UNIVERSITY OF HERTFORDSHIRE

School of Computer Science

Final Report

---

# Tiny Bundles of Knowledge

Minimalistic machines for understanding and making predictions

---

*Author:*
A. D. ROBU

*Supervisor:*
Prof. Daniel POLANI

April 2015

Modular BSc Honours in Computer Science

6COM0282 — Computer Science Project

**Abstract**

The motivation behind this report is the intention to show that machines are a concrete form of knowledge. It extends further by presenting the observations that our operational definition of information does not address the varying demands of agents of relevant information. I derive theorems on the behaviour of various machines and use computer experiments to study knowledge, understanding and relevant information.

*We choose to do these things, not because they are easy, but because they are hard.*

-John F. Kennedy

# Acknowledgements

I would like to praise my supervisor, Daniel Polani for his constant stream of ideas and thank him for making this project a delightful experience for me.

I would also like to thank Daniel's team, for their support and encouragement and to especially acknowledge Dr. Christoph Salge for his observation that relevancy variables can explain the surprising behaviour of the Drop machine. Next, I would like to thank Martin Biehl for pointing that the decomposition of machines should be done against their transition function and not against their states.

Finally, I would also like to thank my friend, Bamdad Fard for all our debates on the nature of knowledge and intelligence, for their intellectual rigour that shaped me.

# Contents

# Chapter 1

# Introduction

## 1.1 Motivation

#### Philosophy of comprehension

We think that our universe is fundamentally comprehensible. This assumption lies at the foundation of science and it's used whenever we make predictions about the physical world.

In simple terms, the purpose of my project is to study how machines understand machines.

#### Value of prediction

In the biological world, evolution rewards agents that make successful predictions about their environment: a gazelle on the African plains will run away from lions, making the implicit prediction that remaining in its current position,, will get itself eaten; an E. coli bacterium climbs glucose gradients which asserts its belief [2] [20, p. 177] that it will benefit from a higher concentration of glucose and showing a "baked in" understanding of the physical laws of motion (swing the flagellum to go forward).

#### Astronomical cycles and plants

Astronomical cycles are large scale patterns in our world that have a significant impact on Earth's biosphere. Most plants respond to these patterns. During the night, legumes droop their leaves and many flowers close-up [21]. The cucumber could be attempting to reduce its risk of freezing and the flower could be preserving its odour, waiting for the insects to come out.

#### Human perspective

Throughout history, humans have devised many mechanisms to keep track of time. There are many incentives for reliable and accurate timekeeping like the safe running of trains and enhanced celestial navigation.

#### Timekeeping methods

It is reasonable to assume that the first technique that humans used to tell the time was simply looking at the sky [1]. This evolved into the first clock - the sundial, which works by extracting information about time from the environment. Then, we got the hourglass, candles and water clocks - devices with memory[1] that measure time spans which we use to count astronomical cycles or to identify our place inside one. We also combine different methods of telling the time: using a 12 hour clock and the knowledge of whether the sun is up we can identify the hour within the 24 hour day. And we can also tell something about time by using proxies: we know that plants follow the seasons so if we see a blooming cherry tree it's probably not autumn (does this mean that cherry trees are clocks...?).

It is interesting to note that all of these clocks are finite and can only distinguish between a finite number of time divisions. This is a problem since we might want to treat time as being unbounded[2]. This

[1] Devices like the hourglass rely on being stateful, in comparison to something like sundial, whose history does not affect its operation.

[2] To clarify, we might want to distinguish large spans of time, without having to use a large mechanism.

dissonance can be resolved in two ways: either accept the limitation that the mechanism will only work for a finite amount of time and then get "used up" just like a candle or match the mechanism to a cycle that we find important, like the day, and use it to identify the step in the cycle, like the hour in the day. Notice how useless a 13 hour clock would be...

#### Biological Clocks

Most living organisms have a biological clock [4] called the circadian rhythm. In the case of humans [5] this clock consists of a gene expression cycle and the suprachiasmatic nuclei, a region of the brain that involves roughly 20,000 neurons [13]. In cyanobacteria, on the other hand, the circadian clock is only made up of three proteins and is so simple that it has been reconstituted in vitro [12].

#### Models

A scientific model is a collection of empirical results whose pattern has been folded up into a formal system. A scientific model might tell me that it will probably rain tomorrow. This may in itself be useful but the prediction itself is only a part of why we value science. Even if my model's prediction is not perfect, the mechanism of the model, itself, can be thought of as an explanation for the phenomena that it is predicting. This is why I personally consider models to be a concrete form of knowledge.

