Bindings as Bounded Natural Functors

Jasmin Blanchette, Lorenzo Gheri, Andrei Popescu, Dmitriy Traytel

Vrije Universiteit Amsterdam  Middlesex University London  ETH Zürich
Bindings as Bounded Natural Functors

Jasmin Blanchette, Lorenzo Gheri, Andrei Popescu, Dmitriy Traytel

https://www.youtube.com/watch?v=gkq20NqQ2MI
Modular framework for datatypes with bindings
  - Complex variable binders
  - Infinitary syntax too (including coinductive datatypes)

Formalized in the Isabelle/HOL proof assistant

It is being implemented as a definitional package
What is a Binder?

Several very expressive syntactic formats:

- CαMl [Pottier 2006]
- Ott [Sewell et al. 2010]
- Unbound [Weirich et al. 2011]
- Isabelle Nominal2 [Urban and Kaliszyk 2012]
- Needle&Knot [Keuchel et al. 2016]

Proposal: Binder = Operator on sets

\[ F : \text{Set}^m \times \text{Set}^n \to \text{Set} \]

plus binding dispatcher relation \( \theta \subseteq \{1,\ldots,m\} \times \{1,\ldots,n\} \).

Think:

\[ F( V_1, \ldots, V_m, T_1, \ldots, T_n ) \]

combines variables \( v_i \in V_i \) and terms \( t_j \in T_j \) such that \( v_i \in V_i \) binds in \( t_j \in T_j \) if \( (i, j) \in \theta \).

\[
\lambda v. t \quad m = 1, n = 1
\]

\[
F(V, T) = V \times T
\]

\[
\theta = \{(1, 1)\}
\]

\[
\text{let } v = t_1 \text{ in } t_2 \quad m = 1, n = 2
\]

\[
F(V, T_1, T_2) = V \times T_1 \times T_2
\]

\[
\theta = \{(1, 2)\}
\]

\[
\text{let rec } v_1 = t_1 \text{ and } \ldots \text{ and } v_k = t_k \text{ in } t \quad m = 1, n = 1
\]

\[
F(V, T) = \text{List}(V \times T) \times T
\]

\[
\theta = \{(1, 1)\}
\]


What is a Binder?

Several very expressive syntactic formats:

\( \text{C\alpha Ml [Pottier 2006] } \)
\( \text{Ott [Sewell et al. 2010] } \)
\( \text{Unbound [Weirich et al. 2011] } \)
\( \text{Isabelle Nominal2 [Urban and Kaliszyk 2012] } \)
\( \text{Needle\&Knot [Keuchel et al. 2016] } \)
What is a Binder?

Binder = Mechanism for combining any variables with any terms.

\[ \lambda v. t \]

let \( v = t_1 \) in \( t_2 \)

let rec \( v_1 = t_1 \) and \( \ldots \) and \( v_k = t_k \) in \( t \)
What is a Binder?

Binder = Mechanism for combining any variables with any terms.

Proposal: Binder = Operator on sets $F : \text{Set}^m \times \text{Set}^n \rightarrow \text{Set}$
plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Think: $F (V_1, \ldots, V_m, T_1, \ldots, T_n)$ combines variables $v_i \in V_i$ and terms $t_j \in T_j$ such that $v_i \in V_i$ binds in $t_j \in T_j$ if $(i, j) \in \theta$.

$$\lambda v. \ t$$

let $v = t_1$ in $t_2$

let rec $v_1 = t_1$ and $\ldots$ and $v_k = t_k$ in $t$
What is a Binder?

Binder = Mechanism for combining any variables with any terms.

Proposal: Binder = Operator on sets $F : \text{Set}^m \times \text{Set}^n \to \text{Set}$ plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Think: $F (V_1, \ldots, V_m, T_1, \ldots, T_n)$ combines variables $v_i \in V_i$ and terms $t_j \in T_j$ such that $v_i \in V_i$ binds in $t_j \in T_j$ if $(i, j) \in \theta$.

\[
\lambda v . t
\]

\[
\text{let } v = t_1 \text{ in } t_2
\]

\[
\text{let rec } v_1 = t_1 \text{ and } \ldots \text{ and } v_k = t_k \text{ in } t
\]
Structure of Binders

Proposal: Binder = Operator on sets $F : \text{Set}^m \times \text{Set}^n \to \text{Set}$

plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.
Structure of Binders

Proposal: Binder = Operator on sets $F : \text{Set}^m \times \text{Set}^n \to \text{Set}$

plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

$F$ “Natural” (Container-like)
Structure of Binders

Proposal: Binder = Operator on sets $F : \text{Set}^m \times \text{Set}^n \to \text{Set}$

plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Finitary?

