# Foundational Extensible Corecursion

Jasmin Blanchette

Andrei Popescu

Dmitriy Traytel

Technische Universität München

# Foundational Extensible Corecursion

Jasmin Blanchette

Andrei Popescu

Dmitriy Traytel

Isabelle HOL

Inria Nancy Grand Est

Loria Laboratoire lorrain de recherche en informatique et ses applications

mpi max planck institut informatik

Middlesex University London

ETH zürich

# Am I Productive?

```
s = 0 : s
```

```
s = 0 : s
```

✓

primitive corecusion

```
s = 0 : tail s
```

```
s = 0 : tail s
```

✗

`tail` evil

```
s = 0 : 1 : s
```

```
s = 0 : 1 : s
```

✓

corecursion up to constructors

```
eo s = head s : eo (tail (tail s))
eo s = head s : eo (tail (tail s))
```

✓

primitive corecusion

```
s = 0 : 1 : eo s
```

```
s = 0 : 1 : eo s
```

✗

eo evil

```
s ⊕ t = (head s + head t) : (tail s ⊕ tail t)
```

```
s ⊕ t = (head s + head t) : (tail s ⊕ tail t)
```

✓

primitive corecusion

```
s ⊗ t = (head s * head t) : (tail s ⊗ t ⊕ s ⊗ tail t)
```

```
s ⊗ t = (head s ∗ head t) : (tail s ⊗ t ⊕ s ⊗ tail t)
```

✓

corecursion up to ⊕

```
s = (0 : 1 : s) ⊕ (0 : s)
```

$$s = (0 : 1 : s) \oplus (0 : s)$$

✓

corecursion up to constructors and $\oplus$

```
s n =  if n > 0
       then s (n - 1) ⊕ (0 : s (n + 1))
       else 1 : s 1
```

```
s n =  if n > 0
       then s (n - 1) ⊕ (0 : s (n + 1))
       else 1 : s 1
```

✓

mixed recursion/corecursion up to ⊕

# Contribution

Foundational framework for

defining all the green stuff and more

# Contribution

Foundational **framework** for

defining **all** the green stuff **and more**

in an **LCF-style** proof assistant (Isabelle HOL)

# Contribution

Foundational framework for

defining all the green stuff and more

in an LCF-style proof assistant ( Isabelle HOL )

Burden on the user: prove $\left\{ \begin{array}{c} \text{parametricity} \\ \text{or} \\ \text{termination} \end{array} \right\}$ here and there

# Contribution

Foundational framework for

defining all the green stuff and more

in an LCF-style proof assistant ( Isabelle HOL )

Burden on the user: prove $\left\{\begin{array}{c}\text{parametricity}\\ \text{or}\\ \text{termination}\end{array}\right\}$ here and there

Most of the time: automatic

# Context

## LCF Philosophy: Reduce everything to a small trusted kernel

Kernel of Isabelle HOL $\approx \left\{ \begin{array}{l} \text{simply typed lambda calculus} + \\ \text{classical higher-order logic (axioms)} + \\ \text{nonrecursive constant definition} + \\ \text{nonrecursive type definition} \end{array} \right.$

# Context

## LCF Philosophy: Reduce everything to a small trusted kernel

Kernel of Isabelle HOL $\approx$

- simply typed lambda calculus $+$
- classical higher-order logic (axioms) $+$
- nonrecursive constant definition $+$
- nonrecursive type definition

**Our agenda** make Isabelle/HOL a (co)recursion-friendly environment

LICS'12 ITP'14 IJCAR'14 ESOP'15 ICFP'15

# Related Work

A lot

# Related Work

## Guarded Coprogramming/Proof Assistants

| | | |
|---|---|---|
| Isabelle | primitive corecursion | corecursor |

# Related Work

## Guarded Coprogramming/Proof Assistants

| | | |
|---|---|---|
| Isabelle | primitive corecursion | corecursor |
| Coq | corecursion up-to constructors | built-in |

# Related Work

## Guarded Coprogramming/Proof Assistants

| | | |
|---|---|---|
| Isabelle | primitive corecursion | corecursor |
| Coq | corecursion up-to constructors | built-in |
| Agda | copatterns + sized types | built-in + type system |

