Relational realizability model for QTT

Our aim is to build a concrete realizability model for QTT which allows for metareasoning with results derived from parametricity. To that end, a construction of a quantified category with families (QCwF[**fill**]) is presented. We follow the relational approach to types introduced by Reynolds for typed lambda calculus[**fill**] and later refined for dependent types theories.[**fill**] Once and for all fix a usage semiring R and an R-linear combinatory algebra $\mathscr{A}$[1].

Taking stocks

Definition 1 (Assembly)**.** An assembly Γ is a pair $(|\Gamma|, e)$ where $|\Gamma|$ is a carrier set and e is a realizability function $|\Gamma| \to \mathcal{P}(\mathscr{A})$, s.t $e(\gamma)$ is nonempty for every $\gamma \in |\Gamma|$.

We interpret $E(\gamma)$ as a set of witnesses for the existence of γ and write $a \vDash_\Gamma \gamma$ to denote $a \in e(\gamma)$.
A morphism between two assemblies $(|\Gamma|, E_\Gamma)$ and $(|\Delta|, E_\Delta)$ is a function $f : |\Gamma| \to |\Delta|$ that is realizable - there exists $a_f \in \mathscr{A}$, s.t the following holds for every $\gamma \in |\Gamma|$ and $a_\gamma \in \mathscr{A}$:

$$a_\gamma \vDash_\Gamma \gamma \implies a_f.a_\gamma \vDash_\Delta f(\gamma)$$

We say that a_f tracks f. Note that only an existence of a realizer for f is stipulated in the definition - multiple realizers do not induce multiple morphisms.
Using these notions we can construct a category $\mathcal{A}sm(\mathscr{A})$.

Definition 2 (Reflexive graph)**.** A reflexive graph (r.g.) G is a tuple $(G_O, G_R, G_{refl}, G_{src}, G_{tgt})$, where G_O and G_R are sets, G_{src} and G_{tgt} are functions $G_R \to G_O$ and G_{refl} is a function $G_O \to G_R$, s.t $G_{src} \circ G_{refl} = id_{G_O} = G_{tgt} \circ G_{refl}$.

G_O and G_R stand for a set of objects and a set of relations, respectively. In general, reflexive graphs are less structured than categories as relations need not compose.

A morphism between reflexive graphs G and H is a pair of functions $(f_o : G_O \to H_O, f_r : G_R \to H_R)$, s.t all of the depicted squares commute:

$$\begin{tikzcd}
G_O \arrow[r, "f_o"] \arrow[d, "G_{refl}" description] & H_O \arrow[d, "H_{refl}" description] \\
G_R \arrow[u, bend left, "G_{src}"] \arrow[u, bend right, "G_{tgt}"'] \arrow[r, "f_r"] & H_R \arrow[u, bend left, "H_{src}"] \arrow[u, bend right, "H_{tgt}"']
\end{tikzcd}$$

Reflexive graphs equipped with r.g. morphisms form a category $\mathcal{R}Gph(\mathcal{S}et)$.
We use reflexive graphs to give a dyadic interpretation of types in the spirit of [**fill**].

[1] In case some non-trivial properties of $\mathscr{A}$ are required, we will tacitly assume that $\mathscr{A}$ is a graph model(see [**fill**]) - **also to fix**

Assembling a reflexive graph

One could easily generalize reflexive graphs by considering object and relation components from an arbitrary category $\mathcal{C}$ instead of $\mathcal{S}et$. As our purpose is to build a realizability model, we pick $\mathcal{A}sm(\mathscr{A})$ as the category of interest and identify two appropriate notions of a family of reflexive graphs of assemblies and internal r.g. in $\mathcal{A}sm(\mathscr{A})$.

Definition 3 (Internal reflexive graph in $\mathcal{A}sm(\mathscr{A})$)**.** An internal reflexive graph G in $\mathcal{A}sm(\mathscr{A})$ is a pair of assemblies (G_O, G_R) and a triple of morphisms $(G_{refl} : G_O \to G_R,\ G_{src} : G_R \to G_O,\ G_{tgt} : G_R \to G_O)$, such that the identities in Definition 2 are satisfied.

