Streams for Cubical Type Theory

Andrea Vezzosi vezzosi@chalmers.se

August 17, 2017

We extend cubical type theory [2] with a notion of stream type where two bisimilar streams can be proven equal according to path equality. In the style of copatterns, corecursive definitions for streams will only compute when head or tail projections are applied to them. Semantics are given by final coalgebras in (arbitrary) cubical sets. This extension is implemented as part of Agda "–cubical"[1].

In short:

- Given a Kan cubical set A, we build the Kan cubical set of streams SA by using a final coalgebra of $A\times$ in arbitrary cubical sets and then show by the universal property that it supports a composition structure.
- Since SA's universal property is with regard to arbitrary cubical sets we can build maps $\mathbb{I} \to SA$, and so paths, by corecursion, exposing the coinductive nature of the propositional equality of streams.
- In cubical type theory we internalize the above by an extended unfold combinator for streams; and in the Agda implementation we do the same by allowing path abstraction within copattern definitions.

Following [3] we could derive a form of dependent elimination for streams that does not break subject reduction, by transporting along the propositional η-law.

The Agda implementation supports a general notion of (coinductive) records, we conjecture that they can be justified by a semantics analogous to the one we give here for the stream type.

Ahrens et al.[1] have already shown M-types are definable within homotopy type theory (as homotopy limits of chains). The current note differs by providing the expected universal property for a primitive stream type, whose representation is more amenable to efficient computation.

1 Typing and Conversion rules

In the following we omit the congruence and commuting with substitions rules for the conversion relation. The expression $\mathsf{S}\ A$ denotes the type of streams with elements in A, together with head and tail projections $t.\mathsf{hd}$ and $t.\mathsf{tl}$.

$$\frac{\Gamma \vdash A}{\Gamma \vdash \mathsf{S}\ A} \qquad \frac{\Gamma \vdash t : \mathsf{S}\ A}{\Gamma \vdash t.\mathsf{hd} : A} \qquad \frac{\Gamma \vdash t : \mathsf{S}\ A}{\Gamma \vdash t.\mathsf{tl} : \mathsf{S}\ A}$$

[1]on Agda's master branch

We then provide an unfold combinator to build streams, this keeps the calculus small and easy to give semantics to while the implementation provides definitions by copatterns, which are more convenient to use.

$$\frac{\begin{array}{c}\Gamma \vdash X \qquad \Gamma \vdash a : X \to A \times X \qquad \Gamma \vdash t : X \\ \Gamma \vdash \varphi \qquad \Gamma, \varphi \vdash u : X \to \mathsf{S}\ A \\ \Gamma, \varphi, x : X \vdash (u\ x).\mathsf{hd} = (a\ x).1 \\ \Gamma, \varphi, x : X \vdash (u\ x).\mathsf{tl} = u\ (a\ x).2\end{array}}{\Gamma \vdash \mathsf{unfold}_X\ [\varphi \mapsto u]\ a\ t : \mathsf{S}\ A[\varphi \mapsto u\ t]}$$

$$\begin{array}{l}\Gamma \vdash (\mathsf{unfold}\ [\varphi \mapsto u]\ a\ t).\mathsf{hd} = (a\ t).1 : \mathsf{S}\ A \\ \Gamma \vdash (\mathsf{unfold}\ [\varphi \mapsto u]\ a\ t).\mathsf{tl}\ = \mathsf{unfold}\ [\varphi \mapsto u]\ a\ (a\ t).2 : \mathsf{S}\ A\end{array}$$

Our unfold takes an extra $[\varphi \mapsto u]$ argument because it not only witnesses that from any coalgebra $a : X \to A \times X$ we can build a map $X \to \mathsf{S}\ A$ that strictly commutes with a[2], but also that any such partial map u can be extended to a total one. In other words unfold shows the contractibility of the type[3] of strict coalgebra morphisms into $\mathsf{S}\ A$.

Composition for streams is characterized by commuting with the head and tail projections.

$$\begin{array}{l}\Gamma \vdash (\mathsf{comp}^i\ (\mathsf{S}\ A)\ [\varphi \mapsto u]\ a_0).\mathsf{hd} = \mathsf{comp}^i\ \mathrm{A} \qquad [\varphi \mapsto u.\mathsf{hd}]\ a_0.\mathsf{hd} : A \\ \Gamma \vdash (\mathsf{comp}^i\ (\mathsf{S}\ A)\ [\varphi \mapsto u]\ a_0).\mathsf{tl}\ = \mathsf{comp}^i\ (\mathsf{S}\ A)\ [\varphi \mapsto u.\mathsf{tl}\ ]\ a_0.\mathsf{tl}\ : \mathsf{S}\ A\end{array}$$

2 Examples

Bisimulation.

$$\frac{\begin{array}{c}\Gamma \vdash A \qquad \Gamma \vdash R : (x\,y : \mathsf{S}\ A) \to U \\ \Gamma \vdash R_{\mathsf{hd}} : (x\,y : \mathsf{S}\ A) \to R\,x\,y \to \mathsf{Path}_A\ x.\mathsf{hd}\ y.\mathsf{hd} \\ \Gamma \vdash R_{\mathsf{tl}} : (x\,y : \mathsf{S}\ A) \to R\,x\,y \to R\ x.\mathsf{tl}\ y.\mathsf{tl}\end{array}}{\begin{array}{c}\Gamma \vdash \mathsf{bisim}_{R,R_{\mathsf{hd}},R_{\mathsf{tl}}} = \lambda x\,y\,r.\,\langle i\rangle\ \mathsf{unfold}\ S\ (\lambda(x,y,r).\,R_{\mathsf{hd}}\ r\ i, (x.\mathsf{tl}, y.\mathsf{tl}, R_{\mathsf{tl}}\ r))\ (x,y,r) \\ : (x\,y : \mathsf{S}\ A) \to R\ x\ y \to \mathsf{Path}_{\mathsf{S}\ A}\ x\ y\end{array}}$$

where $S = [(i=0) \mapsto \lambda(x,y,r).\,x, (i=1) \mapsto \lambda(x,y,r).\,y]$

Note that we can derive the following equalities that show how bisim computes under head and tail, these are also how bisim can be defined by direct recursion in the implementation[4].

$$\begin{array}{l}(\mathsf{bisim}_{R,R_{\mathsf{hd}},R_{\mathsf{tl}}}\ x\ y\ r\ i).\mathsf{hd} = R_{\mathsf{hd}}\ x\ y\ r\ i \\ (\mathsf{bisim}_{R,R_{\mathsf{hd}},R_{\mathsf{tl}}}\ x\ y\ r\ i).\mathsf{tl}\ = \mathsf{bisim}_{R,R_{\mathsf{hd}},R_{\mathsf{tl}}}\ x.\mathsf{tl}\ y.\mathsf{tl}\ (R_{\mathsf{tl}}\ x\ y\ r)\ i\end{array}$$

Uniqueness of unfold Here we prove that $\mathsf{unfold}_X\ a : X \to \mathsf{S}\ A$ is the unique coalgebra morphism from (X, a).

[2] as a coalgebra morphism would
[3] although non-fibrant
[4] `https://github.com/Saizan/cubical-demo/blob/master/Stream.agda`

$$\frac{\begin{array}{c}\Gamma \vdash X \qquad \Gamma \vdash a : X \to A \times X \qquad \Gamma \vdash h : X \to \mathsf{S}\ A \\ \Gamma \vdash p_{\mathsf{hd}} : (x : X) \to \mathsf{Path}_A\ (h\ x).\mathsf{hd}\ (a\ x).1 \\ \Gamma \vdash p_{\mathsf{tl}} : (x : X) \to \mathsf{Path}_{\mathsf{S}\ A}\ (h\ x).\mathsf{tl}\ (h\ (a\ x).2)\end{array}}{\begin{array}{c}\Gamma \vdash \eta_{\mathsf{S}} = \lambda x.\langle i\rangle\ \mathsf{unfold}_{(x:X)\times(s:\mathsf{S}\ A)\times\mathsf{Path}_{\mathsf{S}\ A}\ s\ (h\ x)}\ S\ b\ (x, h\,x, \langle i\rangle\ h\,x) \\ : \mathsf{Path}_{X\to\mathsf{S}\ A}\ h\ (\mathsf{unfold}_X\ a)\end{array}}$$