# Related Work

## Guarded Coprogramming/Proof Assistants

| | | |
|---|---|---|
| Isabelle | primitive corecursion | corecursor |
| Coq | corecursion up-to constructors | built-in |
| Agda | copatterns + sized types | built-in + type system |
| - | FRP (Krishnaswami & Benton, ...) | type system |
| - | clocks (Atkey & McBride) | type system |
| - | guards (Clouston et al.) | type system |

# Related Work

## Guarded Coprogramming/Proof Assistants

| | | |
|---|---|---|
| Isabelle | primitive corecursion | corecursor |
| Coq | corecursion up-to constructors | built-in |
| Agda | copatterns + sized types | built-in + type system |
| - | FRP (Krishnaswami & Benton, . . .) | type system |
| - | clocks (Atkey & McBride) | type system |
| - | guards (Clouston et al.) | type system |
| Isabelle’ | corecursion up-to *friendly* operations<br>mixed with recursion | smart corecursor<br>+ wellfounded recursion |

# Primitive Corecursor

`codatatype` $\mathit{Stream} = \mathit{Int} : \mathit{Stream}$

# Primitive Corecursor

$\texttt{codatatype}\ \mathit{Stream} = \mathit{Int} : \mathit{Stream}$

- $\mathit{Stream} \stackrel{\sim}{=} \texttt{gfp}\,(\mathit{Int} \times -)$
- $\mathsf{corec}^{\mathsf{P}} :: (A \to \mathit{Int} \times A) \to A \to \mathit{Stream}$

# Primitive Corecursor

$\texttt{codatatype}\ \mathit{Stream} = \mathit{Int} : \mathit{Stream}$

- $\mathit{Stream} \overset{\sim}{=} \texttt{gfp}\,(\mathit{Int} \times -)$
- $\mathrm{corec}^{\mathrm{P}} :: (A \to \mathit{Int} \times A) \to A \to \mathit{Stream}$

$\texttt{primcorec}\ s \oplus t = \quad (\mathrm{head}\ s + \mathrm{head}\ t) : (\mathrm{tail}\ s \oplus \mathrm{tail}\ t)$

# Primitive Corecursor

$\texttt{codatatype}\ \mathit{Stream} = \mathit{Int} : \mathit{Stream}$

- $\mathit{Stream} \overset{\sim}{=} \texttt{gfp}\,(\mathit{Int} \times -)$
- $\mathrm{corec}^{\mathrm{P}} :: (A \to \mathit{Int} \times A) \to A \to \mathit{Stream}$

$\texttt{primcorec}\ s \oplus t = \quad (\mathrm{head}\ s + \mathrm{head}\ t) : (\mathrm{tail}\ s \oplus \mathrm{tail}\ t)$

- $s \oplus t = \mathrm{corec}^{\mathrm{P}}\ (\lambda(s,t).\,((\mathrm{head}\ s + \mathrm{head}\ t)\ ,\ (\mathrm{tail}\ s\ \ ,\ \ \mathrm{tail}\ t)))\ (s,t)$

# Primitive Corecursor

$\texttt{codatatype}\ C = \cdots$

- $C \stackrel{\sim}{=} \texttt{gfp}\ F$
- $\mathrm{corec}^{\mathrm{P}} :: (A \to F\ A) \to A \to C$

$\texttt{primcorec}\ f\ \overline{x} = \quad \cdots$

- $f\ \overline{x} {=} \mathrm{corec}^{\mathrm{P}}\ (\lambda(\overline{x}).\ \cdots)\ (\overline{x})$

(Assuming $F$ is a bounded natural functor)

# Smart Corecursor

$\text{corec}^{P} :: (A \to F\ A) \to A \to C$

# Smart Corecursor

$\mathrm{corec}^P :: (A \to F\ A) \to A \to C$

$\mathrm{corec}^S_0 :: (A \to \text{🟩}\,(F\,(\text{🟦}\,A))) \to A \to C$

# Smart Corecursor

$$\mathrm{corec}^{P} :: (A \to F\ A) \to A \to C$$

$$\mathrm{corec}_0^{S} :: (A \to \blacksquare\ (F\ (\blacksquare\ A))) \to A \to C$$

$$\mathrm{corec}_1^{S} :: (A \to \boxed{\oplus}\ (F\ (\boxed{\oplus}\ A))) \to A \to C$$

# Smart Corecursor

$\mathrm{corec}^{P} :: (A \to F\ A) \to A \to C$

$\mathrm{corec}^{S}_{0} :: (A \to \blacksquare\ (F\ (\blacksquare\ A))) \to A \to C$

