On μ-calculus

Gaëlle Fontaine

March 19, 2009

Outline

Quick introduction to the μ-calculus

Syntactic characterization of semantic fragments of the μ-calculus

Expressivity of GL on finite trees

Quick introduction to the μ-calculus

Syntactic characterization of semantic fragments of the μ-calculus

Expressivity of GL on finite trees

Introduction to μ-calculus

- Syntax: $\top \mid \bot \mid p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \mu x.\varphi \mid \nu x.\varphi$

Introduction to μ-calculus

- Syntax: $\top \mid \bot \mid p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid |\Box\varphi \mid \mu x.\varphi \mid \nu x.\varphi$
- Semantic: Given a model (W, R, V) and a variable x, φ induces a map

$$\begin{aligned} \varphi_x : \quad & \mathcal{P}(W) \to \mathcal{P}(W) \\ & S \mapsto \{s : \varphi \text{ is true at } s \text{ if } V(x) = S\} \end{aligned}$$

Introduction to μ-calculus

- Syntax: $\top \mid \bot \mid p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid |\Box\varphi \mid \mu x.\varphi \mid \nu x.\varphi$
- Semantic: Given a model (W, R, V) and a variable x, φ induces a map

$$\begin{aligned} \varphi_x : \quad & \mathcal{P}(W) \to \mathcal{P}(W) \\ & S \mapsto \{s : \varphi \text{ is true at } s \text{ if } V(x) = S\} \end{aligned}$$

- Example

Introduction to μ-calculus

- Syntax: $\top \mid \bot \mid p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid |\Box\varphi \mid \mu x.\varphi \mid \nu x.\varphi$
- Semantic: Given a model (W, R, V) and a variable x, φ induces a map

$$\begin{aligned} \varphi_x : \quad & \mathcal{P}(W) \to \mathcal{P}(W) \\ & S \mapsto \{s : \varphi \text{ is true at } s \text{ if } V(x) = S\} \end{aligned}$$

- Example
- Definition: S is a fixpoint of φ_x if $\varphi_x(S) = S$
  $\mathcal{M}, s \models \mu x.\varphi$ if s belongs to the least fixpoint of φ_x

Introduction to μ-calculus

- Semantic: Given a model (W, R, V) and a variable x, φ induces a map

$$\begin{aligned} \varphi_x : \quad & \mathcal{P}(W) \to \mathcal{P}(W) \\ & S \mapsto \{s : \varphi \text{ is true at } s \text{ if } V(x) = S\} \end{aligned}$$

- Definition: S is a fixpoint of φ_x if $\varphi_x(S) = S$
  $\mathcal{M}, s \models \mu x.\varphi$ if s belongs to the least fixpoint of φ_x
- Thm.[Knaster-Tarski] The least fixpoint of φ_x is the set $\bigcap\{S \subset W : \phi_x(S) = S\}$.

Semantics via games

Goal: check that validity of φ at a point s_0 in model (W, R, V)
Initial position: (φ, s_0)

Position	Player	Possible moves
(p, s) $(s \in V(p))$	$\forall$	$\emptyset$
(p, s) $(s \notin V(p))$	$\exists$	$\emptyset$
$(\neg p, s)$ $(s \in V(p))$	$\forall$	$\emptyset$
$(\neg p, s)$ $(s \in V(p))$	$\exists$	$\emptyset$
$(\psi \vee \chi, s)$	$\exists$	$\{(\psi, s), (\chi, s)\}$
$(\psi \wedge \chi, s)$	$\forall$	$\{(\psi, s), (\chi, s)\}$
$(\Diamond\psi, s)$	$\exists$	$\{(\psi, t) : sRt\}$
$(\Box\psi, s)$	$\forall$	$\{(\psi, t) : sRt\}$
$(\mu x.\psi, s)$	-	$\{(\psi.s)\}$
$(\nu x.\psi, s)$	-	$\{(\psi.s)\}$
(x, s)	-	$\{\sigma x.\varphi_x\}$

where $\sigma x.\varphi_x$ is the only surformula of φ starting by μx or νx.