where

$$\begin{array}{ll} S & = [(i=0) \mapsto \lambda(x,s,p).\, s, (i=1) \mapsto \lambda(x,s,p).\, \mathsf{unfold}_X\ a\ x] \\ b & = \lambda(x,s,p).\, (\mathsf{trans}\,(\langle i\rangle\ (p\,i).\mathsf{hd})\,(p_{\mathsf{hd}}\ x)\,i, ((a\ x).2, s.\mathsf{tl}, \mathsf{trans}\,(\langle i\rangle\ (p\,i).\mathsf{tl})\,(p_{\mathsf{tl}}\ x))) \\ \mathsf{trans}_{A,x,y,z} & = \lambda p\,q.\langle i\rangle\ \mathsf{comp}^j\ A\,[(i=0)\mapsto x, (i=1)\mapsto q\,j]\,(p\,i) \end{array}$$

It might have been more natural to take η_{S} and $\mathsf{unfold}_X : (A \to X \times A) \to \mathsf{S}\ A$ as the primitives, instead of our more general unfold, however we believe that the resulting system would have been less convenient in the sense that equality proofs for streams would be less direct.

3 Semantics

We provide semantics in cubical sets by using final coalgegras in sets and lifting them pointwise[5]. The computation rules for $\mathsf{comp}^i\ (\mathsf{S}\ A)$ and unfold already suggest how to intepret them via coalgebras.

For any set A we fix a final coalgebra of the functor $A\times$ and call it $(\mathcal{S}\ A, \langle \mathsf{hd}, \mathsf{tl}\rangle)$ and call $\mathsf{ana}_A\ (X, f) : X \to \mathcal{S}\ A$ the unique coalgebra morphism from any coalgebra (X, f). So that $\mathsf{hd}\ (\mathsf{ana}_A\ (X, f)\ x) = (f\ x).1$ and $\mathsf{tl}\ (\mathsf{ana}_A\ (X, f)\ x) = \mathsf{ana}_A\ (X, f)\ (f\ x).2$ where we reuse the .1 and .2 notation for set theoretic pairs. We will often omit the subscript A.

Definition 1 (Stream type)**.** Given $A \in \mathrm{Ty}(\Gamma)$ we define $\mathsf{S}(A) \in \mathrm{Ty}(\Gamma)$ by setting $\mathsf{S}(A)\rho = \mathcal{S}\ (A\rho)$ for $\rho \in \Gamma(I)$.

Given $t \in \mathsf{S}(A)\rho$ and $f : I \to J$ we define tf as $\mathsf{ana}_{A(\rho f)}\ (\mathcal{S}\ (A\rho), s \mapsto (\mathsf{hd}(s)f, \mathsf{tl}(s)))\ t$.

Remark 2 (Naturality of projections)*.* For $\rho \in \Gamma(I)$ and $s \in \mathsf{S}(A)\rho$, we have $\mathsf{hd}(s)f = \mathsf{hd}(sf)$ and $\mathsf{tl}(s)f = \mathsf{tl}(sf)$.

Proof.

$$\begin{array}{l} \mathsf{hd}(s)f \\ = \mathsf{hd}(\mathsf{ana}_{A(\rho f)}\ (\mathcal{S}\ (A\rho), s \mapsto (\mathsf{hd}(s)f, \mathsf{tl}(s)))\ t) \\ = \mathsf{hd}(sf) \end{array}$$

$$\begin{array}{l} \mathsf{tl}(s)f \\ = \mathsf{ana}_{A(\rho f)}\ (\mathcal{S}\ (A\rho), s \mapsto (\mathsf{hd}(s)f, \mathsf{tl}(s)))\ (\mathsf{tl}(s)) \\ = \mathsf{tl}(\mathsf{ana}_{A(\rho f)}\ (\mathcal{S}\ (A\rho), s \mapsto (\mathsf{hd}(s)f, \mathsf{tl}(s)))\ s) \\ = \mathsf{tl}(sf) \end{array}$$