F “Natural” (Container-like)
Structure of Binders

Proposal: Binder = Operator on sets $F : \text{Set}^m \times \text{Set}^n \to \text{Set}$
plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Bounded
F “Natural” (Container-like)

\[
p \in F(\overline{V}, \overline{T})
\]

\[
\begin{array}{c}
\{v, \ldots\} \\
\downarrow \\
\{t, \ldots\}
\end{array}
\]
Structure of Binders

Proposal: Binder = Operator on sets $F : \text{Set}^m \times \text{Set}^n \rightarrow \text{Set}$
plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Bounded

F “Natural” (Container-like)

Functor?

\[ p \in F (\overline{V}, \overline{T}) \]

\[ \{v, \ldots\} \]

\[ \{t, \ldots\} \]
Structure of Binders

Proposal: Binder = Operator on sets $F : \text{Set}^m \times \text{Set}^n \to \text{Set}$

plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Bounded

F "Natural" (Container-like)

Functor?

<table>
<thead>
<tr>
<th>let rec $(v = t)^*$ in $t$</th>
<th>$F(V, T) =$</th>
</tr>
</thead>
<tbody>
<tr>
<td>let rec $v = t_1$ and $v = t_2$ in $t$</td>
<td>List $(V \times T) \times T$</td>
</tr>
<tr>
<td>$\theta = {(1, 1)}$</td>
<td>$[(v, t_1), (v, t_2)] \in \text{List} (V \times T)$</td>
</tr>
</tbody>
</table>

$p \in F(V, T)$

\[ \begin{array}{c}
{v, \ldots}\rightarrow \{t, \ldots\} \\
\downarrow \quad \downarrow \\
{v, \ldots} \quad {t, \ldots}
\end{array} \]
Structure of Binders

Proposal: Binder = Operator on sets $F : \text{Set}^m \times \text{Set}^n \rightarrow \text{Set}$ plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Bounded

F “Natural” (Container-like)

Functor?

| Let rec $(v = t)^*$ in t | $F(V, T) =$ | let rec $v = t_1$ and $v = t_2$ in $t$
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>List $(V \times T)^@v \times T$</td>
<td>$\theta = {(1, 1)}$</td>
<td>$[(v, t_1), (v, t_2)] \in \text{List}(V \times T)$</td>
</tr>
</tbody>
</table>

$p \in F(V, T)$

\[ \begin{array}{c}
\text{let rec } v = t_1 \text{ and } v = t_2 \\
\text{in } t
\end{array} \]

\[ \begin{array}{c}
\text{List } (V \times T)^@v \times T
\end{array} \]

\[ \begin{array}{c}
\theta = \{(1, 1)\}
\end{array} \]

\[ \begin{array}{c}
[(v, t_1), (v, t_2)] \in \text{List}(V \times T)
\end{array} \]
Structure of Binders

Proposal: Binder = Operator on sets $F : \text{Set}^m \times \text{Set}^n \rightarrow \text{Set}$

plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Bounded

F “Natural” (Container-like)

Functor on (binding) variable arguments only w.r.t. injections

\[
\begin{align*}
\text{let rec } (v = t)^* \text{ in } t & \quad F(V, T) = \text{let rec } v = t_1 \text{ and } v = t_2 \text{ in } t \\
\theta = \{(1, 1)\} & \quad \text{List}(V \times T)^@ v \times T \\
\end{align*}
\]

\[
\begin{align*}
p \in F(\overline{V}, \overline{T}) & \quad [(v, t_1), (v, t_2)] \in \text{List}(V \times T) \\
\{t, \ldots\} & \quad \{v, \ldots\}
\end{align*}
\]
Structure of Binders

Proposal: Binder = Operator on sets \( F : \text{Set}^m \times \text{Set}^n \rightarrow \text{Set} \)

plus binding dispatcher relation \( \theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\} \).

Bounded

F “Natural” (Container-like)

Functor on (binding) variable arguments only w.r.t. injections

\[
w(\nu).p
\]

\[
p \in F(\overline{V}, \overline{T})
\]

\[
\begin{array}{c}
v \\
\downarrow \\
\{v, \ldots\}
\end{array} \quad \quad \rightarrow \quad \quad \begin{array}{c}
t, \ldots
\end{array}
\]
Structure of Binders

Proposal: Binder = Operator on sets $F : \text{Set}^p \times \text{Set}^m \times \text{Set}^n \rightarrow \text{Set}$

plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Bounded

F “Natural” (Container-like)

Functor on (binding) variable arguments only w.r.t. injections

\[
\begin{align*}
  w(v). p & \\
  p = m = n = 1 & \\
  F(W, V, T) = W \times V \times T & \\
  \theta = \{(1, 1)\}
\end{align*}
\]

\[
p \in F(V, T)
\]

\[
\{w, \ldots\} \rightarrow \{t, \ldots\}
\]

\[
w \downarrow \\
{\{v, \ldots\}}
\]
More About Container-Like Functors

\[ F : \text{Set} \to \text{Set} \]

Functor
\[ \text{map}_F : \prod_{A, B \in \text{Set}} (A \to B) \to F(A) \to F(B) \]

Relator
\[ \text{rel}_F : \prod_{A, B \in \text{Set}} \mathcal{P}(A \times B) \to \mathcal{P}(F(A) \times F(B)) \]

Container types [Hoogendijk and de Moor 2000]
Containers [Abbott et al. 2005]
Bounded Natural Functors (BNFs) [Traytel et al. 2012]
More About Container-Like Functors

\[ F : \text{Set} \to \text{Set} \]

