REPRESENTING GUARDEDNESS IN CALL-BY-VALUE

SERGEY GONCHAROV
FAU ERLangen-NÜRNberg

FSCD 2023, ROME, JULY 5, 2023
How do we know that automata are equivalent?
Equation $(ab)^* = a(ba)^*b + 1$ is true, because $a(ba)^*b + 1$ is a fixpoint of the same map.
Equation \((ab)^* = a(ba)^*b + 1\) is true, because \(a(ba)^*b + 1\) is a fixpoint of the same map\(^1\):

\[
a(ba)^*b + 1 = a((ba)(ba)^* + 1)b + 1
\]

\(^1\)A. Salomaa, Two Complete Axiom Systems for the Algebra of Regular Events, 1966
Proof "by Coinduction"

Equation $(ab)^* = a(ba)^*b + 1$ is true, because $a(ba)^*b + 1$ is a fixpoint of the same map:\(^1\)

$$a(ba)^*b + 1 = a((ba)(ba)^* + 1)b + 1$$
$$= a(ba)(ba)^*b + a1b + 1$$

\(^1\)A. Salomaa, Two Complete Axiom Systems for the Algebra of Regular Events, 1966
Equation \((ab)^* = a(ba)^*b + 1\) is true, because \(a(ba)^*b + 1\) is a fixpoint of the same map\(^1\):

\[
a(ba)^*b + 1 = a((ba)(ba)^* + 1)b + 1 \\
= a(ba)(ba)^*b + a1b + 1 \\
= (ab)a(ba)^*b + ab + 1
\]

\(^1\)A. Salomaa, Two Complete Axiom Systems for the Algebra of Regular Events, 1966
Equation $ (ab)^* = a(ba)^* b + 1 $ is true, because $ a(ba)^* b + 1 $ is a fixpoint of the same map\(^1\):

\[
\begin{align*}
(a(ba)^* b + 1) &= a((ba)(ba)^* + 1)b + 1 \\
n &= a(ba)(ba)^* b + a1b + 1 \\
n &= (ab)a(ba)^* b + ab + 1 \\
n &= (ab)(a(ba)^* b + 1) + 1 
\end{align*}
\]

---

\(^1\)A. Salomaa, Two Complete Axiom Systems for the Algebra of Regular Events, 1966
Equation $(ab)^* = a(ba)^* b + 1$ is true, because $a(ba)^* b + 1$ is a fixpoint of the same map:\(^1\):

$$a(ba)^* b + 1 = a((ba)(ba)^* + 1)b + 1$$
$$= a(ba)(b)^*b + a1b + 1$$
$$= (ab)a(ba)^*b + ab + 1$$
$$= (ab)(a(ba)^*b + 1) + 1$$

\(^1\)A. Salomaa, Two Complete Axiom Systems for the Algebra of Regular Events, 1966
Equation \((ab)^* = a(ba)^*b + 1\) is true, because \(a(ba)^*b + 1\) is a fixpoint of the same map:

\[
\begin{align*}
(a(ba)^*b + 1) &= a((ba)(ba)^* + 1)b + 1 \\
&= a(ba)(ba)^*b + a1b + 1 \\
&= (ab)a(ba)^*b + ab + 1 \\
&= (ab)(a(ba)^*b + 1) + 1
\end{align*}
\]

- This only works because \(x \mapsto abx + 1\) is guarded
- \(x \mapsto (a + 1)x + 1\) is un-guarded and has infinitely many fixpoints

\(^1\)A. Salomaa, Two Complete Axiom Systems for the Algebra of Regular Events, 1966
Bouncing ball is a simple Newtonian system specified by differential equation
\[ \ddot{h} = -g \left( g \approx 9.8 \right) \]
whose solution is
\[ h(t) = h_0 + v_0 t - \frac{gt^2}{2} \]
with initial values:
- \( v_0 = 0, h_0 \neq 0 \) (peak height)
- \( h_0 = 0, v_0 \neq 0 \) (zero height)

This system is **progressive**: every iteration consumes non-zero time (although it keeps getting smaller – Zeno behaviour)

Non-progressive (chattering) behaviour is often regarded a modelling artefact
**Scenario # 3: Process Algebra**

Basic Process Algebra (BPA):

\[ P, Q, \ldots := \exists a \in A \mid P + Q \mid P \cdot Q \]

E.g. we can specify a 2-cell FIFO, storing bits:

\[
\begin{align*}
B_0 &= \text{in}_0 \cdot B_1^0 + \text{in}_1 \cdot B_1^1 \\
B_1^i &= \text{in}_0 \cdot B_2^{0,i} + \text{in}_1 \cdot B_2^{1,i} + \text{out}_i \cdot B_0 \\
B_2^{i,j} &= \text{out}_j \cdot B_1^i
\end{align*}
\]