Semantics via games

- the player who is stuck looses
- what happens if the game is infinite?
  give a rank to every variable in φ such that
  - μ-variable have odd ranks
  - ν-varialble have even ranks
  - if $\sigma x.\psi$ is a subformula of $\delta y.\chi$, then x has bigger rank than y

Semantics via games

- the player who is stuck looses
- what happens if the game is infinite?
  give a rank to every variable in φ such that
  - μ-variable have odd ranks
  - ν-varialble have even ranks
  - if $\sigma x.\psi$ is a subformula of $\delta y.\chi$, then x has bigger rank than y

The winning condition is a parity condition: $\exists$ wins if the smallest rank appearing infinitely often is even

Examples

- Reachability: $\mu x.(p \vee \Diamond x)$
- Existential until: $\mu x.p \vee (q \wedge \Diamond x)$
- Almost always p on some path: $\mu y.\nu x.(p \wedge \Diamond x) \vee \Diamond y$

Automata

A μ-automaton is a tuple $A = (Q, q_0, \delta, \Omega)$ where

- Q is the set of states, q_0 is the initial state
- $\delta : Q \times \mathcal{P}(Prop) \to \mathcal{P}\mathcal{P}(Q)$
- $\Omega : Q \to \mathbb{N}$ is the priority map

Automata

A μ-automaton is a tuple $A = (Q, q_0, \delta, \Omega)$ where

- Q is the set of states, q_0 is the initial state
- $\delta : Q \times \mathcal{P}(Prop) \to \mathcal{P}\mathcal{P}(Q)$
- $\Omega : Q \to \mathbb{N}$ is the priority map

The acceptance game for A in (W, R, M) with starting point s_0

Position	Player	Possible moves
$(t, q) \in \mathcal{M} \times Q$	$\exists$	(t, D) s.t. $D \in \delta(q, L(t))$
(t, D)	$\exists$	(t, m) s.t. $m : Q \to Succ(t)$ satisfies - $\forall q' \in D, \exists tRv$ s.t $v \in m(q')$ - $\forall tRv, \exists q' \in D$ s.t $v \in m(q')$
(t, m)	$\forall$	(v, q') s.t. tRv and $v \in m(q')$

Automata

- The player who gets stuck looses
- The winning condition for an infinite match $(s_0, q_0), (s_1, q_1), \ldots$ is a parity condition: $\exists$ wins if the smallest parity appearing infinitely often in the sequence $\Omega(q_0), \Omega(q_1), \ldots$ is even.

Automata

- The player who gets stuck looses
- The winning condition for an infinite match $(s_0, q_0), (s_1, q_1), \ldots$ is a parity condition: $\exists$ wins if the smallest parity appearing infinitely often in the sequence $\Omega(q_0), \Omega(q_1), \ldots$ is even.

Thm [Janin and Walukiewicz]:

- For every formula φ, there is an automaton A such that for all pointed models (M, s), A accets (M, s) iff $M, s \models \varphi$.
- For every automaton A, there is a formula φ such that for all pointed models (M, s), A accets (M, s) iff $M, s \models \varphi$

Results

- there is a complete axiomatization: modal logic + axiomatization of fixpoints (Walukiewicz)

Results

- there is a complete axiomatization: modal logic + axiomatization of fixpoints (Walukiewicz)
- it has some nice properties like the finite model property (Kozen), interpolation property (Hollenberg)

Results

- there is a complete axiomatization: modal logic + axiomatization of fixpoints (Walukiewicz)
- it has some nice properties like the finite model property (Kozen), interpolation property (Hollenberg)
- the satisfiability problem is EXP-time complete (Emerson and Jutla)

Quick introduction to the μ-calculus

Syntactic characterization of semantic fragments of the μ-calculus

Expressivity of GL on finite trees

Syntactic characterization of semantic fragments of the μ-calculus

Known results:

- A μ-formula is monotone iff it is positive (D'Agostino and Hollenberg)
- A μ-formula is preserved under substructures iff it is universal (D'Agostino and Hollenberg)
- A μ-formula is distributive in p iff it is equivalent to a formula of the form $< \Pi > p$, where Π is a p-free μ-program (Hollenberg).

