REPRESENTING GUARDEDNESS

SERGEY GONCHAROV
FAU Erlangen-Nürnberg

FEBRUARY 17, 2022
Setting the Stage: Notion of Guardedness
How do we know that automata $q_1$ start $q_2$ are equivalent?

are equivalent?
Proof "by Coinduction"

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
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
Proof "by Coinduction"

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

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

\(^1\)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\(^1\):

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

\(^1\)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\(^1\):

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

\(^1\)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:\(^1\)

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

- 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 $\ddot{h} = -g \ (g \approx 9.8)$ 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
Basic Process Algebra (BPA):

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

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

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

\[(i \in \{0, 1\}) \quad (i, j \in \{0, 1\})\]

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

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

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

1. \(TX = \mathcal{P}(A^* \times X)\)
   - an automaton over \(n\) states is a Kleisli map \(h : n \to \mathcal{P}(A^* \times (1 + n))\)
   - languages, recognized by states: \(h^\uparrow : n \to \mathcal{P}(A^*)\)

2. \(TX = \mathbb{R}_{\geq 0} \times X + \mathbb{R}_{\geq 0}\)
   - \(\mathbb{R}_{\geq 0}\) and \(\mathbb{R}_{\geq 0}\) model finite and infinite durations
   - For \(h : X \to T(2 \times X) \cong T(X + X)\), \(h^\uparrow : X \to TX\) is a while-loop on \(h\)

3. \(TX = \nu_\gamma. \mathcal{P}_{\omega_1} (X + A \times \gamma)\) (\(\nu_\gamma. F_\gamma\) is a final \(F\)-coalgebra)
   - \(h : n \to T(\{\sqrt{\} + n)\) is a system of \(n\) recursive process definitions
   - \(h^\uparrow : n \to \nu_\gamma. \mathcal{P}_{\omega_1} (\{\sqrt{\} + A \times \gamma)\) is a solution
Guarded v.s. Unguarded

Most of the time, guarded fixpoint operators 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
A guardedness predicate identifies for all objects $X, Y, Z$ guarded morphisms $C\diamondsuit(X, Y, Z) \subseteq C(X, Y + Z)$, such that

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

where $f : X \to Y \triangleright Z$ means $f \in C\diamondsuit(X, Y, Z)$

- A category with a guardedness predicate is called guarded
- A monad is guarded if its Kleisli category is guarded
Every category/monad is guarded with $f : X \to Y \triangleright Z$ iff $f$ factors through \text{inl}: Y \to Y + Z$ (trivial iteration)

Every category/monad is guarded with $C\diamond(X, Y, Z) = C(X, Y + Z)$ (total iteration)

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

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

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

- **Sorts** $A, B, C, \ldots$
- **Signature** $\Sigma_v$ of pure programs $f : A \to B$, and signature $\Sigma_c$ of 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**:
  \[
  \frac{f : A \to B \in \Sigma_v}{\Gamma \vdash_v v : A} \quad \frac{f : A \to B \in \Sigma_c}{\Gamma \vdash_c f(v) : B} \\
  \frac{\Gamma \vdash_v f(v) : B}{\Gamma \vdash_v x : A} \quad \frac{\Gamma \vdash_c return v : A}{\Gamma \vdash_c p : A \quad x : A \vdash_c q : B}
  \]
  \[
  \frac{x : A \vdash_v x : A}{\Gamma \vdash_v x : A} \quad \frac{\Gamma \vdash_c return v : A}{\Gamma \vdash_c p : A \quad x : A \vdash_c q : B}
  \]
  \[
  \frac{\Gamma \vdash_c p : A \quad x : A \vdash_c q : B}{\Gamma \vdash_c x \leftarrow p ; q : B}
  \]
- **Semantics**:
  - $\llbracket x : A \vdash_v x : A \rrbracket = \text{id}$
  - $\llbracket \Gamma \vdash_v f(v) : B \rrbracket = \llbracket f \rrbracket \llbracket \Gamma \vdash_v v : A \rrbracket$
  - $\llbracket \Gamma \vdash_c f(v) : B \rrbracket = \llbracket f \rrbracket J \llbracket \Gamma \vdash_v v : A \rrbracket$
  - $\llbracket \Gamma \vdash_c \text{return} v : A \rrbracket = J \llbracket \Gamma \vdash_v v : A \rrbracket$
  - $\llbracket \Gamma \vdash_c x \leftarrow p ; q : B \rrbracket = \llbracket x : A \vdash_c q : B \rrbracket \llbracket \Gamma \vdash_c p : A \rrbracket$
Variable contexts can normally carry a list of variables \( \Gamma = \langle x_1 : A_1, \ldots, x_n : A_n \rangle \), e.g. variable introduction now looks like this:

\[
\frac{
}{x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i}
\]

This yields fine-grained call-by-value (FGCBV)\(^2\) metalanguage. It can be interpreted over a Freyd category:

- \( V \) is a category with finite products
- \textbf{action} \( V \times C \to C \) of \( V \) on \( C \)
- \( J : V \to C \) is an identity-on-objects functor, preserving the action

\(^2\)P. Levy, J. Power, H. Thielecke, Modelling Environments in Call-By-Value Programming Languages, 2002
Originally, Moggi\textsuperscript{3} interpreted call-by-value over strong monads

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

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

\[
\begin{align*}
(X \otimes Y) \otimes FZ & \xrightarrow{\tau} F((X \otimes Y) \otimes Z) \\
\tau \downarrow & \\
X \otimes (Y \otimes FY) & \xrightarrow{\tau} X \otimes F(Y \otimes Z) \\
\tau \downarrow & \\
F(X \otimes (Y \otimes Z)) & \\
\end{align*}
\]

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

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

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

Theorem (\textsuperscript{4})

In monoidal closed categories, strength is equivalent to enrichment

\textsuperscript{3}E. Moggi, Notions of Computation and Monads, 1991

\textsuperscript{4}A. Kock, Strong Functors and Monoidal Monads, 1972
If we want to implement higher order:

\[
\begin{align*}
\Gamma, x : A &\vdash_c p : B \\
\Gamma &\vdash \lambda x. p : A \to B \quad &\quad \Gamma \vdash f : A \to B \\
\Gamma &\vdash f v : B
\end{align*}
\]

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

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

naturally in \(A\)

---

**Theorem (5)**

The following are equivalent:

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

---

5Essentially: 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**

*Given id-on-objects functor $J: V \to C$, the following are equivalent:*

- $J$ is a left adjoint
- Presheaves $C(J(-), B): V^{op} \to \text{Set}$ are representable
- $C$ is isomorphic to a Kleisli category of a monad

and then:

**Theorem**

*Given a Freyd category $J: V \to C$, the following are equivalent:*

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

- Monads → Strong monads
- Representability
- Id.-on-obj. functors → Freyd categories
- Strength
Call-by-Value Meets Guardedness
We stick to guarded $C$, and identity-on-objects $J : V \to C$ strictly preserving finite coproducts.

Modify the type system:

<table>
<thead>
<tr>
<th>$\emptyset$ $\diamond$-type</th>
<th>$A$ $\diamond$-type</th>
<th>$A \diamond$-type $B \diamond$-type</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A \diamond$-type</td>
<td>$A \diamond$-type</td>
<td>$A + B \diamond$-type</td>
</tr>
<tr>
<td>$\Diamond A$ type</td>
<td>$A$ type</td>
<td>$A + B$ type</td>
</tr>
</tbody>
</table>

So, e.g. $A + (\Diamond B + \Diamond C)$ indicates that we are guarded in $B$ and in $C$.

Let $\preceq$ be partial order on types, generated by the rules:

- $\Diamond A \preceq A$
- $A \preceq C, B \preceq D \Rightarrow A + B \preceq C + D$

This allows us to compute “guarded” $[A] \diamond$ and “un-guarded” $[A] \diamond$ part of every type $A$, and $[A \preceq B] \in V([A] \diamond, [B] \diamond) \times V([A] \diamond, [B] \diamond + [B] \diamond)$. 
We interpret

\[ [x : A \vdash v : B] \in \mathbf{V}([A] \Diamond, [B] \Diamond) \times \mathbf{V}([A] \Diamond, [B] \Diamond + [B] \Diamond), \]

\[ [x : A \vdash_c p : B] \in \mathbf{C}^\Diamond([A] \Diamond, [B] \Diamond, [B] \Diamond) \times \mathbf{C}([A] \Diamond, [B] \Diamond + [B] \Diamond). \]

For example, variable introduction incorporates weakening:

\[
\frac{A \trianglelefteq B}{x : A \vdash v : B}
\]

So, \([x : A \vdash v : B] = [A \trianglelefteq B]\)

We then can type guarded iteration:

\[
\frac{\Gamma \vdash_c p : A \quad x : A \vdash_c q : B + \Diamond A}{\Gamma \vdash_c \text{iter } x \leftarrow p; q : B}
\]
Definition

Given \( J : V \to C \), as before and guarded \( C \), we call the guardedness predicate \( C \otimes (J-) \) representable if for all \( A, B \in |C| \) the presheaves

\[
C \otimes (J(-), A, B) : V^{\text{op}} \to \text{Set}
\]

are representable

Note that \( C \otimes (X, A, \emptyset) \cong C(X, A) \), hence

Lemma

If \( C \otimes \) is representable, \( J \) is a left adjoint. In this case, \( C \) is a Kleisli category of some monad on \( V \)
Recall, that a bifunctor \( \# : V \times V \to V \) is a **parametrized monad**\(^6\) if

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

**Theorem (Main)**

\( C^\downarrow \) is representable iff

1. There is a parametrized monad \( \# \) on \( V \), such that \( C \) is isomorphic to the Kleisli category of \((-) \# \emptyset\)

2. There is a family of monics \( \epsilon_{X,Y} : X \# Y \to (X + Y) \# \emptyset \) natural in \( X, Y \), such every \( \epsilon_{-} : - \# X \to (- + X) \# \emptyset \) is a monad morphism

3. There is natural \( \zeta_{X,Y} : (X \# Y) \# ((X + Y) \# \emptyset) \to X \# Y \) such that

\[
\begin{align*}
(X \# Y) \# ((X + Y) \# \emptyset) & \xrightarrow{\epsilon_{X \# Y, (X + Y) \# \emptyset}} (X \# Y + (X + Y) \# \emptyset) \# \emptyset \\
& \xrightarrow{\mu_{X + Y, \emptyset}} (X + Y) \# \emptyset \\
& \xrightarrow{\epsilon_{X,Y}} X \# Y
\end{align*}
\]

Then, up to isomorphism, \( f : X \to Y \to Z \) iff \( f \) factors through \( \epsilon \)

---

\(^6\)T. Uustalu, Generalizing Substitution, 2003
Examples

- Trivial guardedness corresponds to $X \# Y = TX$, i.e. $X \# Y$ does not depend on the parameter
- Total guardedness corresponds to $X \# Y = T(X + Y)$ (exception monad transformer)
- Automata: $X \# Y = \mathcal{P}(A^* \times X + A^+ \times Y)$
- Hybrid systems: $X \# Y = \mathbb{R}_{\geq 0} \times X + \mathbb{R}_{\geq 0} \times Y + \mathbb{I}\mathbb{R}_{\geq 0}$
- Generalized processes: $X \# Y = T(X + H(\nu \gamma. T((X + Y) + H\gamma))))$
If $J$ is not a left adjoint, guardedness is not representable. But this is boring. Is there other counterexamples?

**Theorem**

Suppose, every morphism in $\mathcal{V}$ factorizes as a regular epic, followed by a monic. Let $\mathbf{T}$ be a guarded monad on $\mathcal{V}$. Then a family of monos $(\epsilon_{X,Y} : X \# Y \hookrightarrow T(X + Y))_{X,Y \in \mathcal{V}}$ 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 $fg$ is guarded then $f$ is guarded

**Example**

In the category of topological spaces, let $f : X \to Y \upharpoonright Z$ if $Z$ is compact. This is not $\text{Id}$-representable, since there is no “largest compact subspace of $X$” (e.g. $X = \mathbb{N}$ under cofinite topology)
Most sophisticated case:
- \( J : V \to C \) is a Freyd category
- \( V \) and \( C \) are co-Cartesian and \( J \) strictly preserves coproducts

**Theorem**

Presheaves

\[ C(\delta(J(-\times A), B, C)) : V^{\text{op}} \to \text{Set} \]

*are representable iff*
- There is a guarded parametrized monad \( \# \) on \( V \)
- The monad \( (-) \# \emptyset \) is strong
- \( C \) is isomorphic to the Kleisli category of \( (-) \# \emptyset \)
- Exponentials \( (B \# C)^A \) exist

**Proof Idea.**

\[ C(\delta(J(-\times A), B, C)) \cong V(-\times A, B \# C) \cong V(-, (B \# C)^A) \]
Let additionally:

- \( V \) is distributive i.e. \( A \times (B + C) \cong A \times B + A \times C \)
- The action of \( V \) on \( C \) preserves guardedness: \( f \in V(A, B), g \in C\uplus(X, Y, Z), \)

\[
A \times X \xrightarrow{f \otimes g} B \times (Y + Z) \cong B \times Y + B \times Z
\]

is in \( C\uplus(A \times X, B \times Y, B \times Z) \)

We then call \( J \) guarded Freyd category

**Proposition**

If \( C\uplus \) is \( J \)-representable then the induced \( \# \) becomes equipped with strength:

\[
X \times (Y \# Z) \to (X \times Y) \# (X \times Z)
\]

Call such \( \# \) **strong guarded parametrized monad**
Total Summary

guarded parametrized monads

monads

id.-on-obj. functors

guarded Freyd categories

strong guarded parametrized monads

strong monads

Freyd categories

guarded id.-on-obj. functors

representability

strength

guardedness
Computational metalanguage a la Moggi w.r.t. guarded parametrized monads?

Combine guardedness with more advanced structures (e.g. effect handling)

Dualize: representation of guarded recursion by comonads
  - It is known already that (guarded) recursion is categorically dual to (guarded) iteration
  - What are instances of comonadic guarded recursion?
  - Representing recursion on casual streams/course-of-value recursion

Implementation