Comprehensive parametric polymorphism

Fredrik Nordvall Forsberg
University of Strathclyde, Glasgow
fredrik.nordvall-forsberg@strath.ac.uk

LFCS Seminar, Edinburgh, 3 May 2016
Joint work with...

Neil Ghani (Strathclyde)  Alex Simpson (Ljubljana)
Parametric polymorphism [Strachey, 1967]

- A polymorphic program
  \[ t : \forall \alpha. A \]
  
  is parametric if it applies the same uniform algorithm at all instantiations \( t[B] \) of its type parameter.

- Typical example:
  \[ \text{reverse} : \forall \alpha. \text{List } \alpha \rightarrow \text{List } \alpha \]
Reynolds insight: relational parametricity [1983]

- Turn the **negative** statement “not distinguishing types” into the **positive** statement “preserves all relations”.

$$\forall \alpha. \text{A is relationally parametric if for all } R \subseteq B \times B', (t[B], t[B']) \in \langle A \rangle (R)$$

where $$\langle A \rangle (R) \subseteq A(B) \times A(B')$$ is the relational interpretation of the type A.
Reynolds insight: relational parametricity [1983]

- Turn the negative statement “not distinguishing types” into the positive statement “preserves all relations”.

- A polymorphic program $t : \forall \alpha. A$ is relationally parametric if for all relations $R \subseteq B \times B'$,

\[
(t[B], t[B']) \in \langle A \rangle(R)
\]

where $\langle A \rangle(R) \subseteq A(B) \times A(B')$ is the relational interpretation of the type $A$.

- E.g. reverse : $\forall \alpha. \text{List} \ \alpha \rightarrow \text{List} \ \alpha$ is relationally parametric.
Applications of relational parametricity

Relational parametricity enables:

- Reasoning about abstract data types.
- Correctness (universal properties) of encodings of data types.
- ‘Theorems for free!’ [Wadler, 1989].
- Concretely, a specific example: if $t : \forall \alpha. \alpha \rightarrow \alpha$ then $t = \Lambda \alpha. \lambda x. x$.

Usually in the setting of Girard’s/Reynold’s $\lambda 2$ (System F) — serves as a model type theory for (impredicative) polymorphism.
What is the fundamental category-theoretic structure needed to model relational parametricity for $\lambda^2$?
What is the fundamental category-theoretic structure needed to model relational parametricity for $\lambda 2$?

- Perhaps surprising that this question does not have an established answer.
What is the fundamental category-theoretic structure needed to model relational parametricity for $\lambda 2$?

- Perhaps surprising that this question does not have an established answer.

- We know the fundamental structure needed for $\lambda 2$ ($\lambda 2$ fibrations [Seely, 1987]).
What is the fundamental category-theoretic structure needed to model relational parametricity for $\lambda 2$?

- Perhaps surprising that this question does not have an established answer.

- We know the fundamental structure needed for $\lambda 2$ ($\lambda 2$ fibrations [Seely, 1987]).

- We also know the fundamental structures used for relational parametricity (reflexive graph categories [Robinson and Rosolini, 1994], parametricity graphs [Dunphy and Reddy, 2004]).
What is the fundamental category-theoretic structure needed to model relational parametricity for $\lambda^2$?

- Perhaps surprising that this question does not have an established answer.

- We know the fundamental structure needed for $\lambda^2$ ($\lambda^2$ fibrations [Seely, 1987]).

- We also know the fundamental structures used for relational parametricity (reflexive graph categories [Robinson and Rosolini, 1994], parametricity graphs [Dunphy and Reddy, 2004]).

- So why not just combine the two?
So why not just combine the two?

- When doing so, the expected consequences of parametricity are only derivable if the underlying category is *well-pointed*.

- **Recall**: A category \( \mathcal{C} \) is well-pointed when \( f = g : A \to B \) in \( \mathcal{C} \) iff \( f \circ e = g \circ e : 1 \to B \) for all global elements \( e : 1 \to A \).

- This rules out many interesting categories, e.g. functor categories.
So why not just combine the two?

- When doing so, the expected consequences of parametricity are only derivable if the underlying category is well-pointed.

  **Recall:** A category $\mathcal{C}$ is well-pointed when $f = g : A \rightarrow B$ in $\mathcal{C}$ iff $f \circ e = g \circ e : 1 \rightarrow B$ for all global elements $e : 1 \rightarrow A$.

- This rules out many interesting categories, e.g. functor categories.

- Existing solutions (e.g. Birkedal and Møgelberg [2005]) circumvent this by adding significant additional structure to models (enough to model the full logic of Plotkin and Abadi).

- We seek instead a minimal solution still based on the idea of directly combining models of $\lambda^2$ with structure for relational parametricity.
A minimal solution

- We achieve this in a perhaps unexpected way: we change the notion of model of $\lambda^2$.

- $\lambda^2$ fibrations satisfying Lawvere's comprehension property.
A minimal solution

- We achieve this in a perhaps unexpected way: we change the notion of model of $\lambda 2$.

- $\lambda 2$ fibrations satisfying Lawvere’s comprehension property.

- This allows us to combine such comprehensive $\lambda 2$ fibrations with reflexive graph structure to model relational parametricity for $\lambda 2$.

- Validating expected consequences, also for non-well-pointed categories.

- Proof involves novel ingredients due to minimality of structure:
  - definability of direct image relations,
  - arguments without use of equality relations, and
  - only weak forms of graph relations available (‘pseudographs’).
Outline

1. The type theory $\lambda_2$

2. Modelling $\lambda_2$ using (comprehensive) $\lambda_2$ fibrations

3. Modelling relational parametricity using (comprehensive) parametricity graphs

4. Reasoning about parametricity using a type theory $\lambda_2R$
The type theory $\lambda^2$
The polymorphic lambda calculus $\lambda^2$ (System F) [Girard, 1972; Reynolds, 1974]

- Four judgements:

  $\Gamma \text{ ctx}$ \quad $\Gamma$ is a context

  $\Gamma \vdash A$ type \quad $A$ is a type in context $\Gamma$

  $\Gamma \vdash t : A$ \quad term $t$ has type $A$ in context $\Gamma$

  $\Gamma \vdash t = s : A$ \quad judgemental equality

- Types and terms generated by grammars

  $A, B ::= \alpha \mid A \to B \mid \forall \alpha. A$ \quad types

  $t, s ::= x \mid \lambda x. t \mid t \ s \mid \Lambda \alpha. t \mid t[B]$ \quad terms

- Equality generated by ($\beta$) and ($\eta$) for both term and type abstraction.
Only unusual feature of our presentation

- We use a single context with type and term variables interleaved.
- Standard from a dependent types perspective.
- Hence two different context extensions:

\[
\frac{\Gamma \text{ctxt}}{\Gamma, \alpha \text{ctxt}} (\alpha \notin \Gamma)
\]

\[
\frac{\Gamma \text{ctxt}}{\Gamma, x : A \text{ctxt}} (x \notin \Gamma)
\]
Models of $\lambda^2$
\( \lambda^2 \) fibrations [Seely, 1987; see also Jacobs, 1999]

**Definition (\( \lambda^2 \) fibration)**

A \( \lambda^2 \) fibration is a fibration \( p : T \to C \), where the base category \( C \) has finite products, and the fibration:

1. is fibred cartesian closed;
2. has a generic object \( U \) — we write \( \Omega \) for \( pU \);
3. and has fibred-products along projections \( X \times \Omega \to X \) in \( C \).
A \( \lambda^2 \) fibration is a split fibration \( p : \mathcal{T} \to \mathcal{C} \), where the base category \( \mathcal{C} \) has finite products, and the fibration:

1. is fibred cartesian closed;
2. has a split generic object \( U \) — we write \( \Omega \) for \( pU \);
3. and has fibred-products along projections \( X \times \Omega \to X \) in \( \mathcal{C} \).

