How to win a lottery?

Priyaa Varshinee Srinivasan
Topos Institute, Berkeley CA

Polynomial Functors Workshop 2024

Abstract

This article describes a specific coalgebra of Lottery monad. Lottery Monads are polynomial functors which can be used to describe stochastic processes. This article sketches how to use lottery monad to specify "true randomness" and how a system can access the randomness for its state transitions. The entire story turns out to be a $[\mathsf{Lott}, y]$-coalgebra where Lott is the Lottery monad.

1 Introduction

During the workshop, David introduced many cool stuff! One of them was Lottery monads which encodes all possible probability distributions over all sets as a polynomial functor. There is something neat about this monad that caught my attention, so I spent a significant portion of the research week working through this monad.

A Lottery monad is defined to be following functor:

$$\mathsf{Lott} := \sum_{N \in \mathbb{N}} \sum_{p \in \Delta_X} y^{N_p}$$

where, Δ_N is the set of all possible distributions over a set of cardinality N. That is, $\Delta_N := \{p : N \to [0,1] |$ for all x:X, $\sum_{p(n)} = 1\}$. N_p refers to a set of a N elements with probablity distribution p over them.

To understand the monadic structure over Lott, see, I recommend David's blog on Lottery monad. But for this article, we won't need it.

For definition of polynomial functors, polynomial functor maps and monoidal structures, see [2, 1].

2 Theoretically Stochastic State Machines

A stochastic state machine is a state machine which transitions its states stochastically. This section will describe such state machines using Lott.

Punchline first! A stochastic state machine is a natural transformation between polynomial functors:

$$\varphi : Sy^S \to \mathsf{Lott}$$

where $S : \mathsf{Set}$. Ok let's see why this punchline is true!

Consider the following stochastic state machine in Figure 1.

The machine has 3 states: A, B, and C. The machine transitions from one state to another with some probability.

Transtion table:

	A	B	C
A	0.7	0.2	0.1
B	0.4	0.6	
C	1		

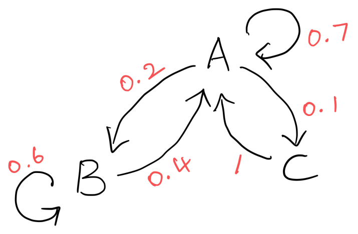

Figure 1: A stochastic state machine

The question is how to represent this state machine as a polynomial functor.

Firstly a state machine is a Moore machine given by a natural transformation of polynomial functors:

$$m : \{A,B,C\}y^{\{A,B,C\}} \longrightarrow y$$

The above map reads: " For each state $\{A,B,C\}$, give a new state from $\{A,B,C\}$. " Any state machine specified as above is deterministic. For each state, there is exactly one state to which the machine can transition to!

The question is how to specify a Moore machine which is non-deterministic, more specifically, stochastic? It is using the Lott monad. The stochastic state machine in Figure 1 is polynomial functors map:

$$\varphi : \{A,B,C\}y^{\{A,B,C\}} \longrightarrow \mathsf{Lott} \tag{1}$$

or

$$\varphi : \{A,B,C\}y^{\{A,B,C\}} \longrightarrow \sum_{N\in\mathbb{N}} \sum_{p\in\Delta_X} y^{N_p}$$

So, φ, in the forward direction is a function between positions. So in forward direction, φ picks a lottery for each state:

$$\varphi_1 = \varphi(1) : \{A,B,C\}y^{\{A,B,C\}}(1) \longrightarrow \sum_{N\in\mathbb{N}} \sum_{p\in\Delta_X} y^{N_p}(1)$$

We can now define this function using the data in Figure 1:

$$\begin{aligned}
\varphi(1) : \{A,B,C\} &\rightarrow \sum_{N:\mathbb{N}} \sum_{p:\Delta_N} 1 \\
A &\mapsto \{0.7, 0.2, 0.1\} \\
B &\mapsto \{0.4, 0.6\} \\
C &\mapsto \{1\}
\end{aligned}$$

φ in the forward direction is a (dependent) function of directions. Given a state $s \in \{A,B,C\}$,

$$\varphi_s^{\#} : \varphi_1(s) \longrightarrow \{A,B,C\}$$

Abuse of notation WARNING: In the above function, $\varphi_1(s)$ refers to set of tickets in lottery $\varphi_1(s)$.

For each state in $\{A,B,C\}$, for the lottery corresponding to that state, for each ticket in that lottery, the corresponding dependent functions a state to that ticket, thereby encoding probablistic transition of the state machine.

