Additive cellular automata monadically

Silvio Capobianco   Tarmo Uustalu

Department of Software Science, Tallinn University of Technology

6-8 May 2022

Overview

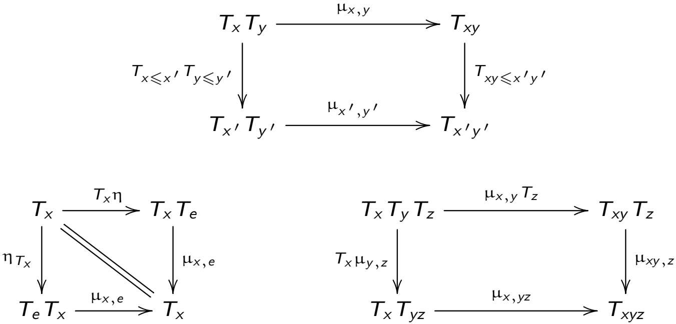
- Cellular automata (CA) are models of parallel computation on regular grids of state machines where a local update rule, whose output depends on the current state of a neighborhood, is applied synchronously at every node.
- This process determines a global transition function of the space of configurations of the entire grid.
- CA are a *comonadic* notion of computation, where local rules are coKleisli maps of a comonad, and this comonad is graded.
- If the set of states is a commutative monoid, then a notion of additive CA arises naturally.
- Local rules of additive CA, in addition, are the Kleisli maps of a *graded monad*.

Graded monads

Let $M = (M, \leqslant, e, \cdot)$ be a partially ordered monoid.
An *$\mathcal{M}$-graded monad* on a category $\mathcal{C}$ is

- for every $x \in M$, an endofunctors T_x on $\mathcal{C}$, together with
- for every $x \leqslant y$, a natural transformation $T_{x \leqslant y} : T_x \rightarrow T_y$
- a natural transformation $\eta : \mathrm{Id}_{\mathcal{C}} \rightarrow T_e$,
- a family of natural transformations $\mu_{x,y} : T_x T_y \rightarrow T_{xy}$

such that $T_{x \leqslant x} = \mathrm{id}_{Tx}$, $T_{y \leqslant z} \circ T_{x \leqslant y} = T_{x \leqslant z}$, and the following diagrams commute:



The Kleisli category of a graded monad

Let T be an M-graded monad on a category $\mathcal{C}$.

1. Objects of $\mathrm{Kl}(T)$ are pairs (x, A) with $x \in M$ and $A \in |\mathcal{C}|$.
2. A morphism from (x, A) to (y, B) in $\mathrm{Kl}(T)$ is a pair (z, f) such that $xz \leqslant y$ and $f : A \to T_z B$, up to equivalence given by:

$$\frac{z \leqslant z'}{(z, xz \leqslant xz' \leqslant y, f) \sim (z', xz' \leqslant y, T_{z \leqslant z'} f)}$$

Configurations and translations

Let $G = (G, 1_G, \cdot)$ be a monoid (of nodes in a grid).

- A *configuration* over G with a set of states A is a function $c : G \to A$.
- The *translation*, or *shift*, is the right action G on A^G defined by:

$$c \triangleright g = \lambda(h : G).c(gh) \text{ for every } g \in G$$

The set of states can be given one of the following:

1. the *discrete topology* made of all the subsets of A;
2. the *discrete uniformity* made of all the supersets of $\Delta_A = \{(a, a) \mid a \in A\}$.

In this case, A^G is assumed to be *(uniformly) prodiscrete* (product of (uniformly) discrete).

Cellular automata

A *cellular automaton* (CA) over a monoid G is a quadruple $\langle A, B, \mathcal{N}, f \rangle$ where:

- the *source alphabet* A and the *target alphabet* B are uniformly discrete spaces;
- the *neighborhood index* $\mathcal{N}$ is a finite subset of G;
- $f : A^{\mathcal{N}} \to B$ is a *local update rule*.

Synchronous application of the local update rule induces a *global transition function* from A^G to B^G:

$$F : A^G \to B^G \,, \quad F(c) = \lambda(g : G).f(\,(c \triangleright g)|_{\mathcal{N}})$$

The monoid G is considered (uniformly) discrete, which makes it an exponentiable object.

Curtis-Lyndon-Hedlund theorem

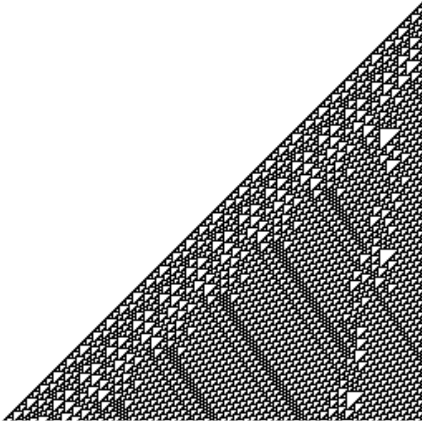
The global functions of CA with source alphabet A and target alphabet B are all and only the *uniformly continuous* functions from A^G to B^G which commute with shifts.

