Lecture 12

Lecture by Dexter Kozen
Notes by Luke Serafin

March 5, 2024

Our goal for this lecture is to view automata in terms of coalgebra. This is a relatively recent viewpoint in that it was first taken in this millenium (more-or-less). A seminal paper by Rutten will be posted on the website. Silva's thesis really brought this theory to fruition; she is a colleague of Kozen.

We shall see a dual relationship between algebra and coalgebra, and this will take us to insights far beyond Kleene's theorem. There are many manifestations of this duality, including the relationship between operational and denotational semantics for programming languages, and between big-step and small-step semantics. Here is a table summarizing related notions in algebra and coalgebra.

Algebra	**Coalgebra**
Signature: Constructors used to build new terms	Signature: Destructors used to break apart coterms
Homomorphisms: Maps which preserve structure	Homomorphisms: Maps which preserve structure
Congruence relations, which are kernels of homomorphisms	Bisimulations, which are kernels of homomorphisms
Free/initial algebra, with a unique morphism to any other algebra with the same generators	Cofree/final coalgebra, with a unique incoming morphism from any coalgebra with the same generators
Equations These collapse elements via a congruence, and are a syntactic impediment to injectivity of the unique incoming morphism from a free algebra	Coequations, which we may think of as syntactic impediments to surjectivity of the cofree morphism
Terms, which are built from constructors	Coterms, which can be thought of as (possibly) infinite terms
Proof by induction, which uses least fixed points of set maps	Proof by coinduction, which uses greatest fixed points of set maps

Next we take a more general view of what an "algebra" is.

A Category-theoretic View of Algebrae

Let $\mathcal{C}$ be a category and $F : \mathcal{C} \to \mathcal{C}$ be a functor (for us usually $\mathcal{C} = \mathbf{Set}$, but for example one could consider topological algebrae by taking $\mathcal{C} = \mathbf{Top}$, the category of topological spaces).

Remark. Such a functor F is called an *endofunctor* of $\mathcal{C}$, in analogy to the terminology "endomorphism" for a morphism from an object to itself.

Definition 1. An *F-algebra* is a pair (X, α) where

- X is an object of $\mathcal{C}$
- α is a morphism $FX \to X$.

The morphism α is called the *structure map* of the algebra.

Now we can define a new category, which is the category of F-algebrae. The objects of this category are F-algebrae (X, α), and the morphisms $h:(X,\alpha) \to (Y,\beta)$ are morphisms of $\mathcal{C}$ such that the following diagram commutes:

$$\begin{tikzcd}
FX \arrow[r, "Fh"] \arrow[d, "\alpha"] & FY \arrow[d, "\beta"] \\
X \arrow[r, "h"] & Y
\end{tikzcd}$$

But what does this definition have to do with familiar notions of algebra? Let's consider the case of groups. The signature of groups is $(\cdot, ^{-1}, 1)$ with arities $(2,1,0)$. A group is a tuple $(G, \cdot, ^{-1}, 1)$ satisfying some universal equational axioms. For our endofunctor $F:\mathbf{Set} \to \mathbf{Set}$ take $FX = X^2 + X + 1$, where $X^2 = X \times X$ and $X + Y$ denotes disjoint union, which could be implemented as

$$\{(0,x) \mid x \in X\} \cup \{(1,y) \mid y \in Y\}.$$

Here we have intuitively "tagged" elements of $X + Y$ according to whether they came from X or from Y. Any tagging functions in_ℓ, in_r would do, in which case we have that the disjoint union is

$$\{\mathrm{in}_\ell(x) \mid x \in X\} \cup \{\mathrm{in}_r(y) \mid y \in Y\}.$$

All of these realizations of the disjoint union are isomorphic, provided we define the notion of a pair of "tagging functions" appropriately.

Cartesian product and disjoint union are realizations in **Set** of a more general, category-theoretic notion of product and coproduct. A *product* in a category $\mathcal{C}$ is an operation $(A,B) \mapsto A \times B$ such that there are projections[1] $\pi_1 : A \times B \to A$ and $\pi_2 : A \times B \to B$ such that for any object C and morphisms $f : C \to A$ and $g : C \to B$ there is a unique morphism (called a *mediating morphism*) $\langle f, g\rangle : C \to A \times B$ such that the following commutes:

$$\begin{tikzcd}
 & C \arrow[dl, "f"'] \arrow[d, "\langle f,g\rangle"] \arrow[dr, "g"] & \\
A & A \times B \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & B
\end{tikzcd}$$

A general coproduct is the dual of this.

$$\begin{tikzcd}
A \arrow[r, "\iota_1"] \arrow[dr, "f"] & A + B \arrow[d, "{[f,g]}"] & B \arrow[l, "\iota_2"'] \arrow[dl, "g"] \\
 & C &
\end{tikzcd}$$

Remark. The maps ι_1, ι_2 are called coprojections. Sometimes people call them injections, but they are not necessarily injective.

In "OCaml" one would write the mediating morphism $[f, g]$ as

```
[f,g](x) = match x with
              | inl(a) --> f(a)
              | inr(b) --> g(b).
```

Categories need not have products or coproducts. A partial order can be thought of as a category, with a unique arrow from x to y iff $x \le y$. In this case one can check that products are infima and coproducts are suprema, and for general partial orders neither of these need exist. For a category $\mathcal{C}$ with products and coproducts, $\times : \mathcal{C}^2 \to \mathcal{C}$ and $+ : \mathcal{C}^2 \to \mathcal{C}$.

[1] Here this means nothing more than "morphism;" the properties which make them projections are about to be stated.

Now we can define an abstract group[2] as an F-algebra (G, α) where $FX = X^2 + X + 1$ and $\alpha : FX \to X$. The map α represents $(\alpha_2, \alpha_1, \alpha_0)$ where

$$\begin{aligned} \alpha_2 &: X^2 \to X \\ \alpha_1 &: X \to X \\ \alpha_0 &: 1 \to X. \end{aligned}$$

(Here $1 = \{*\}$ is some 1-element set.)

What is a homomorphism of these abstract groups? Recall that a homomorphism $h : G_1 \to G_2$ of concrete groups is a set function such that for $x, y \in G$,

$$\begin{aligned} h(xy) &= h(x)h(y) \\ h(x^{-1}) &= h(x)^{-1} \\ h(1_{G_1}) &= 1_{G_2}. \end{aligned}$$

In the abstract case we have that the following diagram commutes:

$$\begin{tikzcd} X^2 + X + 1 \arrow[r, "Fh = h^2 + h + 1"] \arrow[d, "\alpha = {[\alpha_2, \alpha_1, \alpha_0]}"'] & Y^2 + Y + 1 \arrow[d, "\beta = {[\beta_2, \beta_1, \beta_0]}"] \\ X \arrow[r, "h"'] & Y. \end{tikzcd}$$

Here $h^2 : X^2 \to X$ is given by $h^2(x, y) = (h(x), h(y))$, $h^1 = h$, and $h^0 = 1 : \{*\} \to \{*\}$. An object in $X^2 + X + 1$ is of precisely one of the forms $(a, b) \in X^2$, $a \in X$, or $* \in \{*\}$. We shall check what commutativity in the case of (a, b) yields; the other cases are similar and easier. Consider $(a, b) \in X^2$. Going left and then down through the diagram, we have

$$(a, b) \mapsto (h(a), h(b)) \mapsto \beta_2(h(a), h(b)) = h(a)h(b).$$

And going down and then right leads us to

$$(a, b) \mapsto \alpha_2(a, b) = ab \mapsto h(ab).$$

By commutativity of the diagram $h(ab) = h(a)h(b)$, which is indeed one of the usual requirements for a homomorphism of groups. Consider it an exercise that commutativity of the diagram also induces $h(x^{-1}) = h(x)^{-1}$ and $h(1) = 1$.

One can have even more fun with algebrae, for example a monad structure on the underlying category leads to the notion of an Eilenberg-Moore algebra.

Coalgebra over a Category

Fix a category $\mathcal{C}$ and an endofunctor $F : \mathcal{C} \to \mathcal{C}$.

Definition 2. An *F-coalgebra* over $\mathcal{C}$ is a pair (X, γ) such that X is an object of $\mathcal{C}$ and $\gamma : X \to FX$ in $\mathcal{C}$.

Think of γ as destructing its input. F-coalgebrae are objects in a category F-**Coalg** with morphisms $h : (X, \gamma) \to (Y, \delta)$ which are morphisms of $\mathcal{C}$ such that the following commutes:

$$\begin{tikzcd} X \arrow[r, "h"] \arrow[d, "\gamma"'] & Y \arrow[d, "\delta"] \\ FX \arrow[r, "Fh"'] & FY \end{tikzcd}$$

[2] Actually, an abstract algebra with the signature of a group. The group axioms, such as associativity, would require additional diagrams to commute.

Examples

Streams (infinite strings) over an alphabet Σ. For example Σ^ω for Σ a finite alphabet, which is the set of infinite strings in the alphabet Σ. The coalgebra is $(\Sigma^\omega, \mathrm{hd}, \mathrm{tl})$, where $\mathrm{hd}\colon \Sigma^\omega \to \Sigma$ is given by

$$\mathrm{hd}(x_0, x_1, x_2, \ldots) = x_0$$

and $\mathrm{tl}\colon \Sigma^\omega \to \Sigma^\omega$ is given by

$$\mathrm{tl}(x_0, x_1, x_2, x_3, \ldots) = (x_1, x_2, x_3, \ldots).$$

These operations have an inverse $\mathrm{cons}\colon \Sigma \times \Sigma^\omega \to \Sigma^\omega$ defined by

$$\mathrm{cons}(x, (x_0, x_1, x_2, \ldots)) = (x, x_0, x_1, x_2, \ldots).$$

Note that $(\mathrm{hd}, \mathrm{tl})\colon \Sigma^\omega \to \Sigma \times \Sigma^\omega$.

In general, a coalgebra for the functor $FX = \Sigma \times X$ is an object (X, γ) with $\gamma = (\mathrm{obs}, \mathrm{cont})$ where $\mathrm{obs}\colon X \to \Sigma$ is called *observation* and $\mathrm{cont}\colon X \to X$ is called *continuation.* Think of X as a set of states, with information about a state encoded by a letter in the alphabet. Automata are like this. Observations say whether the current state is final or not, and continuations give the next letter.

Streams are the final coalgebra of this signature. For any other such coalgebra, there is a unique coalgebra homomorphism to streams, called the *behaviour* of the coalgebra. Intuitively, the behaviour of a coalgebra is everything observable about the coalgebra.

For (X, γ) a coalgebra with $\gamma = (\mathrm{obs}, \mathrm{cont})$,

$$(X, \mathrm{obs}, \mathrm{cont}) \xrightarrow{\theta} (\Sigma^\omega, \mathrm{hd}, \mathrm{tl})$$

is given by

$$\theta(x) = (\mathrm{obs}(x), \mathrm{obs}(\mathrm{cont}(x)), \mathrm{obs}(\mathrm{cont}^2(x)), \ldots).$$

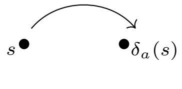
Tracing through the following diagram, one can check that this is a coalgebra homomorphism.

$$\begin{tikzcd}
X \arrow[r, "\theta"] \arrow[d, "\mathrm{obs}\\ \mathrm{cont}"'] & \Sigma^\omega \arrow[d, "\mathrm{hd}\\ \mathrm{tl}"] \\
\Sigma \times X \arrow[r, "\mathrm{id}_\Sigma \times \theta"'] & \Sigma \times \Sigma^\omega.
\end{tikzcd}$$

What are automata? We start with the deterministic case. Fix an alphabet Σ. An automaton is a coalgebra for the functor $\Phi X = 2 \times X^\Sigma$. Before we had $(Q, \Sigma, \delta, s, F)$, but now Σ is understood and we don't care about the start state. The structure $(Q, (\varepsilon, \delta))$ is a coalgebra, where $(\varepsilon, \delta)\colon Q \to \Phi Q = 2 \times Q^\Sigma$ is given by $\varepsilon\colon Q \to 2 = \{0, 1\}$, which encodes the final states via

$$\varepsilon(q) = \begin{cases} 1 & q \in F \\ 0 & q \notin F, \end{cases}$$

and $\delta\colon Q \to Q^\Sigma$, which we can think of as $\delta\colon Q \to (\Sigma \to Q)$, or equivalently as $\delta\colon \Sigma \to (Q \to Q)$, so for $a \in \Sigma$, $\delta_a\colon Q \to Q$.



This is a coalgebra for F.

There is a final coalgebra. The behaviour of a state is the set of strings which would be accepted, were that state the start state. This is $\{x \in \Sigma^* \mid \hat{\delta}(s, x) \in F\}$. The final coalgebra (of behaviours) is $(2^{\Sigma^*}, e, d)$ where $e\colon 2^{\Sigma^*} \to 2$ and $d_a\colon 2^{\Sigma^*} \to 2^{\Sigma^*}$ for $a \in \Sigma$ are defined as follows for $A \subseteq \Sigma^*$:

$$e(A) = \begin{cases} 1 & \epsilon \in A \\ 0 & \epsilon \notin A \end{cases}$$

$$d(A) = \{x \in \Sigma^* : ax \in A\}.$$

We claim that the following diagram commutes:

$$\begin{tikzcd}
Q \arrow[r, "\theta"] \arrow[d, "\delta_a"'] & \Sigma^* \arrow[d, "d_a"] \\
Q \arrow[r, "\theta"'] & \Sigma^*.
\end{tikzcd}$$

This says the map θ is a coalgebra homomorphism with respect to δ, and is verified by using the definition of $\hat{\delta}$ to compute, for $s \in Q$,

$$\begin{aligned}
\theta(\delta_a(s)) &= \{x \mid \hat{\delta}(\delta_a(s) = \delta(s,a), x) \in F\} \\
&= \{x \mid \hat{\delta}(s, ax) \in F\} \\
&= \{x \mid ax \in \theta(s)\} \\
&= d_a(\theta(s)).
\end{aligned}$$

For ε, we need $e(\theta(s)) = \varepsilon(s)$, which follows from

$$e(\theta(s)) = \begin{cases} 1 & \epsilon \in \theta(s) \\ 0 & \epsilon \notin \theta(s) \end{cases}$$

because ϵ is accepted at s if and only if s is a final state.

The value of all this is a very general, coalgebraic view of automata. We saw earlier that automata can be encoded as matrices in Kleene algebrae. Thus coalgebrae for Φ yield algebrae for a different functor corresponding to the signature $(+, \cdot, *, 0, 1)$ of Kleene algebrae. (Namely $GX = X^2 + X^2 + X + 1 + 1$.) Thus we can pass form an automaton coalgebra to an associated Kleene algebra. To go the other direction, we shall show that a free Kleene algebra yields an automaton/coalgebra $(\operatorname{Exp}\Sigma, E, D)$ such that the behaviour of a regular expression is the regular set it represents. Here D is called the Brzozwski derivative. The canonical interpretation $R_\Sigma : \operatorname{Exp}\Sigma \to 2^{\Sigma^*}$ will be the unique morphism into the coalgebra 2^{Σ^*}.