Sampling, Splitting, and Merging in Coinductive Stream Calculus

Jan Rutten, CWI & VUA

Abstract

That this paper may stay out of the hands of the wonderfully cruel folklore referee who once said:

Your paper contains many new and interesting results.

Unfortunately, the new results are not interesting, and the interesting results are not new ;-)

Sampling, Splitting, and Merging in Coinductive Stream Calculus

Jan Rutten, CWI & VUA

Abstract

That this paper may stay out of the hands of the wonderfully cruel folklore referee who once said:

Your paper contains many new and interesting results.

Unfortunately, the new results are not interesting, and the interesting results are not new ;-)

Sampling, Splitting, and Merging in Coinductive Stream Calculus

Jan Rutten, CWI & VUA

Abstract

That this paper may stay out of the hands of the wonderfully cruel folklore referee who once said:

Your paper contains many new and interesting results.

Unfortunately, the new results are not interesting, and the interesting results are not new ;-)

Sampling, Splitting, and Merging in Coinductive Stream Calculus

Jan Rutten, CWI & VUA

Abstract

That this paper may stay out of the hands of the wonderfully cruel folklore referee who once said:

Your paper contains many new and interesting results.

Unfortunately, the new results are not interesting, and the interesting results are not new ;-)

Overview

1. Periodic stream samplers
2. Splitting and merging of streams
3. Rationality
4. Stream circuits

1. Periodic stream samplers
   - are relevant for stream processing systems and
   - digital signal processing applications.

Examples of periodic stream samplers

- Let $A^\omega = \{\sigma \mid \sigma : \mathbb{N} \to A\}$ be the set of streams

$$\sigma = (\sigma(0), \sigma(1), \sigma(2), \ldots)$$

- The function $\mathit{even} : A^\omega \to A^\omega$ is given by

$$\mathit{even}(\sigma) = (\sigma(0), \sigma(2), \sigma(4), \ldots)$$

- The drop operator $D_4^2 : A^\omega \to A^\omega$ is given by

$$D_4^2(\sigma) = (\sigma(0), \sigma(1), \sigma(3), \sigma(4), \sigma(5), \sigma(7), \ldots)$$

Periodic stream samplers, traditionally

- A monotone function $f : \mathbb{N} \to \mathbb{N}$ determines a *stream sampler* $S_f : A^\omega \to A^\omega$ given by

$$S_f(\sigma)(n) = \sigma(f(n))$$

- For which $f : \mathbb{N} \to \mathbb{N}$ is S_f a *periodic* stream sampler?
- One formal answer [Mak 2005, TU/e]: *periodic block maps*.
- Tricky and non-intuitive; leads to difficult proofs.

Periodic stream samplers, coinductively

- Recall: $\sigma^{(k)} = (\sigma(k), \sigma(k+1), \sigma(k+2), \ldots)$.
- A *periodic stream sampler* $S : A^\omega \to A^\omega$ is defined by the following stream differential equation:

$$S(\sigma)^{(k)} = S(\sigma^{(l)})$$

We call $l > 1$ the *period* and $1 \leq k \leq l$ the *block size*.

- Furthermore, we specify the initial values

$$S(\sigma)(0) = \sigma(n_0), \quad \ldots \quad , S(\sigma)(k-1) = \sigma(n_{k-1})$$

where $0 \leq n_0 < n_1 < \cdots < n_{k-1} < l$.

Periodic stream samplers, coinductively

- Recall: $\sigma^{(k)} = (\sigma(k), \sigma(k+1), \sigma(k+2), \ldots)$.
- A *periodic stream sampler* $S : A^\omega \to A^\omega$ is defined by the following stream differential equation:

$$S(\sigma)^{(k)} = S(\sigma^{(l)})$$

We call $l > 1$ the *period* and $1 \leq k \leq l$ the *block size*.

- Furthermore, we specify the initial values

$$S(\sigma)(0) = \sigma(n_0), \ \ldots \ , S(\sigma)(k-1) = \sigma(n_{k-1})$$

where $0 \leq n_0 < n_1 < \cdots < n_{k-1} < l$.