Moreover, the reindexing functors given by the splitting should preserve the above-specified structure in fibres on the nose.
Structure in detail (i)

- Fibration $p : \mathbb{T} \to \mathbb{C}$, $\mathbb{C}$ has finite products.
  - $\mathbb{C}$ category of type variable contexts and substitutions.
  - Products are context concatenation.

\[ \begin{array}{c}
\mathbb{T} \\
\downarrow p \\
\mathbb{C}
\end{array} \]
Structure in detail (i)

- **Fibration** $p : \mathbb{T} \to \mathbb{C}$, $\mathbb{C}$ has finite products.
  - $\mathbb{C}$ category of type variable contexts and substitutions.
  - Products are context concatenation.
  - **Fibre** $\mathbb{T}_\Gamma$ category of types in context $\Gamma$. 
Structure in detail (i)

- **Fibration** \( p : \mathbb{T} \to \mathbb{C} \), \( \mathbb{C} \) has finite products.
  - \( \mathbb{C} \) category of type variable contexts and substitutions.
  - Products are context concatenation.
  - Fibre \( \mathbb{T}_\Gamma \) category of types in context \( \Gamma \).
Structure in detail (i)

- **Fibration** \( p : \mathbb{T} \to \mathbb{C} \), \( \mathbb{C} \) has finite products.
  - \( \mathbb{C} \) category of type variable contexts and substitutions.
  - Products are context concatenation.
  - Fibre \( \mathbb{T}_\Gamma \) category of types in context \( \Gamma \).

\[ \text{Diagram:} \]

\[ \mathbb{T}_\Gamma \quad \sigma \quad \mathbb{T}_{\Gamma'} \]

\[ \mathbb{C} \quad \Gamma \quad \Gamma' \]
Structure in detail (i)

- Fibration \( p : \mathbb{T} \to \mathbb{C} \), \( \mathbb{C} \) has finite products.
  - \( \mathbb{C} \) category of type variable contexts and substitutions.
  - Products are context concatenation.
  - Fibre \( \mathbb{T}_\Gamma \) category of types in context \( \Gamma \).
  - Reindexing is substitution.
Structure in detail (ii)

... is fibred cartesian closed;
  ▶ Each fibre is closed under exponentials.
  ▶ Needed for →.

... has a split generic object we write $\Omega$ for $\mathcal{U}$;

Every object $A$ in $\mathcal{T}$ arises as $A \sim \sigma^*(\mathcal{U})$ for some $\sigma$:

$\mathcal{P}(A) - \Omega$.

Every type arises uniquely by substitution from a generic type.

The generic type $\mathcal{U}$ is a type variable $\alpha$ in context $\Omega = \alpha$.

Needed for type variables.

... and has broad products along projections $\Gamma \times \Omega - \Gamma$ in $C$.

Each reindexing functor $\pi^*\Omega: \mathcal{T}\Gamma \rightarrow \mathcal{T}\Gamma \times \Omega$ has a right adjoint $\Pi\Omega: \mathcal{T}\Gamma \times \Omega \rightarrow \mathcal{T}\Gamma$.

Needed for $\forall$. 
Structure in detail (ii)

- ... is fibred cartesian closed;
  - Each fibre is closed under exponentials.
  - Needed for $\to$.

- ... has a split generic object $U$ — we write $\Omega$ for $pU$;
  - Every object $A$ in $\mathbb{T}$ arises as $A \cong \sigma^*(U)$ for some $\sigma : p(A) \to \Omega$.
  - “Every type arises uniquely by substitution from a generic type”.
  - The generic type $U$ is a type variable $\alpha$ in context $\Omega = \alpha$.
  - Needed for type variables.
Structure in detail (ii)

- ... is fibred cartesian closed;
  - Each fibre is closed under exponentials.
  - Needed for $\to$.

- ... has a split generic object $U$ — we write $\Omega$ for $pU$;
  - Every object $A$ in $\mathcal{T}$ arises as $A \cong \sigma^*(U)$ for some $\sigma : p(A) \to \Omega$.
  - “Every type arises uniquely by substitution from a generic type”.
  - The generic type $U$ is a type variable $\alpha$ in context $\Omega = \alpha$.
  - Needed for type variables.

- ... and has fibred-products along projections $\Gamma \times \Omega \to \Gamma$ in $\mathcal{C}$.
  - Each reindexing functor $\pi^*_\Omega : \mathcal{T}_\Gamma \to \mathcal{T}_{\Gamma \times \Omega}$ has a right adjoint $\prod_\Omega : \mathcal{T}_{\Gamma \times \Omega} \to \mathcal{T}_\Gamma$.
  - Needed for $\forall$. 
Old-fashioned interpretation

Given context $\Gamma$, let $\Theta = \alpha_1, \ldots, \alpha_n$ and $\Delta = x_1 : A_1, \ldots, x_m : A_m$ be the type and term variable components of $\Gamma$. 
Old-fashioned interpretation

- Given context $\Gamma$, let $\Theta = \alpha_1, \ldots, \alpha_n$ and $\Delta = x_1 : A_1, \ldots, x_m : A_m$ be the type and term variable components of $\Gamma$.

- Type variable context $\Theta = \alpha_1, \ldots, \alpha_n$ interpreted as $[\Theta] = \Omega^n$ in $\mathbb{C}$.
Old-fashioned interpretation

- Given context $\Gamma$, let $\Theta = \alpha_1, \ldots, \alpha_n$ and $\Delta = x_1 : A_1, \ldots, x_m : A_m$ be the type and term variable components of $\Gamma$.

- Type variable context $\Theta = \alpha_1, \ldots, \alpha_n$ interpreted as $[\Theta] = \Omega^n$ in $\mathbb{C}$.

- Type $A$ in context $\Theta$ is interpreted as an object in $T_{[\Theta]}$. 
Old-fashioned interpretation

- Given context $\Gamma$, let $\Theta = \alpha_1, \ldots, \alpha_n$ and $\Delta = x_1 : A_1, \ldots, x_m : A_m$ be the type and term variable components of $\Gamma$.

- Type variable context $\Theta = \alpha_1, \ldots, \alpha_n$ interpreted as $[\Theta] = \Omega^n$ in $\mathbb{C}$.

- Type $A$ in context $\Theta$ is interpreted as an object in $T_{[\Theta]}$.

- Term variable context $\Delta = x_1 : A_1, \ldots, x_m : A_m$ interpreted as $[\Delta] = [A_1] \times \ldots \times [A_m]$ in $T_{[\Theta]}$. 
Old-fashioned interpretation

- Given context $\Gamma$, let $\Theta = \alpha_1, \ldots, \alpha_n$ and $\Delta = x_1 : A_1, \ldots, x_m : A_m$ be the type and term variable components of $\Gamma$.

- Type variable context $\Theta = \alpha_1, \ldots, \alpha_n$ interpreted as $[\Theta] = \Omega^n$ in $\mathbb{C}$.

- Type $A$ in context $\Theta$ is interpreted as an object in $T_{[\Theta]}$.

- Term variable context $\Delta = x_1 : A_1, \ldots, x_m : A_m$ interpreted as $[\Delta] = [A_1] \times \ldots \times [A_m]$ in $T_{[\Theta]}$.

- Term $\Gamma \vdash t : A$ is interpreted as morphism

$$[t]_{\Theta;\Delta} : [\Delta] \to [A] \quad \text{in} \quad T_{[\Theta]}$$
Old-fashioned interpretation

Given context $\Gamma$, let $\Theta = \alpha_1, \ldots, \alpha_n$ and $\Delta = x_1 : A_1, \ldots, x_m : A_m$ be the type and term variable components of $\Gamma$.

Type variable context $\Theta = \alpha_1, \ldots, \alpha_n$ interpreted as $[[\Theta]] = \Omega^n$ in $\mathcal{C}$.

Type $A$ in context $\Theta$ is interpreted as an object in $T_{[[\Theta]]}$.

Term variable context $\Delta = x_1 : A_1, \ldots, x_m : A_m$ interpreted as $[[\Delta]] = [[A_1]] \times \ldots \times [[A_m]]$ in $T_{[[\Theta]]}$.

Term $\Gamma \vdash t : A$ is interpreted as morphism

$$[[t]]_{\Theta;\Delta} : [[\Delta]] \to [[A]] \quad \text{in} \quad T_{[[\Theta]]}$$

The combined context made things awkward; let’s fix that by modifying the notion of model and giving a new interpretation.
Our modification: one new ingredient

We take inspirations from models of dependent types, where separated contexts are not possible.

Definition (Comprehensive $\lambda$2 fibration)

A $\lambda$2 fibration $p : \mathbb{T} \to \mathbb{C}$ is comprehensive if it enjoys the comprehension property: the fibred-terminal-object functor $X \mapsto 1_X : \mathbb{C} \to \mathbb{T}$ has a specified right adjoint $K : \mathbb{T} \to \mathbb{C}$.