\( i \in \{0, 1\} \)

Solutions are unique for **guarded** specifications. Otherwise not: \( X = X \) has infinitely many solutions
We can model previous examples with monads, augmented with partially defined iteration operators

\[
\begin{align*}
  f &: X \to T(Y + X) \\
  f^\dagger &: X \to TY
\end{align*}
\]

w.r.t. a co-Cartesian category (=category with finite coproducts)

1. Automata: \( TX = \mathcal{P}(A^* \times X) \)
2. Hybrid time: \( TX = \mathbb{R}_{\geq 0} \times X + \mathbb{R}_{\geq 0} \)
3. BPA: \( TX = \nu \gamma. \mathcal{P}_{\omega_1} (X + A \times \gamma) \) (final \( F \)-coalgebra)

Note that a monad carries information about computational effects, but not about guardedness
Most of time, guarded fixpoints are restrictions of unguarded ones. But the guarded ones are better behaved:

- Often unique, hence enable reasoning by coinduction
- If not unique, often computed as least fixpoints
- Foundation-independent
- Simpler to define and to work with

This motivates a type discipline for propagating guardedness over structures
Iteration and Recursion
Iteration vs. Recursion

- **Iteration operator:**

\[
f : X \to Y + X \\
\hat{f} : X \to Y
\]

- Dually: **recursion operator:**

\[
f : \Gamma \times X \to X \\
\hat{f} : \Gamma \to X
\]

equivalently: \( \text{fix} : (X \to X) \to X \), e.g. in the \( \lambda \)-calculus

- **Guarded recursion:** \( \text{fix} : (\triangleright X \to X) \to X \)
  - Curry-Howard counterpart of the Löb rule
  - Familiar model: topos of trees \( \text{Set}^{\omega_{\text{op}}} \)
  - Notion of guardedness is **representable:** \( f : \Gamma \times X \to Y \) is guarded (=contractive) iff \( f \) factors as

\[
\begin{align*}
\Gamma \times X & \xrightarrow{f} X \\
\Gamma \times \text{next} & \downarrow \\
\Gamma \times \triangleright X &
\end{align*}
\]

---

Problem

Sticking to iteration, can we generally define representable guardedness?

- Maybe (?) we need an endofunctor $\triangleright$, and then $f : X \to Y + X$ is guarded if it factors

$$
X \xrightarrow{f} Y + X
$$

- Example: $f : X \to (Y + X) \times \mathbb{N} \cong Y \times \mathbb{N} + X \times \mathbb{N}$ is guarded iff it factors through $(Y \times \mathbb{N} + X \times \text{suc})$

- However, e.g. $f : X \to (Y + X)^*$ should be guarded if in every $f(x) = [e_1, \ldots, e_n]$ every $e_n \in X$ is preceded by some $e_k \in Y$

  $\Rightarrow \triangleright$ may depend both on $X$ and on $Y$
An identity-on-object functor \( J : V \rightarrow C \) has a right adjoint iff

- \( C \) is isomorphic to Kleisli category of a monad on \( V \)
- all presheaves \( C(J, A) : V^{op} \rightarrow \text{Set} \) are representable

Fine-grain call-by-value\(^4\) was interpreted over Freyd categories, which are certain identity-on-object functors \( J : V \rightarrow C \) where

- \( V \) is a category of values
- \( C \) is a category of computations

All \( J(- \times A) : V \rightarrow C \) have right adjoints iff

- \( C \) is isomorphic to a Kleisli category of a strong monad \( T \), and all Kleisli exponentials \( B^{TA} \) exist
- all presheaves \( C(J(- \times A), B) : V^{op} \rightarrow \text{Set} \) are representable

Here: representability of guardedness in fine-grain call-by-value

---

\(^3\)D. Schumacher, Minimale und Maximale Tripelerzeugende und eine Bemerkung zur Tripelbarkeit, 1969

\(^4\)P. Levy, J. Power, H. Thielecke, Modelling Environments in Call-By-Value Programming Languages, 2002
REPRESENTING GUARDEDNESS
A guardedness predicate identifies for all objects $X, Y, Z$ guarded morphisms $\mathbf{C} \cdot (X, Y, Z) \subseteq \mathbf{C}(X, Y + Z)$, such that

\[
\begin{align*}
\text{(trv}_{+}) & \quad f : X \to Y \\
\text{inl} \ f : X \to Y \triangleright Z
\end{align*}
\]

\[
\begin{align*}
\text{(par}_{+}) & \quad f : X \to V \triangleright W \\
g : Y \to V \triangleright W \\
[f, g] : X + Y \to V \triangleright W
\end{align*}
\]

\[
\begin{align*}
\text{(cmp}_{+}) & \quad f : X \to Y \triangleright Z \\
g : Y \to V \triangleright W \\
h : Z \to V + W \\
[g, h] f : X \to V \triangleright W
\end{align*}
\]

where $f : X \to Y \triangleright Z$ means $f \in \mathbf{C} \cdot (X, Y, Z)$

- A category with a guardedness predicate is called guarded
- A monad is guarded if its Kleisli category is guarded
**Examples**

- $f : X \to \mathcal{P}(A^* \times (Y + Z))$ is guarded if it factors through
  $$\mathcal{P}(A^* \times Y + A^+ \times Z) \hookrightarrow \mathcal{P}(A^* \times Y + A^* \times Z)$$
  $$\cong \mathcal{P}(A^* \times (Y + Z))$$

- $f : X \to \mathbb{R}_{\geq 0} \times (Y + Z) + \mathbb{R}_{\geq 0}$ is guarded if it factors through
  $$\mathbb{R}_{\geq 0} \times Y + \mathbb{R}_{\geq 0} \times Z + \mathbb{R}_{\geq 0} \hookrightarrow \mathbb{R}_{\geq 0} \times Y + \mathbb{R}_{\geq 0} \times Z + \mathbb{R}_{\geq 0}$$
  $$\cong \mathbb{R}_{\geq 0} \times (Y + Z) + \mathbb{R}_{\geq 0}$$

- $f : X \to \nu \gamma. T((Y + Z) + H \gamma)$ is guarded if it factors through
  $$T(Y + H(\nu \gamma. \ldots)) \hookrightarrow T((Y + Z) + H(\nu \gamma. \ldots))$$
  $$\cong \nu \gamma. T((Y + Z) + H \gamma)$$
Call-by-Value with Effects
**Very Simple Metalanguage (VSML)**

- **Sorts** $A, B, C, \ldots$
- **Signatures** $\Sigma_v, \Sigma_c$ of pure and effectful programs $f: A \to B$
- **Semantics** of $(\Sigma_v, \Sigma_c)$ w.r.t. identity-on-objects functor $J: V \to C$:
  - an object $\llbracket A \rrbracket \in |V|$ to each sort $A$
  - a morphism $\llbracket f \rrbracket \in V(\llbracket A \rrbracket, \llbracket B \rrbracket)$ to each $f: A \to B \in \Sigma_v$
  - a morphism $\llbracket f \rrbracket \in C(\llbracket A \rrbracket, \llbracket B \rrbracket)$ to each $f: A \to B \in \Sigma_c$
- **Terms in single-variable (!) context:**

  $f: A \to B \in \Sigma_v \quad \Gamma \vdash_v v: A$  
  \[ \Gamma \vdash_v f(v): B \]  
  \[ x: A \vdash_v x: A \quad \Gamma \vdash_v v: A \]  
  \[ \Gamma \vdash_c \text{return } v: A \]

  $f: A \to B \in \Sigma_c \quad \Gamma \vdash_v v: A$  
  \[ \Gamma \vdash_c f(v): B \]  
  \[ \Gamma \vdash_c p: A \quad x: A \vdash_c q: B \]  
  \[ \Gamma \vdash_c x \leftarrow p; q: B \]

- $\llbracket - \rrbracket$ extends easily
The fine-grain call-by-value (FGCBV) is obtained by enabling multivariable contexts $\Gamma = (x_1 : A_1, \ldots, x_n : A_n)$, e.g. variable term formation:

$$x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i$$

FGCBV can be interpreted over a Freyd category:

- $V$ is a category with finite products
- **action** $V \times C \to C$ of $V$ on $C$
- $J : V \to C$ is an identity-on-objects functor, preserving the action
Originally, Moggi\textsuperscript{5} interpreted call-by-value over strong monads. $T$ is strong if it comes with strength:

$$\tau : X \times TY \rightarrow T(X \times Y)$$

which satisfies a number of coherence conditions. We then can interpret

$$f := [\Gamma \vdash c_{p : A}] \quad g := [\Gamma, x : A \vdash c_{q : B}]$$

$$[\Gamma \vdash c_{x \leftarrow p ; q : B}] : [\Gamma] \xrightarrow{\langle \Gamma, f \rangle} [\Gamma] \times T[A] \xrightarrow{\tau} T[\Gamma \times A] \xrightarrow{g^*} T[B]$$