Functor \( \text{map}_F : \prod_{A,B \in \text{Set}} (A \to B) \to F(A) \to F(B) \)

Relator \( \text{rel}_F : \prod_{A,B \in \text{Set}} \mathcal{P}(A \times B) \to \mathcal{P}(F(A) \times F(B)) \)

Container types [Hoogendijk and de Moor 2000]
Containers [Abbott et al. 2005]
Bounded Natural Functors (BNFs) [Traytel et al. 2012]
Proposal: Binder $= \text{Operator on sets } F : \text{Set}^p \times \text{Set}^m \times \text{Set}^n \to \text{Set}$
that is a Map-Restricted Bounded Natural Functor (MRBNF):

- w.r.t. arbitrary functions on the $p$ free-variable arguments
- w.r.t. injections on the $m$ binding-variable arguments
- w.r.t. arbitrary functions on the $n$ “term” arguments

plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$. 

Term-agnostic: Binds any hypothetical terms.
Structure of Binders (Summary)

Proposal: Binder = Operator on sets $F : \text{Set}^p \times \text{Set}^m \times \text{Set}^n \rightarrow \text{Set}$ that is a Map-Restricted Bounded Natural Functor (MRBNF):

- w.r.t. small-support endofunctions on the $p$ free-variable arguments
- w.r.t. small-support endobijections on the $m$ binding-vars. arguments
- w.r.t. arbitrary functions on the $n$ “term” arguments

plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$. 
Proposal: Binder = Operator on sets \( F : \text{Set}^p \times \text{Set}^m \times \text{Set}^n \rightarrow \text{Set} \)

that is a Map-Restricted Bounded Natural Functor (MRBNF):

w.r.t. small-support endofunctions on the \( p \) free-variable arguments

w.r.t. small-support endobijections on the \( m \) binding-vars. arguments

w.r.t. arbitrary functions on the \( n \) “term” arguments

plus binding dispatcher relation \( \theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\} \).

Term-agnostic: Binds any hypothetical terms.
Constructing Terms from Binders

\[ F : \text{Set} p \times \text{Set} m \times \text{Set} \to \text{Set} \]

Assume \( p = m \).

\[ T(V) = \mu A. F(V, V, A) \]

\[ \text{Alpha-quotiented terms: } T(V) = T(V)/\equiv_\theta \]
Constructing Terms from Binders

\[ F : \text{Set}^p \times \text{Set}^m \times \text{Set} \rightarrow \text{Set} \]

Assume \( p = m \).

\[
\overline{T(V)} = \mu \bar{A}. F(\overline{V}, \overline{V}, \bar{A})
\]
Constructing Terms from Binders

\[ F : \text{Set}^p \times \text{Set}^m \times \text{Set} \to \text{Set} \]

Assume \( p = m \).

Raw terms: \( \overline{T(V)} = \mu \overline{A}. F(\overline{V}, \overline{V}, \overline{A}) \)
Constructing Terms from Binders

\[ F : \text{Set}^p \times \text{Set}^m \times \text{Set} \to \text{Set} \]

Assume \( p = m \).

Raw terms: \( \overline{\text{T}(\overline{V})} = \mu \overline{A}. \overline{F(\overline{V}, \overline{V}, \overline{A})} \)

Alpha-quotiented terms: \( \overline{\text{T}(\overline{V})} = \overline{\text{T}(\overline{V})/\equiv_\theta} \)
Inductive Definition of Alpha-Equivalence

Equality on the top free variables
Possible bijective renamings of top binding variables
Recursive call factoring in the renamings
Abstract Characterization of Alpha-Quotinented Terms?

\[
\begin{align*}
T(\overline{V}) &= \mu A. F(\overline{V}, \overline{V}, A) \quad \text{OK} \\
T(\overline{V}) &= T(\overline{V}) / \equiv_{\theta} \quad \text{too low-level}
\end{align*}
\]
Abstract Characterization of Alpha-Quotinented Terms?

\[
T(\overline{V}) = \mu A. F(\overline{V}, \overline{V}, A) \quad \text{OK}
\]
\[
T(\overline{V}) = T(\overline{V}) / \equiv_{\theta} \quad \text{too low-level}
\]

Operators on \( T \):
- \text{ctor} : F(\overline{V}, \overline{V}, T(\overline{V})) \to T(\overline{V}) \text{ non-injective constructor}
- \text{FVars}_i : T(\overline{V}) \to V_i
- \text{map}_T \text{ functorial action on } T \text{ w.r.t. bijections}

Theorem: \((T, \text{FVars}, \text{map}_T, \text{ctor})\) is the \underline{initial object} in a category of models \( \mathcal{U} = (U, \text{UFVars}_i, \text{Umap}, \text{Uctor}) \) satisfying:
- \text{Umap functorial on bijections}
- \text{Umap and UVars}_i \text{ distribute over Uctor}
- \text{Umap satisfies congruence w.r.t. UVars}_i
Abstract Characterization of Alpha-Quotinented Terms?