$\mathrm{corec}^{S}_{1} :: (A \to \boxed{\oplus}\ (F\ (\boxed{\oplus}\ A))) \to A \to C$

$$\texttt{corec}\ s \otimes t = (\mathrm{head}\ s * \mathrm{head}\ t) : (\mathrm{tail}\ s \otimes t\ \oplus\ s \otimes \mathrm{tail}\ t)$$

- $s \otimes t = \mathrm{corec}^{S}_{1}\ (\lambda(s,t).$
  $\eta((\mathrm{head}\ s * \mathrm{head}\ t)\ ,\ \eta(\mathrm{tail}\ s,\ t)\overline{\oplus}\eta(s,\ \mathrm{tail}\ t)))\ (s,t)$
- $\overline{\oplus} :: \boxed{\oplus}\ A \to \boxed{\oplus}\ A \to \boxed{\oplus}\ A$
- $\eta :: A \to \boxed{\oplus}\ A$

# Smart Corecursor

$\mathrm{corec}^{P} :: (A \to F\ A) \to A \to C$

$\mathrm{corec}^{S}_{0} :: (A \to \blacksquare\ (F\ (\blacksquare\ A))) \to A \to C$

$\mathrm{corec}^{S}_{1} :: (A \to \boxed{\oplus}\ (F\ (\boxed{\oplus}\ A))) \to A \to C$

$\mathrm{corec}^{S}_{2} :: (A \to \boxed{\circledast}\ (F\ (\boxed{\circledast}\ A))) \to A \to C$

$\mathtt{corec}\ s \otimes t = (\mathrm{head}\ s * \mathrm{head}\ t) : (\mathrm{tail}\ s \otimes t\ \oplus\ s \otimes \mathrm{tail}\ t)$

- $s \otimes t = \mathrm{corec}^{S}_{1}\ (\lambda(s,t).$
  $\eta((\mathrm{head}\ s * \mathrm{head}\ t)\ ,\ \eta(\mathrm{tail}\ s,\ t)\overline{\oplus}\eta(s,\ \mathrm{tail}\ t)))\ (s,t)$
- $\overline{\oplus} :: \boxed{\oplus}\ A \to \boxed{\oplus}\ A \to \boxed{\oplus}\ A$
- $\eta :: A \to \boxed{\oplus}\ A$

# $\otimes :: C \to C \to C$ has to be friendly

A friendly function can destroy
one constructor to produce
at least one constructor.

# $\otimes :: C \to C \to C$ has to be friendly

$\exists$ parametric $\rho_{\otimes} :: (A \times F\ A) \to (A \times F\ A) \to F\ (\boxed{\circledast}\ A)$ s.t.

$$s \otimes t = \cdots\ (\rho_{\otimes}\ (\cdots\ (s, t)))$$

# $\otimes :: C \to C \to C$ has to be friendly

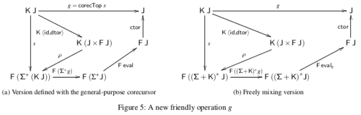
$\exists$ parametric $\rho_{\otimes} :: (A \times F\ A) \to (A \times F\ A) \to F\ (\boxed{\circledast}\ A)$ s.t.

$$s \otimes t = \cdots\ (\rho_{\otimes}\ (\cdots\ (s, t)))$$

$$\rho_{\otimes} :: (A \times (Int \times A)) \to (A \times (Int \times A)) \to (Int \times \boxed{\circledast}\ A)$$

$$\rho_{\otimes}\ (s, hs, ts)\ (t, ht, tt) = (hs * ht, \eta\ ts \mathbin{\overline{\otimes}} \eta\ t\ \overline{\oplus}\ \eta\ s \mathbin{\overline{\otimes}} \eta\ tt)$$

# In the paper



$$\begin{tikzcd}
\mathsf{K}\,\mathsf{J} \arrow[rr, "g = \mathsf{corecTop}\ s"] \arrow[dr, "\mathsf{K}\,\langle \mathsf{id}, \mathsf{dtor} \rangle"'] \arrow[dd, "s"] & & \mathsf{J} \\
 & \mathsf{K}\,(\mathsf{J} \times \mathsf{F}\,\mathsf{J}) \arrow[dl, "\rho"'] & \mathsf{F}\,\mathsf{J} \arrow[u, "\mathsf{ctor}"] \\
\mathsf{F}\,(\Sigma^*\,(\mathsf{K}\,\mathsf{J})) \arrow[r, "\mathsf{F}\,(\Sigma^* g)"] & \mathsf{F}\,(\Sigma^* \mathsf{J}) \arrow[ur, "\mathsf{F}\ \mathsf{eval}"'] &
\end{tikzcd}$$