In essence, sets and functions are directly replaced by the objects and morphisms from $\mathcal{A}sm(\mathscr{A})$. The same transformation is carried out when defining morphisms between internal reflexive graphs as well. With these components, we obtain a category $\mathcal{R}Gph(\mathcal{A}sm(\mathscr{A}))$. By considering internal r.g. of shape (X, X, id_X, id_X, id_X), we identify isomorphic copy of $\mathcal{A}sm(\mathscr{A})$ inside $\mathcal{R}Gph(\mathcal{A}sm(\mathscr{A}))$.
A terminal object in $\mathcal{R}Gph(\mathcal{A}sm(\mathscr{A}))$ is a tuple $(1, 1, id, id, id)$, where 1 is the terminal assembly $(\{\star\}, f)$, with f defined as $\star \mapsto \{I\}$.

Definition 4 (Family of reflexive graphs of assemblies)**.** Let $\mathcal{C}$ be a category with a terminal object. Given a reflexive graph $\Gamma \in Ob(\mathcal{C})$, a family of internal r.g. over Γ is a tuple $S := (S_O, S_R, S_{refl}, S_{src}, S_{tgt})$, where:

- $S_O : \Gamma_O \to \mathcal{A}sm(\mathscr{A})$
- $S_R : \Gamma_R \to \mathcal{A}sm(\mathscr{A})$
- a Γ-indexed collection of morphisms $S_{refl} := \{f : S_O(\gamma) \to S_R(\Gamma_{refl}(\gamma))\}_{\gamma \in \Gamma_O}$
- $S_{src} := \{f : S_R(\gamma) \to S_O(\Gamma_{src}(\gamma))\}_{\gamma \in \Gamma_R}$
- $S_{tgt} := \{f : S_R(\gamma) \to S_O(\Gamma_{tgt}(\gamma))\}_{\gamma \in \Gamma_R}$

such that

- each identity in the following collection is satisfied:

$$S_\sigma(\Gamma_{refl}(\gamma)) \circ S_{refl}(\gamma) = id \text{ for every } \gamma \in \Gamma_O, \sigma \in \{src, tgt\}$$

- there exists an element $a_\sigma \in \mathscr{A}$ which tracks each morphism in the collection S_σ for $\sigma \in \{src, tgt, refl\}$

We are only interested in cases when $\mathcal{C} = \mathcal{S}et$ or $\mathcal{C} = \mathcal{A}sm(\mathscr{A})$.
A morphism M between two families S and T of internal r.g. over Γ is a pair of Γ-indexed collection of functions:

- $M_O := \{f : |S_O(\gamma)| \to |T_O(\gamma)|\}_{\gamma \in \Gamma_O}$
- $M_R := \{f : |S_R(\gamma)| \to |T_R(\gamma)|\}_{\gamma \in \Gamma_O}$

such that:

- $\check{T_{refl}}(M_O(\gamma)(x)) = M_R(\Gamma_{refl}(\gamma))(|S_{refl}(\gamma)|(x))$ for every $\gamma \in \Gamma_O$, $x \in S_O(\gamma)$
- $\check{T_{src}}(M_R(\gamma)(r)) = M_O(\Gamma_{src}(\gamma))(|S_{src}(\gamma)|(r))$ for every $\gamma \in \Gamma_R$, $r \in S_R(\gamma)$
- $\check{T_{tgt}}(M_R(\gamma)(r)) = M_O(\Gamma_{tgt}(\gamma))(|S_{tgt}(\gamma)|(r))$ for every $\gamma \in \Gamma_R$, $r \in S_R(\gamma)$

where given a morphism $f : A \to B$ between assemblies, $\check{f}$ is the underlying set-theoretic function.
A terminal family of r.g. over Γ, 1_Γ, consists of two constant functions, mapping $\gamma \in \Gamma$ to a terminal assembly 1, and three Γ-indexed collections with a sole element id_1.