## 1.2 Scenarios

### 1.2.1 Probabilistic Ant

Here's a scenario: say I am some kind of foraging insect and I go off to forage and I want to return to the nest at some later time. Now this sometimes causes problems for ants: ants get stuck in infinite loops [17]. I want to be better than that. I want some kind of clock that tells whether "its been awhile" since I left the nest. Now since I am just a little insect, I cannot devote a lot of resources constructing fancy mechanisms that tell me "everything" about time. It turns out that the simplest machine that can tell me if "its been awhile" is a two state automaton that has a probability to decay to the ground state. Now it seems that all I have to do it is adjust the decay rate of my automation to the match the span of time that I am interested in...

### 1.2.2 Discriminating Sea Sponge

Suppose I were a sea sponge. Then, I would be very interested in when the next full moon is due, after all, I want to make sure I am releasing my gametes at the right time, and not interested at all in whether it's the year 2066 or not, or whether or not it's a Tuesday... or whether we have a solution to the Collatz conjecture. So what do I do? How do I make a machine that tells me exactly what I want without wasting resources in building or supporting a complex machinery? Can I even build one that doesn't burden me with unnecessary and potentially confusing information?

### 1.2.3 Comprehensible Pac-Man

Here is another scenario. Say there's a man, call him Bob, in an arcade, playing Pac-Man. Now say there's another man, Charlie, *inside* that big, bulky machine watching all the bits in the memory flipping back and forth. As he plays the game and sees the action happen on the screen, Bob will understand how Pac-Man moves; how he eats pellets; how Pinky the Ghost moves, etc. Charlie's job, on the other hand is more difficult; what he can see is an almost incomprehensible mess of numbers, but in time, he will start to make sense of the patterns, in the machine, of the regularity behind the changes. Given enough time, both Bob and Charlie will build up a mental model of Pac-Man and will be able to make predictions about the behaviour of the game. They might also be able to propose explanations for it, even though they might put them in a different context. Bob's experience was similar to the visual confusion that people recovering from blindness would experience [8] or to the experience of a software cracker reverse engineering a binary executable.

The compiled binary was a piece of code that had all of its subroutines and variable names removed. It had all the code jumbled up and any hint (that would help anyone assign meaning to its computation) has been squeezed out. Yet, only from observing patterns in the computational process itself, people can comprehend that machine.

The point that I am trying to make is that before we understand a process, while it is still a "confusing mess", all I can do is distinguish one state from another and keep track of that process. But as I gain some understanding of it, I form mental models that describe a part of that machine or a part of its behaviour. Therefore, the states of machines can be thought of as encoded tuples of the states of smaller machines.

## 1.3 Previous Work

Many of the topics that motivate this project have been approached before.

Information Bottleneck Method" [18] motivated by measuring relevant information. Martin et al. published "Some ways to see two in one" [11] interested in decomposing a system into subsystems.

The idea in the *cherry tree clock* thought experiment, of using proxy machines for making predictions has already been approached in a paper [16] on *digested information*.

The problem of reverse engineering computer programs has been studied in the paper [3] on *concept assignment*.

## 1.4 Mathematical Preliminaries

### 1.4.1 Machines

For the purpose A *Markov chain* is a machine that has a probability distribution amongst a finite number of states. In this report, the probability distribution is described by a column vector. The machine is evolved in time by the application of a matrix, which has the probabilities to transition from each state to every other state.

The *Kronecker Product* is defined as follows:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}.$$

In this report, I use the Kronecker product to *compose machines*. I justify this because it can encode every combination of states and every combination of every transition. Another justification is that it is used in Quantum Computing in a similar way to to encode composite states [14].

### 1.4.2 Probability

*Marginalisation* is the operation that produces the probability distribution of a variable from the joint distribution with another variable.

$$p(x) = \sum_{y} p(x,y)$$

The *expectation* of a random variable is a probability weighted average of a random variable. It is calculated by the following equation:

$$\mathrm{E}[X] = \int_a^b p(x)x\mathrm{d}x.$$

*Laplace's Principle of Insufficient Reason* is a subtle concept, with a rich history [6, Chap. IV], For the purpose of this report, this principle states that as long as nothing else is known about a random variable, a uniform probability distribution should be assigned to it.

### 1.4.3 Information

*Mutual information* is the amount of information one would gain about one variable form observing another. One way to calculate this value is with the following equation:

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}.$$

For the purpose of this report, a *Relevancy Variable* is a random variable whose probability distribution is of interest to an agent. They are used as a kind of filter for relevant information, to hide detail that

is not of interest in the probability distribution of another variable. Notice that mutual information is symmetrical: the amount of information that one variable has about another is the same with the variables swapped.

# Chapter 2

# Models

This chapter presents the machines that have been investigated in this project.

## 2.1 Drop

The Drop machine is a *Markov chain* with two states, labelled $u$ and $d$ and whose time evolution is guided by a probabilistic transition function. A diagram of this machine is provided in Figure 2.1.

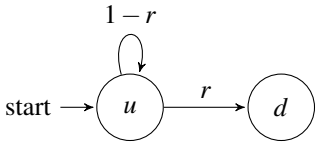


Figure 2.1: Drop machine

The machine is initially in state $u$. After each time-step there is a probability $r$ the machine decays (transitions from state $u$ to state $d$) and a probability $1-r$ that it doesn't. Once the machine has decayed, it remains in state $d$ forever.

The machine does not have any inherent knowledge about what the time is, except what it can infer from its state.

There are two variations of the Drop machine:

**Big Bang** In this variation, the machine starts out in a well defined state where $P(u \mid t = 0) = 1$. The reason why this variation is named *Big Bang* is because its initial well defined state which is analogous to the Big Bang of our universe: a determined state that implies the evolution of the physical system.

**Pre-Decayed** This variation similar to the previous one differing in the fact that this machine is given a chance to decay before $t = 0$. Essentially, its simulation starts one time-step earlier. Because of this, $P(u \mid t = 0) < 1$ for $r > 0$ This model does not have the exceptional, well defined start state.

**Theorem 1.** *The long term behaviour of the Big Bang machine is given by* $P(u \mid t) = (1-r)^t$.

*Proof.* We have been given the recurrence relation $P(u \mid t) = (1-r)P(u \mid t-1)$ and the base case $P(u \mid t = 0) = 1$. Expanding the equation gives:

$$P(u \mid t) = 1\overbrace{(1-r)(1-r)\dots(1-r)}^{t \text{ times}}.$$

It suggests the following generalised form:

$$P(u \mid t) = (1-r)^t.$$

We can test this solution by substituting it into the recurrence relation:

$$P(u \mid t) = (1-r)P(u \mid t-1),$$

$$P(u \mid t) = (1-r)(1-r)^{t-1},$$
$$(1-r)^t = (1-r)^t,$$

which confirms the solution. □

**Corollary 1.1.** *The long term behaviour of the Pre-Decayed machine is given by* $P(u \mid t) = (1-r)^{t+1}$.

## 2.2 Cyclic

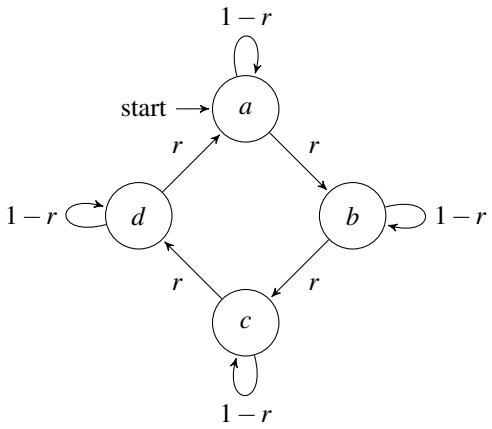


Figure 2.2: Cyclic machine for $N = 4$

This is a machine that follows a cycle in $N$ states. It starts from state 0 and from then, at each time step, the machine has a chance $r$ to transition to state $n+1 \bmod N$ and a chance $r-1$ to remain in the same state. This mechanism can be expressed by the following recurrence relation:

$$p(n \mid t=0) = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise} \end{cases},$$
$$p(n \mid t) = p(n \mid t-1)(1-r) + p(n-1 \mid t-1)r.$$

**Theorem 2.** *The behaviour of the Cyclic machine is given by the following equation:*

$$p(n \mid t) = \sum_{c=0}^{\lfloor \frac{t-n}{N} \rfloor} \binom{t}{n+cN} (1-r)^{t-cN-n} r^{n+cN}$$

*Proof.* The main part of this proof is based on all of the possible different sequences of transitions that can be made to arrive at a state at a given time. To assist our proof, we make a diagram of these transitions, to produce Figure 2.4.

A superficial examination of Figure 2.4 (which does not consider the transitions that wrap around from the last state to state $a$) would imply Equation (2.1).

$$p(n \mid t) = \binom{t}{n} (1-r)^{t-n} r^n. \tag{2.1}$$

This is because there are $\binom{t}{n}$ paths going from the start state to the state of interest and the transition probability of any of these sequences is the probability of each decay repeated by the number of decays times the probability of not decaying times the number of times that a decay does not happen. In this simplification, $n$ gives the number of decays. The next step is to generalise Equation (2.1) to take into account the wrap-around that happens from the last state to state $a$. In our example, some of the transition sequences do not loop around at all, while some of the sequences loop around once. In the general case, the maximum number of times that the clock could loop around is $\lfloor \frac{t-n}{N} \rfloor$. This will be the limit of our

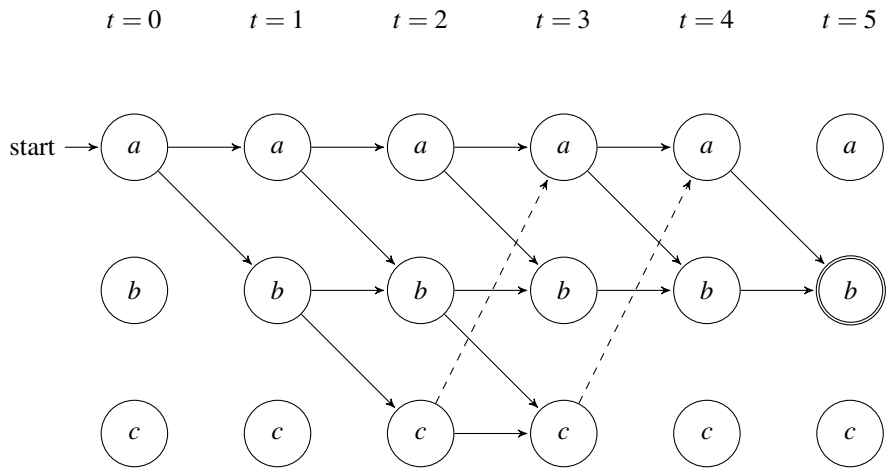


Figure 2.3: Topological map of possible transitions

sum. Because we allow cycles, the number of decays is now $n+cN$. Putting these modifications back into Equation (2.1) we obtain:

$$p(n \mid t) = \sum_{c=0}^{\lfloor \frac{t-n}{N} \rfloor} \binom{t}{n+cN} (1-r)^{t-cN-n} r^{n+cN}.$$

□

## 2.3 Parallel

The *Parallel* machine is a tuple of *Drop* machines used in conjunction. For convenience, in the context of the Parallel machine, we use the following variables and terms:

**bit** To avoid confusion, the Drop machines that make up the Parallel machine will be called bits.

**n** This variable counts the number of bits in a tuple that are up

**N** This variable counts the total bits in the tuple.

**Theorem 3.** *The behaviour of the parallel machine is given by Equation* (2.2).

$$p(n \mid t) = \binom{N}{n} \sum_{k=n}^{N} \frac{p(k \mid t-1)}{\binom{K}{k}} (1-r)^n r^{k-n} \binom{N-n}{k-n} \tag{2.2}$$

*Proof.* The main part of this proof is based on the combinatorics of the decay process, the mapping from the tuples at one time-step, to the tuples at the next time-step. We can visualise this mapping with the aid of Figure 2.4 which displays all of the incoming decays into the $n=2$ level for $N=4$.

From Figure 2.4 we notice how every level has $\binom{N}{n}$ tuples. We introduce the notation $p_n$ and $p_s$ to distinguish between the probability that the machine is in the level $n$ and the probability that the machine is in the specific tuple $s$ respectively. The relation between these two values is

$$p_n(n \mid t) = p_s(n \mid t) \binom{N}{n}. \tag{2.3}$$

We notice that from level $k$ to level $n$ there are $\binom{N-n}{k-n}$ transitions. To calculate the probability of a transition from level $k$ to level $n$ we notice that $k-n$ bits must decay, while $n$ bits have to stay up. Keeping the notation from the Drop machine, we denote the probability for a bit to decay with $r$ and the probability of that bit to stay up with $1-r$. This makes the transition probability from one tuple to another to be $(1-r)^n r^{k-n}$. From any level to another level there are $\binom{N-n}{k-n}$ logically possible transitions. This is because in the target level, $n$ bits must be up, we can consider those bits that remain up to not

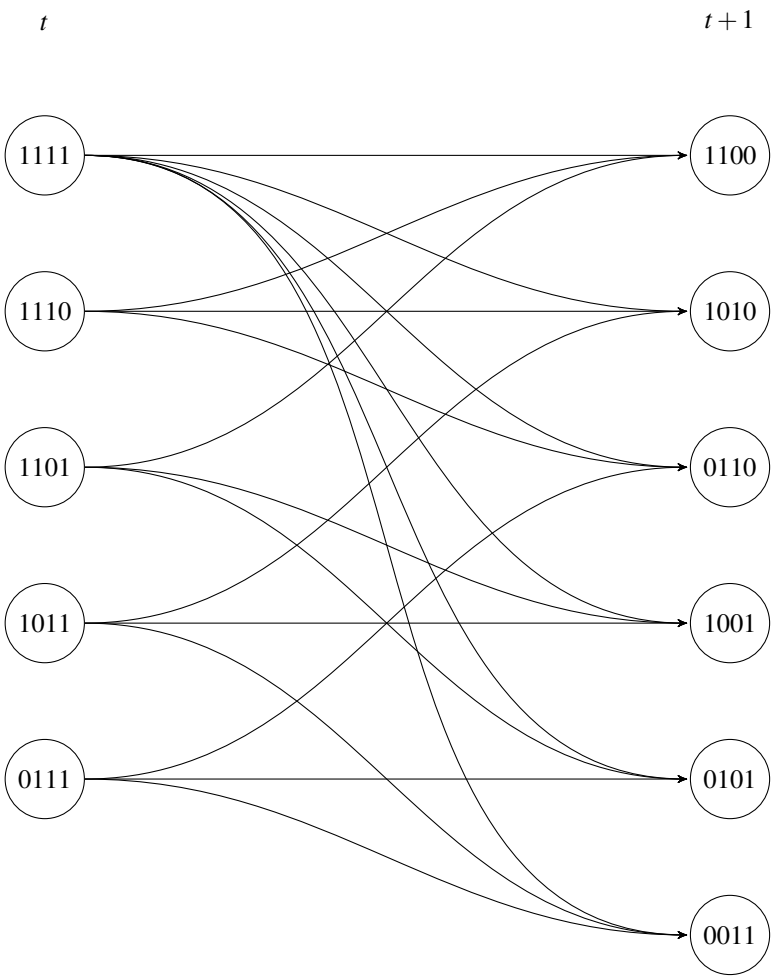


Figure 2.4: Topological mapping of decays

participate in the interaction at all. This leaves us with free slots in the source layer and $k-n$ free bits and the number or arrangements for for $k-n$ bits in $N-n$ slots is $\binom{N-n}{k-n}$.

We also notice that a tuple from level $n$ can remain in level $n$ at the next time-step and that a tuple from $n+1$ can decay to $n$ and so on up to $N$. This suggests that in order to calculate $p_s$, we must sum up all the incoming transitions from level $n$ up to level $k$. When we put these remarks together, we get:

$$p_s(n \mid t) = \sum_{k=n}^{N} p_s(k \mid t-1)(1-r)^n r^{k-n} \binom{N-n}{k-n}.$$

We combine this with Equation (2.3).

$$p(n \mid t) = \binom{N}{n} \sum_{k=n}^{N} \frac{p(k \mid t-1)}{\binom{K}{k}} (1-r)^n r^{k-n} \binom{N-n}{k-n}$$

□

## 2.4 Flip-Flop

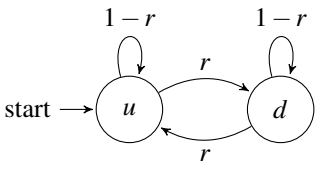


Figure 2.5: Flip-Flop machine

This machine is an extension of the *Drop* machine. Instead of decaying permanently from state $u$ to state $d$, this machine flips between $u$ and $d$. As with the *Drop* machine, the Flip-Flop machine has a probability $r$ to change state.

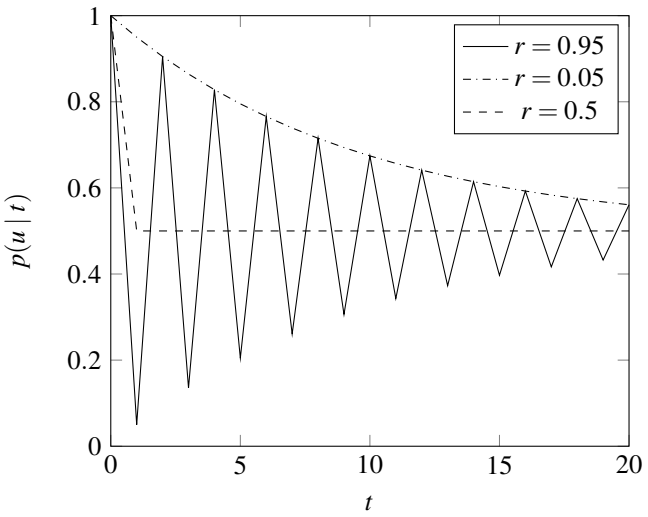


Figure 2.6: Probability to be in state $u$ for different $r$.