(a) Version defined with the general-purpose corecursor

$$\begin{tikzcd}
\mathsf{K}\,\mathsf{J} \arrow[rr, "g"] \arrow[dr, "\mathsf{K}\,\langle \mathsf{id}, \mathsf{dtor} \rangle"'] \arrow[dd, "s"] & & \mathsf{J} \\
 & \mathsf{K}\,(\mathsf{J} \times \mathsf{F}\,\mathsf{J}) \arrow[dl, "\rho"'] & \mathsf{F}\,\mathsf{J} \arrow[u, "\mathsf{ctor}"] \\
\mathsf{F}\,((\Sigma + \mathsf{K})^*\ \mathsf{J}) \arrow[r, "\mathsf{F}\,((\Sigma + \mathsf{K})^* g)"] & \mathsf{F}\,((\Sigma + \mathsf{K})^* \mathsf{J}) \arrow[ur, "\mathsf{F}\,\mathsf{eval}_g"'] &
\end{tikzcd}$$

(b) Freely mixing version

Figure 5: A new friendly operation $g$

# In the paper

$$\begin{tikzcd}
\mathsf{K}\ \mathsf{J} \arrow[rr, "g = \mathsf{corecTop}\ s"] \arrow[dr, "\mathsf{K}\ \langle \mathsf{id}, \mathsf{dtor} \rangle"'] \arrow[dd, "s"] & & \mathsf{J} \\
& \mathsf{K}\ (\mathsf{J} \times \mathsf{F}\ \mathsf{J}) \arrow[dl, "\rho"'] & \mathsf{F}\ \mathsf{J} \arrow[u, "\mathsf{ctor}"'] \\
\mathsf{F}\ (\Sigma^*\ (\mathsf{K}\ \mathsf{J})) \arrow[r, "\mathsf{F}\ (\Sigma^* g)"] & \mathsf{F}\ (\Sigma^* \mathsf{J}) \arrow[ur, "\mathsf{F}\ \mathsf{eval}"'] &
\end{tikzcd}$$

(a) Version defined with the general-purpose corecursor

$$\begin{tikzcd}
\mathsf{K}\ \mathsf{J} \arrow[rr, "g"] \arrow[dr, "\mathsf{K}\ \langle \mathsf{id}, \mathsf{dtor} \rangle"'] \arrow[dd, "s"] & & \mathsf{J} \\
& \mathsf{K}\ (\mathsf{J} \times \mathsf{F}\ \mathsf{J}) \arrow[dl, "\rho"'] & \mathsf{F}\ \mathsf{J} \arrow[u, "\mathsf{ctor}"'] \\
\mathsf{F}\ ((\Sigma + \mathsf{K})^*\ \mathsf{J}) \arrow[r, "\mathsf{F}\ ((\Sigma + \mathsf{K})^* g)"] & \mathsf{F}\ ((\Sigma + \mathsf{K})^* \mathsf{J}) \arrow[ur, "\mathsf{F}\, \mathsf{eval}_g"'] &
\end{tikzcd}$$

(b) Freely mixing version

Figure 5: A new friendly operation $g$

$$\begin{tikzcd}
A \arrow[r, "g"] \arrow[d, "s"'] & \Sigma^*\ (\mathsf{F}\ (\Sigma^* A)) \\
\Sigma^*\ (\mathsf{F}\ (\Sigma^* A) + \Sigma^* A) \arrow[r, "\Sigma^*\ (\mathsf{id} + \Sigma^* g)"] & \Sigma^*\ (\mathsf{F}\ (\Sigma^* A) + \Sigma^*\ (\Sigma^*\ (\mathsf{F}\ (\Sigma^* A))))) \arrow[u, "\mathsf{reduce}"']
\end{tikzcd}$$

