μ-calculus

Formal Methods

Giuseppe De Giacomo

Sapienza Università di Roma
Laurea Magistrale in Ingegneria Informatica

μ-calculus intro

The (modal) μ-calculus is basically constituted by three kinds of components:

- **Propositions** to denote properties of the global store in a given configuration.
- **Modalities** to denote the capability of performing certain actions in a given configuration.
- **Least and greatest fixpoint constructs** to denote "temporal" properties of the system, typically defined by **induction** and **coinduction**.

μ-calculus syntax

Formulae of μ-calculus are formed inductively from action in some fixed set $\mathcal{A}$, primitive (or atomic) propositions in some fixed set $\mathcal{P}$, and variable symbols in some fixed set *Var*, according to the following abstract syntax:

μ-calculus syntax

$$\Phi ::= A \mid \mathit{true} \mid \mathit{false} \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid \langle a \rangle \Phi \mid [a]\Phi \mid \mu X.\Phi \mid \nu X.\Phi \mid X$$

where A is a primitive proposition in $\mathcal{P}$, X is a variable symbol in *Var*, and a is an action in $\mathcal{A}$.
The symbols μ and ν can be considered as quantifiers, and we make use of notions of scope, bound and free occurrences of variables, closed formulas, etc. The definitions of these notions are the same as in first-order logic, treating μ and ν as quantifiers.

μ-calculus syntax

For formulae of the form $\mu X.\Phi$ and $\nu X.\Phi$, we require the **syntactic monotonicity** of Φ wrt X:

Syntactic monotonicity of Φ wrt X

Every occurrence of the variable X in Φ must be within the scope of an even number of negation signs.

Syntactic monotonicity implies **monotonicity**, guaranteeing, by Tarski-Knaster theorem the actual existence of least and greatest fixpoints.

Existence of least and greatest fixpoints

In μ-calculus, given the requirement of syntactic monotonicity, the least fixpoint $\mu X.\Phi$ and the greatest fixpoint $\nu X.\Phi$ always exist.

μ-calculus semantics: transition systems and valuation

The semantics of μ-calculus is based on the notions of **transition system** (i.e., Kripke structure) and variables' **valuation**.

Definition

Transition system Given a set $\mathcal{P}$ of propositions, and set $\mathcal{A}$ of atomic actions, a **transition system** is a triple $\mathcal{T} = (\mathcal{S}, \{\mathcal{R}_a | a \in \mathcal{A}\}, \Pi)$, with a set of states $\mathcal{S}$, a family of transition relations $\mathcal{R}_a \in \mathcal{S} \times \mathcal{S}$, and a mapping Π from $\mathcal{P}$ to subsets of $\mathcal{S}$.

Definition

Valuation Given a transition system $\mathcal{T}$, a **valuation** $\mathcal{V}$ on $\mathcal{T}$ is a mapping from variables in *Var* to subsets of the states in $\mathcal{T}$.

Given a valuation $\mathcal{V}$, we denote by $\mathcal{V}[X \leftarrow \mathcal{E}]$, the valuation identical to $\mathcal{V}$ except for $\mathcal{V}[X \leftarrow \mathcal{E}](X) = \mathcal{E}$, i.e. for every variable Y,

$$\mathcal{V}[X \leftarrow \mathcal{E}](Y) = \begin{cases} \mathcal{E} & \text{if } Y = X \\ \mathcal{V}(Y) & \text{if } Y \neq X \end{cases}$$

μ-calculus semantics: extension function

Let $\mathcal{T} = (\mathcal{S}, \{\mathcal{R}_\alpha | \alpha \in 2^{\mathcal{A}}\}, \Pi)$ be a transition system, and $\mathcal{V}$ a valuation on $\mathcal{T}$. We assign meaning to μ-calculus formulae by associating to $\mathcal{T}$ and $\mathcal{V}$ an **extension function** $(\cdot)^{\mathcal{T}}_{\mathcal{V}}$, which maps μ-calculus formulae to subsets of $\mathcal{S}$.