Being assisted by the plot of $p(u \mid t)$ in Figure 2.6, we can distinguish between different regimes in which the machine operates. As we label them, we notice the analogy with *Dampened Harmonic Oscillation*.

**Stuck** For $r = 0$, the machine is stuck in state $u$.

**Overdampened** For $0 < r < 0.5$, the probability distribution behaves like an overdampened oscillator. It is interesting to note that, in this regime, the probability distribution of the Flip-Flop machine has the same shape as that of the Drop machine. It has the added disadvantage of being objectively worst for the purpose of time-keeping.

**Critically Dampened** For $r = 0.5$, the machine acts like a critically dampened oscillator: it reaches the equilibrium in the shortest possible amount of time — one time step.

**Underdampened** For $0.5 < r < 1$, the probability distribution behaves like an underdampened oscillator. My interpretations is that the machine is initially, "synchronised" with time like a clock, but that the synchronisation gets lost as the machine evolves until there is no correlation between $t$ and $s$.

**Undampened** For $r = 1$ the machine alternates between $u$ and $d$.

**Theorem 4.** *The long term behaviour of the Flip-Flop machine is given by the following complex exponential function:*

$$p(u \mid t) = \frac{1}{2}\left((1-2r)^t + 1\right)$$

*Proof.* We begin by constructing $p(u \mid t)$ from the rules given in Figure 2.5.

$$p(u \mid t) = p(u \mid t-1)(1-r) + p(d \mid t-1)r$$

We substitute $p(d \mid t) = 1 - p(u \mid t)$ and expand.

$$p(u \mid t) = p(u \mid t-1) - p(u \mid t-1)r + p(u \mid t-1) - p(u \mid t-1)r$$

$$p(u \mid t) = p(u \mid t-1)(1-2r) + r$$

Then we use a more succinct, series notation. We let $\alpha = 1-2r$ and $g_n = p(u \mid t)$:

$$g_n = \alpha g_{n-1} + r$$

We can then telescope the sequence starting at $g_3$:

$$g_3 = \alpha\left(\alpha\left(\alpha + r\right) + r\right) + r$$

$$g_3 = \alpha^3 + \alpha^2 r + \alpha^1 r + r$$

And from here extrapolate the sum.

$$g_n = \alpha^n + r\sum_{k=0}^{n-1} \alpha^k$$

Solving the sum gives:

$$g_n = \alpha^n + r\left(\frac{1-\alpha^n}{1-\alpha}\right).$$

We remember that $\alpha = 1-2r$,

$$g_n = \alpha^n + r\left(\frac{1-\alpha^n}{1-(1-2r)}\right).$$

And rearrange to obtain:

$$g_n = \frac{1}{2}\left(\alpha^n + 1\right).$$

We plug our guess back into the original recurrence relation:

$$\frac{1}{2}\left(\alpha^n + 1\right) = \alpha\left(\frac{1}{2}\left(\alpha^{n-1} + 1\right)\right) + r,$$

$$\frac{1}{2}\left(\alpha^n + 1\right) = \frac{1}{2}\left(\alpha^n + 1\right),$$

which confirms the solution. □

### 2.4.1 Asymmetrical Generalisation

We can generalise our model to allow asymmetrical decay rates. Here we switch our formalism to use Markov matrices instead of recurrence relations. We let $S$ be the probability distribution amongst states

$$S = \begin{bmatrix} u \\ d \end{bmatrix},$$

and let $T$ be the transition matrix,

$$T = \begin{bmatrix} 1-\alpha & \beta \\ \alpha & 1-\beta \end{bmatrix}.$$

This allows us to write Equation (2.4) which expresses the time evolution of the machine as the application of the matrix as follows.

$$S_{t+1} = TS \tag{2.4}$$

One thing we can show is a similarity between the Flip-Flop and this Asymmetrical Generalisation. I do this by showing that the Asymmetrical Flip-Flop also has an equilibrium point. This implies that the generalisation behaves similarly to the symmetrical Flip-Flop with the exception that it settles in a different equilibrium. The following proof shows how to calculate it.