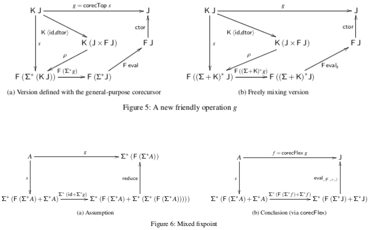
(a) Assumption

$$\begin{tikzcd}
A \arrow[r, "f = \mathsf{corecFlex}\ g"] \arrow[d, "s"'] & \mathsf{J} \\
\Sigma^*\ (\mathsf{F}\ (\Sigma^* A) + \Sigma^* A) \arrow[r, "\Sigma^*\ (\mathsf{F}\ (\Sigma^* f) + \Sigma^* f)"] & \Sigma^*\ (\mathsf{F}\ (\Sigma^* \mathsf{J}) + \Sigma^* \mathsf{J}) \arrow[u, "\mathsf{eval}_{\_(\mathsf{F}\ \_+\_)}"']
\end{tikzcd}$$

(b) Conclusion (via corecFlex)

Figure 6: Mixed fixpoint

# In the paper



$$\begin{tikzcd}
\mathsf{K\,J} \arrow[rr, "g = \mathsf{corecTop}\ s"] \arrow[dr, "\mathsf{K}\ \langle \mathsf{id},\mathsf{dtor}\rangle"] \arrow[dd, "s"] & & \mathsf{J} \\
 & \mathsf{K}\ (\mathsf{J} \times \mathsf{F\,J}) \arrow[dl, "\rho"] & \mathsf{F\,J} \arrow[u, "\mathsf{ctor}"] \\
\mathsf{F}\ (\Sigma^*\ (\mathsf{K\,J})) \arrow[r, "\mathsf{F}\ (\Sigma^* g)"] & \mathsf{F}\ (\Sigma^* \mathsf{J}) \arrow[ur, "\mathsf{F\ eval}"] &
\end{tikzcd}$$

(a) Version defined with the general-purpose corecursor

$$\begin{tikzcd}
\mathsf{K\,J} \arrow[rr, "g"] \arrow[dr, "\mathsf{K}\ \langle \mathsf{id},\mathsf{dtor}\rangle"] \arrow[dd, "s"] & & \mathsf{J} \\
 & \mathsf{K}\ (\mathsf{J} \times \mathsf{F\,J}) \arrow[dl, "\rho"] & \mathsf{F\,J} \arrow[u, "\mathsf{ctor}"] \\
\mathsf{F}\ ((\Sigma + \mathsf{K})^*\ \mathsf{J}) \arrow[r, "\mathsf{F}\ ((\Sigma+\mathsf{K})^* g)"] & \mathsf{F}\ ((\Sigma + \mathsf{K})^* \mathsf{J}) \arrow[ur, "\mathsf{F\,eval}_g"] &
\end{tikzcd}$$

(b) Freely mixing version

Figure 5: A new friendly operation $g$

$$\begin{tikzcd}
A \arrow[r, "g"] \arrow[d, "s"] & \Sigma^*\ (\mathsf{F}\ (\Sigma^* A)) \\
\Sigma^*\ (\mathsf{F}\ (\Sigma^* A) + \Sigma^* A) \arrow[r, "\Sigma^*\ (\mathsf{id} + \Sigma^* g)"] & \Sigma^*\ (\mathsf{F}\ (\Sigma^* A) + \Sigma^*\ (\Sigma^*\ (\mathsf{F}\ (\Sigma^* A))))) \arrow[u, "\mathsf{reduce}"]
\end{tikzcd}$$

(a) Assumption

$$\begin{tikzcd}
A \arrow[r, "f = \mathsf{corecFlex}\ g"] \arrow[d, "s"] & \mathsf{J} \\
\Sigma^*\ (\mathsf{F}\ (\Sigma^* A) + \Sigma^* A) \arrow[r, "\Sigma^*\ (\mathsf{F}\ (\Sigma^* f) + \Sigma^* f)"] & \Sigma^*\ (\mathsf{F}\ (\Sigma^* \mathsf{J}) + \Sigma^* \mathsf{J}) \arrow[u, "\mathsf{eval}_{\_(\mathsf{F}\ \_ + \_)}"]
\end{tikzcd}$$

(b) Conclusion (via corecFlex)

Figure 6: Mixed fixpoint

The following Isabelle-like theory fragment gives a flavor of the envisioned functionality from the user's point of view:

```
codatatype Stream A = SCons (head: A) (tail: Stream A)

corec (friendly) ⊕ : Stream → Stream → Stream
  xs ⊕ ys = SCons (head xs + head ys) (tail xs ⊕ tail ys)

corec (friendly) ⊗ : Stream → Stream → Stream
  xs ⊗ ys = SCons (head xs × head ys)
                  ((xs ⊗ tail ys) ⊕ (tail xs ⊗ ys))
```