- Given $A \in \mathbb{T}_\Gamma$, think of $K(A)$ as the extended context $\Gamma, x : A$.
- For $A \in \mathbb{T}_\Gamma$, write $\kappa_A = p(\varepsilon_A) : K(A) \to \Gamma$ for the ‘projection’ map obtained by applying $p$ to the counit $\varepsilon_A : 1_{K(A)} \to A$ in $\mathbb{T}$.
Interpretation in a comprehensive \( \lambda \) fibration

- Contexts \( \Gamma \) interpreted as object \([\Gamma]\) in \( \mathcal{C} \).
- Type \( \Gamma \vdash A \) type interpreted as object \([A]_\Gamma\) in \( \mathcal{T}_{\Gamma} \).
Interpretation in a comprehensive $\lambda 2$ fibration

- Contexts $\Gamma$ interpreted as object $[\Gamma]$ in $\mathcal{C}$.

- Type $\Gamma \vdash A$ type interpreted as object $[A]_{\Gamma}$ in $\mathcal{T}[\Gamma]$.

- Mutually defined, simultaneously with maps $\pi^\alpha_{\Gamma}: [\Gamma] \rightarrow \Omega$ for every context $\Gamma$ containing $\alpha$.

$$
\begin{align*}
[\cdot] &= 1 \\
[\Gamma, \alpha] &= [\Gamma] \times \Omega \\
[\Gamma, x : A] &= K[A]_{\Gamma} \\
[\forall \alpha. A] &= \prod_{\Omega} [A]_{\Gamma, \alpha} \\
\pi^\alpha_{\Gamma, \alpha} &= \pi_2 \\
\pi^\alpha_{\Gamma, \beta} &= \pi^\alpha_{\Gamma} \circ \pi_1 (\beta \neq \alpha) \\
\pi^\alpha_{\Gamma, x : A} &= \pi^\alpha_{\Gamma} \circ \kappa [A]_{\Gamma}
\end{align*}
$$
Interpretation in a comprehensive $\lambda 2$ fibration

- Contexts $\Gamma$ interpreted as object $[\Gamma]$ in $C$.

- Type $\Gamma \vdash A$ type interpreted as object $[A]_\Gamma$ in $T[\Gamma]$.

- Mutually defined, simultaneously with maps $\pi^\alpha_\Gamma : [\Gamma] \rightarrow \Omega$ for every context $\Gamma$ containing $\alpha$.

\[
\begin{align*}
[\cdot] &= 1 \\
[\Gamma, \alpha] &= [\Gamma] \times \Omega \\
[\Gamma, x : A] &= K[A]_\Gamma \\
[\forall \alpha. A] &= \prod_\Omega [A]_\Gamma, \alpha
\end{align*}
\]

\[
\begin{align*}
\pi^\alpha_{\Gamma, \alpha} &= \pi_2 \\
\pi^\alpha_{\Gamma, \beta} &= \pi^\alpha_{\Gamma} \circ \pi_1 (\beta \neq \alpha) \\
\pi^\alpha_{\Gamma, x : A} &= \pi^\alpha_{\Gamma} \circ \kappa[A]_\Gamma
\end{align*}
\]

- Term $\Gamma \vdash t : A$ is interpreted as global element

\[
[t]_\Gamma : 1_{[\Gamma]} \rightarrow [A]_\Gamma \text{ in } T[\Gamma]
\]
For future reference

Compare the interpretation of terms in standard and comprehensive \( \lambda 2 \) fibrations:

- \([t]_{\Theta; \Delta} : [\Delta] \rightarrow [A] \) in \( T_{[\Theta]} \) (old-fashioned, standard)

- versus global element

\[
[t]_\Gamma : 1_{[\Gamma]} \rightarrow [A]_\Gamma \quad \text{in} \quad T_{[\Gamma]}
\]

(comprehensive)
Soundness and completeness

Theorem (Soundness for $\lambda 2$)

If $\Gamma \vdash t_1 = t_2 : A$ then, in every comprehensive $\lambda 2$ fibration, we have $\llbracket t_1 \rrbracket_\Gamma = \llbracket t_2 \rrbracket_\Gamma$.

Theorem (Full completeness for $\lambda 2$)

There exists a comprehensive $\lambda 2$ fibration satisfying:

1. for every type $\Gamma \vdash A$ type, every global point $1_{[\Gamma]} \longrightarrow \llbracket A \rrbracket_\Gamma$ is the denotation $\llbracket t \rrbracket_\Gamma$ of some term $\Gamma \vdash t : A$; and

2. for all terms $\Gamma \vdash t_1, t_2 : A$ satisfying $\llbracket t_1 \rrbracket_\Gamma = \llbracket t_2 \rrbracket_\Gamma$, we have $\Gamma \vdash t_1 = t_2 : A$. 
(Comprehensive) parametricity graphs
Incorporating relational parametricity

- These models do not model parametricity.

- In order to do so, we combine with the structure of reflexive graph categories [Ma and Reynolds, 1992; Robinson and Rosolini, 1994; O’Hearn and Tennent, 1995; ...].

- Simple category-theoretic structure for modelling relations.
Reflexive graph categories

- Categories $\mathbb{V}$ and $\mathbb{E}$, where we think of $\mathbb{E}$ as the category of relations over objects of $\mathbb{V}$.

- The functors $\nabla_1$, $\nabla_2$ are ‘projection’ functors giving source and target of relations, respectively, and $\Delta$ maps an object to its ‘identity relation’.
Reflexive graph categories

- Categories $\mathbb{V}$ and $\mathbb{E}$, where we think if $\mathbb{E}$ as category of relations over objects of $\mathbb{V}$.

- The functors $\nabla_1$, $\nabla_2$ are ‘projection’ functors giving source and target of relations, respectively, and $\Delta$ maps an object to its ‘identity relation’.

- Notation: $R: A \leftrightarrow B$ means $R \in \mathbb{E}$ and $\nabla_1 R = A$, $\nabla_2 R = B$.

- Similarly, write $f \times g: R \to S$ if there is $h: R \to S$ in $\mathbb{E}$ with $\nabla_1 h = f$ and $\nabla_2 h = g$. (Will soon assume $h$ is unique, if it exists.)
Parametricity graphs [Dunphy, 2002; Dunphy and Reddy, 2004]

We need to add further conditions to ensure that the objects of $\mathbb{E}$ behave sufficiently like relations.
Parametricity graphs [Dunphy, 2002; Dunphy and Reddy, 2004]

We need to add further conditions to ensure that the objects of $\mathcal{E}$ behave sufficiently like relations.

- **Relational** if $\langle \nabla_1, \nabla_2 \rangle : \mathcal{E} \rightarrow \mathcal{V} \times \mathcal{V}$ is faithful. Intuitively, relations are proof-irrelevant.
Parametricity graphs [Dunphy, 2002; Dunphy and Reddy, 2004]

We need to add further conditions to ensure that the objects of $E$ behave sufficiently like relations.

- **Relational** if $\langle \nabla_1, \nabla_2 \rangle : E \rightarrow V \times V$ is faithful. Intuitively, relations are proof-irrelevant.

- **Identity property** if for every $h : \Delta A \rightarrow \Delta B$ in $E$, it holds that $\nabla_1 h = \nabla_2 h$. Allows one to think of $\Delta A$ as an identity relation on $A$. 
Parametricity graphs [Dunphy, 2002; Dunphy and Reddy, 2004]

\[ \begin{array}{c}
\nabla_1 \\
\mathbb{E} \\
\Delta \\
\nabla_2 \\
\mathbb{V} \\
\end{array} \]