\[
T(\overline{V}) = \mu A. \ F(\overline{V}, \overline{V}, A) \quad \text{OK}
\]

\[
T(\overline{V}) = T(\overline{V}) / \equiv_\theta \quad \text{too low-level}
\]

Operators on \( T \):
- \( \text{ctor} : F(\overline{V}, \overline{V}, T(\overline{V})) \to T(\overline{V}) \) non-injective constructor
- \( F\text{Vars}_i : T(\overline{V}) \to V_i \)
- \( \text{map}_T \) functorial action on \( T \) w.r.t. bijections

Theorem: \((T, F\text{Vars}, \text{map}_T, \text{ctor})\) is the initial object in a category of models \( \mathcal{U} = (U, U\text{FVars}_i, U\text{map}, U\text{ctor}) \) satisfying:
- \( U\text{map} \) functorial on bijections
- \( U\text{map} \) and \( U\text{FVars}_i \) distribute over \( U\text{ctor} \)
- \( U\text{map} \) satisfies congruence w.r.t. \( U\text{FVars}_i \)

\[\Downarrow\]

Recurser generalizing the state-of-the-art nominal recursors
Abstract Characterization of Alpha-Quotinenced Terms?

Notation: \( T(\overline{V}) = \mu_\theta A. F(\overline{V}, \overline{V}, A) \)

Operators on \( T \):
- \( \text{ctor} : F(\overline{V}, \overline{V}, T(\overline{V})) \rightarrow T(\overline{V}) \) non-injective constructor
- \( \text{FVars}_i : T(\overline{V}) \rightarrow V_i \)
- \( \text{map}_T \) functorial action on \( T \) w.r.t. bijections

Theorem: \((T, \text{FVars}, \text{map}_T, \text{ctor})\) is the initial object in a category of models \( \mathcal{U} = (U, \text{UFVars}, \text{Umap}, \text{Uctor}) \) satisfying:
- \( \text{Umap} \) functorial on bijections
- \( \text{Umap} \) and \( \text{UFVars}_i \) distribute over \( \text{Uctor} \)
- \( \text{Umap} \) satisfies congruence w.r.t. \( \text{UFVars}_i \)

\( \downarrow \)

Closure Properties for MRBNFs

Include standard type constructors: sums, products, . . .

Include nonfree type constructors: fin. sets, bags, prob. distrib., . . .

Closed under standard least/greatest fixpoints: lists, trees, . . .

Closed under linearization

Closed under binding-aware least/greatest fixpoints (modulo binding dispatchers)

“Side effects”: Binding-aware (co)recursors and (co)induction principles (obeying Barendregt’s variable convention)
Closure Properties for MRBNFs

Include standard type constructors: sums, products, …

Include nonfree type constructors: fin. sets, bags, prob. distrib., …

Closed under standard least/greatest fixpoints: lists, trees, …

Closed under linearization

Closed under binding-aware least/greatest fixpoints (modulo binding dispatchers)

“Side effects”: Binding-aware (co)recursors and (co)induction principles (obeying Barendregt’s variable convention)

Modular and flexible specification framework for binding datatypes

Plug and play

Complex binders made easy
Example: POPLmark Syntax Fragment

Type-variable $\alpha$, term-variables $x$, labels $l$

Types $\sigma ::= \alpha | \ldots$

Patterns $p ::= x : \sigma | \{ l_i = p_i \mid i \in 1 \ldots n \}$

Terms $t ::= x | \Lambda \alpha. \; t | \text{let } p = t_1 \text{ in } t_2$

Assumptions: Term-variables are pairwise distinct in any pattern. In terms, term-variables coming from patterns and type-variables near $\Lambda$'s are binding.
Example: POPLmark Syntax Fragment

Type-variable \( \alpha \), term-variables \( x \), labels \( l \)

Types \( \sigma ::= \alpha | \ldots \)  
Patterns \( p ::= x : \sigma | \{ l_i = p_i \mid i \in 1 \ldots n \} \)  
Terms \( t ::= x | \Lambda \alpha. \ t | \text{let } p = t_1 \text{ in } t_2 \)

Assumptions: Term-variables are pairwise distinct in any pattern. In terms, term-variables coming from patterns and type-variables near \( \Lambda \)'s are binding.

Type \((A) = \ldots \)  
Pattern \((A, X) = (\mu P. \ X \times \text{Type}(A) + \text{FinPFunc}(\text{Label}, P))^{@X} \)
Example: POPLmark Syntax Fragment

Type-variable $\alpha$, term-variables $x$, labels $l$

Types

$\sigma \ ::= \ \alpha \mid \ldots$

Patterns

$p \ ::= \ x : \sigma \mid \{l_i = p_i \mid i \in 1\ldots n\}$

Terms

$t \ ::= \ x \mid \Lambda \alpha.\ t \mid \text{let } _p_ = t_1 \text{ in } t_2$

Assumptions: Term-variables are pairwise distinct in any pattern. In terms, term-variables coming from patterns and type-variables near $\Lambda$'s are binding.

Type $(A) = \ldots$

Pattern $(A, X) = (\mu P.\ X \times \text{Type} (A) + \text{FinPFunc} (\text{Label}, P)) \circ X$

Term $(A, X) = \mu \theta T.\ X + A \times T + \text{Pattern} (A, X) \times T^2$
Example: POPLmark Syntax Fragment

Type-variable $\alpha$, term-variables $x$, labels $l$

Types $\sigma ::= \alpha \mid \ldots$

Patterns $p ::= x : \sigma \mid \{l_i = p_i \mid i \in 1 \ldots n\}$

Terms $t ::= x \mid \Lambda \alpha . t \mid \text{let } _p_ = t_1 \text{ in } t_2$

Assumptions: Term-variables are pairwise distinct in any pattern.
In terms, term-variables coming from patterns and type-variables near $\Lambda$'s are binding.

\[\text{Type } (A) = \ldots\]
\[\text{Pattern } (A, X) = (\mu P . X \times \text{Type } (A) + \text{FinPFunc } (\text{Label}, P))^{@X}\]
\[\text{Term } (A, X) = \mu \theta T . X + A \times T + \text{Pattern } (A, X) \times T^2\]
\[= \mu \theta T . F (A, X, A, X, T)\]

where:
\[F (A', X', A, X, T) = X' + A \times T + \text{Pattern } (A', X) \times T^2\]
\[\theta = \{(1, 1), (2, 1)\}\]
Related Work: 1999

A lot of work on categorical generalizations of the “nameless”, De Bruijn representation, pioneered by:

Fiore et al. (LICS’99)
Hofmann (LICS’99)
Bird and Paterson (J. Func. Prog. ’99)
Altenkirch and Reus (CSL’99)

By contrast, our work is within the “nameful” paradigm, generalizing Nominal Logic (Gabbay and Pitts (LICS’99))

(Higher Order Abstract Syntax (HOAS) – the third main paradigm)
Related Work: Relevant Classes of Functors

- Dependent Polynomial = Indexed Container
- MRBNF
- Accessible = Quotient of Polynomial
- BNF
- (Infinitary) Analytic = Quotient Container
- Polynomial = Container
Isabelle Nominal2 [Urban and Kaliszyk 2012]
- Good user support
- Complex binders via syntactic format

Our improvements (once the implementation is ready):
- Expressiveness
- Modularity
- Better integration with Isabelle’s standard datatypes (which are based on BNFs)
Bindings as Bounded Natural Functors

Jasmin Blanchette, Lorenzo Gheri, Andrei Popescu, Dmitriy Traytel

https://www.youtube.com/watch?v=gkq2ONqQ2MI
Reserve Slides
The Linearization Operator \(^{©}\)

\[ \text{List}(A)^{©A} = \{xs \in \text{List}(A) \mid \forall i, j. i \neq j \rightarrow xs_i \neq xs_j\} \]
The Linearization Operator

\[ \text{List} (A)^@A = \{xs \in \text{List} (A) \mid \forall i, j. i \neq j \rightarrow xs_i \neq xs_j \} \]

Let \( F : \text{Set} \rightarrow \text{Set} \) be a BNF.

\( !_A : A \rightarrow \text{Unit} = \{\ast\} \)

\( \text{shape} = \text{map}_F !_A : F(A) \rightarrow F(\text{Unit}) \)

How about: \( p \in F(A) \) linear

\( \ldots \text{if } \forall q. \text{shape } q = \text{shape } p \rightarrow |\text{set}_F q| \leq |\text{set}_F p| \)
The Linearization Operator

List \((A)^@A\) = \(\{xs \in \text{List} (A) \mid \forall i, j. \ i \neq j \rightarrow xs_i \neq xs_j\}\)

Let \(F : \text{Set} \rightarrow \text{Set}\) be a BNF.

\[!_A : A \rightarrow \text{Unit} = \{\ast\}\]

\(\text{shape} = \text{map}_F !_A : F (A) \rightarrow F (\text{Unit})\)

How about: \(p \in F (A)\) linear

... if \(\forall q. \ \text{shape} \ q = \text{shape} \ p \rightarrow |\text{set}_F \ q| \leq |\text{set}_F \ p|\)

Works for finitary functors.

Fails in general: For \(F = \text{Stream}\), \([0, 0, 1, 2, 3, \ldots] \in F (\text{IN})\) linear.
The Linearization Operator

\[
\text{List} (A)^{\circ A} = \{xs \in \text{List} (A) \mid \forall i, j. \ i \neq j \rightarrow xs_i \neq xs_j\}
\]

Let \( F : \text{Set} \rightarrow \text{Set} \) be a BNF.

\[
!_A : A \rightarrow \text{Unit} = \{\ast\}
\]

\[
\text{shape} = \text{map}_F !_A : F(A) \rightarrow F(\text{Unit})
\]

Better: \( p \in F(A) \) linear

... if \( \forall q. \ \text{shape} \ q = \text{shape} \ p \rightarrow \exists f : A \rightarrow A. \ \text{map}_F f \ p = q \)

Works in general.

Gives us back a sub-functor, \( F^{\circ} \), of \( F \)'s restriction to bijections.
Modularity

Let \( F : \text{Set}^m \times \text{Set}^m \times \text{Set} \rightarrow \text{Set} \) be an MRBNF. \[
T(\overline{V}) = \mu_\theta A. F(\overline{V}, \overline{V}, A)
\]

Is \( T \) also an MRBNF?
Let \( F : \text{Set}^m \times \text{Set}^m \times \text{Set} \rightarrow \text{Set} \) be an MRBNF. 

\[ T(\overline{V}) = \mu_\theta A. F(\overline{V}, \overline{V}, A) \]

Is \( T \) also an MRBNF?

\[ \text{set}_i^T := \text{FVars}_i \]

On bijections, \( \text{map}_T \) lifted from \( \text{map}_T \). 
But want \( \text{map}_T \) non-bijective/injective functions too!
**Modularity**

Let $F : \text{Set}^m \times \text{Set}^m \times \text{Set} \to \text{Set}$ be an MRBNF.

$$T(\overline{V}) = \mu_\theta A. F(\overline{V}, \overline{V}, A)$$

Is $T$ also an MRBNF?

$\text{set}_i^T := \text{FVars}_i$

On bijections, $\text{map}_T$ lifted from $\text{map}_T$.

But want $\text{map}_T$ non-bijective/injective functions too!

$\Downarrow$

$\text{map}_T := \text{the capture-avoiding substitution}$

Problem: Substitution only behaves well on functions of small support.

So we have $F$ functorial w.r.t. (arbitrary) functions implies $T$ functorial only w.r.t. small-support functions.
Proposal: Binder = Operator on sets $F : \text{Set}^p \times \text{Set}^m \times \text{Set}^n \to \text{Set}$
that is a Map-Restricted Bounded Natural Functor (MRBNF):

- w.r.t. arbitrary functions on the $p$ free-variable arguments
- w.r.t. injections on the $m$ binding-variable arguments
- w.r.t. arbitrary functions on the $n$ “term” arguments

plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Proposal: Binder = Operator on sets $F : \text{Set}^p \times \text{Set}^m \times \text{Set}^n \to \text{Set}$
that is a Map-Restricted Bounded Natural Functor (MRBNF):

- w.r.t. small-support functions on the $p$ free-variable arguments
- w.r.t. small-support injections on the $m$ binding-vars. arguments
- w.r.t. arbitrary functions on the $n$ “term” arguments

plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.
Structure of Binders: More Details

Proposal: Binder $= \text{Operator on sets } F : \text{Set}^m \times \text{Set}^n \to \text{Set}$ plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Can assume more about $F$?

Problem 1: let rec $v_1 = t_1$ and $\ldots$ and $v_k = t_k$ in $t_m = n = 1$ $F(\text{V, T}) = \text{List (\text{V} \times \text{T}) \times \text{T}}$ $\theta = \{(1, 1)\}$

Want to disallow repetitions, as in “let rec $v = t_1$ and $v = t_2$ in $t$” yet $[(v, t_1), (v, t_2)] \in \text{List (\text{V} \times \text{T})}$

Solution: Replace $\text{List (\text{V} \times \text{T})}$ with $\text{List (\text{V} \times \text{T}) \@ \text{V}}$ where “$\@ \text{V}$” means “linearize on $\text{V}$”, i.e., “exclude $\text{V}$-repetitions”
Structure of Binders: More Details

Proposal: Binder = Operator on sets $F : \text{Set}^m \times \text{Set}^n \rightarrow \text{Set}$

plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Can assume more about $F$? How about Finitary Natural Functor (FNF)?

Functor: for $f_i : V_i \rightarrow V'_i$, $g_j : T_j \rightarrow T'_j$

$$\text{map}_F \quad \bar{f} \quad \bar{g} : F (\bar{V}, \bar{T}) \rightarrow F (\bar{V}', \bar{T}')$$

Natural (container-like): there exist the natural transformations

$$\text{set}^i_F : F (\bar{V}, \bar{T}) \rightarrow \mathcal{P} (V_i) \quad \text{set}^{m+j}_F : F (\bar{V}, \bar{T}) \rightarrow \mathcal{P} (T_j)$$

Finitary: $\text{set}^i_F (\ldots)$, $\text{set}^{m+j}_F (\ldots)$ finite
Structure of Binders: More Details

Proposal: Binder = Operator on sets $F : \text{Set}^m \times \text{Set}^n \to \text{Set}$
plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Can assume more about $F$? How about Bounded Natural Functor (BNF)?

**Functor:** for $f_i : V_i \to V'_i$, $g_j : T_j \to T'_j$
$$\text{map}_F \ f \ g : F (V, T) \to F (V', T')$$

Natural (container-like): there exist the natural transformations
$$\text{set}^i_F : F (V, T) \to \mathcal{P} (V_i) \quad \text{set}^{m+j}_F : F (V, T) \to \mathcal{P} (T_j)$$

**Bounded:** For some cardinal $\text{bd}_F$
$$\text{set}^i_F (\ldots) \leq \text{bd}_F \quad \text{set}^{m+j}_F (\ldots) \leq \text{bd}_F$$
Structure of Binders: More Details

Proposal: Binder = Operator on sets $F : \text{Set}^m \times \text{Set}^n \to \text{Set}$

plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Can assume more about $F$? How about **Bounded** Natural Functor (BNF)?