Another semantic fragment

Definition: A formula φ is continuous in p if for all $\mathcal{M} = (M, R, V)$ and all $s \in M$,

$$\mathcal{M}, s \Vdash \varphi \iff \exists\, F \subseteq V(p) \text{ s.t. } F \text{ is finite and } \mathcal{M}[p := F], s \Vdash \varphi.$$

Another semantic fragment

Definition: A formula φ is continuous in p if for all $\mathcal{M} = (M, R, V)$ and all $s \in M$,

$$\mathcal{M}, s \Vdash \varphi \iff \exists\ F \subseteq V(p) \text{ s.t. } F \text{ is finite and } \mathcal{M}[p := F], s \Vdash \varphi.$$

Examples: $\Diamond p$ and $\mu x.\Diamond(x \vee p)$ are continuous
$\Box p$ and $\nu x.\Diamond(x \wedge p)$ are not continuous

Motivations

- it corresponds to a nice algebraic notion: Scott continuity
- this fragment is constructive
- it is linked with PDL

Scott continuity

- in the algebra $(\mathcal{P}(W), \subseteq)$, a Scott open set $\mathcal{O}$ is a family of subsets of of W such that
  - $\mathcal{O}$ is closed under upsets
  - the complement of $\mathcal{O}$ is closed under joins of directed families
- $f : \mathcal{P}(W) \to \mathcal{P}(W)$ is Scott open if for all $\mathcal{O}$ open, $f^{-1}[\mathcal{O}]$ is Scott open

Fact. φ is continuous in p iff the map φ_p induced by φ (with variable p) is Scott continuous

Motivations

- it corresponds to a nice algebraic notion: Scott continuity
- this fragment is constructive
- it is linked with PDL

Definition of constructivity

Recall: φ_x maps $S \subseteq W$ to the set $\{s \in W : M, V[x \mapsto S], s \models \varphi\}$

$M, s \models \varphi$ if s belongs to the least fixpoint of φ_x

Construction of the least fixpoint

$$\varphi_x^\alpha(\emptyset) = \begin{cases} \emptyset & \text{if } \alpha = 0 \\ \varphi_x(\varphi_x^\beta(\emptyset)) & \text{if } \alpha = \beta + 1 \\ \bigcup\{\varphi_x^\beta(\emptyset) : \beta < \alpha\} & \text{otherwise} \end{cases}$$

There is a least α such that $\varphi_x^\alpha(\emptyset) = \varphi_x^{\alpha+1}(\emptyset)$

The least fixpoint of φ_x is $\varphi_x^\alpha(\emptyset)$ and α is the closure ordinal.

Definition of constructivity

Definition: a formula φ is construtive in p if its closure ordinal is ω
That is, in a model M, the least fixpoint of φ_p is $\bigcup\{\varphi_p^n : n \in \omega\}$

Definition of constructivity

Definition: a formula φ is construtive in p if its closure ordinal is ω
That is, in a model M, the least fixpoint of φ_p is $\bigcup\{\varphi_p^n : n \in \omega\}$

Result: If a formula is continuous in p, it is construtive in p.

Definition of constructivity

Definition: a formula φ is construtive in p if its closure ordinal is ω
That is, in a model M, the least fixpoint of φ_p is $\bigcup\{\varphi_p^n : n \in \omega\}$

Result: If a formula is continuous in p, it is construtive in p.

Question: Does continuity imply constructivity?
Counter-examples: $\Box p \wedge \Box\Box\bot$

Definition of constructivity

Definition: a formula φ is construtive in p if its closure ordinal is ω
That is, in a model M, the least fixpoint of φ_p is $\bigcup\{\varphi_p^n : n \in \omega\}$

Result: If a formula is continuous in p, it is construtive in p.

Question: Does continuity imply constructivity?
Counter-examples: $\Box p \wedge \Box\Box\bot$
But is it really a counter-example? $\mu p.(\Box p \wedge \Box\Box\bot) \equiv \mu p.\Box\Box\bot$

Definition of constructivity

Definition: a formula φ is construtive in p if its closure ordinal is ω
That is, in a model M, the least fixpoint of φ_p is $\bigcup\{\varphi_p^n : n \in \omega\}$

Result: If a formula is continuous in p, it is construtive in p.

Question: Does continuity imply constructivity?
Counter-examples: $\Box p \wedge \Box\Box\bot$
But is it really a counter-example? $\mu p.(\Box p \wedge \Box\Box\bot) \equiv \mu p.\Box\Box\bot$
Alternative question: for all constructive formulas φ, is there a continuous formula ψ such that $\mu p.\varphi \equiv \mu p.\psi$

Motivations

- it corresponds to a nice algebraic notion: Scott continuity
- this fragment is constructive
- it is linked with PDL

Link with PDL

- there is a subset Φ of the continuous fragment such that the "language generated" by Φ is PDL
- the inclusion is strict: $\mu x.\Diamond(p \wedge x) \wedge \Diamond(q \wedge x)$ is continuous but not in PDL

Syntactic characterization

Definition. $CF(P)$ is defined by induction by:

$$\varphi \ ::= \ p \mid \psi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid \mu x.\chi,$$

where

- $p \in P$
- no proposition letter of ψ is in P
- $\chi \in CF(P \cup \{x\})$

Syntactic characterization

Theorem. (1) A formula is continuous in p iff it is in $CF(p)$.
(2) It is decidable whether a formula is continuous in p.

Proof

- consider the continuous fragment as the intersection of the finite depth fragment and the finite width fragment

Proof

- consider the continuous fragment as the intersection of the finite depth fragment and the finite width fragment
- for each fragment F, define a transformation T_F mapping an automaton A to an automaton $T_F(A)$ such that

Proof

- consider the continuous fragment as the intersection of the finite depth fragment and the finite width fragment
- for each fragment F, define a transformation T_F mapping an automaton A to an automaton $T_F(A)$ such that
  - φ is in F iff A and $T_F(A)$ are equivalent
  - for every A, $T_F(A)$ corresponds to a formula in the right syntaxic fragment

Proof

- consider the continuous fragment as the intersection of the finite depth fragment and the finite width fragment
- for each fragment F, define a transformation T_F mapping an automaton A to an automaton $T_F(A)$ such that
  - φ is in F iff A and $T_F(A)$ are equivalent
  - for every A, $T_F(A)$ corresponds to a formula in the right syntaxic fragment
- each T_F corresponds to a transformation t_F mapping a formula to a formula and such that φ is in F iff $t_F(\varphi)$ is in the right syntactic fragment

Proof

Example: the finite depth fragment

- transformation for the automata

Proof

Example: the finite depth fragment

- transformation for the automata
- transformation for the formulas:
  $t(\nu x.\psi) = \mu x.(t(\psi) \vee (\nu x.\psi[\bot/p]))$

Questions

- link with constructivity
- can we use the same proof for the result about distributivity (by Hollenberg)?

Quick introduction to the μ-calculus

Syntactic characterization of semantic fragments of the μ-calculus

Expressivity of GL on finite trees

Idea

- Goal: find easy conditions to determine when a MSO formula is equivalent to a GL formulas on finite trees
- Technique: combine the fixpoint theorem for GL (De Jongh, Sambin, van Benthem) with the expressiveness result for the μ-calculus (Janin and Walukiewicz)

Reminder

- GL is the modal logic which is complete w.r.t. to the class of transitive conversely well-founded models

Reminder

- GL is the modal logic which is complete w.r.t. to the class of transitive conversely well-founded models
- MSO (monadic second order logic) is an extension of FO with set quantification

$$xRy \mid \phi \vee \phi \mid \neg\varphi \mid y \in X \mid \exists x.\varphi$$

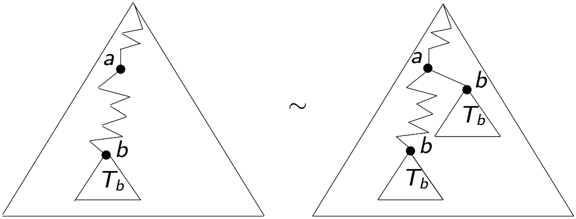
- Semantics: $M, V \models \varphi$:
  $M, V \models y \in X$ if $V(y) \in V(X)$
  $M, V \models \exists X.\varphi$ if there is $S \subseteq M$ s.t. $M, V[X \mapsto S] \models \varphi$

Properties of GL formulas on finite trees

- the truth of a GL formula at a point s only depends of the successors of s
  Notation: T_s is the tree generated by s

Properties of GL formulas on finite trees

- the truth of a GL formula at a point s only depends of the successors of s
  Notation: T_s is the tree generated by s
- a GL formula cannot make a difference between the roots of the two trees

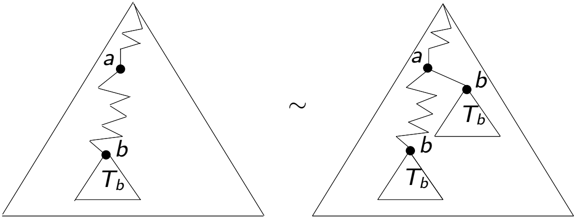


Notation: if T is the tree on the left, we denote by $C_{a,b}(T)$ the tree on the right

Main result

Thm. An MSO formula $\varphi(x)$ is equivalent to a GL formula iff

- for all trees T and points s, $T, s \models \varphi$ iff $T_s, s \models \varphi$
- for all trees T with root r, and with nodes a and b such that b is a descendant of a, $T, r \models \phi$ iff $C_{a,b}(T), r \models \varphi$



Structure of the proof

- show that GL is the transitive bisimulation invariant fragment of MSO on finite trees

  Z is a transitive bisimulation between (W_0, R_0, V_0) and (W_1, R_1, V_1) if Z is a bisimulation between (W_0, R_0^+, V_0) and (W_1, R_1^+, V_1)

Structure of the proof

- show that GL is the transitive bisimulation invariant fragment of MSO on finite trees

  Z is a transitive bisimulation between (W_0, R_0, V_0) and (W_1, R_1, V_1) if Z is a bisimulation between (W_0, R_0^+, V_0) and (W_1, R_1^+, V_1)

- show that a class of finites trees is closed for transitive bisimulations iff it satisfies the two conditions of the previous slides

To show that GL is the transitive bisimulation fragment of MSO on finite trees

Combine these two results:

- Thm [Janin and Walukiewicz] A property is expressible in the μ-calculus iff it is expressible in MSO and bisimulation invariant.

To show that GL is the transitive bisimulation fragment of MSO on finite trees

Combine these two results:

- Thm [Janin and Walukiewicz] A property is expressible in the μ-calculus iff it is expressible in MSO and bisimulation invariant.
- Thm [De Jongh, Sambin, van Benthem] Every formula of GL has a unique fixpoint (w.r.t. a variable x) and this fixpoint is definable in GL.

Extensions of the result

- consider path expressions instead of formulas (which are node expressions)
- replace Godel Lob logic by Grzegorzyk logic