

From De Bruijn to co-De Bruijn using Category Theory

Everybody's Got To Be Somewhere^[2]

Marius Weidner

Chair of Programming Languages, University of Freiburg

February 7, 2024

Outline

1 Getting Started: Scopes and Binders Categorically

- The Category of Scopes
- Intrinsically Scoped De Bruijn Syntax

2 Going Further: From De Bruijn to co-De Bruijn

- The Slice Category of Subscopes
- Sets Indexed by Scopes
- Intrinsically Scoped co-De Bruijn Syntax

3 Wrapping Up: What I've (Not) Told You

- This Is Actually an Agda Paper
- References

The Category of Scopes: Δ_+^X

Definition

Let Δ_+^X be the category of scopes with

- Objects: $\bar{x}, \bar{y}, \bar{s} \in |\Delta_+^X| = X^*$ and
- Morphisms: $f, g \in \Delta_+^X(\bar{x}, \bar{y})$ for $\bar{x}, \bar{y} \in X^*$ are inductively defined by inference rules

$$\frac{}{\varepsilon \sqsubseteq \varepsilon} \cdot \qquad \frac{\bar{x} \sqsubseteq \bar{y}}{\bar{x}x \sqsubseteq \bar{y}x} 1 \qquad \frac{\bar{x} \sqsubseteq \bar{y}}{\bar{x} \sqsubseteq \bar{y}y} 0$$

Remark

Morphisms in Δ_+^X can be represented by *bit vectors* $\bar{b} \in \{0, 1\}^*$ with one bit per variable of the target scope telling whether it has been mapped to or skipped by the source scope.

Objects & Morphisms in Δ_{+}^{\top}

Example

Let $X = \top$, where \top is the set with exactly one element $\langle \rangle$.

Then, objects in Δ_{+}^{\top} represent numbers.

$$3 \xrightarrow{10101} 5$$

• ————— •

◦

• ————— •

◦

• ————— •

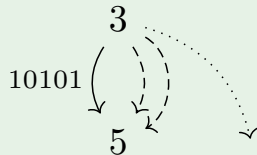
1

0

1

0

1



Identity and Composition in Δ_+^\top

Example

$$5 \xrightarrow{id} 5$$

• ————— •

• ————— •

• ————— •

• ————— •

• ————— •

Example

$$2 \xrightarrow{101} 3$$

• ————— •

◦ ;

• ————— •

$$3 \xrightarrow{10101} 5$$

• ————— •

◦

• ————— •

◦

• ————— •

=

$$2 \xrightarrow{10001} 5$$

• ————— •

◦

• ————— •

◦

• ————— •

Δ_+^X is in Fact a Category

Lemma

In Δ_+^X every object $\bar{x} \in X^*$ has an identity morphism, i.e. we can construct an identity morphism for every object \bar{x} using the inference rules.

Proof.

$$id \quad : \quad \bar{x} \sqsubseteq \bar{x}$$

$$id \varepsilon = \cdot$$

$$id \bar{x}x = id1 \quad \square$$

Corollary

$$id - l \quad : \quad id; f = f$$

$$id - r \quad : \quad f; id = f$$

Definition

In Δ_+^X two morphisms $f : \bar{x} \sqsubseteq \bar{y}$ and $g : \bar{y} \sqsubseteq \bar{z}$ compose to a morphism $f; g : \bar{x} \sqsubseteq \bar{z}$, i.e. we can construct a morphism $f; g$ from f and g using the inference rules:

$$\frac{}{-; -} \quad : \quad \bar{x} \sqsubseteq \bar{y} \rightarrow \bar{y} \sqsubseteq \bar{z} \rightarrow \bar{x} \sqsubseteq \bar{z}$$

$$\cdot \quad ; \quad \cdot \quad = \quad \cdot$$

$$f1 \quad ; \quad g1 \quad = \quad (f; g)1$$

$$f0 \quad ; \quad g1 \quad = \quad (f; g)0$$

$$f \quad ; \quad g0 \quad = \quad (f; g)0$$

Corollary

$$assoc \quad : \quad f; (g; h) = (f; g); h$$

$$antisym \quad : \quad (f : \bar{x} \sqsubseteq \bar{y}) \rightarrow (g : \bar{y} \sqsubseteq \bar{x}) \rightarrow \bar{x} = \bar{y} \wedge f = g = id \bar{x}$$