□

Definition 3 (.hd and .tl)**.** Given then $t \in \mathrm{Ter}(\Gamma; \mathsf{S}(A))$ we define $.\mathsf{hd}(t) \in \mathrm{Ter}(\Gamma; A)$ and $.\mathsf{tl}(t) \in \mathrm{Ter}(\Gamma; \mathsf{S}(A))$ by setting $.\mathsf{hd}(t)\rho = \mathsf{hd}(t\rho)$ and $.\mathsf{tl}(t)\rho = \mathsf{tl}(t\rho)$ for $\rho \in \Gamma(I)$.

[5] as one would do for limits in presheaf categories, non-coincidentally

Definition 4 (unfold)**.** Given $X \in \mathrm{Ty}(\Gamma)$, $\varphi \in \mathrm{Ter}(\Gamma; \mathbb{F})$, $u \in \mathrm{Ter}(\Gamma, \varphi; X \to \mathsf{S}(A))$, $a \in \mathrm{Ter}(\Gamma; A \to X \times A)$, $t \in \mathrm{Ter}(\Gamma; X)$, we define $\mathsf{unfold}(X, \varphi, u, a, t) \in \mathrm{Ter}(\Gamma; \mathsf{S}(A))$ by setting $\mathsf{unfold}(X, \varphi, u, a, t)\rho = \mathsf{ana}\ (X\rho, (a\rho)_1)\ (t\rho)$ for $\rho \in \Gamma(I)$.

Theorem 5 (unfold under φ)**.** *By the previous assumptions about u and a and*

$$\forall \rho \in (\Gamma, \varphi)(I) \text{ and } t \in X\rho, \quad \begin{array}{l} \mathsf{hd}((u\rho)_1\ t) = ((a\rho)_1\ t).1 \\ \mathsf{tl}((u\rho)_1\ t) = (u\rho)_1\ ((a\rho)_1\ t).2 \end{array}$$

we have $\mathsf{unfold}(X, \varphi, u, a, t)\rho = (u\rho)_1\ (t\rho)$ *for* $\rho \in (\Gamma, \varphi)(I)$.

Proof. By uniqueness of $\mathsf{ana}\ (X\rho, (a\rho)_1)$. □

The computation rules for unfold follow directly from the behaviour of ana.

Definition 6 (Composition)**.** Given $A \in \mathrm{Ty}(\Gamma)$, we give a composition structure for $\mathsf{S}(A) \in \mathrm{Ty}(\Gamma)$. Assuming I, $i \notin I$, $\rho \in \Gamma(I, i)$, $\varphi \in \mathbb{F}(I)$, $u \in \mathrm{Ter}(\mathbf{y}(I, i), \varphi; \mathsf{S}(A)\rho\iota_\varphi)$, and $a_0 \in \mathsf{S}(A)\rho(i0)$ with $a_0\iota_\varphi = u(i0)$, we define $\mathsf{comp}_{\mathsf{S}(A)}(I, i, \rho, \varphi, u, a_0) \in \mathsf{S}(A)\rho(i1)$ by $\mathsf{comp}_{\mathsf{S}(A)}(I, i, \rho, \varphi, u, a_0) = \mathsf{ana}\ (X_{I,i,\rho,\varphi}, \mathsf{coalg}_{I,i,\rho,\varphi})\ (u, a_0)$ where

$$X_{I,i,\rho,\varphi} = \{(u, a_0) | u \in \mathrm{Ter}(\mathbf{y}(I, i), \varphi; \mathsf{S}(A)\rho\iota_\varphi), a_0 \in \mathsf{S}(A)\rho(i0) \text{ with } a_0\iota_\varphi = u(i0)\}$$

$$\mathsf{coalg}_{I,i,\rho,\varphi}(u, a_0) = (\mathsf{comp}_A(I, i, \rho, \varphi, \mathsf{hd}(u), \mathsf{hd}(a_0)), (\mathsf{tl}(u), \mathsf{tl}(a_0)))$$

We show that for any $f : I \to J$ and $j \notin J$,

$$(\mathsf{comp}_{\mathsf{S}(A)}(I, i, \rho, \varphi, u, a_0))f = \mathsf{comp}_{\mathsf{S}(A)}(J, j, \rho(f, i = j), \varphi f, u(f, i = j), a_0 f)$$

Proof.

$$(\mathsf{comp}_{\mathsf{S}(A)}(I, i, \rho, \varphi, u, a_0))f$$

$$= (\mathsf{ana}\ (X_{I,i,\rho,\varphi}, \mathsf{coalg}_{I,i,\rho,\varphi})\ (u, a_0))f$$

$$=< \text{by uniqueness} > \mathsf{ana}\ (X_{I,i,\rho,\varphi}, p \mapsto ((\mathsf{coalg}_{I,i,\rho,\varphi}(p).1)f, \mathsf{coalg}_{I,i,\rho,\varphi}(p).2))\ (u, a_0)$$

$$=< \text{by uniqueness} > \mathsf{ana}\ (X_{J,j,\rho(f,i=j),\varphi f}, \mathsf{coalg}_{J,j,\rho(f,i=j),\varphi f})\ (u(f, i = j), a_0 f)$$

$$= \mathsf{comp}_{\mathsf{S}(A)}(J, j, \rho(f, i = j), \varphi f, u(f, i = j), a_0 f)$$

□

By definition the $\mathsf{comp}_{\mathsf{S}(A)}$ map commutes with the hd and tl projections stricly, which in turn verifies the computation rules for $\mathsf{comp}^i\ (\mathsf{S}\ A)$:

$$\begin{array}{l} \mathsf{hd}(\mathsf{comp}_{\mathsf{S}(A)}(I, i, \rho, \varphi, u, a_0)) = \mathsf{comp}_A(I, i, \rho, \varphi, \mathsf{hd}(u), \mathsf{hd}(a_0)) \\ \mathsf{tl}(\mathsf{comp}_{\mathsf{S}(A)}(I, i, \rho, \varphi, u, a_0)) = \mathsf{comp}_{\mathsf{S}(A)}(I, i, \rho, \varphi, \mathsf{tl}(u), \mathsf{tl}(a_0)) \end{array}$$

Theorem 7 (Fibrant Stream type)**.** *Given $A \in \mathrm{FTy}(\Gamma)$ we have $\mathsf{S}(A) \in \mathrm{FTy}(\Gamma)$. Moreover, $\mathsf{S}(A)\sigma = \mathsf{S}(A\sigma) \in \mathrm{FTy}(\Delta)$ for all $\sigma : \Delta \to \Gamma$.*

Proof. We define $\mathsf{S}(A) \in \mathrm{FTy}(\Gamma)$ as $(\mathsf{S}(A) \in \mathrm{Ty}(\Gamma), \mathsf{comp}_{\mathsf{S}(A)})$. Given $\sigma : \Delta \to \Gamma$, both $\mathsf{S}(A)\sigma\rho = \mathsf{S}(A\sigma)\rho$ and $\mathsf{comp}_{\mathsf{S}(A)}(I, i, \sigma\rho, \phi, u, a_0) = \mathsf{comp}_{\mathsf{S}(A\sigma)}(I, i, \rho, \phi, u, a_0)$ follow direcly by expansion, because in the definitions the only uses of the environment are in relation to the type of elements or its composition structure. □

References

[1] B. Ahrens, P. Capriotti, and R. Spadotti. Non-wellfounded trees in homotopy type theory. *CoRR*, abs/1504.02949, 2015.

[2] C. Cohen, T. Coquand, S. Huber, and A. Mörtberg. Cubical type theory: a constructive interpretation of the univalence axiom. *CoRR*, abs/1611.02108, 2016.

[3] C. McBride. Let's see how things unfold: Reconciling the infinite with the intensional. In *Proceedings of the 3rd International Conference on Algebra and Coalgebra in Computer Science*, CALCO'09, pages 113–126, Berlin, Heidelberg, 2009. Springer-Verlag.