- We need to add further conditions to ensure that the objects of \( \mathbb{E} \) behave sufficiently like relations.
- **Relational** if \( \langle \nabla_1, \nabla_2 \rangle : \mathbb{E} \to \mathbb{V} \times \mathbb{V} \) is faithful. Intuitively, relations are proof-irrelevant.
- **Identity property** if for every \( h : \Delta A \longrightarrow \Delta B \) in \( \mathbb{E} \), it holds that \( \nabla_1 h = \nabla_2 h \). Allows one to think of \( \Delta A \) as an identity relation on \( A \).
- **Parametricity graph**: relational, with the identity property, and \( \langle \nabla_1, \nabla_2 \rangle : \mathbb{E} \to \mathbb{V} \times \mathbb{V} \) a fibration. Ensures that there are enough relations by supplying inverse image relations.
Combining reflexive graphs and comprehensive $\lambda^2$ fibrations
A **comprehensive** $\lambda 2$ **parametricity graph** is a reflexive graph of comprehensive $\lambda 2$ fibrations

\[
\begin{align*}
\mathcal{R}(\mathbb{T}) & \xleftarrow{\phantom{\nabla_1^T}} \xrightarrow{\nabla_1^T, \Delta^T, \nabla_2^T} \mathbb{T} \\
\mathcal{R}(\mathbb{C}) & \xleftarrow{\phantom{\nabla_1^C}} \xrightarrow{\nabla_1^C, \Delta^C, \nabla_2^C} \mathbb{C}
\end{align*}
\]

which is “fibrewise” a parametricity graph.
Combining reflexive graphs and comprehensive \( \lambda^2 \) fibrations

**Main definition (Comprehensive \( \lambda^2 \) parametricity graph)**

A comprehensive \( \lambda^2 \) parametricity graph is a reflexive graph of comprehensive \( \lambda^2 \) fibrations

\[
\begin{align*}
\mathcal{R} & \xleftarrow{} \begin{pmatrix} \nabla^T_1, \Delta^T, \nabla^T_2 \end{pmatrix} T \\
\mathcal{R}(\mathbb{T}) & \xrightarrow{} \begin{pmatrix} \nabla^T_1, \Delta^T, \nabla^T_2 \end{pmatrix} T \\
\mathcal{R}(\mathbb{C}) & \xleftarrow{} \begin{pmatrix} \nabla^C_1, \Delta^C, \nabla^C_2 \end{pmatrix} C \\
\mathbb{C} & \xrightarrow{} \begin{pmatrix} \nabla^C_1, \Delta^C, \nabla^C_2 \end{pmatrix} C
\end{align*}
\]

which is “fibrewise” a parametricity graph.

**Note:** Recover “broken” definition by dropping comprehensive.
A type theory for reasoning about parametricity
Reasoning in models: a type theory $\lambda 2R$

- We construct a type theory $\lambda 2R$ which is the ‘internal language’ of comprehensive $\lambda 2$ parametricity graphs.

- By proving soundness and completeness, we can work in $\lambda 2R$ instead of directly in the model.

- $\lambda 2R$ is similar in many respects to System R [Abadi, Cardelli and Curien, 1993] and System P [Dunphy, 2002].
We construct a type theory \( \lambda 2R \) which is the ‘internal language’ of comprehensive \( \lambda 2 \) parametricity graphs.

By proving soundness and completeness, we can work in \( \lambda 2R \) instead of directly in the model.

\( \lambda 2R \) is similar in many respects to System R [Abadi, Cardelli and Curien, 1993] and System P [Dunphy, 2002].

Not a conservative extension of \( \lambda 2 \) — parametric models enjoy much stronger properties than arbitrary models (for which \( \lambda 2 \) is internal language).
New judgement forms

\( \lambda 2R \) extends \( \lambda 2 \) with three new judgements:

\[
\begin{align*}
\Theta & \text{ rctxt} & \Theta \text{ is a relational context} \\
\Theta & \vdash A_1 RA_2 \text{ rel} & R \text{ is a relation between types } A_1 \text{ and } A_2 \\
\Theta & \vdash (t_1 : A_1) R (t_2 : A_2) & t_1 : A_1 \text{ is related to } t_2 : A_2 \text{ by the relation } R
\end{align*}
\]
Relation formation rules

\[ \Theta \vdash \alpha \rho \beta \text{ rel} \quad (\alpha \rho \beta \in \Theta) \]

\[ \Theta \vdash A_1 RA_2 \text{ rel} \quad \Theta \vdash B_1 SB_2 \text{ rel} \]

\[ \Theta \vdash (A_1 \rightarrow B_1)(R \rightarrow S)(A_2 \rightarrow B_2) \text{ rel} \]

\[ \Theta, \alpha \rho \beta \vdash A_1 RA_2 \text{ rel} \]

\[ \Theta \vdash (\forall \alpha. A_1)(\forall \alpha \rho \beta. R)(\forall \beta. A_2) \text{ rel} \]
Relation formation rules

\[ \Theta 
\vdash \alpha \rho \beta \quad \text{rel} \quad (\alpha \rho \beta \in \Theta) \]

\[ \Theta 
\vdash A_1 RA_2 \quad \text{rel} \quad \Theta 
\vdash B_1 SB_2 \quad \text{rel} \]

\[ \Theta 
\vdash (A_1 \to B_1)(R \to S)(A_2 \to B_2) \quad \text{rel} \]

\[ \Theta, \; \alpha \rho \beta 
\vdash A_1 RA_2 \quad \text{rel} \]

\[ \Theta 
\vdash (\forall \alpha. \; A_1)(\forall \alpha \rho \beta. \; R)(\forall \beta. \; A_2) \quad \text{rel} \]

\[ \Theta 
\vdash B_1 RB_2 \quad \text{rel} \]

\[ (\Theta)_1 \vdash t_1 : A_1 \to B_1 \quad (\Theta)_2 \vdash t_2 : A_2 \to B_2 \]

\[ \Theta 
\vdash A_1 ([t_1 \times t_2]^{-1} R)A_2 \quad \text{rel} \]

(Will get back to projections \((-)_{i}\) soon.)
Direct image relations

Direct image relations

\[ \Theta \vdash A_1 \text{ rel} \quad (\Theta)_1 \vdash t_1 : A_1 \to B_1 \quad (\Theta)_2 \vdash t_2 : A_2 \to B_2 \]

\[ \Theta \vdash B_1 ([t_1 \times t_2]! R) B_2 \text{ rel} \]
Direct image relations

Direct image relations

$$\Theta \vdash A_1 RA_2 \text{ rel} \quad (\Theta)_1 \vdash t_1 : A_1 \to B_1 \quad (\Theta)_2 \vdash t_2 : A_2 \to B_2$$

$$\Theta \vdash B_1 ([t_1 \times t_2]! R)B_2 \text{ rel}$$

are definable by the impredicative encoding

$$[t_1 \times t_2]! R := [i_{B_1} \times i_{B_2}]^{-1}(\forall \alpha \rho \beta. ((- \circ t_1) \times (- \circ t_2))^{-1}(R \to \rho)) \to \rho$$

where $$i_B$$ abbreviates $$\lambda b. \lambda \alpha. \lambda t. t b : B \to \forall \alpha. (B \to \alpha) \to \alpha.$$
Direct image relations

$$\Theta \vdash A_1 \text{RA}_2 \rel$$
$$\vdash (\Theta)_1 \vdash t_1 : A_1 \rightarrow B_1$$
$$\vdash (\Theta)_2 \vdash t_2 : A_2 \rightarrow B_2$$
$$\vdash \Theta \vdash B_1 ([t_1 \times t_2]_! R)B_2 \rel$$

are definable by an impredicative encoding.
Direct image relations

\[ \Theta \vdash A_1 R A_2 \text{ rel} \quad (\Theta)_1 \vdash t_1 : A_1 \to B_1 \quad (\Theta)_2 \vdash t_2 : A_2 \to B_2 \]

\[ \Theta \vdash B_1 ([t_1 \times t_2]! R) B_2 \text{ rel} \]

are definable by an impredicative encoding.

Semantically, this means:

**Theorem**

*In any comprehensive \( \lambda 2 \) parametricity graph, the functors*

\[ \langle \nabla^T_1, \nabla^T_2 \rangle \upharpoonright \mathcal{R}(T)_W : \mathcal{R}(T)_W \to T_{\nabla^C_1 W} \times T_{\nabla^C_2 W} \]

*are also opfibrations* (hence bifibrations).
Operations on syntax

- Left and right projections $(\cdot)_1, (\cdot)_2$ from relational contexts to typing contexts.

\[
(\cdot)_i = \cdot \\
(\Theta, \alpha_1 \rho \alpha_2)_i = (\Theta)_i, \alpha_i \\
(\Theta, (x_1 : A_1)R(x_2 : A_2))_i = (\Theta)_i, x_i : A_i
\]
Operations on syntax

- Left and right projections \((\cdot)_1, (\cdot)_2\) from relational contexts to typing contexts.

\[
(\cdot)_i = \cdot \\
(\Theta, \alpha_1 \rho \alpha_2)_i = (\Theta)_i, \alpha_i \\
(\Theta, (x_1 : A_1)R(x_2 : A_2))_i = (\Theta)_i, x_i : A_i
\]

- Conversely, a “doubling” operation takes typing contexts to relational contexts.