Example: Rule 110

This cellular automaton has $G = \mathbb{Z}$, $A = B = \{0, 1\}$, $\mathcal{N} = \{-1, 0, 1\}$, and local rule:

$$f(xyz) = \text{if } x = 1 \text{ then } y + z \text{ else } y \vee z$$

Here is its evolution starting from a single cell at 1. (Source: Wikipedia.)



Local updates as coKleisli maps

In (C. and U. 2010) we define the category LocB_G of *local behaviors* on G as follows:

- Objects are uniform spaces.
- A morphism from A to B in LocB_G is a morphism from A^G to B in $\mathbf{Unif}$.
- The identity of $A \in |\mathrm{LocB}_G|$ is $j_A = \lambda(c : A^G).c(1_G)$.
- To each $f \in \mathbf{Unif}(A^G, B)$ corresponds a unique $f^\dagger \in \mathbf{Unif}(A^G, B^G)$ defined by:

$$f^\dagger(c) ::= \lambda(g : G).f(c \triangleright g) \text{ for every } c \in A^G$$

- Composition in $\mathrm{LocB}(G)$ is the defined as $h \bullet f = h \circ f^\dagger$.

In other words, LocB_G is the *coKleisli category* of the comonad (D, ε, δ) on $\mathbf{Unif}$ defined by:

- $DA = A^G$ for every $A \in |\mathbf{Unif}|$.
- $Df(c) = f \circ c$ for every $f \in \mathbf{Unif}(A, B)$ and $c \in A^G$.
- $\varepsilon_A(c) = c(1_G)$ for every $A \in |\mathbf{Unif}|$.
- $\delta_A(c) = \lambda(g : G).c \triangleright g = \lambda(g : G).\lambda(h : G).c(gh)$ for every $A \in |\mathbf{Unif}|$.

Note that we lose *information on the neighborhood*.
This doesn't matter as long as we *know* that there is one.

Update of configurations

Let $\mathcal{A} = \langle A, B, \mathcal{N}, f \rangle$ be a CA on G with global function F.

- To update c on g, $\mathcal{A}$ must know the states of all the neighbors of g:
- That is, it must know the value of c on $g\mathcal{N}$.
- The state of c outside of $g\mathcal{N}$ has no influence on the state of $F(c)$ on g.

Composition of local rules

Let $\mathcal{A} = \langle A, B, \mathcal{N}, f \rangle$ and $\mathcal{B} = \langle B, C, \mathcal{M}, h \rangle$ be CA on G with global functions F and H, respectively.

- What information do we need to update $c(g)$ by $H \circ F$?
- To update $F(c)$ at g, H must know its value on:

$$g\mathcal{M} = \{gm \mid m \in \mathcal{M}\}$$

- But to update c at gn, F must know its value on:

$$gm\mathcal{N} = \{gmn \mid n \in \mathcal{N}\}$$

- Then, to update c at g, $H \circ F$ must know its value *on $g\mathcal{M}\mathcal{N}$*.

So there's a grading too

We observed that if $f : A^G \to B$ is defined by the neighborhood $\mathcal{N}$ and $h : B^G \to C$ is defined by the neighborhood $\mathcal{M}$, then $g \bullet f$ is defined by the neighborhood $\mathcal{M}\mathcal{N}$.
This gives a structure of *graded comonad* on $\mathbf{Set}$ as follows:

- The grading pomonoid is the set $\mathcal{P}_{\mathrm{fin}}(G)$ of finite subsets of G with set inclusion as the partial order and the *Frobenius product* as the multiplication:

$$\mathcal{M}\mathcal{N} = \{mn \mid m \in \mathcal{M}, n \in \mathcal{N}\}$$

- The object component becomes $D_{\mathcal{N}} A = A^{\mathcal{N}}$ instead of A^G.
- The morphism component doesn't change.
- The new counit $\varepsilon_A : D_{\{1_G\}} A \to A$ is:

$$\varepsilon_A(c : \{1_G\}) = c(1_G)$$

- The new comultiplication $\delta_{\mathcal{M},\mathcal{N},A} : D_{\mathcal{M}\mathcal{N}} A \to D_{\mathcal{M}} D_{\mathcal{N}} A$ is:

$$\delta_{\mathcal{M},\mathcal{N},A}(c : A^{\mathcal{M}\mathcal{N}}) = \lambda(m : \mathcal{M}).\lambda(n : \mathcal{N}).c(mn)$$

- CA local rules are still coKleisli maps.