Intrinsically Scoped De Bruijn Syntax via Δ_+^\top

Definition

Let $Tm : \mathbb{N} \rightarrow Set$ be the set of lambda calculus terms inductively defined by

$$\frac{1 \sqsubseteq n}{Tm\ n} \#$$

$$\frac{Tm\ n \quad Tm\ n}{Tm\ n} \$$$

$$\frac{Tm\ (n + 1)}{Tm\ n} \lambda$$

Example

$$\mathbb{K} = \lambda x. \lambda y. x \quad = \lambda \lambda \#1$$

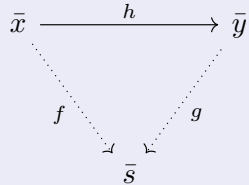
$$\mathbb{S} = \lambda x. \lambda y. \lambda z. x\ z\ (y\ z) = \lambda \lambda \lambda \#2 \#0 (\#1 \#0)$$

The Slice Category of Subscopes: $\Delta_+^X \setminus \bar{s}$

Definition

Let $\Delta_+^X \setminus \bar{s}$ be the category of subsopes for a given $\bar{s} \in X^*$ with

- Objects: $\bar{b}, (\bar{x}, f) \in |\Delta_+^X \setminus \bar{s}| = (\bar{x} : X^* \times \Delta_+^X(\bar{x}, \bar{s}))$ and
- Morphisms: $h \in [\Delta_+^X \setminus \bar{s}]((\bar{x}, f), (\bar{y}, g))$ such that $f = h; g$

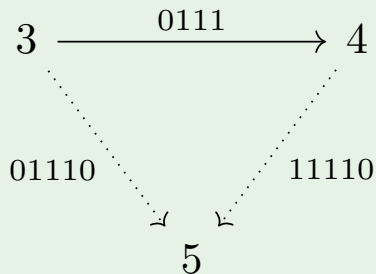


Remark

Objects in $\Delta_+^X \setminus \bar{s}$ can be represented by *bit vectors* $\bar{b} \in \{0, 1\}^*$ with one bit per variable of scope \bar{s} , telling whether it has been selected.

Objects & Morphisms in $\Delta_+^T \setminus 5$

Example



Alternatively:

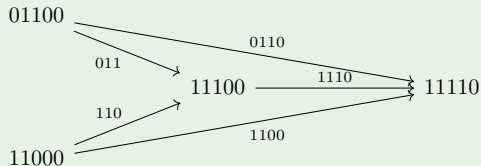
- $(3, 01110) \xrightarrow{0111} (4, 11110)$
- $01110 \xrightarrow{0111} 11110$

The Curious Case of Coproducts in $\Delta_+^X \setminus \bar{s}$

Theorem

Objects in the slice category $\bar{b}_1, \bar{b}_2 \in |\Delta_+^X \setminus \bar{s}|$ have a coproduct object $\bar{b}_1 + \bar{b}_2$, i.e. there exist morphisms $l \in [\Delta_+^X \setminus \bar{s}](\bar{b}_1, \bar{b}_1 + \bar{b}_2)$ and $r \in [\Delta_+^X \setminus \bar{s}](\bar{b}_2, \bar{b}_1 + \bar{b}_2)$ such that every pair $f \in [\Delta_+^X \setminus \bar{s}](\bar{b}_1, \bar{b}_3)$ and $g \in [\Delta_+^X \setminus \bar{s}](\bar{b}_2, \bar{b}_3)$ factor through a unique $h \in [\Delta_+^X \setminus \bar{s}](\bar{b}_1 + \bar{b}_2, \bar{b}_3)$ such that $f = l; h$ and $g = r; h$.

Example



Remark

The coproduct $\bar{b}_1 + \bar{b}_2$ of two subsopes \bar{b}_1, \bar{b}_2 corresponds to the minimal subscope covering both \bar{b}_1 and \bar{b}_2 . The coproduct $\bar{b}_1 + \bar{b}_2$ can be computed by pointwise disjunction of \bar{b}_1 and \bar{b}_2 .