- Mutually defined with a “relational interpretation” \(\langle A \rangle\) of types \(A\).

\[
\begin{align*}
\langle \cdot \rangle &= \cdot \\
\langle \Gamma, \alpha \rangle &= \langle \Gamma \rangle, \alpha \rho^\alpha \alpha \\
\langle \Gamma, x : A \rangle &= \langle \Gamma \rangle, (x : A)\langle A \rangle (x : A) \\
\langle A \rightarrow B \rangle &= \langle A \rangle \rightarrow \langle B \rangle \\
\langle \forall \alpha. A \rangle &= \forall \alpha \rho^\alpha \alpha. \langle A \rangle
\end{align*}
\]
Operations on syntax

- Left and right projections $(\cdot)_1, (\cdot)_2$ from relational contexts to typing contexts.

\[ (\cdot)_i = \cdot \]

\[ (\Theta, \alpha_1 \rho \alpha_2)_i = (\Theta)_i, \alpha_i \]

\[ (\Theta, (x_1 : A_1)R(x_2 : A_2))_i = (\Theta)_i, x_i : A_i \]

- Conversely, a “doubling” operation takes typing contexts to relational contexts.

- Mutually defined with a “relational interpretation” $\langle A \rangle$ of types $A$.

\[ \langle \cdot \rangle = \cdot \]

\[ \langle \alpha \rangle = \rho^\alpha \]

\[ \langle \Gamma, \alpha \rangle = \langle \Gamma \rangle, \alpha \rho^\alpha \alpha \]

\[ \langle A \rightarrow B \rangle = \langle A \rangle \rightarrow \langle B \rangle \]

\[ \langle \Gamma, x : A \rangle = \langle \Gamma \rangle, (x : A)\langle A \rangle (x : A) \]

\[ \langle \forall \alpha. A \rangle = \forall \alpha \rho^\alpha \alpha. \langle A \rangle \]

- **Note:** Left and right hand side treated separately, so e.g. $\alpha \rho^\alpha \alpha$ equivalent to $\alpha \rho \beta$ if everything fresh.
Reflexive graph structure on syntax

Lemma

1. If $\Theta \vdash (t_1 : A_1)R(t_2 : A_2)$ then $(\Theta)_i \vdash t_i : A_i$.
2. If $\Gamma \vdash t : A$ then $\langle \Gamma \rangle \vdash (t : A)\langle A \rangle(t : A)$.

Second item is Reynolds’ Abstraction Theorem in our setting.
Relatedness rules: standard relation formers

\[ \Theta \vdash (x_1 : A_1)R(x_2 : A_2) \quad ((x_1 : A_1)R(x_2 : A_2) \in \Theta) \]

\[ \Theta, (x_1 : A_1)R(x_2 : A_2) \vdash (t_1 : B_1)S(t_2 : B_2) \]

\[ \Theta \vdash (\lambda x_1. t_1 : A_1 \to B_1)(R \to S)(\lambda x_2. t_2 : A_2 \to B_2) \]

\[ \Theta \vdash (s_1 : A_1 \to B_1)(R \to S)(s_2 : A_2 \to B_2) \quad \Theta \vdash (t_1 : A_1)R(t_2 : A_2) \]

\[ \Theta \vdash (s_1 \ t_1 : B_1)S(s_2 \ t_2 : B_2) \]

\[ \Theta, \alpha\rho\beta \vdash (t_1 : A_1)R(t_2 : A_2) \]

\[ \Theta \vdash (\wedge \alpha. t_1 : \forall \alpha. A_1)(\forall \alpha\rho\beta. R)(\wedge \beta. t_2 : \forall \beta. A_2) \]

\[ \Theta \vdash (t_1 : \forall \alpha. A_1)(\forall \alpha\rho\beta. R)(t_2 : \forall \beta. A_2) \quad \Theta \vdash B_1 SB_2 \text{ rel} \]

\[ \Theta \vdash (t_1[B_1] : A_1[\alpha \mapsto B_1])R[\alpha\rho\beta \mapsto B_1 SB_2](t_2[B_2] : A_2[\beta \mapsto B_2]) \]
Relatedness rules: standard relation formers

\[
\Theta \vdash (x_1 : A_1)R(x_2 : A_2) \quad ((x_1 : A_1)R(x_2 : A_2) \in \Theta)
\]

\[
\Theta, (x_1 : A_1)R(x_2 : A_2) \vdash (t_1 : B_1)S(t_2 : B_2)
\]

\[
\Theta \vdash (\lambda x_1. t_1 : A_1 \to B_1)(R \to S)(\lambda x_2. t_2 : A_2 \to B_2)
\]

\[
\Theta \vdash (s_1 : A_1 \to B_1)(R \to S)(s_2 : A_2 \to B_2) \quad \Theta \vdash (t_1 : A_1)R(t_2 : A_2)
\]

\[
\Theta \vdash (s_1 t_1 : B_1)S(s_2 t_2 : B_2)
\]

\[
\Theta, \alpha \rho \beta \vdash (t_1 : A_1)R(t_2 : A_2)
\]

\[
\Theta \vdash (\land \alpha. t_1 : \forall \alpha. A_1)(\forall \alpha \rho \beta. R)(\land \beta. t_2 : \forall \beta. A_2)
\]

\[
\Theta \vdash (t_1 : \forall \alpha. A_1)(\forall \alpha \rho \beta. R)(t_2 : \forall \beta. A_2) \quad \Theta \vdash B_1 SB_2 \text{ rel}
\]

\[
\Theta \vdash (t_1 [B_1] : A_1[\alpha \mapsto B_1])R[\alpha \rho \beta \mapsto B_1 SB_2](t_2[B_2] : A_2[\beta \mapsto B_2])
\]
Relatedness rules: standard relation formers

\[
\Theta \vdash (x_1 : A_1)R(x_2 : A_2) \quad ((x_1 : A_1)R(x_2 : A_2) \in \Theta)
\]

\[
\Theta, (x_1 : A_1)R(x_2 : A_2) \vdash (t_1 : B_1)S(t_2 : B_2)
\]

\[
\Theta \vdash (\lambda x_1 . t_1 : A_1 \to B_1)(R \to S)(\lambda x_2 . t_2 : A_2 \to B_2)
\]

\[
\Theta \vdash (s_1 : A_1 \to B_1)(R \to S)(s_2 : A_2 \to B_2) \quad \Theta \vdash (t_1 : A_1)R(t_2 : A_2)
\]

\[
\Theta \vdash (s_1 t_1 : B_1)S(s_2 t_2 : B_2)
\]

\[
\Theta, \alpha \rho \beta \vdash (t_1 : A_1)R(t_2 : A_2)
\]

\[
\Theta \vdash (\land \alpha . t_1 : \forall \alpha . A_1)(\forall \alpha \rho \beta . R)(\land \beta . t_2 : \forall \beta . A_2)
\]

\[
\Theta \vdash (t_1 : \forall \alpha . A_1)(\forall \alpha \rho \beta . R)(t_2 : \forall \beta . A_2) \quad \Theta \vdash B_1 SB_2 \text{ rel}
\]

\[
\Theta \vdash (t_1 [B_1] : A_1[\alpha \mapsto B_1])R[\alpha \rho \beta \mapsto B_1 SB_2](t_2[B_2] : A_2[\beta \mapsto B_2])
\]
Relatedness rules: inverse image relations and substitution

\[ \Theta \vdash (t_1 \ u_1 : B_1) R (t_2 \ u_2 : B_2) \]

\[ \Theta \vdash (u_1 : A_1) ([t_1 \times t_2]^{-1} R) (u_2 : A_2) \]

\[ \Theta \vdash (t_1 : A_1) R (t_2 : A_2) \quad \Theta_1 \vdash t_1 = s_1 : A_1 \quad \Theta_2 \vdash t_2 = s_2 : A_2 \]

\[ \Theta \vdash (s_1 : A_1) R (s_2 : A_2) \]
One more rule: the parametricity rule

- The system get its power from inverse image relations together with the parametricity rule.

- Recall: If $\Gamma \vdash t : A$ then $\langle \Gamma \rangle \vdash (t : A)\langle A \rangle(t : A)$.

- So $\langle A \rangle$ is the equality relation? No! Only in closed contexts.

- In fact, for open types, $\langle \alpha \rangle = \alpha \rho \beta$. 
One more rule: the parametricity rule

- The system gets its power from inverse image relations together with the parametricity rule.

- Recall: If $\Gamma \vdash s = t : A$ then $\langle \Gamma \rangle \vdash (s : A)\langle A \rangle(t : A)$.
One more rule: the parametricity rule

- The system get its power from inverse image relations together with the **parametricity rule**.

- Recall: If $\Gamma \vdash s = t : A$ then $\langle \Gamma \rangle \vdash (s : A)\langle A\rangle(t : A)$.

- Parametricity rule states converse:

\[
\frac{\langle \Gamma \rangle \vdash (s : A)\langle A\rangle(t : A)}{\Gamma \vdash s = t : A}
\]
One more rule: the parametricity rule

- The system gets its power from inverse image relations together with the parametricity rule.

- Recall: If $\Gamma \vdash s = t : A$ then $\langle \Gamma \rangle \vdash (s : A)\langle A\rangle (t : A)$.

- Parametricity rule states converse:

$$
\Gamma \vdash (s : A)\langle A\rangle (t : A) \\
\Gamma \vdash s = t : A
$$

- So $\langle A \rangle$ is the equality relation? No! Only in closed contexts.

- In fact, for open types, $\langle A \rangle$ is not even a homogeneous relation, since $\langle \alpha \rangle = \alpha \rho \beta$. 
Interpretation in comprehensive $\lambda 2$ parametricity graphs

$\nabla^T_1, \Delta^T, \nabla^T_2$

$\mathcal{R}(\mathbb{T}) \xrightarrow{\nabla^T_1, \Delta^T, \nabla^T_2} \mathbb{T}$

$R \left( \mathcal{R}(\mathbb{T}) \right) \xrightarrow{\nabla^C_1, \Delta^C, \nabla^C_2} \mathcal{R}(\mathbb{C})$

$\lambda 2$ interpreted in $p$, as before.

Relational context $\Theta$ interpreted as an object $[\Theta]$ in $\mathcal{R}(\mathbb{C})$.

Syntactic relation $\Theta \vdash ARB$ rel interpreted as a semantic relation $[R]_\Theta : [A]_{(\Theta)_1} \leftrightarrow [B]_{(\Theta)_2}$ in $\mathcal{R}(\mathbb{T})_{[\Theta]}$ using $\lambda 2$ structure.
Interpretation of inverse image relations

- Inverse-image relation $\Theta \vdash A_1([t_1 \times t_2]^{-1} R)A_2$ rel interpreted using the \textit{fibration} property of the parametricity graph:

- Have

\[
\begin{align*}
[t_1]_{(\Theta)_1} : 1 & \twoheadrightarrow [A_1]_{(\Theta)_1} \Rightarrow [B_1]_{(\Theta)_1} \\
[t_2]_{(\Theta)_2} : 1 & \twoheadrightarrow [A_2]_{(\Theta)_2} \Rightarrow [B_2]_{(\Theta)_2}
\end{align*}
\]
Interpretation of inverse image relations

- Inverse-image relation $\Theta \vdash A_1([t_1 \times t_2]^{-1} R)A_2$ rel interpreted using the *fibration* property of the parametricity graph:

- Have

$$\llbracket t_1 \rrbracket_{(\Theta)_1} : 1 \times \llbracket A_1 \rrbracket_{(\Theta)_1} \longrightarrow \llbracket B_1 \rrbracket_{(\Theta)_1}$$

$$\llbracket t_2 \rrbracket_{(\Theta)_2} : 1 \times \llbracket A_2 \rrbracket_{(\Theta)_2} \longrightarrow \llbracket B_2 \rrbracket_{(\Theta)_2}$$
Interpretation of inverse image relations

- Inverse-image relation $\Theta \vdash A_1([t_1 \times t_2]^{-1} R)A_2$ rel interpreted using the *fibration* property of the parametricity graph:

- Have

$$\llbracket t_1 \rrbracket(\Theta)_1 : \llbracket A_1 \rrbracket(\Theta)_1 \rightarrow \llbracket B_1 \rrbracket(\Theta)_1$$
$$\llbracket t_2 \rrbracket(\Theta)_2 : \llbracket A_2 \rrbracket(\Theta)_2 \rightarrow \llbracket B_2 \rrbracket(\Theta)_2$$
Interpretation of inverse image relations

- Inverse-image relation $\Theta \vdash A_1([t_1 \times t_2]^{-1} R)A_2$ rel interpreted using the \textit{fibration} property of the parametricity graph:

- Have

$$\llbracket t_1 \rrbracket_{\Theta_1} : \llbracket A_1 \rrbracket_{\Theta_1} \rightarrow \llbracket B_1 \rrbracket_{\Theta_1}$$
$$\llbracket t_2 \rrbracket_{\Theta_2} : \llbracket A_2 \rrbracket_{\Theta_2} \rightarrow \llbracket B_2 \rrbracket_{\Theta_2}$$

$$\llbracket R \rrbracket : \llbracket B_1 \rrbracket_{\Theta_1} \leftrightarrow \llbracket B_2 \rrbracket_{\Theta_2}$$
Interpretation of inverse image relations

- Inverse-image relation $\Theta \vdash A_1([t_1 \times t_2]^{-1}R)A_2$ rel interpreted using the fibration property of the parametricity graph:

- Have

\[
\begin{align*}
[t_1](\Theta)_1 : [A_1](\Theta)_1 & \rightarrow [B_1](\Theta)_1 \\
[t_2](\Theta)_2 : [A_2](\Theta)_2 & \rightarrow [B_2](\Theta)_2
\end{align*}
\]

- Reindex $[R] : [B_1](\Theta)_1 \leftrightarrow [B_2](\Theta)_2$ in the fibration along these maps to interpret $[[t_1 \times t_2]^{-1}R] : [A_1](\Theta)_1 \leftrightarrow [A_2](\Theta)_2$. 
Why didn’t this work before?

- If we try to replay the interpretation in the old-fashioned semantics without comprehension, we get:

\[
[t_1]' : ([\Delta])_1 \rightarrow ([A_1])_1 \Rightarrow ([B_1])_1 \\
[t_2]' : ([\Delta])_2 \rightarrow ([A_2])_2 \Rightarrow ([B_2])_2
\]
Why didn’t this work before?

If we try to replay the interpretation in the old-fashioned semantics without comprehension, we get:

\[
\begin{align*}
[t_1]': ([\Delta])_1 \times ([A_1])_1 &\rightarrow ([B_1])_1 \\
[t_2]': ([\Delta])_2 \times ([A_2])_2 &\rightarrow ([B_2])_2
\end{align*}
\]
Why didn’t this work before?

- If we try to replay the interpretation in the old-fashioned semantics without comprehension, we get:

\[
\begin{align*}
\llbracket t_1 \rrbracket' : (\llbracket \Delta \rrbracket)_1 & \times (\llbracket A_1 \rrbracket)_1 \rightarrow (\llbracket B_1 \rrbracket)_1 \\
\llbracket t_2 \rrbracket' : (\llbracket \Delta \rrbracket)_2 & \times (\llbracket A_2 \rrbracket)_2 \rightarrow (\llbracket B_2 \rrbracket)_2
\end{align*}
\]

- Reindexing along this does not give a relation \((\llbracket A_1 \rrbracket)_1 \leftrightarrow (\llbracket A_2 \rrbracket)_2\)!
Why didn’t this work before?

- If we try to replay the interpretation in the old-fashioned semantics without comprehension, we get:

\[
\begin{align*}
\llbracket t_1 \rrbracket' : (\llbracket \Delta \rrbracket)_1 \times (\llbracket A_1 \rrbracket)_1 & \rightarrow (\llbracket B_1 \rrbracket)_1 \\
\llbracket t_2 \rrbracket' : (\llbracket \Delta \rrbracket)_2 \times (\llbracket A_2 \rrbracket)_2 & \rightarrow (\llbracket B_2 \rrbracket)_2
\end{align*}
\]

- Reindexing along this does not give a relation \((\llbracket A_1 \rrbracket)_1 \leftrightarrow (\llbracket A_2 \rrbracket)_2\)!

- So things work because in the new semantics, \(\llbracket t_i \rrbracket(\Theta)_i\) are global points. Possible because of use of comprehension.
Theorem (Soundness for \( \lambda 2R \))

In every comprehensive \( \lambda 2 \) parametricity graph:

1. if \( \Gamma \vdash t_1 = t_2 : A \) then \( \llbracket t_1 \rrbracket_\Gamma = \llbracket t_2 \rrbracket_\Gamma \); and
2. if \( \Theta \vdash (t_1 : A_1)R(t_2 : A_2) \) then \( \llbracket t_1 \rrbracket_\Theta_1 \times \llbracket t_2 \rrbracket_\Theta_2 : 1[\Theta] \rightarrow \llbracket R \rrbracket_\Theta \).
Soundness

Theorem (Soundness for $\lambda2R$)

In every comprehensive $\lambda2$ parametricity graph:

1. if $\Gamma \vdash t_1 = t_2 : A$ then $\llbracket t_1 \rrbracket_\Gamma = \llbracket t_2 \rrbracket_\Gamma$; and

2. if $\Theta \vdash (t_1 : A_1) R (t_2 : A_2)$ then $\llbracket t_1 \rrbracket_{(\Theta)_1} \times \llbracket t_2 \rrbracket_{(\Theta)_2} : \mathbf{1}_{[\Theta]} \longrightarrow [R]_\Theta$.

Substitution in relations sound by relational property.

Parametricity rule sound by identity property.

Inverse image rules sound by fibration property.
Theorem (Full completeness for \( \lambda 2R \))

There exists a comprehensive \( \lambda 2 \) parametricity graph satisfying the following.

1. For every type \( \Gamma \vdash A \) type, every global point \( 1_{[\Gamma]} \rightarrow [A]_{\Gamma} \) is the denotation \([t]_{\Gamma}\) of some term \( \Gamma \vdash t : A \).

2. For all terms \( \Gamma \vdash t_1, t_2 : A \) satisfying \([t_1]_{\Gamma} = [t_2]_{\Gamma}\), we have \( \Gamma \vdash t_1 = t_2 : A \).

3. For every relation \( \Theta \vdash A_1 RA_2 \) type, every global point \( 1_{[\Theta]} \rightarrow [R]_{\Theta} \) arises as \([t_1](\Theta)_1 \times [t_2](\Theta)_2\) for terms \( t_1, t_2 \) such that \( \Theta \vdash (t_1 : A_1)R(t_2 : A_2) \).
Deriving the expected consequences
Warm-up: $\forall \alpha. \alpha \rightarrow \alpha$ is terminal

- Want to prove $\Gamma, z : \forall \alpha. \alpha \rightarrow \alpha \vdash z = \Lambda \alpha. \lambda x. x : \forall \alpha. \alpha \rightarrow \alpha$. 
Warm-up: \( \forall \alpha. \alpha \rightarrow \alpha \) is terminal

- Want to prove \( \Gamma, z : \forall \alpha. \alpha \rightarrow \alpha \vdash z = \Lambda \alpha. \lambda x. x : \forall \alpha. \alpha \rightarrow \alpha. \)

- By extensionality, it is enough to show
  \[
  \Gamma, z : \forall \alpha. \alpha \rightarrow \alpha, \alpha, x : \alpha \vdash z[\alpha] x = x : \alpha
  \]
Warm-up: $\forall \alpha. \alpha \to \alpha$ is terminal

- Want to prove $\Gamma, z : \forall \alpha. \alpha \to \alpha \vdash z = \Lambda \alpha. \lambda x. x : \forall \alpha. \alpha \to \alpha$.

- By extensionality, it is enough to show

  \[ \Gamma, z : \forall \alpha. \alpha \to \alpha, \alpha, x : \alpha \vdash z[\alpha]x = x : \alpha \]

- Further by the parametricity rule, it is enough to show

  \[ \langle \Gamma, z : \forall \alpha. \alpha \to \alpha, \alpha, x : \alpha \rangle \vdash (z[\alpha]x : \alpha)\langle \alpha \rangle(x : \alpha) \]
Warm-up: ∀α. α → α is terminal

- Want to prove Γ, z : ∀α. α → α ⊢ z = Λα. λx. x : ∀α. α → α.

- By extensionality, it is enough to show

  \[ \Gamma, z : \forall \alpha. \alpha \rightarrow \alpha, \alpha, x : \alpha \vdash z[\alpha] x = x : \alpha \]

- Further by the parametricity rule, it is enough to show

  \[ \langle \Gamma \rangle, z(\forall \alpha \rho \beta. \rho \rightarrow \rho) w, \alpha \rho \beta, (x : \alpha) \rho(y : \beta) \vdash (z[\alpha] x : \alpha) \rho(y : \beta) \]
Warm-up: \( \forall \alpha. \alpha \rightarrow \alpha \) is terminal

- Want to prove \( \Gamma, z : \forall \alpha. \alpha \rightarrow \alpha \vdash z = \Lambda \alpha. \lambda x. x : \forall \alpha. \alpha \rightarrow \alpha. \)

- By extensionality, it is enough to show

\[
\Gamma, z : \forall \alpha. \alpha \rightarrow \alpha, \alpha, x : \alpha \vdash z[\alpha] x = x : \alpha
\]

- Further by the parametricity rule, it is enough to show

\[
\langle \Gamma \rangle, z(\forall \alpha \rho \beta. \rho \rightarrow \rho) w, \alpha \rho \beta, (x : \alpha) \rho(y : \beta) \vdash (z[\alpha] x : \alpha) \rho(y : \beta)
\]

- \( (x : \alpha)R(w : \forall \alpha. \alpha \rightarrow \alpha) \) where \( R = ([id \times (\lambda_. y)]^{-1} \rho) \), since \( x \rho y \).
Warm-up: $\forall \alpha. \alpha \rightarrow \alpha$ is terminal

- Want to prove $\Gamma, z : \forall \alpha. \alpha \rightarrow \alpha \vdash z = \Lambda \alpha. \lambda x. x : \forall \alpha. \alpha \rightarrow \alpha$.

- By extensionality, it is enough to show

$$\Gamma, z : \forall \alpha. \alpha \rightarrow \alpha, \alpha, x : \alpha \vdash z[\alpha] x = x : \alpha$$

- Further by the parametricity rule, it is enough to show

$$\langle \Gamma \rangle, z(\forall \alpha \rho \beta. \rho \rightarrow \rho) w, \alpha \rho \beta, (x : \alpha) \rho(y : \beta) \vdash (z[\alpha] x : \alpha) \rho(y : \beta)$$

- $(x : \alpha) R(w : \forall \alpha. \alpha \rightarrow \alpha)$ where $R = ([id \times (\lambda _ . y)]^{-1} \rho)$, since $x \rho y$.

- Since $z(\forall \rho. \rho \rightarrow \rho) w$, by instantiating $\alpha \rho \beta = \alpha R(\forall \beta. \beta \rightarrow \beta)$

$$(z[\alpha])(R \rightarrow R)(w[\forall \beta. \beta \rightarrow \beta])$$
Warm-up: \( \forall \alpha. \alpha \rightarrow \alpha \) is terminal

- Want to prove \( \Gamma, z : \forall \alpha. \alpha \rightarrow \alpha \vdash z = \Lambda \alpha. \lambda x. x : \forall \alpha. \alpha \rightarrow \alpha. \)

- By extensionality, it is enough to show
  \[
  \Gamma, z : \forall \alpha. \alpha \rightarrow \alpha, \alpha, x : \alpha \vdash z[\alpha] \ x = x : \alpha
  \]

- Further by the parametricity rule, it is enough to show
  \[
  \langle \Gamma \rangle, z(\forall \alpha \rho \beta. \rho \rightarrow \rho) \ w, \alpha \rho \beta, (x : \alpha) \rho(y : \beta) \vdash (z[\alpha] \ x : \alpha) \rho(y : \beta)
  \]

- \((x : \alpha)R(w : \forall \alpha. \alpha \rightarrow \alpha)\) where \(R = ([id \times (\lambda_\_ y)]^{-1} \rho)\), since \(x \rho y\).