\textsuperscript{5}E. Moggi, Notions of Computation and Monads, 1991
If we want to implement higher order:

\[
\frac{\Gamma, x : A \vdash c \, p : B}{\Gamma \vdash \lambda x. \, p : A \rightarrow B}
\]

we need to have a semantics \([A \rightarrow B] = U([A], [B])\), such that

\[C(J(X \times A), B) \cong V(X, U(A, B))\]

naturally in \(A\)

---

**Theorem (6)**

The following are equivalent:

1. \(C(J(X \times A), B) \cong V(X, U(A, B))\) for some \(U : V \times C \rightarrow V\), naturally in \(A\)
2. Presheaves \(C(J(X \times (-)), B) : V^{op} \rightarrow Set\) are representable
3. \(C\) is isomorphic to a Kleisli category of a strong monad \(T\) on \(V\) and all exponentials \((TB)^A\) exist

---

6Essentially: P. Levy, J. Power, H. Thielecke, Modelling Environments in Call-By-Value Programming Languages, 2002
If we do not care about strength, we have a simpler characterization

**Theorem (Schumacher)**

Given id-on-objects functor $J: V \rightarrow C$, the following are equivalent:
- $J$ is a left adjoint
- Presheaves $C(J(-), B): V^{\text{op}} \rightarrow \text{Set}$ are representable
- $C$ is isomorphic to a Kleisli category of a monad

and then:

**Theorem**

Given a Freyd category $J: V \rightarrow C$, the following are equivalent:
- $J$ is a left adjoint
- Presheaves $C(J(-), B): V^{\text{op}} \rightarrow \text{Set}$ are representable
- $C$ is isomorphic to a Kleisli category of a **strong** monad
**Intermediate Summary**

- **Monads** → **Strong Monads**
- **Id.-on-obj. Functors** → **Freyd Categories**
- **Representability**
- **Strength**
Call-by-Value Meets Guardedness
**Definition**

Given $J: V \to C$, as before and guarded $C$, call the guardedness predicate $C_\bullet$ \textit{(J-)representable} if for all $A, B \in |C|$ the presheaves

$$C_\bullet(J(-), A, B): V^{op} \to \text{Set}$$

are representable

Note that $C_\bullet(X, A, \emptyset) \cong C(X, A)$, hence

**Lemma**

\textit{If} $C_\bullet$ \textit{is representable,} $J$ \textit{is a left adjoint. In this case, $C$ is a Kleisli category of some monad on $V$}
GUARDED PARAMETRIZED MONADS

Recall that a bifunctor \( \# : V \times V \to V \) is a **parametrized monad**\(^7\) if

- Every \((-) \# X\) is a monad
- Every \((-) \# f\) is a monad morphism

**Definition**

A **guarded parametrized monad** on a symmetric monoidal \((V, \otimes, I, \rho, \lambda, \alpha, \gamma)\) consists of a bifunctor \(\# : V \times V \to V\), natural transformations

\[
\eta : A \to A \# I \\
\upsilon : A \# (B \otimes C) \to (A \otimes B) \# C \\
\xi : (A \# B) \# C \to A \# (B \otimes C) \\
\chi : A \# B \otimes C \# D \to (A \otimes C) \# (B \otimes D) \\
\zeta : A \# (B \# C) \to A \# (B \otimes C)
\]

plus a bunch of commutative diagrams

- **Intuition:** in \(X \# Y\), \(Y\) is the guarded part, \(X\) is (possibly) unguarded part
- **Coherence property:** if \(f, g : \mathcal{E}_1 \to \mathcal{E}_2 \# \mathcal{E}'_2\) and
  
  - \(f, g\) are made of \(\eta, \epsilon, \upsilon, \xi, \zeta, \rho, \lambda, \alpha, \gamma, \text{id} , \otimes, \#\)
  - object letters do not repeat either in \(\mathcal{E}_1\) or in \(\mathcal{E}_2 \# \mathcal{E}'_2\)
  - \(\mathcal{E}_2\) and \(\mathcal{E}'_2\) do not contain \(\#\)

  then \(f = g\)

\(^7\)T. Uustalu, Generalizing Substitution, 2003
Theorem

Given co-Cartesian $\mathbf{V}$ and an identity-on-object functor $J : \mathbf{V} \to \mathbf{C}$ strictly preserving coproducts, $\mathbf{C}$ is guarded and $\mathbf{C}_*$ is representable iff

