POLYNOMIAL APPROXIMATIONS FOR MODEL CHECKING

(Extended abstract of a SHORT TALK)

N. V. Shilov[1,2], N. O. Garanina[2]

[1] Visiting Erskine Fellow, University of Canterbury, Christchurch, New Zealand
e-mail: nikolai.shilov@cosc.canterbury.ac.nz
[2] Institute of Informatics Systems, Novosibirsk, Russia,
e-mail: shilov@iis.nsk.su

Key words: μ-Calculus, model checking, complexity.

Abstract. The μ-Calculus of D.Kozen (1983) is a very powerful propositional program logic with fixpoints. It is widely used for specification and verification. Model checking is a very popular automatic approach for verification of specifications of finite state systems. The most efficient algorithms that have been developed so far for model checking the μ-Calculus in finite state systems have exponential upper bounds. A.Emerson, C.Jutla, and P.Sistla studied (1993) the first fragment of the μ-Calculus that permits arbitrary nesting and alternations of fixpoints, and polynomial model checking in finite state systems. In contrast we study lower and upper approximations for model checking that are computable in polynomial time, and that can give correct semantics in finite models for formulae with arbitrary nesting and alternations. A.Emerson, C.Jutla, and P.Sistla proved also that the model checking problem for the μ-Calculus in finite state systems is in $\mathcal{NP} \cap co\text{-}\mathcal{NP}$. We develop another proof (that we believe is a new one) as a by-product of our study.

1 Preliminaries

The μ-Calculus of D.Kozen (μC) [8] is a very powerful propositional program logic with fixpoints. It is widely used for specification and verification of properties of finite state systems[2]. Due to this reason we restrict ourself in this paper by finite state systems also. Please refer to [9] for the elementary introduction to μC. The comprehensive definition of μC can be found in a recent textbook [1].)

The syntax of μC is constructed from two disjoint alphabets of propositional variables (Prp) and action symbols (Act). It consists of formulae: $\phi ::= (\neg\phi) \mid (\phi \wedge \psi) \mid \langle a \rangle \phi \mid ([a]\phi) \mid (\mu p.\phi) \mid (\nu p.\phi)$, ($p \in Prp$, $a \in Act$, no negative instances of bounded variables). The semantics of μC is defined in models. A model is a triple (D, R, V), where the domain D is a non-empty set, the interpretation R is a total mapping $R : Act \to 2^{D \times D}$, the valuation V is another total mapping $V : Prp \to 2^D$. In every model $M = (D, R, V)$, for every formula ϕ, the semantics

$M(\phi)$ is a subset of the domain D that is defined by induction by the formula structure.

A propositional variable is said to be a propositional constant in a formula iff it is free in the formula. A formula is said to be in a normal form iff negation is applied to propositional constants in the formula only. Due to standard De Morgan laws and duality of [ ] and $\langle\rangle$, μ and ν, every formula of μC is equivalent to some formula in the normal form that can be constructed in polynomial time.

Let us extend μC by some new features: total S5 modalities $\Box$ and $\Diamond$, and second order (SO) quantifiers $\forall$ and $\exists$. The syntax: ϕ ::= μC | $(\Box\phi)$ | $(\Diamond\phi)$ | $(\forall\ p.\ \phi)$ | $(\exists\ p.\ \phi)$. The semantics:

- $M(\Box\psi) = \begin{cases} D, \text{ if } M(\psi) = D \\ \emptyset \text{ otherwise} \end{cases}$, $M(\Diamond\psi) = \begin{cases} D, \text{ if } M(\psi) \neq \emptyset \\ \emptyset \text{ otherwise} \end{cases}$,
- $M(\forall p.\psi) =$ the greatest lower bound of $\{M_{S/p}(\psi)\ :\ S \subseteq D\}$,
  $M(\exists p.\psi) =$ the least upper bound of $\{M_{S/p}(\psi)\ :\ S \subseteq D\}$,

where ψ, p range over formulae and propositional variables, and $M_{S/p}$ denotes the model that agrees with M everywhere but p: $V_{S/p}(p) = S$.

We also use a variation of the following classical theorem of R. Fagin [5]. (Please refer to [6] for the elementary introduction to descriptive complexity in general and Fagin theorem in particular. For further details refer to [7].)

> *A set SET of finite structures is in $\mathcal{NP}$ iff there exists a first-order formula FRM that $SET = \{STR : STR \models (\exists P_1 ... \exists P_m\ FRM)\}$, where P_1, ... P_m are predicate symbols in FRM.*

The theorem implies that for every first-order formula FRM the following set of finite structures $\{STR : STR \models (\forall P_1 ... \forall P_m\ FRM)\}$ is in co-$\mathcal{NP}$. We generalize this claim a little bit.

Assume FRM be a first-order formula with predicate symbols P_1, ... P_m and Q_1, ... Q_n. Assume also that we are interested in finite structures STR where interpretations of Q_1, ... Q_n are some explicit second-order functions G_1, ... G_n of interpretations of P_1, ... P_m. In this case we say that FRM is a first-order formula with predicate symbols P_1, ... P_m and second-order functions G_1, ... G_n.

Proposition 1. *For all predicate symbols P_1, ... P_m, for every first-order formula FRM with predicate symbols P_1, ... P_m and some second-order functions, the following set of finite structures $\{\ STR\ :\ STR \models (\forall P_1 ... \forall P_m\ FRM)\}$ is in co-$\mathcal{NP}$ provided that all used second-order functions are computable in time polynomial on the structure size.*

2 Upper Bound

The model checking problem for the propositional μ-Calculus in finite state systems is to decide the following set $\{\ (\phi, M, s)\ :\ \phi$ is a μC formula, M is a finite model, and $s \in M(\phi)\}$. The best known complexity class for the model checking problem for μC in finite models is $\mathcal{NP} \cap$ co-$\mathcal{NP}$ [4]. (It is not known

whether the problem is (in)complete in any of $\mathcal{NP}$ and *co-*$\mathcal{NP}$.) A new proof for this upper bound is sketched below.

Proposition 2.

- *For every propositional variable p, every μC+S5 formula θ in the normal form, and every its subformula $(\mu p.\phi)$ that is not nested within the scope of other μ or ν, the following is a tautology:* $\theta \leftrightarrow \Big(\forall p.\big(\Box(\phi \rightarrow p) \rightarrow \theta^{p}_{(\mu p.\phi)}\big)\Big)$.
- *For every propositional variable q, every μC+S5 formula θ in the normal form, and every its subformula $(\nu q.\psi)$ that is not nested within the scope of other μ or ν, the following is a tautology:* $\theta \leftrightarrow \Big(\exists q.\big(\Box(q \rightarrow \psi) \wedge \theta^{q}_{(\nu q.\psi)}\big)\Big)$.

(Here and throughout the paper $\rightarrow$ and $\leftrightarrow$ stay for standard abbreviations for implication and equivalence, X^{Y}_{Z} is the formula obtained by substituting Y for all occurrences of Z in X.)

A formula is said to be in the special prefix form iff it looks as follows:

$$(\star)\ \underbrace{\forall p_n.\ \exists q_n.\ ...\ \forall p_1.\ \exists q_1.}_{n \text{ times}}$$
$$\underbrace{\Big(\Box(\phi_1 \rightarrow p_1) \rightarrow \Big(\Box(q_1 \rightarrow \psi_1) \wedge \big(...(\Box(\phi_n \rightarrow p_n) \rightarrow (\Box(q_n \rightarrow \psi_n) \wedge\ \theta))...\big)\Big)\Big)}_{n \text{ times}}$$

where n is some integer, all formulae θ, ϕ_1, ψ_1, ..., ϕ_n, ψ_n are quantifier- and fixpoint-free, and without any instance of any of propositional variables p_1, q_1, ... p_n, q_n under negations.

Corollary 3. *Every formula of the μ-Calculus is equivalent to some formula in the special prefix form that can be constructed in polynomial time.*

In the following proposition we use symbols of second order functions that are interpreted in models by second order functions on sets of states. We introduce these symbols and functions for Skolemization of existential quantifiers in formulae that are in the special prefix forms.

Proposition 4. *Let ξ be an arbitrary formula in the special prefix form $(\star)$. Let g_n, ... g_1 be disjoint symbols of second order functions of 1, ... n arguments respectively. For every model M there exist total second order functions on sets of states $G_n : (2^D)^1 \rightarrow 2^D$, ... $G_1 : (2^D)^n \rightarrow 2^D$, with values computable in polynomial time, that the following equality holds:*

$$M(\xi) \ = \ M_{(G_1/g_1)...(G_n/g_n)}\Big(\forall p_1.\ ...\ \forall p_n.\ \Big(\Box\big(\textstyle\bigwedge_{i=1}^{i=n}(\phi_i \rightarrow p_i)\big)\ \rightarrow \theta\Big)^{g_n(p_n)...g_1(p_1,...p_n)}_{q_n...q_1}\Big).$$

In combination with proposition 1, it implies the following theorem.

Theorem 5. *The model checking problem for the propositional μ-Calculus in finite models is in $\mathcal{NP} \cap$ co-$\mathcal{NP}$.*

3 Approximating Model Checking

The most efficient algorithms that have been developed so far for model checking the μC in finite models have exponential upper bounds (ex., [3]). A.Emerson, C.Jutla, and P.Sistla studied the first fragment of the μ-Calculus that permits arbitrary nesting and alternations of fixpoints, and polynomial model checking in finite state systems [4]. The fragment comprises the formulae in the normal form where only closed subformulae can occur in the range of box [ ] and at most one conjunct in every conjunction $\wedge$ is not a closed formula.

We suggest another "fragment" that enjoys polynomial model checking in finite models in spite of nesting and amount of alternations. In contrast to the syntactical definition of the above fragment, the new one has a computational characterization. It exploits a polynomial approximation algorithm presented below. The algorithm calculates lower and upper approximations for semantics of formulae in finite models. If for some formula in some model both approximations are equal, then the algorithm succeeds in calculation of semantics of the formula in the model.

Input a formula ϕ of μC and a finite model M.

Preprocessing. Convert ϕ into the equivalent normal form and then into the equivalent special prefix form (see $\star$). Let two vectors of formulae $\Phi = (\phi_1, ...\phi_n)$ and $\Psi = (\psi_1, ...\psi_n)$ collect all ϕ's and ψ's in ξ.

Let $\underline{P}^j = (\underline{P}^j_1, ...\underline{P}^j_n)$ and $\underline{Q}^j = (\underline{Q}^j_1, ...\underline{Q}^j_n)$ (where $j \geq 0$) be vectors of disjoint variables for sets of states.

Let $\overline{P}^k = (\overline{P}^k_1, ...\overline{P}^k_n)$ and $\overline{Q}^k = (\overline{Q}^k_1, ...\overline{Q}^k_n)$ (where $k \geq 0$) be some other vectors of disjoint variables also for sets of states.

Let $j := 0$ and $k := 0$, $\underline{Q}^0 := (\emptyset, ...\emptyset)$ and $\overline{P}_0 := (D, ...D)$.

Processing.

Repeat

$\underline{P}^j :=$ the least fixpoint of $\lambda S_n...S_1 .\big(M_{(\underline{Q}^j_n/q_n)...(\underline{Q}^j_1/q_1)(S_n/p_n)...(S_1/p_1)}(\Phi)\big)$,

$\underline{Q}^{j+1} :=$ the greatest fixpoint of $\lambda S_n...S_1 .\big(M_{(S_n/q_n)...(S_1/q_1)(\underline{P}^j_n/p_n)...(\underline{P}^j_1/p_1)}(\Psi)\big)$,

$j := j + 1$ **until** $\underline{Q}^j = \underline{Q}^{j-1}$.

Repeat chemu ravno $\overline{Q}^k_i$???

$\overline{P}^k :=$ the least fixpoint of $\lambda S_n...S_1 .\big(M_{(\overline{Q}^k_n/q_n)...(\overline{Q}^k_1/q_1)(S_n/p_n)...(S_1/p_1)}(\Phi)\big)$,

$\overline{Q}^{k+1} :=$ the greatest fixpoint of $\lambda S_n...S_1 .\big(M_{(S_n/q_n)...(S_1/q_1)(\overline{P}^k_n/p_n)...(\overline{P}^k_1/p_1)}(\Psi)\big)$,

$k := k + 1$ **until** $\overline{Q}^k = \overline{Q}_{k-1}$.

Let $j := j - 1$ and $k := k - 1$.

Output two sets of states: $\underline{M}(\phi) \equiv M_{(\underline{Q}^j_n/q_n)...(\underline{Q}^j_1/q_1)(\underline{P}^j_n/p_n)...(\underline{P}^j_1/p_1)}(\theta)$ and $\overline{M}(\phi) \equiv M_{(\overline{Q}^k_n/q_n)...(\overline{Q}^k_1/q_1)(\overline{P}^k_n/p_n)...(\overline{P}^k_1/p_1)}(\theta)$.

Proposition 6. *For every formula ϕ of the propositional μ-Calculus, and for every finite Kripke model M, the approximation algorithm returns lower and upper bounds for semantics of the formula in the model: $\underline{M}(\phi) \subseteq M(\phi) \subseteq \overline{M}(\phi)$.*

It implies the following theorem.

Theorem 7. *For every formula ϕ of the propositional μ-Calculus, and for every finite Kripke model M, if $\underline{M}(\phi) = \overline{M}(\phi)$ then $M(\phi) = \underline{M}(\phi) = \overline{M}(\phi)$.*

Let us give an example of the formula that is in the proposed fragment: $FAIR \equiv \nu q.\Big([a]q \wedge \big(\mu p.(r \vee [a]p)\big)\Big)$. The formula holds in finite models on those states where the propositional constant r holds infinitely often along every infinite a-path that starts from these states. The formula *FAIR* does not belong to the fragment of A.Emerson, C.Jutla, and P.Sistla, and we do not know whether *FAIR* is equivalent to some formula in the fragment of A.Emerson, C.Jutla, and P.Sistla.

Let us remark also that our "fragment" has no restrictions on nesting and alternations. For every $n \geq 1$ the formula $(\mu x_1.\nu y_1.\ ...(\mu x_n.\nu y_n.(\bigwedge_{i=1}^{i=n}(x_i \wedge y_i)))...)$ belongs to the "fragment", it has $(2 \times n - 1)$ alternations of fixpoints and the nesting level $2 \times n$.

The Model Checker that utilizes the approximation algorithm is in process of implementation. It outputs the upper and lower approximations of semantics of the input formulae in the input finite models (that sometimes are their exact semantics). An experimental and theoretical study of its utility is a topic for the further research. In our theoretical studies we hope to develop some syntactical conditions for input formulae and easy-to-check semantic conditions for input models that guarantee that the approximation algorithm calculates the exact semantics. At present we can prove a very simple claim of this kind: the algorithm calculates exact semantics in finite models for all μC formulae without alternations; in particular, it is correct model checking algorithm for the Computation Tree Logic (CTL [2]). In our experimental research we hope to demonstrate an utility of both lower and upper approximations in practical program and system verification.

Acknowledgement: We would like to thanks anonymous reviewers for comments and suggestions. Unfortunately we are unable to provide all proof within this extended abstract due to space limitations. But all proofs and the full-body version of the paper are available upon request.

References

1. Arnold A. and Niwinski D. *Rudiments of μ-calculus.* North Holland, 2001.
2. Clarke E.M., Grumberg O., Peled D. *Model Checking.* MIT Press, 1999.
3. Cleaveland R., Klain M., Steffen B. *Faster Model-Checking for Mu-Calculus.* Lecture Notes in Computer Science, v.663, 1993, p.410-422.
4. Emerson E.A., Jutla C.S., Sistla A.P. *On model-checking for fragments of Mu-Calculus.* Lecture Notes in Computer Science, v.697, 1993, p.385-396.
5. Fagin R. *Generalized First-Order Spectra and Polynomial-Time Recognizable Sets.* Complexity of Computations, SIAM-AMS Proc., v.7, 1974, p.27-41.
6. Immerman N. *Descriptive Complexity: A Logician's Approach to Computation.* Notices of the American Mathematical Society, v.42, n.10, 1995, p.1127 - 1133.
7. Immerman N. *Descriptive Complexity.* 1999, Springer-Verlag

8. Kozen D. *Results on the Propositional Mu-Calculus.* Theoretical Computer Science, v.27, n.3, 1983, p.333-354.
9. Shilov N.V., Yi K. *How to find a coin: propositional program logics made easy.* The Bulletin of the European Association for Theoretical Computer Science, v.75, 2001, p.127-151.

This article was processed using the LaTeX macro package with LLNCS style