related to the results of the previous section, as the following final corollary shows:

Corollary 6. *If there is a polynomial time computable translation from the μ-calculus to modal logic over $S5$ and Theorem 2 specializes to scalar μ-calculus, then the μ-calculus model checking problem is in P.*

Proof. Given a μ-formula ϕ and a model M, first reduce the problem to a μ-formula ϕ' over an $S5$ model M', and then apply the polynomial translation in order to obtain a modal formula ϕ^* with

$$M \models \phi \Leftrightarrow M' \models \phi^*.$$

Since both M' and ϕ^* are obtained in polynomial time from M, ϕ and *modal* model checking is in P, we have done. □

10 Conclusion

In this paper we have investigated some aspects of μ-calculus on $S5$ graphs. Arguably, these graphs are quite simple. However, simplicity of $S5$ graphs gives us better satisfiability bounds than arbitrary graphs, but not better bounds on the model checking problem.

An interesting question is whether there is a direct, natural translation from modal μ systems to modal systems in $S5$: by this we mean that the translation should not go through transforming a system of equations into a single scalar term.

A similar analysis of μ-calculus model checking and satisfiability could be carried over other important classes of graphs. An example is $K4$. Since satisfiability in $K4$ efficiently reduces to general satisfiability, in $K4$ we have the same bound as in K, that is, $EXPTIME$. It would be interesting to see whether a better bound can be given. Note that for model checking, as we have seen, a better bound for $K4$ with respect to arbitrary graphs does not exist.

The same analysis could be done for other interesting classes of graphs, e.g. the longstanding Gödel-Löb class GL (i.e. the transitive wellfounded graphs), or for more recent classes such as graphs of bounded tree width or classes with forbidden minors. A good model checking algorithm for bounded tree width is e.g. [18], but we do not have yet a polynomial time model checking algorithm on graphs of bounded tree width. Apparently there is no result on satisfiability on bounded tree width.

Acknowledgments

This work has been partially supported by the PRIN project n. 20089M932N *Innovative and multi-disciplinary approaches for constraint and preference reasoning.*

References

1. L. Alberucci, Sequent Calculi for the Modal μ-Calculus over S5. J. Log. Comput. 19(6): 971–985 (2009).
2. L. Alberucci and A. Facchini, The modal μ-calculus hierarchy over restricted classes of transition systems, J. Symb. Logic 74 (2009) 1367–1400.
3. A. Arnold and D. Niwinski, Rudiments of μ-calculus, North-Holland, 2001.
4. Edmund M. Clarke, Jr., Orna Grumberg and Doron A. Peled, Model Checking, MIT Press, 1999.
5. A. Dawar and M. Otto, Modal characterisation theorems over special classes of frames, Ann. Pure Appl. Logic 161 (2009), 1–42.
6. G. D'Agostino and G. Lenzi, On the μ-calculus over transitive and finite transitive frames, Theor. Comput. Sci. 411(50): 4273–4290 (2010).
7. E. Allen Emerson: Model Checking and the Mu-calculus. Descriptive Complexity and Finite Models 1996: 185–214.
8. E. A. Emerson and C. S. Jutla, The complexity of tree automata and logics of programs. In Proc. 29th IEEE FOCS 328–337 (1988).
9. E. A. Emerson and C. S. Jutla. Tree Automata, Mu-Calculus and Determinacy (Extended Abstract). FOCS 1991: Pages 368–377.
10. Robert S. Streett and E. Allen Emerson. An Automata Theoretic Decision Procedure for the Propositional Mu-Calculus. Information and Computation 81 (1989), 249–264.
11. R. Fagin, Reasoning about knowledge, MIT Press, 2003.
12. M. J. Fischer and R. E. Ladner, Propositional dynamic logic of regular programs, J. Comput. System Sci., 18 (1979), pp. 194–211.
13. Thomas A. Henzinger, Orna Kupferman, and Rupak Majumdar, On the Universal and Existential Fragments of the Mu-Calculus, Theoretical Computer Science 354:173-186, 2006.
14. M. Jurdzinski, Deciding the winner in parity games is in $UP \cap co - UP$, Inform. Proc. Letters 68 (1998), 119-124.
15. D. Kozen, Results on the Propositional mu-Calculus, Theor. Comput. Sci. 27: 333–354 (1983).
16. A. Mader, Verification of Modal Properties using Boolean Equation Systems, Ph. D. Thesis, 1997.
17. D. A. Martin, Borel determinacy, Ann. Math., 102 (1975), pp. 363–371.
18. J. Obdrzalek, Fast Mu-Calculus Model Checking when Tree-Width Is Bounded. CAV 2003, 80–92.
19. F. Poggiolesi, A cut-free simple sequent calculus for modal logic S5, Review of Symbolic Logic, 1:3–15, 2008.
20. I. Walukiewicz, Completeness of Kozen's Axiomatisation of the Propositional Mu-Calculus, Information and Computation 157 (2000), 142–182.