me
Kleene Monads in a Short While

Sergey Goncharov
FAU Erlangen-Nürnberg

October 24, 2023, Oberseminar Theoretische Informatik
Kleene iteration is iteration of regular expressions, e.g.

\[(0 + 1)^* ; 0 ; (0 + 1)^*\]

Kleene algebra is a lightweight equational theory of (Kleene) iteration, complete over formal languages

Extremely popular, has lots of extensions: hybrid, concurrent, stateful, etc

Kleene monads is a simple categorification of Kleene algebras

Elgot monads: deterministic, very general, highly compositional

But not quite that popular 😞

So, can we combine structure (Elgot) and power (Kleene)?

Previously: Unifying Elgot iteration and while-loops

Here: Categorical notion of iteration with nondeterminism
A **Kleene algebra** is a structure \((S, 0, 1, +, ;, (-))^*\), where

- \((S, 0, 1, +, ;)\) is an idempotent semiring:
  - \((S, 0, +)\) is a **commutative** \((x + y = y + x)\) and **idempotent** \((x + x = x)\) monoid
  - \((S, 1, ;)\) is a monoid
  - **Distributive laws**:
    
    \[
    x ; (y + z) = x ; y + x ; z \\
    (x + y) ; z = x ; z + y ; z
    \]

    (thus, \(S\) is partially ordered: \(x \leq y\) iff \(x + y = y\))

- Kleene iteration satisfies \(x^* = 1 + x ; x^*\), and

\[
\begin{align*}
    x ; y + z & \leq y \\
    x^* ; z & \leq y
\end{align*}
\]

\[
\begin{align*}
    x + z ; y & \leq z \\
    x ; y^* & \leq z
\end{align*}
\]

Equivalently: \(x^* ; z\) is a least fixpoint of \(x ; (-) + z\) and \(z ; y^*\) is a least fixpoint of \((-) ; y + z\)

**Intuition:** 0 is a deadlock, 1 is a neutral program, ; is sequential composition, + is non-deterministic choice
Regular expressions

Algebraic language of finite state machines and beyond

Relational semantics of programs

Relational reasoning and verification, e.g. via dynamic logic

Plenty of extensions:
- modal $\Rightarrow$ modal Kleene algebra (Struth et al.)
- stateful $\Rightarrow$ KAT + B! (Grathwohl, Kozen, Mamouras)
- concurrent $\Rightarrow$ concurrent Kleene algebra (Hoare et al.)
- nominal $\Rightarrow$ nominal Kleene algebra (Kozen et al.)
- with differential equations $\Rightarrow$ differential dynamic logic (Platzer et al.)
- etc., etc., etc.

decidability and completeness properties (most famously w.r.t. formal languages and relational interpretations)
A minimalist extension is **Kleene algebra with tests (KAT)**, which adds

- another Kleene algebra $B$ of **tests**
- an operation-preserving inclusion $B \hookrightarrow S$
- complementation operator $\overline{(-)}$: $B \to B$, such that

$$\overline{a} + a = \top \quad \overline{\overline{a}} = a \quad \overline{a + b} = \overline{a} ; \overline{b} \quad \overline{0} = 1$$

(this makes $B$ into a **Boolean algebra**)

This enables encodings

- **Branching**  
  (if $b$ then $p$ else $q$) as $b ; p + \overline{b} ; q$
- **Looping**  
  (while $b$ do $p$) as $(b ; p)^* ; \overline{b}$
- **Hoare triples**  
  $\{a\} p \{b\}$ as $a ; p ; b = a ; p$

In particular, we can embed **deterministic** semantics to **non-deterministic** semantics
Kleene-Kozen Categories

A category is a Kleene-Kozen category if it has operations

- $0 : \text{Hom}(A, B)$
- $+ : \text{Hom}(A, B) \times \text{Hom}(A, B) \to \text{Hom}(A, B)$
- $(-)^* : \text{Hom}(A, A) \to \text{Hom}(A, A)$

that together with identity $(1)$ and composition $(;)$ satisfy Kleene algebra laws

Fact 1

Kleene algebra is Kleene-Kozen category on one object

Fact 2\(^1\)

Alternative axiomatization: laws of idempotent semirings, plus

\[
1^* = 1 \quad f^* = 1 + f \; f^* \quad (f + g)^* = f^* ; (g ; f^*)^* \quad \frac{h ; f = g ; h}{h ; f^* = g^* ; h}
\]

A **Kleene monad** is a monad $T$, whose Kleisli category is a Kleene-Kozen category

**Example (Powerset)**

Powerset: $T = \mathcal{P}$, Kleisli morphisms $X \to \mathcal{P}Y = \text{relations}$

**Example (State Transformer)**

If $T$ is a Kleene monad, $(T(- \times S))^S$ yields a Kleene monad

**Example (Writer Transformer)**

If $T$ is a (strong) Kleene monad and $M$ is a monoid, $T(M \times \text{Id})$ yields a Kleene monad. E.g. $\mathcal{P}(A^* \times \text{Id})$, "formal language monad" is Kleene

😊 This is pretty much it
Definition (Elgot monad)

A (complete) **Elgot monad**\(^2\) in a category with binary coproducts (!) is a monad \(T\) equipped with an Elgot iteration operator

\[ (-)\dagger : \text{Hom}(X, T(Y \oplus X)) \to \text{Hom}(X, TY), \]

satisfying four laws: fixpoint, uniformly, naturality and codiagonal

- **Elgotness** is robust and stable under many monad transformers
  - \( T \mapsto T(M \times -) \) (writer)
  - \( T \mapsto T(- \oplus E) \) (exception)
  - \( T \mapsto (T(- \times S))^S \) (state)
  - \( T \mapsto \nu\gamma. T(- \oplus H\gamma) \) (resumption)

- **Laws** go back to Elgot\(^3\), except for uniformity

---


\(^3\)Elgot, “Monadic Computation And Iterative Algebraic Theories”, 1975.
**Fixpoint** \((f : X \rightarrow T(Y \oplus X)):\)

\[
\begin{array}{c}
\begin{array}{c}
\xrightarrow{x}
\end{array}
\begin{array}{c}
f \quad x
\end{array}
\begin{array}{c}
\quad y
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\xrightarrow{x}
\end{array}
\begin{array}{c}
f 
\end{array}
\begin{array}{c}
\quad y
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\xrightarrow{x}
\end{array}
\begin{array}{c}
f 
\end{array}
\begin{array}{c}
\quad y
\end{array}
\end{array}
\]

**Uniformity** \((f : X \rightarrow T(Z \oplus X), g : Y \rightarrow T(Z \oplus Y), h : X \rightarrow Y):\)

\[
\begin{array}{c}
\begin{array}{c}
\xrightarrow{x}
\end{array}
\begin{array}{c}
\eta h 
\quad z
\end{array}
\begin{array}{c}
g 
\quad y
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\xrightarrow{x}
\end{array}
\begin{array}{c}
\eta h 
\quad z
\end{array}
\begin{array}{c}
g 
\quad y
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\xrightarrow{x}
\end{array}
\begin{array}{c}
\eta h 
\quad z
\end{array}
\begin{array}{c}
g 
\quad y
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\xrightarrow{x}
\end{array}
\begin{array}{c}
f 
\quad y
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\xrightarrow{x}
\end{array}
\begin{array}{c}
\eta h 
\quad z
\end{array}
\begin{array}{c}
g 
\quad y
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\xrightarrow{x}
\end{array}
\begin{array}{c}
\eta h 
\quad z
\end{array}
\begin{array}{c}
g 
\quad y
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\xrightarrow{x}
\end{array}
\begin{array}{c}
f 
\quad y
\end{array}
\end{array}
\]
Naturality \((f : X \to T(Y \oplus X), g : Y \to TZ)\):

\[
\begin{array}{c}
\begin{array}{c}
\text{f} \quad x \\
\text{y} \\
z
\end{array}
\end{array}
\quad = 
\begin{array}{c}
\begin{array}{c}
f \\
\text{y} \\
\text{g} \\
\text{z}
\end{array}
\end{array}
\]

Codiagonal \((f : X \to T(Y \oplus (X \oplus X)))\):

\[
\begin{array}{c}
\begin{array}{c}
\text{f} \quad x \\
\text{y} \\
\text{z}
\end{array}
\end{array}
\quad = 
\begin{array}{c}
\begin{array}{c}
f \\
\text{y} \\
\end{array}
\end{array}
\]

Naturality and Codiagonal are basically coherence laws
Kleene Iteration as Elgot Iteration

Theorem

A category $C$ is a Kleene-Kozen category if

1. $C$ is enriched over bounded join-semilattices and strict join-preserving morphisms
2. $C$ supports Elgot iteration that satisfies fixpoint, naturality, codiagonal and strong uniformity:

$$
\begin{align*}
&\begin{array}{c}
&h \\
&\downarrow
\end{array}
\begin{array}{c}
g \\
\end{array}
\begin{array}{c}
y \\
z
\end{array}
= \\
\begin{array}{c}
f \\
\end{array}
\begin{array}{c}
h \\
\end{array}
\begin{array}{c}
y \\
z
\end{array}
\end{align*}
$$

where $h$ is strict: $h ; 0 = 0$
3. $T$ satisfies the law $(\text{inl} + \text{inr})^\dagger = 1$, equivalently $1^* = 1$
What we want:

- Single-category notion of iteration
- Robustness under monad transformers
- Nondeterminism
- $0 ; f = 0$
- $(f + g) ; h = f ; h + g ; h$

What we do not want:

- $f ; 0 = 0$ (because of $\text{raise } e_1 = \text{raise } e_1 ; 0 = \text{raise } e_2 ; 0 = \text{raise } e_2$)
- $f ; (g + h) = f ; g + f ; h$ (because of process algebra semantics)
- Coincidence of deadlock and divergence (think of $\mathcal{P}(\text{Id} + 1)$)
- $1^* = 1$ (again $\mathcal{P}(\text{Id} + 1)$, but there are others)
Problem with coproducts
  ▶ Can we make do without coproducts, sticking the the principle: “kleene-iteration-algebra” is a single-object “kleene-iteration-category”?

Problem with tests
  ▶ Instead of coproducts, Kleene algebra models control with tests. What are tests in our case?

Problem with uniformity
  ▶ Uniform w.r.t. what we need?
Picture of Cat
Call maps w.r.t. which we want to be uniform **tight**

- Smallest candidate (Elgot monads): maps that factor through monad unit
- Largest candidate (Kleene monads): all maps
- Unless we demand tight maps to be strict \((f; 0 = 0)\), we can easily show that non-trivial exception monads fail uniformity

So, tight maps must be somewhere between

- The class generated from coproduct injections by copairing
- The class of all strict maps

**Fact**

There is an Elgot monad \(T\), uniform w.r.t. strict morphisms, such that its exception monad transform is not
**Linear Maps**

Call a category non-deterministic if the hom-sets are semi-lattices under $(0, +)$, and

$$0; f = 0, \quad (g + h); f = g; f + h; f$$

In such a category, call $f$ linear if for all $g, h$

$$f; 0 = 0, \quad f; (g + h) = f; g + f; h$$

If there are binary coproducts, require coproduct injections to be linear

**Theorem**

Let $T$ be an Elgot monad with non-deterministic Kleisli category, and uniform w.r.t. linear maps.

- Every exception monad transform of $T$ is uniform w.r.t. linear maps
- If $T$ has the property $f; T! = 0 \implies f = 0$ then $\nu T(\ominus H \gamma)$ is uniform w.r.t. linear maps
To model control with coproducts, it suffices to use decisions, i.e. morphisms of type $X \to X \oplus X$, e.g.

$$d \in C(X, X \oplus X) \quad f, g \in C(X, Y)$$

$$\text{if } d \text{ then } f \text{ else } g = d ; [g, f] \in C(X, Y)$$

---

**Theorem**

Let $C$ be a non-deterministic category with binary coproducts; for every $X$, let $C^b(X) \subseteq C(X, X)$ be a Boolean algebra under $\cup$ and $+$

1. Maps $\diamond: C^b(X) \to C(X, X \oplus X)$, $\bowtie: C(X, X \oplus X) \to C^b(X)$:

$$\diamond b = \overline{b} ; \text{inl} + b ; \text{inr} \quad d \bowtie = d ; [0, 1]$$

form a retraction

2. Every $d$ in the image of $\diamond$ is linear

3. $(\text{if } e \text{ then } \text{inr} \text{ else } d) \bowtie = d \bowtie + e \bowtie$, $(\text{if } e \text{ then } d \text{ else } \text{inl}) \bowtie = e \bowtie ; d \bowtie$

4. For every $d$ in the image of $\diamond$, $(\text{if } d \text{ then } \text{inl} \text{ else } \text{inr}) \bowtie = \overline{d} \bowtie$
**Proposition**

Let

- $C$ be a non-deterministic category
- $D$ be a wide subcategory of $C$ with coproducts preserved by inclusion
- $C^b(X) \subseteq D(X, X)$ for all $X$
  - form Boolean algebras under $\sqcup$ and $+$
  - contain $[\text{inl}, 0]$ and $[0, \text{inr}]$ whenever $X = X_1 \sqcup X_2$.

Then $C$ supports uniform w.r.t. $D$ Elgot iteration iff it supports

$$b \in C^b(X) \quad f \in C(X, X) \quad \text{while } b \text{ do } f \in C(X, X)$$

such that

$$\text{while } b \text{ do } f = \text{if } b \text{ then } f \; (\text{while } b \text{ do } f) \; \text{else } 1$$
$$\text{while } (b \lor c) \text{ do } f = (\text{while } b \text{ do } f) ; \text{ while } c \text{ do } (f ; \text{ while } b \text{ do } f)$$

$$u ; \overline{b} = \overline{c} ; v \quad u ; b ; f = c ; g ; u \quad u ; \text{ while } b \text{ do } f = (\text{while } c \text{ do } g) ; v$$

$(u, v \in D)$
A triple \((\mathcal{C}, \mathcal{C}^\ell, \mathcal{C}^b)\) is **Kleene-iteration category with tests (KiCT)** if

- \(\mathcal{C}\) is a non-deterministic category
- \(\mathcal{C}^\ell\) is a wide subcategory of **tight** morphisms, which are all linear
- \(\mathcal{C}^b\) is a family \((\mathcal{C}^b(X) \subseteq \mathcal{C}^\ell(X, X))_{X \in |\mathcal{C}|}\) of **tests** where every \(\mathcal{C}^b(X)\) is a Boolean algebra under \(\land\) and \(+\)
- for every \(X \in |\mathcal{C}|\) there is \((-)^* : \mathcal{C}(X, X) \to \mathcal{C}(X, X)\) such that

  \[
  \begin{align*}
  (*\text{-Fix}) & \quad f^* = 1 + f \ ; \ f^* \\
  (*\text{-Or}) & \quad (f + g)^* = f^* \ ; \ (g \ ; \ f^*)^* \\
  (*\text{-Uni}) & \quad \frac{u \ ; \ f = g \ ; \ u}{u \ ; \ f^* = g^* \ ; \ u} \quad (u \in \mathcal{C}^\ell)
  \end{align*}
  \]

PLUS (!) an unidentified set of principles like \((*\text{-Uni})\)
Fact 1
A Kleene algebra with tests is precisely a single-object Kleene-iteration category with tests, such that all morphisms are linear and $1^* = 1$

Fact 2
Let $C, C^\ell$ and $C^b$ be as follows:
- $C$ is a non-deterministic category with binary coproducts (!)
- $C^\ell$ is a wide subcategory of $C$ with binary coproducts, consisting of linear morphisms only, such that the inclusion preserves coproducts
- For every $X \in |C|$, $C^b(X) \subseteq C^\ell(X, X)$
  - forms a Boolean algebra under $;$ and $+$
  - contains $[\text{inl}, 0]$ and $[0, \text{inr}]$ whenever $X = X_1 \oplus X_2$

Then $(C, C^\ell, C^b)$ is a KiCT iff $C$ supports Elgot iteration uniform w.r.t. $C^\ell$
Unfortunately, the coproduct problem is deeper

For example, we seem to require coproducts to prove identities like

\[ h^* = (h ; (h + 1))^* \]

Basically, because we can go 2d and instantiate \( f \) and \( g \) in

\[ (f + g)^* = f^* ; (g ; f^*)^* \]

with

An ultimate yardstick for a correct axiomatization could be a completeness result
TOWARDS COMPLETENESS
Kozen’s Completeness

- Kozen\(^4\) showed completeness of Kleene algebra w.r.t. regular events (=regular languages).
- In fact, he did more: he showed that for a fixed signature \(\Sigma\), regular languages \(\text{Reg}(\Sigma)\) is a free Kleene algebra on \(\Sigma\), hence

\[
[p] = [q] \implies p = q
\]

where \([p]\) is the language, generated by \(p\).

- Idea of the proof: given \([p] = [q]\), arrange a series of equivalent non-deterministic automata \(A_1, \ldots, A_n\), such that

\[
\begin{array}{c}
A_1 \\ ~ \\ \downarrow \\
p = p_1 \sim p_2 \sim p_3 \cdots p_n = q
\end{array}
\]

- Key technical step: for any Kleene algebra \(A\), matrices \(\text{Mtx}_n(A)\) of size \(n \times n\) over \(A\) again form a Kleene algebra.

Given a Kleene-Kozen category $\mathcal{C}$, let $\text{Mtx}(\mathcal{C})$ be as follows:

- Objects are non-empty lists $\langle A_1, \ldots, A_n \rangle$ of objects of $\mathcal{C}$
- A morphism $f : \langle A_1, \ldots, A_n \rangle \to \langle B_1, \ldots, B_m \rangle$ in $\text{Mtx}(\mathcal{C})$ is given by a family $\langle f_{i,j} : A_i \to B_j \rangle_{i \leq n, j \leq m}$ of morphisms in $\mathcal{C}$
- The identity morphism over $\langle A_1, \ldots, A_n \rangle$ is the family $\langle \delta_{i,j} \rangle_{i,j \leq n}$ where $\delta_{i,i} = 1$ and $\delta_{i,j} = 0$ if $i \neq j$
- Morphism composition as for matrices: given $f : \langle A_1, \ldots, A_n \rangle \to \langle B_1, \ldots, B_m \rangle$ and $g : \langle B_1, \ldots, B_m \rangle \to \langle C_1, \ldots, C_k \rangle$,

  $$(f ; g)_{i,j} = f_{i,1} ; g_{1,j} + \ldots + f_{i,m} ; g_{m,j}$$

**Fact**

$\text{Mtx}(\mathcal{C})$ is a Kleene-Kozen category with strict binary coproducts, and $\mathcal{C}$ fully embeds to it
Free Kleene-Kozen category with Coproducts

- Fix
  - Set of sorts $S$, and
  - Signature $\Sigma$ of symbols, together with types of the form $A \rightarrow B$

- These can be interpreted in any Kleene-Kozen category $D$: $[A] \in |D|$, $[f : A \rightarrow B] \in D([A], [B])$

- Free Kleene-Kozen category with coproducts $\text{Free}^+(S, \Sigma)$ over $(S, \Sigma)$ is characterized by the universal properly:

$$\text{Free}^+(S, \Sigma) \quad \xrightarrow{[-]} D \quad \text{Free}^+(S, \Sigma) \quad \xrightarrow{[-]} D$$
Objects of $\text{Free}^\oplus(S, \Sigma)$ are lists of sorts $\langle A_1, \ldots, A_n \rangle$ from $S$.

A morphism from $\langle A_1, \ldots, A_n \rangle$ to $\langle B_1, \ldots, B_m \rangle$ is a tuple $\langle t_i^{A_i, B_1, \ldots, B_m} \rangle_{i \leq n}$ of rational trees, specified with the grammar:

$$t^{A, B_1, \ldots, B_m} ::= \langle o \subseteq \{ i \in \mathbb{N} \mid A = B_i \}, \langle t^{C, B_1, \ldots, B_m} \rangle_f : A \rightarrow C \in S \rangle$$

where “rational” means:

- infinite
- number of subtrees are finite

Intuitively, a rational tree $t^{A, B_1, \ldots, B_m}$ represents the behaviour of a deterministic automaton with $m$ kinds of acceptance, with states classified by sorts, such that an action from $q_1$ of sort $A$ to $q_2$ of sort $B$ must be some $f : A \rightarrow B \in \Sigma$. 
**Constructing Free$^\oplus(S, \Sigma)$**

$$t^{A,B_1,\ldots,B_m} := \langle o \subseteq \{i \in \mathbb{N} \mid A = B_i \}, \langle t^{C,B_1,\ldots,B_m} \rangle_f : A \rightarrow C \in S \rangle$$

- Sum of rational trees is computed (co-)recursively, by joining the $o$'s
- $0$ is the tree with all the $o$’s empty
- $; \; \; \; \; \text{is a bit tricky: it is like substitution, combined with } +$
- $(-)^\dagger \; \; \; \; \text{is defined as a least fixpoint (iteration laws follow)}$
- To define $[-]^\#_D : \text{Free}^\#(S, \Sigma) \rightarrow D$ we need to fold trees into finite expressions and use uniformity to show that $[-]^\#_D$ is structure preserving
Define \( \text{Free}(S, \Sigma) \hookrightarrow \text{Free}^{\oplus}(S, \Sigma) \) as a full subcategory on \( S \)

Let \( D \) be any Kleene-Kozen category, which interprets \( (S, \Sigma) \) by \( [-]_D \)

The composition of \( [-]^\#_{\text{Mtx}(D)} \) with \( \text{Free}(S, \Sigma) \hookrightarrow \text{Free}^{\oplus}(S, \Sigma) \) factors through the inclusion of \( D \) to \( \text{Mtx}(D) \), yielding \( [-]^\#_D : \text{Free}(S, \Sigma) \to D \):
The general problem of organizing the totality of valid laws of iteration is solved by one of few known feats:

- by assuming linearity globally (Kozen)
- by relying on coproducts (Bloom and Ésik)
- “coalgebraic” approach\(^5\), relying on uniqueness of (some) fixpoints

The case of general Kleene-iteration categories with tests is open

---

Questions?


Definition (Monad)

A monad $T$ (on a category $C$) is given by a Kleisli triple $(T, 1, -^*)$ where

- $T : |C| \to |C|$
- $1$ is a family of morphisms $1_X : X \to TX$, forming monad unit
- $(-)^*$ assigns to each $f : X \to TY$ a morphism $f^* : TX \to TY$

satisfying the laws: $1^* = 1$, $f^* 1 = f$, $(f^* g)^* = f^* g^*$

This entails that

- $T$ is a functor, $1$ is a natural transformation
- $f : X \to TY$ and $g : Y \to TZ$ (regarded as programs) can be Kleisli composed to $f ; g = g^* f : X \to TZ$

By varying $T$ we obtain various ‘generalized programs’ $f : X \to TY$ while programs of the form $1f$ can be seen as ‘pure programs’

Example: $T = \text{powerset} \Rightarrow$ generalized programs = non-deterministic programs, pure programs = deterministic programs = functions