There is an apparent discrepancy in definition of family of r.g of assemblies and morphisms between them - morphisms ignore any realizability information. This is due to the role they play in interpretation of QTT - semantic terms in the 0-fragment of the theory are interpreted as morphisms from the terminal family and hence do not require any realizers.
It is then tempting to continue discarding realizability information in Definition 4. However, as families of r.g of assemblies will model the semantic types in QCwF, the approach seems infeasible.

- types provide realizability blueprint for resourced semantic terms. Thus we cannot substitute $\mathcal{S}et$ for $\mathcal{A}sm$ ($\mathscr{A}$).
- given a context $\Gamma \in Ob(\mathcal{R}Gph)$, $\rho \in R$ and a type $S \in Ty(\Gamma)$, the resourced context extension $\Gamma.\rho S \in Ob(\mathcal{R}Gph(\mathcal{A}sm(\mathscr{A})))$ makes use of the trackers for src, tgt and $refl$ morphisms. (Note that these are not needed when building an ordinary CwF).

A CwF from families of reflexive graphs of assemblies

As a first step toward obtaining a relationally parametric realizability model of QTT, we construct a concrete CwF using families of internal graphs of assemblies.
Consider the category $\mathcal{R}Gph$ with terminal object $1 := (\{\star\}, \{\star\}, id, id, id)$.
Let $\Gamma, \Delta \in Ob(\mathcal{R}Gph)$, define:

- the collection of semantic types $Ty(\Gamma)$ as the collection of families of internal r. g. over Γ.
- given a type $S \in Ty(\Gamma)$, let $Tm(\Gamma, S) := Hom(1_\Gamma, S)$.
  Spelling this out and ignoring the contribution of the terminal family, we get:
  An element $M \in Tm(\Gamma, S$ is a pair of functions $(M_O : \forall \gamma \in \Gamma_O.S_O(\gamma), M_R : \forall \gamma \in \Gamma_R.S_R(\gamma))$, such that

$$\forall \gamma \in \Gamma_O.\check{S_{refl}}(M_O(\gamma)) = M_R(\Gamma_{refl}(\gamma))$$
$$\forall \gamma \in \Gamma_R.\check{S_{src}}(M_R(\gamma)) = M_O(\Gamma_{src}(\gamma))$$
$$\forall \gamma \in \Gamma_R.\check{S_{tgt}}(M_R(\gamma)) = M_O(\Gamma_{tgt}(\gamma))$$

- given $f : \Gamma \to \Delta$, substitutions in types and terms is a precomposition with f on the object and relation components of types and terms respectively. Clearly, these operations are compatible with identity and composition in $\mathcal{R}Gph$
- context extension : Suppose $S \in Ty(\Gamma)$, construct a r.g. $\Gamma.S$ as :

$$\begin{aligned}(\Gamma.S)_O &= \{(\gamma, x) : \gamma \in \Gamma_O, x \in S_O(\gamma)\}\\ (\Gamma.S)_R &= \{(\gamma, r) : \gamma \in \Gamma_R, r \in S_R(\gamma)\}\\ (\Gamma.S)_{refl}(\gamma, x) &= (\Gamma_{refl}(\gamma), S_{refl}(\gamma)(x))\\ (\Gamma.S)\sigma(\gamma, r) &= (\Gamma\sigma(\gamma), S\sigma(\gamma)(r)), \quad \sigma \in \{src, tgt\}\end{aligned}$$

Claim. *$Hom_{\mathcal{R}Gph}(\Delta, \Gamma.S) \cong \{(f, M) : f : \Delta \to \Gamma, M \in TM(\Delta, S\{f\}\}$, natural in Δ.*

Upgrading to a QCwF

We simultenously recall the definition of a QCwF and add structure to our CwF.