- Since \(z(\forall \rho. \rho \rightarrow \rho) \ w\), by instantiating \(\alpha \rho \beta = \alpha R(\forall \beta. \beta \rightarrow \beta)\)
  \[
  (z[\alpha])(R \rightarrow R)(w[\forall \beta. \beta \rightarrow \beta])
  \]
  hence
  \[
  (z[\alpha] \ x)R(w[\forall \beta. \beta \rightarrow \beta] \ w)
  \]
Warm-up: \( \forall \alpha. \alpha \rightarrow \alpha \) is terminal

- Want to prove \( \Gamma, z : \forall \alpha. \alpha \rightarrow \alpha \vdash z = \Lambda \alpha. \lambda x. x : \forall \alpha. \alpha \rightarrow \alpha \).

- By extensionality, it is enough to show
  \[
  \Gamma, z : \forall \alpha. \alpha \rightarrow \alpha, \alpha, x : \alpha \vdash z[\alpha] x = x : \alpha
  \]

- Further by the parametricity rule, it is enough to show
  \[
  \langle \Gamma \rangle, z(\forall \alpha \rho \beta. \rho \rightarrow \rho) w, \alpha \rho \beta, (x : \alpha) \rho(y : \beta) \vdash (z[\alpha] x : \alpha) \rho(y : \beta)
  \]

- \((x : \alpha) R(w : \forall \alpha. \alpha \rightarrow \alpha)\) where \( R = ([id \times (\lambda _. y)]^{-1} \rho) \), since \( x \rho y \).

- Since \( z(\forall \rho. \rho \rightarrow \rho) w \), by instantiating \( \alpha \rho \beta = \alpha R(\forall \beta. \beta \rightarrow \beta) \)
  \[
  (z[\alpha])(R \rightarrow R)(w[\forall \beta. \beta \rightarrow \beta])
  \]
  hence
  \[
  (z[\alpha] x)([id \times (\lambda _. y)]^{-1} \rho)(w[\forall \beta. \beta \rightarrow \beta] w)
  \]
Warm-up: $\forall \alpha. \alpha \to \alpha$ is terminal

- Want to prove $\Gamma, z : \forall \alpha. \alpha \to \alpha \vdash z = \Lambda \alpha. \lambda x. x : \forall \alpha. \alpha \to \alpha$.

- By extensionality, it is enough to show
  $\Gamma, z : \forall \alpha. \alpha \to \alpha, \alpha, x : \alpha \vdash z[\alpha] x = x : \alpha$.

- Further by the parametricity rule, it is enough to show
  $\langle \Gamma \rangle, z(\forall \alpha \rho \beta. \rho \to \rho) w, \alpha \rho \beta, (x : \alpha) \rho(y : \beta) \vdash (z[\alpha] x : \alpha) \rho(y : \beta)$

- $(x : \alpha) R(w : \forall \alpha. \alpha \to \alpha)$ where $R = ([id \times (\lambda_. y)]^{-1} \rho)$, since $x \rho y$.

- Since $z(\forall \rho. \rho \to \rho) w$, by instantiating $\alpha \rho \beta = \alpha R(\forall \beta. \beta \to \beta)$
  $$ (z[\alpha])(R \to R)(w[\forall \beta. \beta \to \beta]) $$

  hence
  $$ (z[\alpha] x)([id \times (\lambda_. y)]^{-1} \rho)(w[\forall \beta. \beta \to \beta] w) $$

  i.e.
  $$ (z[\alpha] x : \alpha) \rho(y : \beta). $$
The expected consequences

Theorem (Consequences of Parametricity)

System $\lambda 2R$ proves:

1. $\forall \alpha. \alpha \rightarrow \alpha$ is 1.

2. $\forall \alpha. (A \rightarrow B \rightarrow \alpha) \rightarrow \alpha$ is $A \times B$.

3. $\forall \alpha. \alpha$ is 0.

4. $\forall \alpha. (A \rightarrow \alpha) \rightarrow (B \rightarrow \alpha) \rightarrow \alpha$ is $A + B$.

5. $\forall \alpha. (\forall \beta. (T(\beta) \rightarrow \alpha)) \rightarrow \alpha$ is $\exists \alpha. T(\alpha)$.

6. The type $\forall \alpha. (T(\alpha) \rightarrow \alpha) \rightarrow \alpha$ is the carrier of the initial $T$-algebra for all functorial type expressions $T(\alpha)$.

7. The type $\exists \alpha. (\alpha \rightarrow T(\alpha)) \times \alpha$ is the carrier of the final $T$-coalgebra for all functorial type expressions $T(\alpha)$.

8. Terms of type $\forall \alpha. F(\alpha, \alpha) \rightarrow G(\alpha, \alpha)$ for mixed-variance type expressions $F$ and $G$ are dinatural.
Some comments on the proof

- As usual, relations representing graphs of functions play a key role.
Some comments on the proof

- As usual, relations representing graphs of functions play a key role.

- Two ways to define concrete graphs:
  
  - $(x : A) gr_*(f) (y : B)$ if $f x = y$.
Some comments on the proof

- As usual, relations representing graphs of functions play a key role.

- Two ways to define concrete graphs:
  
  - $(x:A) gr_*(f) (y:B)$ if $f \times x = y$.
  
  - $(x:A) gr!(f)(y:B)$ if there exists $w:A$ such that $x = w$ and $y = f w$.
Some comments on the proof

- As usual, relations representing graphs of functions play a key role.

- Two ways to define concrete graphs:
  
  - $(x : A) \, gr_*(f) (y : B)$ if $(f \times B) \langle B \rangle (y : B)$.
  
  - $(x : A) \, gr_!(f) (f \times B)$ if there exists $w : A$ such that $(x : A) \langle A \rangle (w : A)$.
  
  - Since we only have pseudo-identities, these do not coincide in general.
Some comments on the proof

- As usual, relations representing graphs of functions play a key role.

- Two ways to define concrete graphs:
  - \((x : A) \text{gr}_*(f) (y : B)\) if \((f x : B) (B) (y : B)\).
  - \((x : A) \text{gr}_!(f) (f w : B)\) if there exists \(w : A\) such that \((x : A) (A) (w : A)\).
  - Since we only have pseudo-identities, these do not coincide in general.

- \(\text{gr}_*(f) := [f \times \text{id}]^{-1} (B)\) defined using fibrational structure,
  \(\text{gr}_!(f) := [\text{id} \times f]! (A)\) using derived opfibrational structure.
Some comments on the proof

- As usual, relations representing graphs of functions play a key role.

- Two ways to define concrete graphs:
  - \((x : A) \, gr_*(f) (y : B)\) if \((f \times B) \langle B \rangle (y : B)\).
  - \((x : A) \, gr_f(f) (f \, w : B)\) if there exists \(w : A\) such that \((x : A) \langle A \rangle (w : A)\).
  - Since we only have pseudo-identities, these do not coincide in general.

- \(gr_*(f) := [f \times id]^{-1} \langle B \rangle\) defined using fibrational structure,
  \(gr_f(f) := [id \times f]_! \langle A \rangle\) using derived opfibrational structure.

- Subtlety: initial algebras use inverse image pseudographs, final coalgebras direct image ones.
Summary

- $\lambda 2$ fibrations with comprehension property as natural models of $\lambda 2$ (sound and complete).

- Comprehensive $\lambda 2$ parametricity graphs form good models of relational parametricity for $\lambda 2$, with usual strong consequences.

- Reasoning in the models using a sound and complete type theory $\lambda 2R$, including inverse image relations.

- Proof of consequences of parametricity involves novel ingredients:
  - direct image relations via impredicative encoding,
  - no identity relations available, and
  - two different pseudo-graph relations (using inverse and direct images).

- **Future work**: Extend to e.g. dependent type theory.

---

Neil Ghani, Fredrik Nordvall Forsberg and Alex Simpson
Comprehensive parametric polymorphism: categorical models and type theory.
FoSSaCS 2016.
Summary

- $\lambda^2$ fibrations with comprehension property as natural models of $\lambda^2$ (sound and complete).
- Comprehensive $\lambda^2$ parametricity graphs form good models of relational parametricity for $\lambda^2$, with usual strong consequences.
- Reasoning in the models using a sound and complete type theory $\lambda^2 R$, including inverse image relations.
- Proof of consequences of parametricity involves novel ingredients:
  ▶ direct image relations via impredicative encoding,
  ▶ no identity relations available, and
  ▶ two different pseudo-graph relations (using inverse and direct images).

Future work:

- Extend to e.g. dependent type theory.

Neil Ghani, Fredrik Nordvall Forsberg and Alex Simpson

Comprehensive parametric polymorphism: categorical models and type theory.

FoSSaCS 2016.