Functor: for $f_i : V_i \to V'_i$, $g_j : T_j \to T'_j$

$$\text{map}_F \ f \ g : F (V, T) \to F (V', T')$$

Natural (container-like): there exist the natural transformations

$$\text{set}_F^i : F (V_i) \to \mathcal{P} (V_i) \quad \text{set}_F^{m+j} : F (T_j) \to \mathcal{P} (T_j)$$

**Bounded**: For some cardinal $\text{bd}_F$

$$\text{set}_F^i (\ldots) \leq \text{bd}_F \quad \text{set}_F^{m+j} (\ldots) \leq \text{bd}_F$$

Problem 1:

let rec $v_1 = t_1$ and \ldots and $v_k = t_k$ in $t$

$m = n = 1$

$$F (V, T) = \text{List} (V \times T) \times T$$

$$\theta = \{(1, 1)\}$$

Want to disallow repetitions, as in “let rec $v = t_1$ and $v = t_2$ in $t$” yet $[(v, t_1), (v, t_2)] \in \text{List} (V \times T)$
Structure of Binders: More Details

Proposal: Binder = Operator on sets $F : \text{Set}^m \times \text{Set}^n \rightarrow \text{Set}$

plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Can assume more about $F$? How about **Bounded** Natural Functor (BNF)?

Functor: for $f_i : V_i \rightarrow V'_i$, $g_j : T_j \rightarrow T'_j$

$$\text{map}_F \ f \ g : F (V, T) \rightarrow F (V', T')$$

Natural (container-like): there exist the natural transformations

$$\text{set}_F^i : F (V, T) \rightarrow \mathcal{P} (V_i) \quad \text{set}_F^{m+j} : F (V, T) \rightarrow \mathcal{P} (T_j)$$

**Bounded:** For some cardinal $bd_F$

$$\text{set}_F^i (\ldots) \leq bd_F \quad \text{set}_F^{m+j} (\ldots) \leq bd_F$$

Problem 1:

let rec $v_1 = t_1$ and \ldots and $v_k = t_k$ in $t$  \hspace{1cm} m = n = 1

$$F (V, T) = \text{List} (V \times T) \times T$$

$$\theta = \{(1, 1)\}$$

Want to disallow repetitions, as in “let rec $v = t_1$ and $v = t_2$ in $t$” yet $[(v, t_1), (v, t_2)] \in \text{List} (V \times T)$

Solution: Replace $\text{List} (V \times T)$ with $\text{List} (V \times T)^\circ V$

where “$^\circ V$” means “linearize on $V$”, i.e., “exclude $V$-repetitions”
Structure of Binders: More Details

Proposal: Binder = Operator on sets $F : \text{Set}^m \times \text{Set}^n \to \text{Set}$

plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Can assume more about $F$? How about Bounded Natural Functor (BNF)?

Functor: for $f_i : V_i \to V'_i$, $g_j : T_j \to T'_j$

$$\text{map}_F \, \bar{f} \, \bar{g} : F (V, T) \to F (V', T')$$

Natural (container-like): there exist the natural transformations

$$\text{set}_F^i : F (V, T) \to \mathcal{P} (V_i) \quad \text{set}_F^{m+j} : F (V, T) \to \mathcal{P} (T_j)$$

Bounded: For some cardinal $bd_F$

$$\text{set}_F^i (\ldots) \leq bd_F \quad \text{set}_F^{m+j} (\ldots) \leq bd_F$$

Problem 1:

Let rec $v_1 = t_1$ and $v_k = t_k$ in $t$ $\quad m = n = 1$

$F (V, T) = \text{List} (V \times T)^@ \times T$

$\theta = \{(1, 1)\}$

Want to disallow repetitions, as in “let rec $v = t_1$ and $v = t_2$ in $t$” yet $[(v, t_1), (v, t_2)] \in \text{List} (V \times T)$

Solution: Replace $\text{List} (V \times T)$ with $\text{List} (V \times T)^@$ where “$^@$” means “linearize on $V$”, i.e., “exclude $V$-repetitions”
Structure of Binders: More Details

Proposal: Binder = Operator on sets $F : \text{Set}^m \times \text{Set}^n \to \text{Set}$
plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Can assume more about $F$? How about **Bounded** Natural Functor (BNF)?