1. A CwF $(\mathcal{C}, 1, Ty, Tm, -.-, \langle -.- \rangle)$.
   Consider the CwF built in the previous section
2. Category $\mathcal{L}$ for context with resource annotation, equipped with a faithful functor $U : \mathcal{L} \to \mathcal{C}$.
   Take $cL := \mathcal{R}Gph(\mathcal{A}sm(\mathscr{A}))$ and let U be the forgetful functor $\mathcal{R}Gph(\mathcal{A}sm(\mathscr{A})) \to \mathcal{R}Gph$.
3. Addition structure - let $\mathcal{L} \times_{\mathcal{C}} \mathcal{L}$ denote the pullback $\mathcal{L} \xrightarrow{U} \mathcal{C} \xleftarrow{U} \mathcal{L}$, we stipulate the existence of a functor $(+) : \mathcal{L} \times_{\mathcal{C}} \mathcal{L} \to \mathcal{L}$, s.t $U(\Gamma' + \Gamma'') = U(\Gamma') = U(\Gamma'')$. Moreover, there is a distinguished object $\Diamond \in \mathcal{L}$, s.t $U(\Diamond) = 1$.
   Suppose Γ', Γ'' are internal r.g. in $\mathcal{A}sm(\mathscr{A})$, such that $|\Gamma'_O| = |\Gamma''_O|$ and $|\Gamma'_R| = |\Gamma''_R|$. Construct the internal r.g. $\Gamma := \Gamma' + \Gamma''$, where:
   - $\Gamma_O := (|\Gamma'_O|, \vDash_\Gamma)$ with $a \vDash_\Gamma \gamma$ iff there exist $x, y \in \mathscr{A}$, s.t. $a = [x, y]$ and $x \vDash_{\Gamma'} \gamma$ and $y \vDash_{\Gamma''} \gamma$.
   - define Γ_R similarly as Γ_O.
   - Γ_{refl}, Γ_{src}, Γ_{tgt} are all inherited from Γ' (or Γ''). *(should check this is indeed well-defined)*
     To wit that addition preserves realizability of functions, wlog take some $\gamma \in \Gamma_R$, s.t $a \vDash_{\Gamma_R} \gamma$. Then by definition there exists $x, y \in \mathscr{A}$, s.t $x \vDash_{\Gamma'_R} \gamma$ and $y \vDash_{\Gamma''_R} \gamma$. Moreover, there exist $b, c \in \mathscr{A}$, s.t $b.x \vDash_{\Gamma'_O} \Gamma'_{src}(\gamma)$ and $c.y \vDash_{\Gamma''_O} \Gamma''_{src}(\gamma)$. Let $a^* = \lambda x \lambda y.[b.x, c.y]$, then a^* tracks Γ_{src}.
     Pick a terminal internal r.g. as an interpretation for $\Diamond$.
4. Scaling with $\rho \in R$ - there is a functor $\rho(-) : \mathcal{L} \to \mathcal{L}$, s.t. $U(\rho(-)) = U(-)$.
   Again we define a internal r.g. $\Gamma := \rho(\Gamma')$ as :
   - $\Gamma_\sigma = (|\Gamma'_\sigma|, \vDash_{\Gamma_\sigma})$ with $a \vDash_{\Gamma_\sigma} \gamma$ iff there is $x \in \mathscr{A}$, s.t $a =!_\rho x$ and $x \vDash_{\Gamma'_\sigma} \gamma$ for $\sigma \in \{O, R\}$

- Γ_σ is just Γ'_σ for $\sigma \in \{src, tgt, rfl\}$
  To see that, for instance, Γ_{refl} retain realizability, let $a \vDash_{\Gamma_O} \gamma$. Therefore, there is some $x, b \in \mathscr{A}$, s.t $x \vDash_{\Gamma'_O} \gamma$ and $b.x \vDash_{\Gamma'_R} \Gamma'_{refl}(\gamma)$. Now consider $a^* := F_\rho.(!_\rho b)$, we have $a^*.x \vDash_{\Gamma_R} \Gamma_{refl}(\gamma)$.
  As the functor ρ modifies only realizability information, it is clear that $U(\rho(-)) = U(-)$ holds.

Type formers

Universe of small types

Dependent function types

Dependent pairs