**Theorem 5.** *The stationary distribution of the Asymmetrical Flip-Flop is*

$$\pi = \begin{bmatrix} \frac{\beta}{\alpha+\beta} \\ \frac{\alpha}{\beta+\alpha} \end{bmatrix}.$$

*Proof.* We solve the eigenvalue equation for $\lambda = 1$.

$$\begin{gathered} \pi = T\pi \\ \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1-\alpha & \beta \\ \alpha & 1-\beta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\ \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} (1-\alpha)x+\beta y \\ \alpha x+(1-\beta)y \end{bmatrix} \\ \begin{bmatrix} \alpha x \\ \beta y \end{bmatrix} = \begin{bmatrix} \beta y \\ \alpha x \end{bmatrix} \end{gathered}$$

We remember the constraint that $\pi$ must be a probability distribution,

$$\sum_i \pi_i = 1,$$

$$x+y = 1.$$

And substitute $x = 1-y$ and $y = 1-x$ into the equations:

$$\begin{gathered} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{\beta}{\alpha}(1-x) \\ \frac{\alpha}{\beta}(1-y) \end{bmatrix}, \\ \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{\beta}{\alpha+\beta} \\ \frac{\alpha}{\alpha+\beta} \end{bmatrix} \end{gathered}$$

□

## 2.5 Deterministic Universe

For the purpose of this report, a deterministic universe is a Markov chain with a deterministic transition function. A graphical example of a deterministic universe is presented in Figure 2.7 along with its transition matrix in Equation (2.5). Notice that the entries in the transition matrix are integers.

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{2.5}$$

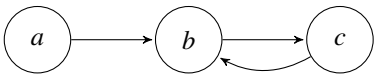


Figure 2.7: Deterministic Universe Example

# Chapter 3

# Investigations

The purpose of the clocks investigation is to make machines that have some mutual information with time. Probabilistic machines are used here because they can tell apart arbitrarily large spans of time. The *Drop* machine is the simplest of these. One experiment shows that the total amount of information given by the Drop machine has some complexity because the Drop machine tells different things in different regimes.

The purpose of the universes investigation is to show that under limited assumptions, arbitrary universes share some mutual information and to attempt to show that some universes can be thought of as being composed of smaller, independent universes.

Then, the *Flip-Flop* machine is generalised to allow asymmetric decay rates. The *Asymmetrical Flip-Flop* machines can also be viewed as a probabilistic universe and this connects the two strands of investigation - these machines can be seen as either clocks or universes.

## 3.1 Fine-Tuning the Clock

This experiment is motivated by the *Probabilistic Ant* scenario in Section 1.2.1, the scenario of an agent that has minimal computational resources, requires a simple yet optimised mechanism that can give it information about time within a time-range. In order to satisfy the simplicity criterion of this scenario, I chose to use the drop machine as the object of study. This is because the Drop machine only has two states which are the minimum number of states that can allow any kind of behaviour. It has an arguably simple transition function,, which could be modelled by any physical decay process.

As I have already chosen the number of states that this machine can have and the structure of its transition function (by choosing to use the Drop machine), there is only one free parameter left: the decay rate. In order to obtain an overview of its behaviour, I have plotted the amount of mutual information between the state of the machine and time for various decay rates and for various time-spans to obtain Figure 3.1.

The next plot is a slice into Figure 3.1 at time-span length 20 which produces Figure 3.2.

Figure 3.2 shows an inflection in the top curve. This is something that surprised me, and it's the reason why there's a second curve, belonging to the pre-decayed drop machine. What I was expecting was a mathematically simple relation, with only one maximum between the time-span size and the best decay rate for that time-span size for the big bang drop machine. I was expecting there to be a correspondence where each time-span has an optimal decay rate resulting from a "match" when the exponential decay curve "fits" the time-span in some way.

The next plot, Figure 3.3 plots the best decay rate for every time-span. It can be thought of as the curve drawn through Figure 3.1 marking the decay rate giving the largest amount of information.

Figure 3.3 shows an interesting behaviour for the Big Bang machine, with a kind of phase transition between the regime of small time-spans to the regime of large time-spans. This transition occurs because of the inflection in the information curve of the Big Bang Drop machine, when one maximum dips down bellow the other. This plot also implies that tuning the clocks accurately for large time-spans requires great precision.