Our earlier examples

- The function $even : A^\omega \to A^\omega$ is given by

$$even(\sigma)^{(1)} = even(\sigma^{(2)}), \qquad even(\sigma)(0) = \sigma(0)$$

- The drop operator $D_4^2 : A^\omega \to A^\omega$ is given by

$$D_4^2(\sigma)^{(3)} = D_4^2(\sigma^{(4)})$$

with initial values

$$D_4^2(\sigma)(0) = \sigma(0), \;\; D_4^2(\sigma)(1) = \sigma(1), \;\; D_4^2(\sigma)(2) = \sigma(3)$$

Our earlier examples

- The function *even* : $A^\omega \to A^\omega$ is given by

$$even(\sigma)^{(1)} = even(\sigma^{(2)}), \qquad even(\sigma)(0) = \sigma(0)$$

- The drop operator $D_4^2 : A^\omega \to A^\omega$ is given by

$$D_4^2(\sigma)^{(3)} = D_4^2(\sigma^{(4)})$$

with initial values

$$D_4^2(\sigma)(0) = \sigma(0), \;\; D_4^2(\sigma)(1) = \sigma(1), \;\; D_4^2(\sigma)(2) = \sigma(3)$$

Easy proofs, coinductively

- If $S, T : A^\omega \to A^\omega$ are two periodic stream samplers then so is $T \circ S$.
- We have:

$$D_2^0 = D_4^0 \circ D_5^2 \circ D_6^4$$

For a proof, define $R \subseteq A^\omega \times A^\omega$ by

$$\begin{aligned} R &= \{< D_2^0(\sigma),\ D_4^0 \circ D_5^2 \circ D_6^4(\sigma)\ >|\ \sigma \in A^\omega\} \\ &\cup \{< D_2^0(\sigma),\ D_4^2 \circ D_5^0 \circ D_6^2(\sigma)\ >|\ \sigma \in A^\omega\} \\ &\cup \{< D_2^0(\sigma),\ D_4^1 \circ D_5^3 \circ D_6^0(\sigma)\ >|\ \sigma \in A^\omega\} \end{aligned}$$

and note that it is a *stream bisimulation*.

2. Splitting and merging of streams

Take and zip operators

- Example: $T_3^2(\sigma) = (\sigma(2), \sigma(5), \sigma(8), \ldots)$
- General: we define the *take operator* $T_l^i : A^\omega \to A^\omega$ by

$$T_l^i(\sigma)' = T_l^i(\sigma^{(l)}) \qquad T_l^i(\sigma)(0) = \sigma(i)$$

for $l \geq 2$ and $0 \leq i < l$.

- Example: $Z_2(\sigma, \tau) = (\sigma(0), \tau(0), \sigma(1), \tau(1), \sigma(2), \tau(2), \ldots)$
- General: we define the *zip operator* $Z_k : (A^\omega)^k \to A^\omega$ by

$$Z_k(\sigma_0, \ldots, \sigma_{k-1})' = Z_k(\sigma_1, \ldots, \sigma_{k-1}, \sigma_0')$$

with initial value $Z_k(\sigma_0, \ldots, \sigma_{k-1})(0) = \sigma_0(0)$.

Take and zip operators

- Example: $T_3^2(\sigma) = (\sigma(2), \sigma(5), \sigma(8), \ldots)$
- General: we define the *take operator* $T_l^i : A^\omega \to A^\omega$ by

$$T_l^i(\sigma)' = T_l^i(\sigma^{(l)}) \qquad T_l^i(\sigma)(0) = \sigma(i)$$

for $l \geq 2$ and $0 \leq i < l$.

- Example: $Z_2(\sigma, \tau) = (\sigma(0), \tau(0), \sigma(1), \tau(1), \sigma(2), \tau(2), \ldots)$
- General: we define the *zip operator* $Z_k : (A^\omega)^k \to A^\omega$ by

$$Z_k(\sigma_0, \ldots, \sigma_{k-1})' = Z_k(\sigma_1, \ldots, \sigma_{k-1}, \sigma_0')$$

with initial value $Z_k(\sigma_0, \ldots, \sigma_{k-1})(0) = \sigma_0(0)$.