Functor: for $f_i : V_i \to V_i$ bijections, $g_j : T_j \to T'_j$

$$\text{map}_F \ f \ g : F(\overline{V}, \overline{T}) \to F(\overline{V}', \overline{T}')$$

Natural (container-like): there exist the natural transformations

$$\text{set}_F^i : F(\overline{V}, \overline{T}) \to \mathcal{P}(V_i) \quad \text{set}_F^{m+j} : F(\overline{V}, \overline{T}) \to \mathcal{P}(T_j)$$

**Bounded**: For some cardinal $\text{bd}_F$

$$\text{set}_F^i (\ldots) \leq \text{bd}_F \quad \text{set}_F^{m+j} (\ldots) \leq \text{bd}_F$$

Problem 1:

let rec $v_1 = t_1$ and \ldots and $v_k = t_k$ in $t$

$$F(\overline{V}, \overline{T}) = \text{List } (V \times T)^{\overline{V}} \times T$$

$$\theta = \{(1, 1)\}$$

Want to disallow repetitions, as in “let rec $v = t_1$ and $v = t_2$ in $t$” yet

$$[(v, t_1), (v, t_2)] \in \text{List } (V \times T)$$

Solution: Replace $\text{List } (V \times T)$ with $\text{List } (V \times T)^{\overline{V}}$
where “$^{\overline{V}}$” means “linearize on $V$”, i.e., “exclude $V$-repetitions”
Structure of Binders: More Details

Proposal: Binder = Operator on sets $F : \text{Set}^p \times \text{Set}^m \times \text{Set}^n \to \text{Set}$

plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Can assume more about $F$? How about **Bounded** Natural Functor (BNF)?

Functor: for $f_i : V_i \to V_i$ bijections, $g_j : T_j \to T'_j$

$$\text{map}_F \bar{f} \bar{g} : F(\overline{V}, \overline{T}) \to F(\overline{V}', \overline{T}')$$

Natural (container-like): there exist the natural transformations

$$\text{set}^i_F : F(\overline{V}, \overline{T}) \to \mathcal{P}(V_i) \quad \text{set}^{m+j}_F : F(\overline{V}, \overline{T}) \to \mathcal{P}(T_j)$$

**Bounded**: For some cardinal $\text{bd}_F$

$$\text{set}^i_F(\ldots) \leq \text{bd}_F \quad \text{set}^{m+j}_F(\ldots) \leq \text{bd}_F$$

Problem 2:

$w(v). p$

Free variables are currently completely ignored.
Structure of Binders: More Details

Proposal: Binder = Operator on sets $F : \text{Set}^p \times \text{Set}^m \times \text{Set}^n \rightarrow \text{Set}$

plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Can assume more about $F$? How about **Bounded** Natural Functor (BNF)?

Functor: for $h_k : W_k \rightarrow W_k, f_i : V_i \rightarrow V_i$ bijections, $g_j : T_j \rightarrow T_j'$

$$\text{map}_F (h \ f \ g) : F (W, V, T) \rightarrow F (W, V, T')$$

Natural (container-like): there exist the natural transformations

$\text{set}^k_F : F (W, V, T) \rightarrow \mathcal{P} (W_k)$

$\text{set}^{p+i}_F : F (W, V, T) \rightarrow \mathcal{P} (V_i)$

$\text{set}^{p+m+j}_F : F (W, V, T) \rightarrow \mathcal{P} (T_j)$

**Bounded**: For some cardinal $\text{bd}_F$

$$\text{set}^k_F (\ldots) \leq \text{bd}_F \quad \text{set}^{p+i}_F (\ldots) \leq \text{bd}_F \quad \text{set}^{p+m+j}_F (\ldots) \leq \text{bd}_F$$

Problem 2:

$$p = m = n = 1$$

$w (v) \cdot p$

$$F (W, V, T) = W \times V \times T$$

$$\theta = \{(1, 1)\}$$

Free variables are currently completely ignored.

Solution: Have **free-variable args.** in addition to **binding-variable args.**
Structure of Binders: More Details

Proposal: Binder = Operator on sets $F : \text{Set}^p \times \text{Set}^m \times \text{Set}^n \rightarrow \text{Set}$
plus binding dispatcher relation $\theta \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$.

Can assume more about $F$? How about **Bounded** Natural Functor (BNF)?

Functor: for $h_k : W_k \rightarrow W_k, f_i : V_i \rightarrow V_i$ bijections, $g_j : T_j \rightarrow T_j'$

$$\text{map}_F \bar{h} \bar{f} \bar{g} : F (\overline{W}, \overline{V}, \overline{T}) \rightarrow F (\overline{W}, \overline{V}, \overline{T'})$$

Natural (container-like): there exist the natural transformations

$$\text{set}^k_F : F (\overline{W}, \overline{V}, \overline{T}) \rightarrow \mathcal{P}(W_k) \quad \text{set}^{p+i}_F : F (\overline{W}, \overline{V}, \overline{T}) \rightarrow \mathcal{P}(V_i) \quad \text{set}^{p+j}_F : F (\overline{W}, \overline{V}, \overline{T}) \rightarrow \mathcal{P}(T_j)$$

**Bounded**: For some cardinal $\text{bd}_F$

$$\text{set}^k_F (...) \leq \text{bd}_F \quad \text{set}^{p+i}_F (...) \leq \text{bd}_F \quad \text{set}^{p+j}_F (...) \leq \text{bd}_F$$

Problem 2: $p = m = n = 1$

$w (v). p$ $F (W, V, T) = W \times V \times T$

$\theta = \{(1, 1)\}$

Free variables are currently completely ignored.

Solution: Have **free-variable args.** in addition to **binding-variable args.**