The extension function $(\cdot)^{\mathcal{T}}_{\mathcal{V}}$ is defined inductively as follows:

μ-calculus semantics

$$\begin{array}{lcl}
(A)^{\mathcal{T}}_{\mathcal{V}} & = & \Pi(A) \subseteq \mathcal{S} \\
(X)^{\mathcal{T}}_{\mathcal{V}} & = & \mathcal{V}(X) \subseteq \mathcal{S} \\
(\mathit{true})^{\mathcal{T}}_{\mathcal{V}} & = & \mathcal{S} \\
(\mathit{false})^{\mathcal{T}}_{\mathcal{V}} & = & \emptyset \\
(\neg\Phi)^{\mathcal{T}}_{\mathcal{V}} & = & \mathcal{S} - (\Phi)^{\mathcal{T}}_{\mathcal{V}} \\
(\Phi_1 \wedge \Phi_2)^{\mathcal{T}}_{\mathcal{V}} & = & (\Phi_1)^{\mathcal{T}}_{\mathcal{V}} \cap (\Phi_2)^{\mathcal{T}}_{\mathcal{V}} \\
(\Phi_1 \vee \Phi_2)^{\mathcal{T}}_{\mathcal{V}} & = & (\Phi_1)^{\mathcal{T}}_{\mathcal{V}} \cup (\Phi_2)^{\mathcal{T}}_{\mathcal{V}} \\
(\langle a \rangle \Phi)^{\mathcal{T}}_{\mathcal{V}} & = & \{s \in \mathcal{S} \mid \exists s'.\ (s, s') \in \mathcal{R}_a \text{ and } s' \in (\Phi)^{\mathcal{T}}_{\mathcal{V}}\} \\
([a]\Phi)^{\mathcal{T}}_{\mathcal{V}} & = & \{s \in \mathcal{S} \mid \forall s'.\ (s, s') \in \mathcal{R}_a \text{ implies } s' \in (\Phi)^{\mathcal{T}}_{\mathcal{V}}\} \\
(\mu X.\Phi)^{\mathcal{T}}_{\mathcal{V}} & = & \bigcap\{\mathcal{E} \subseteq \mathcal{S} \mid \ (\Phi)^{\mathcal{T}}_{\mathcal{V}[X \leftarrow \mathcal{E}]} \subseteq \mathcal{E}\ \} \\
(\nu X.\Phi)^{\mathcal{T}}_{\mathcal{V}} & = & \bigcup\{\mathcal{E} \subseteq \mathcal{S} \mid \ \mathcal{E} \subseteq\ (\Phi)^{\mathcal{T}}_{\mathcal{V}[X \leftarrow \mathcal{E}]}\}
\end{array}$$

μ-calculus semantics: observations

Note that, the semantics shows that not all μ-calculus constructs are independent. In particular, we have:

- The usual boolean abbreviations: $\mathit{false} = A \wedge \neg A$; $\mathit{true} = \neg \mathit{false}$; $\Phi_1 \vee \Phi_2 = \neg(\neg\Phi_1 \wedge \neg\Phi_2)$; and also $\Phi_1 \supset \Phi_2 = \neg\Phi_1 \vee \Phi_2$.
- $[\varrho]\Phi = \neg\langle\varrho\rangle\neg\Phi$;
- $\nu X.\Phi = \neg\mu X.\neg\Phi[X/\neg X]$ where $\Phi[X/\neg X]$ is the formula obtained by substituting all free occurrences of X by the formula $\neg X$.

Note also that if Φ is closed (no free variables are present in Φ) then the extension of $(\Phi)^{\mathcal{T}}_{\mathcal{V}}$ is in fact independent of the valuation $\mathcal{V}$ so we could write $(\Phi)^{\mathcal{T}}$, dropping any reference to $\mathcal{V}$. It is usual to say that **a closed Φ is true in a state s of the transition system $\mathcal{T}$** iff $s \in (\Phi)^{\mathcal{T}}$.