Additive cellular automata

Let $A = (A, 0_A, +_A)$ and $(B, 0_B, +_B)$ be *commutative* monoids and let $G = (G, 1_G, \cdot_G)$ be a monoid.

- The sets A^G and B^G with the pointwise operation:

$$(c + d)(g) = c(g) + d(g)$$

  are commutative monoids too.

- We then say that a cellular automaton $\mathcal{A} = \langle A, B, \mathcal{N}, f \rangle$ is *additive* if its global function $F : A^G \to B^G$ is a homomorphism of commutative monoids:

$$F(c + d) = F(c) + F(d) \text{ for every } c, d \in A^G$$

- Then the behavior of the CA is entirely determined by its behavior on point configurations of the form

$$c_{n,a} = \lambda(g : G).\text{if } g = n \text{ then } a \text{ else } 0 \quad \text{for } n \in \mathcal{N}, a \in A$$

- By the *superposition principle*, this is *equivalent* to saying that there exist a family of morphisms $\phi_n : A \to B$ parameterized by $n \in \mathcal{N}$ such that:

$$f = \lambda(c : A^{\mathcal{N}}).\bigoplus_{n \in \mathcal{N}} \phi_n(c(n))$$

Formal series

Let A be a commutative monoid.

- A *formal series* over G with *coefficients* in A is an expression of the form:

$$\mathcal{S} = \sum_{g \in G} \langle a_g \mid g \rangle \quad \text{with } a_g \in A \text{ for every } g \in G$$

- The space $A[\![G]\!]$ of formal series over G with coefficients in A is a commutative monoid with respect to pointwise addition.
- The *support* of $\mathcal{S}$ is the set $\operatorname{supp} \mathcal{S} = \{g \in G \mid a_g \neq 0\}$.
- $\mathcal{S}$ is a *formal polynomial* if $|\operatorname{supp} \mathcal{S}| < \infty$.
  The monoid of formal polynomials is denoted as $A[G]$.
- A configuration $c \in A^G$ can be identified with the formal series:

$$\mathcal{S}_c = \sum_{g \in G} \langle c(g) \mid g \rangle$$

Formal polynomials as a graded monad?

Our intuition is that the following is a $\mathcal{P}_{\mathrm{fin}}(G)$-graded monad on the category $\mathbf{CommMon}$ of commutative monoids and their homomorphisms:

- $T_{\mathcal{N}} A = A[\mathcal{N}] ::= \{\mathcal{S} \in A[G] \mid \operatorname{supp} \mathcal{S} \subseteq \mathcal{N}\}$.
- $T_{\mathcal{N}} f(\mathcal{S}) = \sum_{g \in G} \langle f(\mathcal{S}(g)) \mid g \rangle$ for every $f : A \to B$ and $\mathcal{S} \in A[\mathcal{N}]$.
- $\eta_A(a) = \langle a \mid 1_G \rangle$
- $\mu_{\mathcal{M},\mathcal{N},A} \left(\sum_g \langle \sum_h \langle a_{g,h} \mid h \rangle \mid g \rangle \right) = \sum_p \left\langle \bigoplus_{gh=p} a_{g,h} \mid p \right\rangle$

DOES THIS WORK?

A matter of support

We want $\mu_{\mathcal{M},\mathcal{N}} : T_{\mathcal{M}} T_{\mathcal{N}} \to T_{\mathcal{M}\mathcal{N}}$.

- An element of $T_{\mathcal{N}} A$ is a formal series with coefficients in A and support contained in $\mathcal{N}$.
- Then an element of $T_{\mathcal{M}} T_{\mathcal{N}} A$ is a formal series such that:
  1. its coefficients are formal series $\mathcal{S}_g$ with coefficients in A;
  2. each of those series has support contained in $\mathcal{N}$; and
  3. every series $\mathcal{S}_g$ such that $g \notin \mathcal{M}$ is identically zero.
- That is, such an object has the form:

$$\mathcal{S} = \sum_{g \in G} \left\langle \underbrace{\sum_{h \in G} \langle a_{g,h} \mid h \rangle}_{\mathcal{S}_g} \mid g \right\rangle$$

  where:
  1. $a_{g,h} = 0$ for every $g \notin \mathcal{M}$, and
  2. $a_{g,h} = 0$ for every $g \in G$ and every $h \notin \mathcal{N}$,

  that is,
  - $a_{g,h} = 0$ for every $(g, h) \notin \mathcal{M} \times \mathcal{N}$.

Formal polynomials as a graded monad

Let $\mathcal{S}$ be like in the previous slide.

- Consider the series $\sum_{p \in G} \left\langle \bigoplus_{gh=p} a_{g,h} \mid p \right\rangle$.
- The only things that might make $\bigoplus_{gh=p} a_{g,h} \neq 0$ are the pairs $(g, h) \in \mathcal{M} \times \mathcal{N}$ such that $gh = p$ and also $a_{g,h} \neq 0$.
- This never happens if $p \notin \mathcal{M}\mathcal{N}$.

Then indeed the multiplication $\mu_{\mathcal{M},\mathcal{N},A} : T_{\mathcal{M}} T_{\mathcal{N}} A \to T_{\mathcal{M}\mathcal{N}} A$ must be:

$$\mu_{\mathcal{M},\mathcal{N},A} \left(\sum_g \left\langle \sum_h \langle \mathcal{S}_g(h) \mid h \rangle \mid g \right\rangle \right) = \sum_p \left\langle \bigoplus_{gh=p} \mathcal{S}_g(h) \mid p \right\rangle \text{ for every } \mathcal{S} \in T_{\mathcal{M}} T_{\mathcal{N}} A$$

Additive CA as Kleisli morphisms?

How can additive CA enter all this? We must be a bit careful:

- Let $\mathcal{A} = \langle A, B, \mathcal{N}, f \rangle$ be a CA on G with global function F.
- The value of c at g influences the value of $F(c)$ at h if and only if g is a neighbor of h.
- But this means $g \in h\mathcal{N}$, that is, $h \in g\mathcal{N}^{-1}$:
  Here we need that G be a *group*, not just a monoid.

A matter of direction

Our intuition is that:

a CA local rule with neighborhood $\mathcal{N} \in \mathcal{P}_{\text{fin}}(G)$, i.e., a coKleisli map

$$f : D_{\mathcal{N}} A \to B$$

corresponds to a Kleisli map with grade $\mathcal{N}^{-1} \in \mathcal{P}_{\text{fin}}(G^{\text{rev}})$

$$\kappa : A \to T_{\mathcal{N}^{-1}} B$$

CoKleisli as Kleisli and vice versa

Given an additive CA local rule for a neighbourhood $\mathcal{N} \in \mathcal{P}_{\mathrm{fin}}(G)$, i.e., a coKleisli map

$$f : A^{\mathcal{N}} \to B$$

we know that f is of the form

$$f(c) = \bigoplus_{n \in \mathcal{N}} \phi_n(c(n))$$

The corresponding Kleisli map has $\mathcal{N}^{-1} \in \mathcal{P}_{\mathrm{fin}}(G^{\mathrm{rev}})$ as the grade:

$$\begin{gathered} \kappa : A \to B\left[\mathcal{N}^{-1}\right] \\ \kappa(a) = \sum_{n \in \mathcal{N}^{-1}} \langle \phi_{n^{-1}}(a) \mid n \rangle \end{gathered}$$

Underlying this is that D is a left adjoint graded comonad to T.

An example

Let $A = B = (\{0,1\}, \vee, 0)$, $G = \mathbb{Z}$, $\mathcal{N} = \{-1, 2\}$.

We can have a local update rule with neighbourhood $\mathcal{N}$, i.e., a coKleisli map for D as follows:

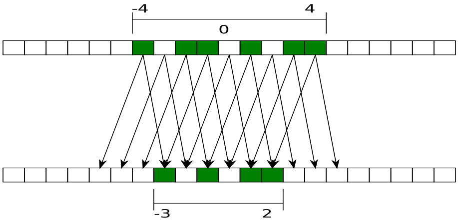
$$f : A^{\{-1,2\}} \to B\,, \quad f(c) = c(-1) \vee c(2)$$

The Kleisli map for T, with grade $\mathcal{N}^{-1}$ is:

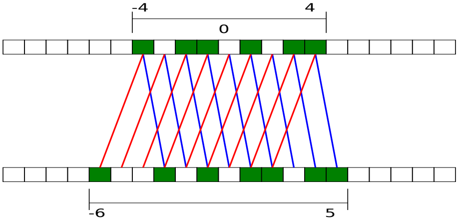
$$\kappa : A \to B\,[\{+1, -2\}]\ , \quad \kappa(b) = \sum_{\nu \in \{1,-2\}} \langle b \mid \nu \rangle$$

A comparison

The coKleisli map is about what is *needed* for the update:



The Kleisli map is about what *might* cause a perturbation:



Conclusion

While general CA can only be analyzed as a comonadic notion of computation, additive CA admit both a comonadic and a monadic analysis.

There is further structure: the comonad and monad are monoidal, etc.

Connections to formal language theory for G a free monoid and A, B semirings. Configurations/formal series are then weighted languages, patterns/formal polynomials are finite weighted languages. Applying a CA global rule is taking a residual of a weighted language.