- $\mathbf{C} \cong \mathbf{V}_{-\#0}$ for a guarded parametrized monad $(\#, \eta, \upsilon, \chi, \xi, \zeta)$
- the compositions

$$X \# Y \cong X \# (Y + \emptyset) \xrightarrow{\upsilon_{X,Y,\emptyset}} (X + Y) \# \emptyset$$

are all monic and

- $f : X \to Y \triangleright Z$ iff $f$ factors through $Y \# (Z + 0) \xrightarrow{\upsilon} (Y + Z) \# \emptyset$

Analogously, representability of

$$\mathbf{C}_*(J(- \times A), B, C) : \mathbf{V}^{\text{op}} \to \mathbf{Set}$$

produces strong guarded parametrized monads
Examples

- Least guardedness: $X \# Y = TX$
- Greatest guardedness: $X \# Y = T(X + Y)$ (exception transformer)
- Automata: $X \# Y = \mathcal{P}(A^* \times X + A^+ \times Y)$
- Hybrid systems: $X \# Y = \mathbb{R}_{\geq 0} \times X + \mathbb{R}_{\geq 0} \times Y + \mathbb{R}_{\geq 0}$
- Generalized processes: $X \# Y = T(X + H(\nu_\gamma. T((X + Y) + H_\gamma)))$
guarded id.-on-obj. functors

guarded parametrized monads

strong guarded parametrized monads

guarded Freyd categories

monads

strong monads

id.-on-obj. functors

Freyd categories

guardedness

representability

strength
Generalize: express quantitative information by typing, e.g. how productive is the program, how much time it consumes, etc.

Dualize: representation of guarded recursion by comonads
  ▶ What are instances of comonadic guarded recursion?
  ▶ Representing recursion on casual streams/course-of-value recursion

Implement (Haskell, Agda, Coq)

Research:
  ▶ Can we prove more general coherence theorem?
  ▶ Are properly monoidal guarded parametrized monads interesting?
  ▶ Can we characterize guardedness by unary functors (like ▷)?
Thank you for your **Attention!**
Originally, Moggi\(^8\) interpreted call-by-value over **strong monads**

- A functor \( F : C \to D \) between monoidal \( C \) and \( D \) is strong if there is (natural in \( A, B \)) **strength** \( \tau_{A,B} : A \otimes FB \to F(A \otimes B) \), such that

\[
\begin{align*}
I \otimes FX & \cong FX \\
\tau \downarrow & \quad \mid \\
F(I \otimes X) & \cong FX
\end{align*}
\]

\[
(\tau \otimes Y) \otimes FZ \quad \tau \quad F((X \otimes Y) \otimes Z)
\]

- A monad \((T, \eta, \mu)\) on \( C \) is strong if \( T \) is strong additionally \( \eta, \mu \) are strong:

\[
\begin{align*}
X \otimes Y & \quad \quad X \otimes Y \\
X \otimes \eta \downarrow & \quad \downarrow \eta \\
X \otimes TY & \quad \tau \quad T(X \otimes Y)
\end{align*}
\]

\[
\begin{align*}
X \otimes TY & \quad \tau \quad T(TX \otimes TY) \quad \tau \quad TT(TX \otimes Y) \\
X \otimes \mu \downarrow & \quad \downarrow \mu \\
X \otimes TY & \quad \tau \quad T(X \otimes Y)
\end{align*}
\]

**Theorem (⁹)**

**In monoidal closed categories, strength is equivalent to enrichment**

---

\(^8\) E. Moggi, Notions of Computation and Monads, 1991

\(^9\) A. Kock, Strong Functors and Monoidal Monads, 1972
Non-Representable Guardedness

If $J$ is not a left adjoint, guardedness is not representable. But this is boring. Is there other counterexamples?

Theorem

Suppose, every morphism in $\mathbf{V}$ factorizes as a regular epic, followed by a monic. Let $T$ be a guarded monad on $\mathbf{V}$. Then a family of monos $(\epsilon_{X,Y} : X \nrightarrow Y \hookrightarrow T(X + Y))_{X,Y \in |\mathbf{V}|}$ extends to a guarded parametrized monad iff

- every $\epsilon_{X,Y}$ is the largest guarded subobject of $T(X + Y)$
- for every $f : X \to T(Y + Z)$ and a regular epic $g : X' \to X$, if $f \circ g$ is guarded then $f$ is guarded

Example

In $\mathbf{Set}$, let $f : X \to Y + Z$ be guarded in $Z$ if $\{z \in Z \mid f^{-1}(\text{inr } z) \neq \emptyset\}$ is finite.

This predicate is not $\text{Id}$-representable, as any $1 \hookrightarrow X \xrightarrow{\text{inr}} \emptyset + X$ is guarded, but $\text{inr}$ is not if $X$ is infinite.