The Category \overline{Set} of Sets Indexed by Scope

Definition

Let \overline{Set} be the category of sets indexed by scopes with

- Objects: $T, S \in |\overline{Set}| = \bar{X} = X^* \rightarrow Set$ and
- Morphisms: $f \in \overline{Set}(T, S) = (\bar{x} \in X^*) \rightarrow T(\bar{x}) \rightarrow S(\bar{x}) = T \rightarrow S$

Definition

Let $_ \uparrow _ : \bar{X} \rightarrow \bar{X} = (T, \bar{x}) \mapsto (T(\bar{s}) \times \bar{s} \sqsubseteq \bar{x})$ be the set of a terms together with a selection of its variables. We write $t \uparrow h$ for elements of $T \uparrow \bar{x}$.

The Notion of Relevant Pairs

Remark

The set $T \uparrow \bar{x}$ packs an set $T \in \bar{X}$ indexed by $\bar{x} \in X^*$ applied to a subscope \bar{s} of \bar{x} , together with a selection $\bar{b} \in |\Delta_+^X \setminus \bar{x}|$ of the variables of T .

Definition

Let $Cov : \bar{x} \sqsubseteq \bar{s} \rightarrow \bar{y} \sqsubseteq \bar{s} \rightarrow Set$ be the set of *coverings* indexed by morphisms \bar{b}_1 and \bar{b}_2

$$\frac{}{Cov \cdot \cdot} \cdot \quad \frac{Cov \bar{b}_1 \bar{b}_2}{Cov \bar{b}_1 1 \bar{b}_2} L \quad \frac{Cov \bar{b}_1 \bar{b}_2}{Cov \bar{b}_1 \bar{b}_2 1} R \quad \frac{Cov \bar{b}_1 \bar{b}_2}{Cov \bar{b}_1 1 \bar{b}_2 1} B$$

Let the set of relevant pairs be defined as

- $_ \times_R _ : \bar{X} \rightarrow \bar{X} \rightarrow \bar{X} = (T, S, \bar{x}) \mapsto ((_ \uparrow \bar{b}_1 : T \uparrow \bar{x}) \times (_ \uparrow \bar{b}_2 : S \uparrow \bar{x}) \times Cov \bar{b}_1 \bar{b}_2)$
- $_,R_ : T \uparrow \bar{x} \rightarrow S \uparrow \bar{x} \rightarrow (T \times_R S) \uparrow \bar{x}$
 $= ((t_1 \uparrow \bar{b}_1), (t_2 \uparrow \bar{b}_2)) \mapsto ((t_1 \uparrow \bar{b}'_1), (t_2 \uparrow \bar{b}'_2), \bar{b}_1 \oplus \bar{b}_2) \uparrow \bar{b}'$

Exploring Relevant Pairs

Remark

Coverings $Cov \bar{b}_1 \bar{b}_2$ hold data about the coproduct of \bar{b}_1 and \bar{b}_2 as well as information about the original appearance of \bar{b}_1 and \bar{b}_2 .

Example

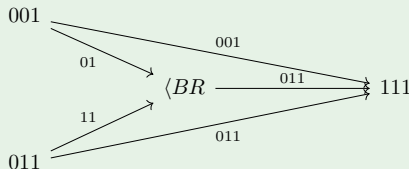
Look at $\lambda x. \lambda y. \lambda z. z \ y = \lambda \lambda \lambda (\#0 \ \#1)$.

The variable terms could also be represented as

- $z' : Tm \uparrow 3 = \#0 \uparrow 001$
- $y' : Tm \uparrow 3 = \#1 \uparrow 011$

And the application term could be a *relevant pair*

- $z',_R y' : (Tm \times_R Tm) \uparrow 3 = (\#0 \uparrow 01, \#1 \uparrow 11, BR) \uparrow 011$



Intrinsically Scoped co-De Bruijn Syntax

Definition

Let $Tm : \mathbb{N} \rightarrow Set$ be inductively defined:

$$\frac{}{Tm\ 1\ \#} \quad \frac{(Tm \times_R Tm)\ n}{Tm\ n}\ \$ \quad \frac{Tm\ (n + 1)}{Tm\ n}\ \lambda$$

Example

$$\begin{aligned} \mathbb{K} &= \lambda x. \lambda y. x &= \not\downarrow \\ \mathbb{S} &= \lambda x. \lambda y. \lambda z. x\ z\ (y\ z) = \lambda\ \lambda\ \lambda\ (\\ &\quad (((\# \uparrow 10),_{[LR]} (\# \uparrow 01)) \uparrow 101),_{[LRB]} \\ &\quad (((\# \uparrow 10),_{[LR]} (\# \uparrow 01)) \uparrow 011)) \uparrow id \\ &\quad) \end{aligned}$$