Let define all the dependent functions corresponding to 1.

$$\begin{aligned}
&\varphi_A : \{0.7, 0.2, 0.1\} \to \{A, B, C\}\\
&\quad 0.7 \mapsto A; \quad 0.2 \mapsto B; \quad 0.1 \mapsto C\\
&\varphi_B : \{0.4, 0.6\} \to \{A, B, C\}\\
&\quad 0.4 \mapsto A; \quad .6 \mapsto B\\
&\varphi_C : \{1\} \to \{A, B, C\}\\
&\quad \bullet \mapsto A
\end{aligned}$$

The machine φ in equation 1 is *theroetical.* That is, φ hows how to produce a lottery for each state, but "once a ticket is picked over that lottery" (this ticket needs to be picked probabilistically) φ know what the next state of the machine is. But, who chooses the ticket? Unless someone gives a ticket from the chosen lottery, the machine cannot transition to the next state.

The question is how to pick a ticket so that the probablities are *satisfied* as the number of state transitions of the machine approaches infinity?

3 A good lottery-ticket-chooser

Let us now design a machine[1] that can pick a "winning ticket" each time – the winning ticket means, given a lottery (N, p), choose a ticket from that lottery such that the ticket sequence will satisfy the probability distribution p as it length approaches infinity.

Such a machine needs to be a random number generator or choose ticket by some sort of pure randomness. Given a lottery (N, p), the machine produce the next winning ticket, that is a $n : N_p$. What winning means is determined by p.

So the define the states of the machine like this:

$$R := \{r \mid r : \mathsf{Lott}(1) \to \mathsf{GoodSeq}(N, p)\}$$

$$\mathsf{GoodSeq}(N, p) := \left\{\sum : \mathbb{N} \to N \mid \text{the sequence sum has maximal entropy by some criteria}^{2}\right\}$$

Think of each $r : R$ as a matrix. Each row of the matrix corresponds to a lottery (N, p) where $N \in N, p \in \Delta_N$. Each row is an infinite sequence of tickets. Each such sequence is a good sequence. So you can see that the column number approaches infinity.

Consider the polynomial,

$$Ry^R$$

In each position, this polynomial has knowledge of exactly one sequence of winning tickets for each lottery.

Lets consider a machine that for every lottery will give a ticket from that lottery. It is a global section:

$$\varphi : \mathsf{Lott} \to y$$

All possible such machines or all possible global sections of Lott are:

$$[\mathsf{Lott} \to y]$$

which is a polynomial. So, we have two polynomial:

1. A poynomial in each of its position has knowledge of good sequences for each lottery.
2. A polynomial in each position knows how to choose a ticket for each lottery.

[1] a polynomial functor map

The machine that will take the knowledge of the first polynomial of what the next winning ticket is and apply it to the second polynomial which chooses tickets is:

$$\theta : Ry^R \longrightarrow [\mathsf{Lott}, y]$$

which is same as

$$\theta : Ry^R \longrightarrow \sum_{t:Lott \longrightarrow y} y^{\mathsf{Lott(1)}}$$

Forward direction - On positions: $\theta(1)$ which will take a $r : R$ and map it to the ticket chooser (global section) which will give the first winning ticket for each lottery. That is, $\theta_1(r) = r[0]$

Backward direction - On directions: For each $r : R$, for each lottery (N, p), the new random number generator $r' : R$ is same as r except at the row (N, p), $r'[N, p][i] = r[N, p][i + 1]$, that is, the tickets are left shifted by one. Imagine the winning ticket of that lottery has been torn off from the sequence. The rest of the sequences remaining ticket since that lottery has not been chosen.

So this machine, gives a winning ticket for the lottery that the user gives (while moving from one state to another). This can go on forever.

So lets give a lottery to this machine!

Summarizing the machines we have so far:

1. Theoretical stochastic state machine: We have a machine which picks a lottery, and once given a ticket from that lottery, moves to another state.
2. Winning-ticket chooser: We have a machine, which when given a lottery picks the next winning ticket!

4 Practically Stochastic State Machine

$$\varphi : Sy^S \longrightarrow \mathsf{Lott}$$
$$\mathsf{Roulette} : Ry^R \longrightarrow [\mathsf{Lott}, y]$$

Then, tensoring them and composing with eval gives

$$SRy^{SR} \xrightarrow{\varphi \otimes \mathsf{Roulette}} \mathsf{Lott} \otimes [\mathsf{Lott}, y] \xrightarrow{\mathsf{eval}} y$$

Read the composite map as follows: For each state and a random number generator pair, a 'next state and the random number generator' pair.

Sequence of states is a good random sequence based on the lottery chosen by φ for the initial state of the machine

And, that is how one wins a Lottery!!

Things to do: Why is this machine good, in other words, universal?

References

[1] Nelson Niu and David I. Spivak. Polynomial functors: A general theory of interaction.

[2] David I. Spivak. A reference for categorical structures on **Poly**, 2024.