Formally $s \in (\Phi)^{\mathcal{T}}$ stands or $s \in (\Phi)^{\mathcal{T}}_{\mathcal{V}}$ for every valuation $\mathcal{V}$ (the extension of Φ is in fact independent of $\mathcal{V}$ with Φ closed).

μ-calculus semantics: intuition

Intuitively, the extension function $(\cdot)^{\mathcal{T}}_{\mathcal{V}}$ assigns to the various constructs of μ-calculus the following meanings:

Intuition on $(\cdot)^{\mathcal{T}}_{\mathcal{V}}$

- The boolean connectives have the expected meaning.
- The extension of $\langle a\rangle\Phi$ includes the states $s \in \mathcal{S}$ such that starting from s, there is an execution of action a that leads to a successive state s' included in the extension of Φ.
- The extension of $[a]\Phi$ includes the states s such that starting from s, each execution of action a leads to some successive state s' included in the extension of Φ.

μ-calculus semantics: intuition

For the fixpoint constructs we have:

Intuition on $(\mu X.\Phi)^{\mathcal{T}}_{\mathcal{V}}$ and $(\nu X.\Phi)^{\mathcal{T}}_{\mathcal{V}}$

- The extension of $\mu X.\Phi$ is the **smallest subset** $\mathcal{E}_\mu$ of $\mathcal{S}$ such that, assigning to X the extension $\mathcal{E}_\mu$, the resulting extension of Φ is contained in $\mathcal{E}_\mu$. That is, the extension of $\mu X.\Phi$ is the **least fixpoint** of the operator $\lambda\mathcal{E}.(\Phi)^{\mathcal{T}}_{\mathcal{V}[X\leftarrow\mathcal{E}]}$.
- Similarly, the extension of $\nu X.\Phi$ is the **greatest subset** $\mathcal{E}_\nu$ of $\mathcal{S}$ such that, assigning to X the extension $\mathcal{E}_\nu$, the resulting extension of Φ contains $\mathcal{E}_\nu$. That is, the extension of $\nu X.\Phi$ is the **greatest fixpoint** of the operator $\lambda\mathcal{E}.(\Phi)^{\mathcal{T}}_{\mathcal{V}[X\leftarrow\mathcal{E}]}$.

The syntactic monotonicity of Φ wrt X guarantees the monotonicity of the operator $\lambda\mathcal{E}.(\Phi)^{\mathcal{T}}_{\mathcal{V}[X\leftarrow\mathcal{E}]}$ and hence, by Tarski-Knaster Theorem, the unique existence of the least fixpoint.

μ-calculus: examples

Let us consider the case we have a single action *next* represent generic transitions. Then:

Example

$$\langle next\rangle true$$

expresses the capability of making a *next*-transition

Example

$$[next]false$$

expresses the inability of executing any *next*-transition.

Example

$$\langle next\rangle true \wedge [next]P$$

says that *next*-transitions are allowed and they all reach states where P holds.

μ-calculus: examples

Example

$$\mu X.P \vee \langle next \rangle X$$

expresses that there **exists an evolution** of the system such that P **eventually** holds. Indeed, its extension $\mathcal{E}_\mu$ is the smallest set that includes (1) the states in the extension of Φ; and (2) the states that can execute a transition leading to a successive state that is in $\mathcal{E}_\mu$. In other words, the extension $\mathcal{E}_\mu$ includes each state s such that there exists a run from s leading eventually (i.e. in a finite number of steps) to a state in the extension of P. Note the inductive nature of this property which is typical of properties defined by least fixpoint.