Take and zip operators

- *Take* and *zip* let us split and merge streams.
- They are sufficient to express all periodic stream samplers $S : A^\omega \to A^\omega$:

$$S(\sigma) = Z_k(\, T_l^{n_0}(\sigma),\; T_l^{n_1}(\sigma),\; \ldots,\; T_l^{n_{k-1}}(\sigma)\,)$$

  of *period* $l > 1$ and *block size* $1 \leq k \leq l$.
- For instance,

$$D_6^4(\sigma) = \; Z_5(\, T_6^0(\sigma),\; T_6^1(\sigma),\; T_6^2(\sigma),\; T_6^3(\sigma),\; T_6^5(\sigma)\,)$$

Take and zip operators

- More generally, we can define periodic stream *transformers*, not only *samplers*. E.g.,

$$Rev_3(\sigma) = (\sigma(2), \sigma(1), \sigma(0), \sigma(5), \sigma(4), \sigma(3), \ldots)$$

which is given by

$$Rev_3(\sigma) = Z_3(T_3^2(\sigma),\ T_3^1(\sigma),\ T_3^0(\sigma))$$

Basic properties

- Take and zip are each other's *inverse*, as follows:

$$Z_k(\,T_k^0(\sigma),\,\ldots,\,T_k^{k-1}(\sigma)\,) = \;\sigma$$

$$T_l^i(Z_l(\sigma_0,\ldots,\sigma_{l-1})) = \;\sigma_i$$

- Also,

$$T_l^i(\sigma) = \;Z_k(\,T_{k\times l}^i(\sigma),\,T_{k\times l}^{l+i}(\sigma),\,\ldots,\,T_{k\times l}^{(k-1)\times l+i}(\sigma)\,)$$

- One can do elementary equational reasoning using these (and similar) laws.

3. Rational streams

- are finitely representable, for instance by: stream circuits, weighted automata, vector spaces of finite dimension.
- cf. rational *languages*.

Stream calculus (on $\mathbb{R}^\omega$)

We use the following constants and operators:

- constants (one for each $r \in \mathbb{R}$): $[r] = (r, 0, 0, 0, \dots)$
- constant (cf. formal variable): $X = (0, 1, 0, 0, 0, \dots)$
- sum: $(\sigma + \tau)(n) = \sigma(n) + \tau(n)$
- convolution (aka Cauchy) product:

$$(\sigma \times \tau)(n) = \sigma(0) \cdot \tau(n) + \cdots + \sigma(n) \cdot \tau(0)$$

- (formal) inverse to product: if $\sigma(0) \neq 0$ then $\exists! \; \frac{1}{\sigma}$ s.t.

$$\sigma \times \frac{1}{\sigma} = (1, 0, 0, 0, \dots)$$

Stream calculus (on $\mathbb{R}^\omega$)

- Note:

$$X \times \sigma = (0, \sigma(0), \sigma(1), \sigma(2), \ldots)$$

- Convention:

$$3 \times X^2 = [3] \times X \times X \quad (= (0, 0, 3, 0, 0, 0, \ldots))$$

- *Polynomial* streams: for instance,

$$2 + 3X - 7X^4 \quad (= (2, 3, 0, 0, -7, 0, 0, 0, \ldots))$$

- *Rational* streams: for instance,

$$\frac{1 + X}{1 - 2X + X^2} \quad (= (1, 3, 5, 7, \ldots))$$

Take and *zip* preserve rationality

An elementary proof is based on the following facts (cf. [BR88]):

- For all $k \geq 1$,

$$Z_k(\sigma_0, \ldots, \sigma_{k-1}) = \sigma_0(X^k) + (X \times \sigma_1(X^k)) + \cdots + (X^{k-1} \times \sigma_{k-1}(X^k))$$

- T_l^i is linear: for all $r, s \in \mathbb{R}$, $\sigma, \tau \in \mathbb{R}^\omega$,

$$T_l^i(\,(s \times \sigma) + (t \times \tau)\,) = (s \times T_l^i(\sigma)\,) + (t \times T_l^i(\tau)\,)$$

- For $1 \leq i \leq l$ and $\sigma \in \mathbb{R}^\omega$,

$$T_l^i(X \times \sigma) = T_l^{i-1}(\sigma) \qquad T_l^0(X \times \sigma) = X \times T_l^{l-1}(\sigma)$$