This is actually an Agda Paper

- All categorical concepts formalized in Agda
- Categorical concepts applied to formally reason over programming languages using theorem provers
- Suitable representations of objects and morphisms that *do not block* are required
- Comes with a *universe of metasyntaxes with binding*
- Defines *hereditary* substitution on metasyntaxes

```

cop (θ o') (φ o') = let !!!!!tl, c, tr = cop θ φ in !!!!!tl to', co', tr to'
cop (θ o') (φ os) = let !!!!!tl, c, tr = cop θ φ in !!!!!tl t's', c c's, tr t'ss
cop (θ os) (φ o') = let !!!!!tl, c, tr = cop θ φ in !!!!!tl t'ss, c c's', tr t's'
cop (θ os) (φ os) = let !!!!!tl, c, tr = cop θ φ in !!!!!tl t'ss, c c'ss, tr t'sss
cop oz oz = !!!!!tzzz, czz, tzzz

```

The `copU` proof goes by induction on the triangles which share ψ' and inversion of the coproduct.

A further useful property of coproduct diagrams is that we can selectively refine them by a thinning into the covered scope.

```

subCOP : (ψ : kz ⊆ kz') → Cover ov θ' φ' →
  Σ λ iz → Σ λ jz → Σ (iz ⊆ kz) λ θ → Σ (jz ⊆ kz) λ φ →
  Σ (iz ⊆ iz') λ ψ0 → Σ (jz ⊆ jz') λ ψ1 → Cover ov θ φ

```



The implementation is a straightforward induction on the diagram.

The payoff from coproducts is the type of *relevant pairs* — the co-de-Brujin touchstone:

```

record _×R_ (S T : K) (ijz : Bwd K) : Set where
  constructor pair
  field outl : S ⇕ ijz;   outr : T ⇕ ijz
  cover : Cover tt (thinning outl) (thinning outr)
  in pair (s ⇕ θ) (t ⇕ φ) c ↑ ψ

```

The corresponding projections are readily definable.

```

outlR : (S ×R T) ⇕ kz → S ⇕ kz
outlR (pair s _ ⇕ ψ) = thin ⇕ ψ s

outrR : (S ×R T) ⇕ kz → T ⇕ kz
outrR (pair _ t ⇕ ψ) = thin ⇕ ψ t

```

7 Monoidal Structure of Order-Preserving Embeddings

Variable bindings extend scopes. The λ construct does just one 'snoc', but binding can be simultaneous, so the monoidal structure on Δ_k induced by concatenation is what we need.

```

_++_ : Bwd K → Bwd K → Bwd K
kz ++ [] = kz
kz ++ (iz ∙ j) = (kz ++ iz) ∙ j

```

Concatenation further extends to `Coverings`, allowing us to build them in chunks.

```

_++c_ : Cover ov θ φ → Cover ov θ' φ' → Cover ov (θ ++c θ') (φ ++c φ')
c ++c (d c's) = (c ++c d) c's
c ++c (d c's') = (c ++c d) c's'
c ++c (c'ss {both = b} d) = c'ss {both = b} (c ++c d)
c ++c czz = c

```

One way to build such a chunk is to observe that two scopes cover their concatenation.

Using co-De Bruijn is not hard, category theory is!

```
data Cov : (k l m : ℕ) → Set where
  · : Cov 0 0 0
  L : Cov k l m → Cov (suc k) l (suc m)
  R : Cov k l m → Cov k (suc l) (suc m)
  B : Cov k l m → Cov (suc k) (suc l) (suc m)
```

```
data Tm : ℕ → Set where
  # : Tm 1
  λ : Tm (suc n) → Tm n
  _$[_]_ : Tm k → Cov k l m → Tm l → Tm m
```

```
_ = λ (λ (λ ((# $[ L (R ·) ] #) $[ L (R (B ·)) ] (# $[ L (R ·) ] #))))
```


References

- [1] Conor McBride. *Cats and types: Best friends?* Aug. 2021. URL: <https://www.youtube.com/watch?v=05IJ3YL8p0s>.
- [2] Conor McBride. “Everybody’s Got To Be Somewhere”. In: *Electronic Proceedings in Theoretical Computer Science* 275 (July 2018), pp. 53–69. ISSN: 2075-2180. DOI: 10.4204/eptcs.275.6. URL: <http://dx.doi.org/10.4204/EPTCS.275.6>.