Bindings as Bounded Natural Functors

JASMIN CHRISTIAN BLANCHETTE, Vrije Universiteit Amsterdam, The Netherlands and Max-Planck-Institut für Informatik, Germany
LORENZO GHERI, Middlesex University London, UK
ANDREI POPESCU, Middlesex University London, UK and Institute of Mathematics Simion Stoilow of the Romanian Academy, Romania
DMITRIY TRAYTEL, ETH Zürich, Switzerland

We present a general framework for specifying and reasoning about syntax with bindings. Abstract binder types are modeled using a universe of functors on sets, subject to a number of operations that can be used to construct complex binding patterns and binding-aware datatypes, including non-well-founded and infinitely branching types, in a modular fashion. Despite not committing to any syntactic format, the framework is "concrete" enough to provide definitions of the fundamental operators on terms (free variables, alpha-equivalence, and capture-avoiding substitution) and reasoning and definition principles. This work is compatible with classical higher-order logic and has been formalized in the proof assistant Isabelle/HOL.

CCS Concepts: • **Theory of computation → Logic and verification; Higher order logic; Type structures; Interactive proof systems**;

Additional Key Words and Phrases: syntax with bindings, inductive and coinductive datatypes, proof assistants

ACM Reference Format:
Jasmin Christian Blanchette, Lorenzo Gheri, Andrei Popescu, and Dmitriy Traytel. 2019. Bindings as Bounded Natural Functors . *Proc. ACM Program. Lang.* 0, 0, Article 0 (July 2019), 3 pages.

Authors' addresses: Jasmin Christian Blanchette, Department of Computer Science, Vrije Universiteit Amsterdam, De Boelelaan 1081a, Amsterdam, 1081 HV, The Netherlands, j.c.blanchette@vu.nl, Research Group 1, Max-Planck-Institut für Informatik, Saarland Informatics Campus E1 4, Saarbrücken, 66123, Germany; Lorenzo Gheri, School of Science and Technology, Middlesex University London, The Burroughs, London, NW4 4BT, UK, lg571@live.mdx.ac.uk; Andrei Popescu, Middlesex University London, School of Science and Technology, The Burroughs, London, NW4 4BT, UK, A.Popescu@mdx.ac.uk, Institute of Mathematics Simion Stoilow of the Romanian Academy, Calea Grivitei 21, Bucharest, 010702, Romania; Dmitriy Traytel, Institute of Information Security, Department of Computer Science, ETH Zürich, Universitätstrasse 6, Zürich, 8092, Switzerland, traytel@inf.ethz.ch.

2019. 2475-1421/2019/7-ART0 $15.00
https://doi.org/

APPENDIX

A USEFUL VARIATIONS OF THE (CO)RECURSION PRINCIPLES

A.1 A Fixed-Parameter Restriction

Recall that our recursors employ a notion of dynamically varying parameter, whose free variables must be avoided. Let us introduce some notation for a useful particular case: that of static (fixed) parameters, more precisely, that of fixed sets of variables that must be avoided. Technically, we assume that the parameter type is a singleton, which is the same as replacing the parameter structure with a tuple $\mathcal{A}$ consiting of fixed small sets of variables $A_i \subseteq \alpha_i$ (each A_i representing the set of variables of the unique parameter).[1] Also, since $\overline{\alpha}\, P$ is a singleton, we can replace $\overline{\alpha}\, P \to \overline{\alpha}\, U$ with $\overline{\alpha}\, U$.

Definition 1. Given a tuple $\mathcal{A}$ of small sets, an $\mathcal{A}$-model is a quadruple $\mathcal{U} = (\overline{\alpha}\, U, \overline{\mathsf{UFVars}}, \mathsf{Umap}, \mathsf{Uctor})$, where:

- $(\overline{\alpha}\, U, \overline{\mathsf{UFVars}}, \mathsf{Umap})$ is a term-like structure
- $\mathsf{Umap} : (\alpha_1 \to \alpha_1) \to \cdots \to (\alpha_m \to \alpha_m) \to \overline{\alpha}\, U \to \overline{\alpha}\, U$

such that the following hold:

(MC) $(\forall i \in [m].\ \mathsf{supp}\ f_i \cap A_i = \varnothing)$
$\longrightarrow \mathsf{Umap}\ \overline{f}\ (\mathsf{Uctor}\ y) \ = \ \mathsf{Uctor}\ (\mathsf{map}_F\ \overline{f}\ \overline{f}\ [\mathsf{Umap}\ \overline{f}]^n\ y)$

(VC) $(\forall i \in [m].\ \mathsf{topBind}_i\ y \cap A_i = \varnothing) \wedge$
$(\forall i \in [m].\ \forall j \in [n].\ \forall u.\ u \in \mathsf{rec}_j\ y.\ \mathsf{UFVars}_i\, u \setminus \mathsf{topBind}_{i,j}\, y \subseteq A_i)$
$\longrightarrow \ \forall i \in [m].\ \mathsf{UFVars}_i\, (\mathsf{Uctor}\ y) \ \subseteq \ A_i$

Then Theorem 20 instantiates to:

Theorem 2. Given a tuple of small sets $\mathcal{A}$ and an $\mathcal{A}$-model $\mathcal{U}$, there exists a unique function $H : \overline{\alpha}\, \boldsymbol{T} \to \overline{\alpha}\, U$ such that:

(C) $(\forall i \in [m].\ \mathsf{nonClash}\ x \ \wedge\ \mathsf{topBind}_i\ x \ \cap\ A_i = \varnothing)$
$\longrightarrow \ H\ (\mathsf{ctor}\ x) \ = \ \mathsf{Uctor}\ (\mathsf{map}_F\ [\mathsf{id}]^{2*m}\ [H]^n\ x)$

(M) $(\forall i \in [m].\ \mathsf{supp}_i\ f_i \cap A_i = \varnothing) \longrightarrow H\, (\mathsf{map}_{\boldsymbol{T}}\ \overline{f}\ t) \ = \ \mathsf{Umap}\ \overline{f}\ (H\ t)$

(V) $\forall i \in [m].\, \mathsf{UFVars}_i\ (H\ t) \ \subseteq\ \mathsf{FVars}_i\ t \ \cup\ A_i$

Since the majority of binding-aware recursive definitions seem to require fixed rather than dynamic parameters, in our Isabelle formalization of the recursor we wire in $\mathcal{A}$ as a primitive (in addition to $\mathcal{P}$)—this avoids the bureaucracy of having to instantiate $\mathcal{P}$ to a singleton for handling fixed parameters.

A.2 The Full-Fledged Primitive (Co)recursor

In Section 7 we have presented a restricted form of (co)recursors that are usually known as (co)iterators. Here we formulate the full-fledged (co)recursors, which constitute a theoretically straightforward but practically useful extension of the (co)iterators.

The difference between a recursor and an iterator is that the former allows the value of a function H applied to a given term ctor x to depend not only on the values of H on the recursive components t of x, but also on the components themselves. To cater for this, we routinely enhance our notions of term-like structure and model with additional term arguments, as highlighted below:

Definition 3. An *extended term-like structure* is a triple $\mathcal{D} = (\overline{\alpha}\, D, \overline{\mathsf{DFVars}}, \mathsf{Dmap})$, where

[1] The smallness of a $A_i \subseteq \alpha_i$ means, as usual, that $|A_i| < |\alpha_i|$.

- $\overline{\alpha}\, D$ is a polymorphic type
- $\overline{\mathsf{DFVars}}$ is a tuple of functions $\mathsf{DFVars}_i : \overline{\alpha}\, P \to \overline{\alpha}\, \boldsymbol{T} \to \alpha_i\ set$ for $i \in [m]$
- $\mathsf{Dmap} : (\alpha_1 \to \alpha_1) \to \cdots \to (\alpha_m \to \alpha_m) \to \overline{\alpha}\, D \to \overline{\alpha}\, \boldsymbol{T} \to \overline{\alpha}\, D$

are such that the following hold:

- $\mathsf{Dmap}\ [\mathsf{id}]^m\ t = \mathsf{id}$
- $\mathsf{Dmap}\ (g_1 \circ f_1)\ \cdots\ (g_m \circ f_m)\ t = \mathsf{Dmap}\ \overline{g}\ t \circ \mathsf{Dmap}\ \overline{f}\ t$
- $(\forall i \in [m].\ \forall a \in \mathsf{DFVars}_i\ t\ d.\ f_i\ a = a) \longrightarrow \mathsf{Dmap}\ \overline{f}\ t\ d = d$
- $a \in \mathsf{DFVars}_i\ (\mathsf{map}_{\boldsymbol{T}}\ \overline{f}\ t)\, (\mathsf{Dmap}\ \overline{f}\ t\ d) \longleftrightarrow f_i^{-1}\ a \in \mathsf{DFVars}_i\ t\ d$

Definition 4. Given a parameter structure $\mathcal{P}$, an extended $\mathcal{P}$-model is a quadruple $\mathcal{U} = (\overline{\alpha}\, U, \overline{\mathsf{UFVars}}, \mathsf{Umap}, \mathsf{Uctor})$, where:

- $(\overline{\alpha}\, U, \overline{\mathsf{UFVars}}, \mathsf{Umap})$ is an extended term-like structure
- $\mathsf{Uctor} : (\overline{\alpha}, \overline{\alpha}, [\ \overline{\alpha}\, \boldsymbol{T} \times (\overline{\alpha}\, P \to \overline{\alpha}\, U)]^n)\, F \to \overline{\alpha}\, P \to \overline{\alpha}\, U$

such that the following hold:

(MC) $\mathsf{Umap}\ \overline{f}\ (\mathsf{ctor}\ x_y)\ (\mathsf{Uctor}\ y\ p) = \mathsf{Uctor}\ (\mathsf{map}_F\ \overline{f}\ \overline{f}\ [\ \langle \mathsf{map}_{\boldsymbol{T}}, \mathsf{Umap}\ \rangle\ \overline{f}]^n\ y)\ (\mathsf{Pmap}\ \overline{f}\ p)$

(VC) $(\forall i \in [m].\ \mathsf{topBind}_i\ y \cap \mathsf{PFVars}_i\ p = \varnothing) \land$
$(\forall i \in [m].\ \forall j \in [n].\ \forall\ t, pu, p.\ (t, pu\ ) \in \mathsf{rec}_j\ y.\ \mathsf{UFVars}_i\ (pu\ p) \setminus \mathsf{topBind}_{i,j}\ y \subseteq$
$\mathsf{FVars}_i\ t \setminus \mathsf{topBind}_{i,j}\ x_y \cup \mathsf{PFVars}_i\ p)$
$\longrightarrow\ \forall i \in [m].\ \mathsf{UFVars}_i\ (\mathsf{Uctor}\ y\ p) \subseteq\ \mathsf{FVars}_i\, (\mathsf{ctor}\ x_y) \cup \mathsf{topFree}\ y\ \cup\ \mathsf{PFVars}_i\ p$

Above, x_y and $x_{y'}$ are shorthands for $\mathsf{map}_F\ [\mathsf{id}]^{2*m}\ [\mathsf{fst}]^n\ y$ and $\mathsf{map}_F\ [\mathsf{id}]^{2*m}\ [\mathsf{fst}]^n\ y'$, respectively. Also recall that fst and snd are the standard first and second projection functions on the product type $\times$. Moreover, $\langle \mathsf{map}_{\boldsymbol{T}}, \mathsf{Umap}\rangle \overline{f}$ denotes the function $\lambda(t, pu).\ (\mathsf{map}_{\boldsymbol{T}}\ \overline{f}\ t, \mathsf{Umap}\ \overline{f}\ t\ pu)$.

Note that, for (VC), the additional structure brought by the extended models makes the presence of $\mathsf{topFree}\ y$ redundant. Indeed, it is easy to check that $\mathsf{topFree}\ y = \mathsf{topFree}\ x_y$, meaning that $\mathsf{topFree}\ y \subseteq \mathsf{FVars}_i\ (\mathsf{ctor}\ y)$. In short, $\mathsf{topFree}\ y$ can be removed from the conclusion of (VC), without affecting this property.

The recursion theorem follows suit with this term-argument extension.

Full-fledged recursion extension of Theorem 20: Given a parameter structure $\mathcal{P}$ and a $\mathcal{P}$-model $\mathcal{U}$, there exists a unique function $H : \overline{\alpha}\, \boldsymbol{T} \to \overline{\alpha}\, P \to \overline{\alpha}\, U$ such that:

(C) $(\forall i \in [m].\ \mathsf{nonClash}\ x\ \land\ \mathsf{topBind}_i\ x \cap \mathsf{PFVars}_i\ p = \varnothing)\ \longrightarrow$
$H\ (\mathsf{ctor}\ x)\ p\ =\ \mathsf{Uctor}\ (\mathsf{map}_F\ [\mathsf{id}]^{2*m}\ [\ \langle \mathsf{id}, H\ \rangle\ ]^n\ x)\ p$

(M) $H\, (\mathsf{map}_{\boldsymbol{T}}\ \overline{f}\ t)\ p\ =\ \mathsf{Umap}\ \overline{f}\ t\ (f\ t\ (\mathsf{Pmap}\ \overline{f}^{-1}\ p))$

(V) $\forall i \in [m].\ \mathsf{UFVars}_i\ t\ (H\ t\ p) \subseteq\ \mathsf{FVars}_i\ t\ \cup\ \mathsf{PFVars}_i\ p$

A similar game can be played with the corecursor, where the additional term inputs occur in the result of the function, with the following intuition: In addition to the option of delving into a corecursive call, we now also have the option to stop the corecursion immediately returning an indicated term. For example, the constructor-like operator Udtor of an extended comodel will have the type $\overline{\alpha}\, U \to ((\overline{\alpha}, \overline{\alpha}, [\ \overline{\alpha}\, \boldsymbol{T} + \overline{\alpha}\, U]^n)\, F)\ set$.

It is easy to infer the extended version of the (co)recursion theorems from their original version. However, in our Isabelle formalization we directly prove the extended versions.