Take and *zip* preserve rationality

For instance, for

$$\sigma = \frac{1}{(1-X)^2} = (1,2,3,\ldots)$$

we have:

$$T_3^0(\sigma) = \frac{1+2X}{(1-X)^2} \qquad T_3^1(\sigma) = \frac{2+X}{(1-X)^2} \qquad T_3^2(\sigma) = \frac{3}{(1-X)^2}$$

Take and *zip* preserve rationality

For instance, for

$$\sigma = \frac{1}{(1-X)^2} = (1, 2, 3, \ldots)$$

we have:

$$\begin{aligned} Rev_3(\sigma) &= Z_3(T_3^2(\sigma),\ T_3^1(\sigma),\ T_3^0(\sigma)) \\ &= Z_3\left(\frac{3}{(1-X)^2}, \frac{2+X}{(1-X)^2}, \frac{1+2X}{(1-X)^2}\right) \\ &= \frac{3}{(1-X^3)^2} + X \times \frac{2+X^3}{(1-X^3)^2} + X^2 \times \frac{1+2X^3}{(1-X^3)^2} \\ &= \frac{3-X-X^2+2X^3}{(1-X)^2(1+X+X^2)} \end{aligned}$$

4. Stream circuits

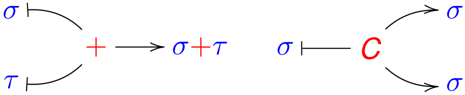
- Finite stream circuits built from adders and registers correspond to rational streams.
- Here we want to add *take* and *zip* gates and see what happens.

Behaviour of stream circuits

- Four basic types of gates:

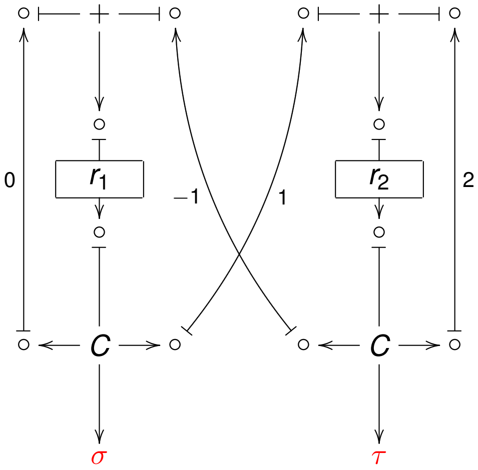
r-multiplier: $\sigma \stackrel{r}{\longmapsto} r \times \sigma$

register: $\sigma \longmapsto \boxed{r} \longrightarrow r + (X \times \sigma) \qquad (= (r, \sigma(0), \sigma(1), \ldots))$

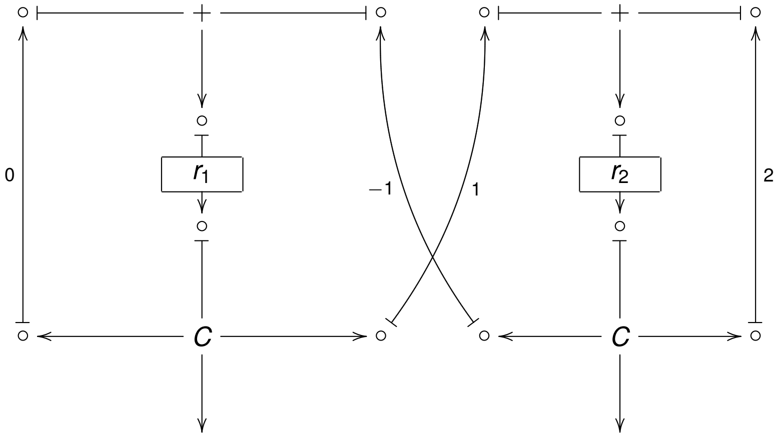


adder copier

Computing streams σ and τ



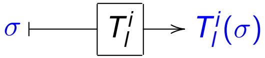
Two rational streams:



$\sigma = \dfrac{r_1 \times (1 - 2X) - (r_2 \times X)}{(1-X)^2}$ $\quad$ $\tau = \dfrac{(r_1 \times X) + r_2}{(1-X)^2}$

Extended stream circuits

- A new type of gate for each *take* operator:



- A new type of gate for each *zip* operator: