On modal μ-calculus in $S5$ and applications

Giovanna D'Agostino
DIMI, Università di Udine, Italy
giovanna.dagostino@dimi.uniud.it

Giacomo Lenzi (speaker)
DipMat, Università di Salerno, Italy
gilenzi@unisa.it

CILC 2011
Pescara, Italy
August 31-September 2

Introduction

Introduction

The modal μ-calculus is modal logic plus extremal fixpoints.

Introduction

The modal μ-calculus is modal logic plus extremal fixpoints.

The modal μ-calculus is important in system verification, because it lies at the heart of the Model Checking technique.

Introduction

The modal μ-calculus is modal logic plus extremal fixpoints.

The modal μ-calculus is important in system verification, because it lies at the heart of the Model Checking technique.

Model theoretic and algorithmic properties of μ-calculus are interesting, both on arbitrary graphs and on subclasses of graphs.

Introduction

The modal μ-calculus is modal logic plus extremal fixpoints.

The modal μ-calculus is important in system verification, because it lies at the heart of the Model Checking technique.

Model theoretic and algorithmic properties of μ-calculus are interesting, both on arbitrary graphs and on subclasses of graphs.

In this talk we will consider two important subclasses of graphs, $S5$ and $K4$.

Contents

Contents

Contents

Contents

Scalar μ-calculus

Scalar μ-calculus

Variables, atoms and negated atoms are formulas;

Scalar μ-calculus

Variables, atoms and negated atoms are formulas;

$\wedge, \vee$ are the boolean operators;

Scalar μ-calculus

Variables, atoms and negated atoms are formulas;

$\wedge, \vee$ are the boolean operators;

$[\;], \langle\;\rangle$ are the modal operators;

Scalar μ-calculus

Variables, atoms and negated atoms are formulas;

$\wedge, \vee$ are the boolean operators;

$[\ ], \langle\ \rangle$ are the modal operators;

$\mu X.\phi(X)$ is the least fixpoint;

Scalar μ-calculus

Variables, atoms and negated atoms are formulas;

$\wedge, \vee$ are the boolean operators;

$[\ ], \langle\ \rangle$ are the modal operators;

$\mu X.\phi(X)$ is the least fixpoint;

$\nu X.\phi(X)$ is the greatest fixpoint.

Vectorial μ-calculus

Vectorial μ-calculus

We consider vectorial μ-terms of the form

Vectorial μ-calculus

We consider vectorial μ-terms of the form

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

Vectorial μ-calculus

We consider vectorial μ-terms of the form

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

where f_i are modal formulas and

Vectorial μ-calculus

We consider vectorial μ-terms of the form

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

where f_i are modal formulas and θ_i is μ or ν.

Vectorial μ-calculus

We consider vectorial μ-terms of the form

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

where f_i are modal formulas and θ_i is μ or ν.

A vectorial μ-term T is by definition equivalent to a n-tuple of formulas $(Sol_1(T), \dots, Sol_n(T))$.

Fixpoint hierarchy

Fixpoint hierarchy

In scalar and vectorial μ-calculus we let:

Fixpoint hierarchy

In scalar and vectorial μ-calculus we let:

$\Pi_0 = \Sigma_0 =$ formulas without fixpoints;

Fixpoint hierarchy

In scalar and vectorial μ-calculus we let:

$\Pi_0 = \Sigma_0 =$ formulas without fixpoints;

$\Pi_{n+1} =$ closure of $\Pi_n \cup \Sigma_n$ under composition and greatest fixpoints;

Fixpoint hierarchy

In scalar and vectorial μ-calculus we let:

$\Pi_0 = \Sigma_0 =$ formulas without fixpoints;

$\Pi_{n+1} =$ closure of $\Pi_n \cup \Sigma_n$ under composition and greatest fixpoints;

$\Sigma_{n+1} =$ closure of $\Pi_n \cup \Sigma_n$ under composition and least fixpoints;

Fixpoint hierarchy

In scalar and vectorial μ-calculus we let:

$\Pi_0 = \Sigma_0 =$ formulas without fixpoints;

$\Pi_{n+1} =$ closure of $\Pi_n \cup \Sigma_n$ under composition and greatest fixpoints;

$\Sigma_{n+1} =$ closure of $\Pi_n \cup \Sigma_n$ under composition and least fixpoints;

$\Delta_n = \Sigma_n \cap \Pi_n$.

Kripke semantics

Kripke semantics

We need valued graphs (G, val), where

Kripke semantics

We need valued graphs (G, val), where

$G = (V, R)$ is a graph;

Kripke semantics

We need valued graphs (G, val), where

$G = (V, R)$ is a graph;

val is a function from atoms and variables to $\mathcal{P}(V)$.

Kripke semantics

We need valued graphs (G, val), where

$G = (V, R)$ is a graph;

val is a function from atoms and variables to $\mathcal{P}(V)$.

For every formula ϕ, $||\phi||(G, val)$ is a subset of V, defined by induction on ϕ.

Bisimulation

Bisimulation

B is a bisimulation between (G, val) and (G', val') if
$B \subseteq V(G) \times V(G')$

Bisimulation

B is a bisimulation between (G, val) and (G', val') if $B \subseteq V(G) \times V(G')$ and whenever xBx':

Bisimulation

B is a bisimulation between (G, val) and (G', val') if $B \subseteq V(G) \times V(G')$ and whenever xBx':

- $x \in val(A)$ if and only if $x' \in val'(A)$;

Bisimulation

B is a bisimulation between (G, val) and (G', val') if $B \subseteq V(G) \times V(G')$ and whenever xBx':

- $x \in val(A)$ if and only if $x' \in val'(A)$;
- if xRy, then there is y' such that $x'R'y'$ and yBy';

Bisimulation

B is a bisimulation between (G, val) and (G', val') if $B \subseteq V(G) \times V(G')$ and whenever xBx':

- $x \in val(A)$ if and only if $x' \in val'(A)$;
- if xRy, then there is y' such that $x'R'y'$ and yBy';
- if $x'R'y'$, then there is y such that xRy and yBy'.

$K4$ and $S5$

$K4$ and $S5$

$K4$ is the class of all transitive graphs.

$K4$ and $S5$

$K4$ is the class of all transitive graphs.

$S5$ is the class of all equivalence relations, i.e. of reflexive, symmetric, transitive graphs.

$K4$ and $S5$

$K4$ is the class of all transitive graphs.

$S5$ is the class of all equivalence relations, i.e. of reflexive, symmetric, transitive graphs.

$K4$ is important in many contexts (e.g. temporal reasoning), whereas $S5$ is often used as an epistemic logic.

Parity games

Parity games

Two players c and d (after Arnold);

Parity games

Two players c and d (after Arnold);

$$G = (V_c, V_d, E, v_0, \Omega : V_c \cup V_d \to \{1, \ldots, n\});$$

Parity games

Two players c and d (after Arnold);

$G = (V_c, V_d, E, v_0, \Omega : V_c \cup V_d \to \{1, \ldots, n\})$;

players move along edges;

Parity games

Two players c and d (after Arnold);

$G = (V_c, V_d, E, v_0, \Omega : V_c \cup V_d \to \{1, \ldots, n\})$;

players move along edges;

if either player has no move, the other wins;

Parity games

Two players c and d (after Arnold);

$$G = (V_c, V_d, E, v_0, \Omega : V_c \cup V_d \to \{1, \dots, n\});$$

players move along edges;

if either player has no move, the other wins;

otherwise, d wins if the largest value of Ω occurring infinitely often is even.

Contents

A reduction theorem

A reduction theorem

Theorem

Given a model (G, val) and a vectorial μ-term T,

A reduction theorem

Theorem

Given a model (G, val) and a vectorial μ-term T, there is a model (G', val') and a vectorial μ-term T', such that:

A reduction theorem

Theorem

Given a model (G, val) and a vectorial μ-term T, there is a model (G', val') and a vectorial μ-term T', such that:

- *T' is existential (i.e. box free);*

A reduction theorem

Theorem

Given a model (G, val) and a vectorial μ-term T, there is a model (G', val') and a vectorial μ-term T', such that:

- *T' is existential (i.e. box free);*
- *G' is of class $S5$;*

A reduction theorem

Theorem

Given a model (G, val) and a vectorial μ-term T, there is a model (G', val') and a vectorial μ-term T', such that:

- *T' is existential (i.e. box free);*
- *G' is of class $S5$;*
- *(G', Val') and T' are built in time polynomial in the size of (G, val) plus the size of T;*

A reduction theorem

Theorem

Given a model (G, val) and a vectorial μ-term T, there is a model (G', val') and a vectorial μ-term T', such that:

- *T' is existential (i.e. box free);*
- *G' is of class $S5$;*
- *(G', Val') and T' are built in time polynomial in the size of (G, val) plus the size of T;*
- *(G, val) verifies $sol_1(T)$ if and only if (G', val') verifies $sol_1(T')$.*

Corollaries

Corollaries

Corollary

If there is a polytime *translation from box-free vectorial μ-calculus in $S5$ to vectorial modal logic in $S5$,*

Corollaries

Corollary

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

Corollaries

Corollary

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

Corollary

If there is a polytime *translation from box-free vectorial μ-calculus in $K4$ to vectorial Π_2 in $K4$,*

Corollaries

Corollary

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

Corollary

If there is a polytime *translation from box-free vectorial μ-calculus in $K4$ to vectorial Π_2 in $K4$, then the μ-calculus model checking problem is in P.*

Satisfiability in $S5$

Satisfiability in $S5$

Lemma

If a μ-calculus formula ϕ has a model in $S5$, then it has one with size linear in ϕ.

Satisfiability in $S5$

Lemma

If a μ-calculus formula ϕ has a model in $S5$, then it has one with size linear in ϕ.

Our proof uses parity games.

Satisfiability in $S5$

Lemma

If a μ-calculus formula ϕ has a model in $S5$, then it has one with size linear in ϕ.

Our proof uses parity games.

Theorem

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

Satisfiability in $S5$

Lemma

If a μ-calculus formula ϕ has a model in $S5$, then it has one with size linear in ϕ.

Our proof uses parity games.

Theorem

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

In fact, *NP*-hardness is because the μ-calculus includes propositional logic;

Satisfiability in $S5$

Lemma

If a μ-calculus formula ϕ has a model in $S5$, then it has one with size linear in ϕ.

Our proof uses parity games.

Theorem

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

In fact, NP-hardness is because the μ-calculus includes propositional logic; an NP algorithm is given by guessing a model of a formula and then running an NP model checking algorithm.

On existential μ-calculus

On existential μ-calculus

Lemma

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

On existential μ-calculus

Lemma

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

In fact, if ϕ has a model M, then it is true on the $S5$ closure of M.

On existential μ-calculus

Lemma

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

In fact, if ϕ has a model M, then it is true on the $S5$ closure of M. From the previous theorem it follows:

On existential μ-calculus

Lemma

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

In fact, if ϕ has a model M, then it is true on the $S5$ closure of M. From the previous theorem it follows:

Corollary

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

On existential μ-calculus

Lemma

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

In fact, if ϕ has a model M, then it is true on the $S5$ closure of M. From the previous theorem it follows:

Corollary

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

Corollary

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

Contents

The Alberucci-Facchini rank

The Alberucci-Facchini rank

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.

The Alberucci-Facchini rank

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.

Define now an ordinal rank on formulas:

The Alberucci-Facchini rank

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.

Define now an ordinal rank on formulas:

- $rank(A) = rank(\neg A) = 1$;
- $rank(\langle\ \rangle\phi) = rank([\ ]\phi) = rank(\phi) + 1$;
- $rank(\phi \wedge \psi) = rank(\phi \vee \psi) = max\{rank(\phi), rank(\psi)\} + 1$;

The Alberucci-Facchini rank

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.

Define now an ordinal rank on formulas:

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

The Alberucci-Facchini translation

The Alberucci-Facchini translation

It is a function t given by induction on the rank:

The Alberucci-Facchini translation

It is a function t given by induction on the rank:

- $t(A) = A, t(\neg A) = \neg A$;
- $t(true) = true$, $t(false) = false$;
- $t(\langle\ \rangle\phi) = \langle\ \rangle t(\phi)$;
- $t([\ ]\phi) = [\ ]t(\phi)$;
- $t(\phi \wedge \psi) = t(\phi) \wedge t(\psi)$;
- $t(\phi \vee \psi) = t(\phi) \vee t(\psi)$;

The Alberucci-Facchini translation

It is a function t given by induction on the rank:

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

The Alberucci-Facchini translation

It is a function t given by induction on the rank:

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

The Alberucci-Facchini translation

It is a function t given by induction on the rank:

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

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 complexity of the translation

The complexity of the translation

Lemma

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

The complexity of the translation

Lemma

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

The exponential bound is tight.

An alternative translation

An alternative translation

Let ϕ be a μ-calculus formula containing a set At of atoms.

An alternative translation

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.

An alternative translation

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. We note that in $S5$, characteristic formulas of models are modal.

An alternative translation

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. We note that in $S5$, characteristic formulas of models are modal.

So, ϕ is equivalent to the finite disjunction of the characteristic formulas of the bisimulation classes of the models of ϕ.

An alternative translation

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. We note that in $S5$, characteristic formulas of models are modal.

So, ϕ is equivalent to the finite disjunction of the characteristic formulas of the bisimulation classes of the models of ϕ. Note that this alternative translation is also (at most) exponential.

A corollary

A corollary

Corollary

If the reduction theorem specializes to scalar μ-calculus

A corollary

Corollary

If the reduction theorem specializes to scalar μ-calculus and there is a polytime *translation from the μ-calculus to modal logic in $S5$,*

A corollary

Corollary

If the reduction theorem specializes to scalar μ-calculus and there is a polytime *translation from the μ-calculus to modal logic in $S5$, then the μ-calculus model checking is in P.*

Conclusion

Conclusion

Simplicity of $S5$ gives good satisfiability bounds, but no good model checking bounds.

Conclusion

Simplicity of $S5$ gives good satisfiability bounds, but no good model checking bounds.

Are there "natural" translations from vectorial to scalar terms in $S5$?

Conclusion

Simplicity of $S5$ gives good satisfiability bounds, but no good model checking bounds.

Are there "natural" translations from vectorial to scalar terms in $S5$?

Complexity of satisfiability of μ in $K4$ can be obtained by reducing to arbitrary graphs; what about better bounds?

Thank you!