# In the paper

$$\begin{tikzcd}
\mathsf{K\,J} \arrow[rr, "g = \mathsf{corecTop}\ s"] \arrow[dr, "\mathsf{K}\ \langle \mathsf{id},\mathsf{dtor}\rangle"'] \arrow[dd, "s"'] & & \mathsf{J} \\
& \mathsf{K}\,(\mathsf{J}\times\mathsf{F\,J}) \arrow[dl, "\rho"'] & \mathsf{F\,J} \arrow[u, "\mathsf{ctor}"] \\
\mathsf{F}\,(\Sigma^*\,(\mathsf{K\,J})) \arrow[r, "\mathsf{F}\,(\Sigma^* g)"] & \mathsf{F}\,(\Sigma^*\,\mathsf{J}) \arrow[ur, "\mathsf{F\ eval}"'] &
\end{tikzcd}$$

(a) Version defined with the general-purpose corecursor

$$\begin{tikzcd}
\mathsf{K\,J} \arrow[rr, "g"] \arrow[dr, "\mathsf{K}\ \langle \mathsf{id},\mathsf{dtor}\rangle"'] \arrow[dd, "s"'] & & \mathsf{J} \\
& \mathsf{K}\,(\mathsf{J}\times\mathsf{F\,J}) \arrow[dl, "\rho"'] & \mathsf{F\,J} \arrow[u, "\mathsf{ctor}"] \\
\mathsf{F}\,((\Sigma+\mathsf{K})^*\,\mathsf{J}) \arrow[r, "\mathsf{F}\,((\Sigma+\mathsf{K})^* g)"] & \mathsf{F}\,((\Sigma+\mathsf{K})^*\,\mathsf{J}) \arrow[ur, "\mathsf{F\,eval}_g"'] &
\end{tikzcd}$$

(b) Freely mixing version

Figure 5: A new friendly operation $g$

$$\begin{tikzcd}
A \arrow[rr, "g"] \arrow[d, "s"'] & & \Sigma^*\,(\mathsf{F}\,(\Sigma^* A)) \\
\Sigma^*\,(\mathsf{F}\,(\Sigma^* A)+\Sigma^* A) \arrow[rr, "\Sigma^*\,(\mathsf{id}+\Sigma^* g)"] & & \Sigma^*\,(\mathsf{F}\,(\Sigma^* A)+\Sigma^*\,(\Sigma^*\,(\mathsf{F}\,(\Sigma^* A))))) \arrow[u, "\mathsf{reduce}"]
\end{tikzcd}$$

(a) Assumption

$$\begin{tikzcd}
A \arrow[rr, "f = \mathsf{corecFlex}\ g"] \arrow[d, "s"'] & & \mathsf{J} \\
\Sigma^*\,(\mathsf{F}\,(\Sigma^* A)+\Sigma^* A) \arrow[rr, "\Sigma^*\,(\mathsf{F}\,(\Sigma^* f)+\Sigma^* f)"] & & \Sigma^*\,(\mathsf{F}\,(\Sigma^*\mathsf{J})+\Sigma^*\mathsf{J}) \arrow[u, "\mathsf{eval}_{\_(\mathsf{F}\ \_+\_)}"]
\end{tikzcd}$$

(b) Conclusion (via corecFlex)

Figure 6: Mixed fixpoint

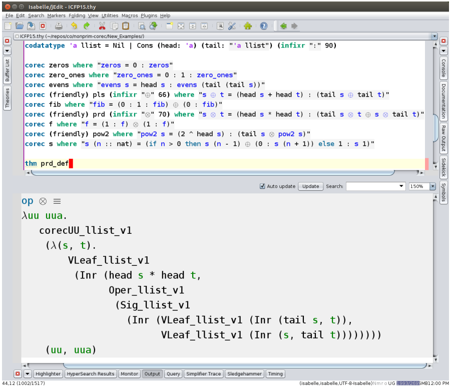
The following Isabelle-like theory fragment gives a flavor of the envisioned functionality from the user's point of view:

$$\texttt{codatatype}\ \mathsf{Stream}\ A = \mathsf{SCons}\ (\mathsf{head}{:}\ A)\ (\mathsf{tail}{:}\ \mathsf{Stream}\ A)$$

$$\texttt{corec (friendly)}\ \oplus : \mathsf{Stream} \to \mathsf{Stream} \to \mathsf{Stream}$$
$$\quad xs \oplus ys = \mathsf{SCons}\ (\mathsf{head}\ xs + \mathsf{head}\ ys)\ (\mathsf{tail}\ xs \oplus \mathsf{tail}\ ys)$$

$$\texttt{corec (friendly)}\ \otimes : \mathsf{Stream} \to \mathsf{Stream} \to \mathsf{Stream}$$
$$\quad xs \otimes ys = \mathsf{SCons}\ (\mathsf{head}\ xs \times \mathsf{head}\ ys)$$
$$\qquad ((xs \otimes \mathsf{tail}\ ys) \oplus (\mathsf{tail}\ xs \otimes ys))$$

# In the meantime



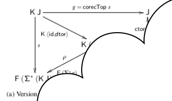
Isabelle/jEdit - ICFP15.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
ICFP15.thy (~/repos/co/nonprim-corec/New_Examples/)
Buffer List
Theories
Console
Documentation
Raw Output
Sidekick
Symbols

```
codatatype 'a llist = Nil | Cons (head: 'a) (tail: "'a llist") (infixr ":" 90)

corec zeros where "zeros = 0 : zeros"
corec zero_ones where "zero_ones = 0 : 1 : zero_ones"
corec evens where "evens s = head s : evens (tail (tail s))"
corec (friendly) pls (infixr "⊕" 66) where "s ⊕ t = (head s + head t) : (tail s ⊕ tail t)"
corec fib where "fib = (0 : 1 : fib) ⊕ (0 : fib)"
corec (friendly) prd (infixr "⊗" 70) where "s ⊗ t = (head s * head t) : (tail s ⊗ t ⊕ s ⊗ tail t)"
corec f where "f = (1 : f) ⊗ (1 : f)"
corec (friendly) pow2 where "pow2 s = (2 ^ head s) : (tail s ⊗ pow2 s)"
corec s where "s (n :: nat) = (if n > 0 then s (n - 1) ⊕ (0 : s (n + 1)) else 1 : s 1)"

thm prd_def
```

☑ Auto update | Update | Search: | 150%

```
op ⊗ ≡
λuu uua.
   corecUU_llist_v1
    (λ(s, t).
        VLeaf_llist_v1
         (Inr (head s * head t,
               Oper_llist_v1
                (Sig_llist_v1
                  (Inr (VLeaf_llist_v1 (Inr (tail s, t)),
                        VLeaf_llist_v1 (Inr (s, tail t))))))))
    (uu, uua)
```

In the paper ... eantime



|  |  |
|---|---|
| Coq | constructor$^+$ |
| Agda | constructor$^+$ · arbitrary (manual proofs) |
| Isabelle | friendly$^*$ · constructor · friendly$^*$ (auto proofs) |

# Thanks for listening!
# Questions?

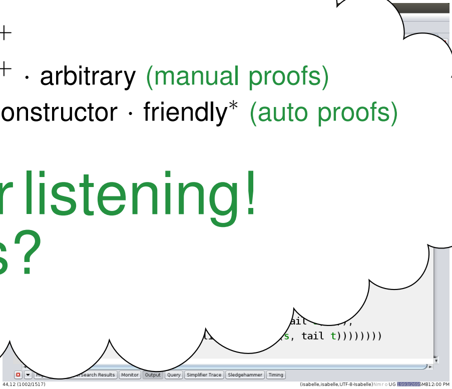
Σ* (

The fo
envisioned

```
codatatype S
```

corec (friendly) ⊕ : St
  $xs \oplus ys$ = SCons (head

corec (friendly) ⊗ : Stream → Strea
  $xs \otimes ys$ = SCons (head $xs$ × head $ys$)
      (($xs \otimes$ tail $ys$) ⊕ (tail $xs \otimes ys$))



```
ail
s, tail t))))))))
```

# Foundational Extensible Corecursion

Jasmin Blanchette

Andrei Popescu

Dmitriy Traytel

Isabelle HOL

Inria Nancy Grand Est

Loria Laboratoire lorrain de recherche en informatique et ses applications

mpi max planck institut informatik

Middlesex University London

ETH zürich

What is s 1?

```
s n = if n > 0 then s (n - 1) ⊕ (0 : s (n + 1)) else 1 : s 1
```