μ-calculus: examples

Example

$$\nu X.P \wedge [next] X$$

i.e. $\neg(\mu X.\neg P \vee \langle next \rangle X)$ – expresses the **invariance** of P under all of the evolutions of the system. Indeed, its extension $\mathcal{E}_\nu$ is the largest set of states in the extension of P from which every transition leads to a successive state which is still in $\mathcal{E}_\nu$. In other words, the extension $\mathcal{E}_\nu$ includes each state s such that every state along every run from s is in the extension of P. Note the coinductive nature of this property which is typical of properties defined by greatest fixpoint.

μ-calculus: examples

Example

$$\mu X.P \vee (\langle next \rangle true \wedge [next]X)$$

expresses that for **all evolutions** of the system, P **eventually** holds. Indeed, its extension $\mathcal{E}_\mu$ is the smallest set that includes (1) the states in the extension of P; and (2) the states that can make a transition and such that every transition leads to a state in $\mathcal{E}_\mu$. In other words, the extension $\mathcal{E}_\mu$ includes each state s such that every run from s leads eventually (i.e. in a finite number of steps) to a state in the extension of P.

Example

$$\nu X.\mu Y.(P \wedge \langle next \rangle X) \vee (\langle next \rangle Y)$$

expresses a **strong fairness** of a run: there exists a run where P is true infinitely often.

In general, μ-calculus allows for expressing very sophisticated properties of dynamic systems, such as very general forms of **liveness**, **safety**, *and* **fairness**.

μ-calculus: example (program correctness)

Let us still consider a single action *next*, which represent the execution of a transition, and an atomic predicate *Final* representing states that are final configurations. Moreover let $\mathcal{T}_\delta$ be the transition system generated by a program δ.

Example

$$\nu X.(Final \wedge Q) \vee (\langle next \rangle true \wedge [next]X)$$

denotes the so-called **weakest (liberal) precondition** $WLP(\delta, Q)$ for Q for the program δ. We can check **Hoare partial correctness** $\{P\}\delta\{Q\}$ by checking over $\mathcal{T}_\delta$ the following formula:

$$P \supset \nu X.(Final \wedge Q) \vee (\langle next \rangle true \wedge [next]X)$$

Example

$$\mu X.(Final \wedge Q) \vee (\langle next \rangle true \wedge [next]X)$$

denotes the so-called **(proper) weakest precondition** $WP(\delta, Q)$ for Q for the program δ. We can check **total correctness**, i.e., $[P]\delta[Q]$ by checking over $\mathcal{T}_\delta$ the following formula:

$$P \supset \mu X.(Final \wedge Q) \vee (\langle next \rangle true \wedge [next]X)$$

μ-calculus: simple properties

Often, we use the notation $\Phi(X)$ to indicate that the variable X occurs free in the formula Φ (other variables could occur free in Φ as well), and the notation $\Phi(\Psi)$, where Ψ is a formula, as a shorthand for the formula obtained by syntactically substituting all free occurrences of X in $\Phi(X)$ by the concept Ψ).

Simple properties

Below σ stands for μ or ν

- $\sigma X.\Phi(X)$ is equivalent to $\sigma Y.\Phi(Y)$, as long as Y is free for X in $\Phi(X)$.
- $\sigma X.\Phi$ and X does not occur in Φ, then $\sigma X.\Phi$ equivalent to Φ.
- $\Phi(\sigma X.\Phi(X))$ is equivalent to $\sigma X.\Phi(X)$, indeed $\sigma X.\Phi(X)$ is a fixpoint.
- $\mu X.\Phi(X)$ logically implies $\nu X.\Phi(X)$, indeed the least fixpoint is always smaller/equal to the greatest fixpoint.

μ-calculus: model checking

The reasoning problem we are interested in is **model checking**:

Definition

Let $\mathcal{T} = (\mathcal{S}, \{\mathcal{R}_a \mid a \in \mathcal{A}\}, \Pi)$ be a transition system, let $s \in \mathcal{S}$ be one of its states, and let Φ be a closed (no free variables are present) μ-calculus formula. The related **model checking** problem is to verify whether

$$s \in (\Phi)^{\mathcal{T}}_{\mathcal{V}}$$

where $\mathcal{V}$ is any valuation, since Φ is closed.

Often we abbreviate $s \in (\Phi)^{\mathcal{T}}_{\mathcal{V}}$ by $\mathcal{T}, s \models \Phi$ or simply by $s \models \Phi$ referring to $\mathcal{T}$ only implicitly.

μ-calculus: complexity of reasoning

Theorem

Checking (closed) a μ-calculus formula Φ over a transition system $\mathcal{T} = (\mathcal{S}, \{\mathcal{R}_a \mid a \in \mathcal{A}\}, \Pi)$ can be done in time

$$O((|\mathcal{T}| \cdot |\Phi|)^k)$$

where $|\mathcal{T}| = |\mathcal{S}| + \Sigma_{a \in \mathcal{A}} |\mathcal{R}_a|$, i.e., the number of states plus the number of transitions of $\mathcal{T}$, $|\Phi|$ is the size of formula Φ (in fact, considering propositional formulas as atomic), and k is the number of nested fixpoints, i.e., fixpoints whose variables are one within the scope of the other.

Also, in general model checking is in $NP \cap coNP$.

Theorem

Checking satifiability/validity/logical implication in μ-calculus is decidable and more precisely EXPTIME-complete.

μ-calculus: model checking algorithm

Given a μ-calculus formula Φ over a transition system $\mathcal{T} = (\mathcal{S}, \{\mathcal{R}_a \mid a \in \mathcal{A}\}, \Pi)$ and a valuation $\mathcal{V}$, the **model checking algorithm** is based on recursively **labeling the states** of the transition systems with the formulas that are true in them, following closely the semantics.

μ-calculus model checking algorithm

$$\begin{aligned}
[\![A]\!]^{\mathcal{T}}_{\mathcal{V}} &= \Pi(A) \\
[\![X]\!]^{\mathcal{T}}_{\mathcal{V}} &= \mathcal{V}(X) \\
[\![true]\!]^{\mathcal{T}}_{\mathcal{V}} &= \mathcal{S} \\
[\![false]\!]^{\mathcal{T}}_{\mathcal{V}} &= \emptyset \\
[\![\neg\Phi]\!]^{\mathcal{T}}_{\mathcal{V}} &= \mathcal{S} - [\![\Phi]\!]^{\mathcal{T}}_{\mathcal{V}} \\
[\![\Phi_1 \wedge \Phi_2]\!]^{\mathcal{T}}_{\mathcal{V}} &= [\![\Phi_1]\!]^{\mathcal{T}}_{\mathcal{V}} \cap [\![\Phi_2]\!]^{\mathcal{T}}_{\mathcal{V}} \\
[\![\Phi_1 \vee \Phi_2]\!]^{\mathcal{T}}_{\mathcal{V}} &= [\![\Phi_1]\!]^{\mathcal{T}}_{\mathcal{V}} \cup [\![\Phi_2]\!]^{\mathcal{T}}_{\mathcal{V}}
\end{aligned}$$

$$\begin{aligned}
[\![\langle a \rangle \Phi]\!]^{\mathcal{T}}_{\mathcal{V}} &= \textsc{PreE}(a, [\![\Phi]\!]^{\mathcal{T}}_{\mathcal{V}}) \\
[\![[a]\Phi]\!]^{\mathcal{T}}_{\mathcal{V}} &= \textsc{PreA}(a, [\![\Phi]\!]^{\mathcal{T}}_{\mathcal{V}})
\end{aligned}$$

$$\begin{aligned}
[\![\mu X.\Phi]\!]^{\mathcal{T}}_{\mathcal{V}} &= \textsc{lfp} X.[\![\Phi]\!]^{\mathcal{T}}_{\mathcal{V}} \\
[\![\nu X.\Phi]\!]^{\mathcal{T}}_{\mathcal{V}} &= \textsc{gfp} X.[\![\Phi]\!]^{\mathcal{T}}_{\mathcal{V}}
\end{aligned}$$

where $\textsc{PreE}, \textsc{PreA}, \textsc{gfp}, \textsc{lfp}$ are defined below.

For the atomic propositions, variables and propositional operator the labeling works in an obvious way.

μ-calculus: model checking algorithm

Let $\mathcal{E} \subseteq \mathcal{S}$ be a set of state and $a \in \mathcal{A}$ an action. Then $\textsc{PreE}$ and $\textsc{PreA}$ label the existential and universal a-preimage of $\mathcal{E}$ respectively.

Existential a-preimage of $\mathcal{E}$

$\textsc{PreE}(a, \mathcal{E})$, i.e., the **existential a-preimage of $\mathcal{E}$**, is defined as follows:

$$\textsc{PreE}(a, \mathcal{E}) = \{s \in \mathcal{S} \mid \exists s'.\ (s, s') \in \mathcal{R}_a \text{ and } s' \in \mathcal{E}\}$$

Universal a-preimage of $\mathcal{E}$

$\textsc{PreA}(a, \mathcal{E})$, i.e., the **universal a-preimage of $\mathcal{E}$**, is defined as follows:

$$\textsc{PreA}(a, \mathcal{E}) = \{s \in \mathcal{S} \mid \forall s'.\ (s, s') \in \mathcal{R}_a \text{ implies } s' \in \mathcal{E}\}$$

Notice the preimage operators follow the semantics of the $\langle a \rangle \cdot$ and $[a] \cdot$ very closely.

μ-calculus: model checking algorithm

Procedures $\textsc{lfp}X.[\![\Phi]\!]^{\mathcal{T}}_{\mathcal{V}}$ and $\textsc{gfp}X.[\![\Phi]\!]^{\mathcal{T}}_{\mathcal{V}}$ apply Tarski-Knaster approximates theorem to compute **least fixpoint** and **greatest fixpoint** of operator $[\![\Phi]\!]^{\mathcal{T}}_{\mathcal{V}}$:

Procedure $\textsc{lfp}X.[\![\Phi]\!]^{\mathcal{T}}_{\mathcal{V}}$

$\mathcal{X}_{old} := [\![\mathit{False}]\!]^{\mathcal{T}}_{\mathcal{V}}$;
$\mathcal{X} := [\![\Phi]\!]^{\mathcal{T}}_{\mathcal{V}[X \leftarrow \mathcal{X}_{old}]}$;
while $(\mathcal{X} \neq \mathcal{X}_{old})$ {
  $\mathcal{X}_{old} := \mathcal{X}$;
  $\mathcal{X} := [\![\Phi]\!]^{\mathcal{T}}_{\mathcal{V}[X \leftarrow \mathcal{X}_{old}]}$;
}
return $\mathcal{X}$;

Procedure $\textsc{gfp}X.[\![\Phi]\!]^{\mathcal{T}}_{\mathcal{V}}$

$\mathcal{X}_{old} := [\![\mathit{True}]\!]^{\mathcal{T}}_{\mathcal{V}}$;
$\mathcal{X} := [\![\Phi]\!]^{\mathcal{T}}_{\mathcal{V}[X \leftarrow \mathcal{X}_{old}]}$;
while $(\mathcal{X} \neq \mathcal{X}_{old})$ {
  $\mathcal{X}_{old} := \mathcal{X}$;
  $\mathcal{X} := [\![\Phi]\!]^{\mathcal{T}}_{\mathcal{V}[X \leftarrow \mathcal{X}_{old}]}$;
}
return $\mathcal{X}$;

*Notice the number of interations of the **while** is at most equal to the number of states $\mathcal{S}$ of the transition system $\mathcal{T}$.*