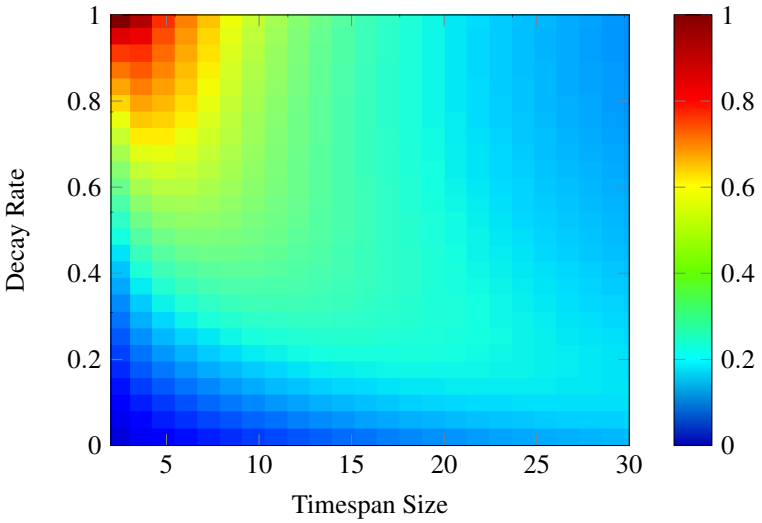


Figure 3.1: Information for different decay rates and time-spans

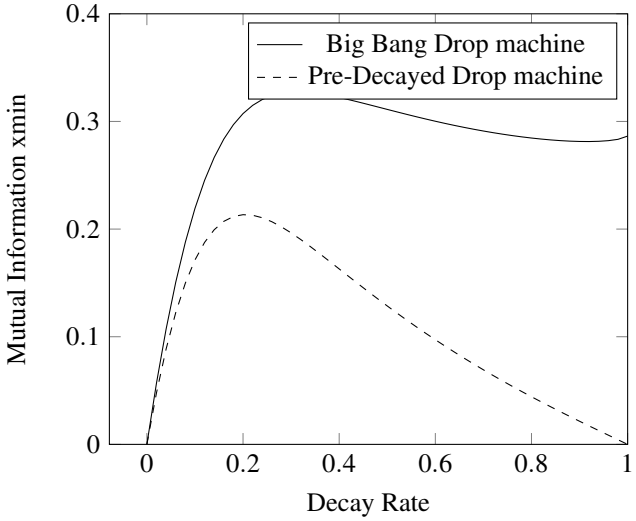


Figure 3.2: Information about a time-span of length 20

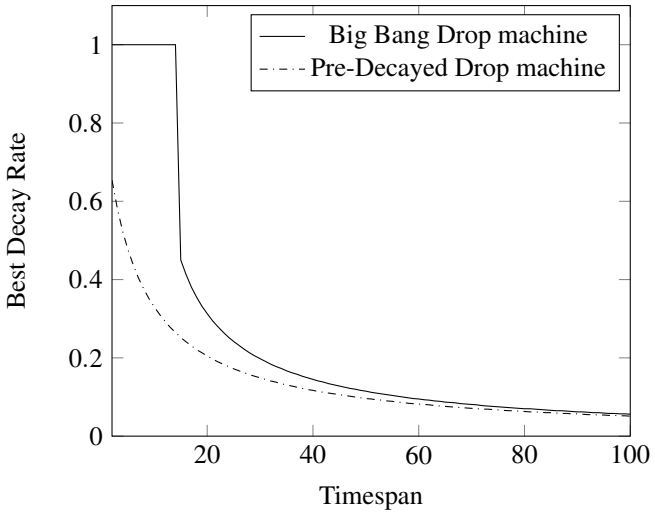


Figure 3.3: Best Decay Rates for Drop machines

## 3.2 Relevant Information

This experiment is motivated by the *Discriminating Sea Sponge* scenario in Section 1.2.2 which is concerned with relevant information and by the plot in Figure 3.2 which displays a curious inflection point. This experiment will break down that information curve of the Drop machine by using *relevancy variables*. I have chosen to use the following relevancy variables:

**Left-Right** This variable splits any time-span into equal halves. This is analogous to the need that the *probabilistic ant* has to know whether it is far away from the nest or not. This variable is shown graphically in Figure 3.5

**Odd-Even** This variable splits time-spans into two alternating parts. This is not meant to be analogous to any kind of information that an agent might be interested in. It is used to filter away information that is easy to obtain anyways[1]. This variable is shown graphically in Figure 3.5

Figure 3.4: Left-Right relevancy variable

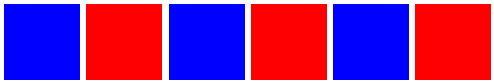

Figure 3.5: Alternating relevancy variable

The last part of this experiment is Figure 3.6. This figure is a breakdown of the amount of mutual information against the two relevancy variables. This breakdown explains the inflection of the Information curve of the Drop machine: different parts of the curve come from knowing different things and because in some regimes one dominates the other.

## 3.3 Narrow Distribution

The purpose of this experiment is to explore the behaviour of the Parallel machine.

[1] A probabilistic machine is not required for an agent to know whether it is at an odd or even time. A deterministic machine flipping back and forth between one state and another will suffice.

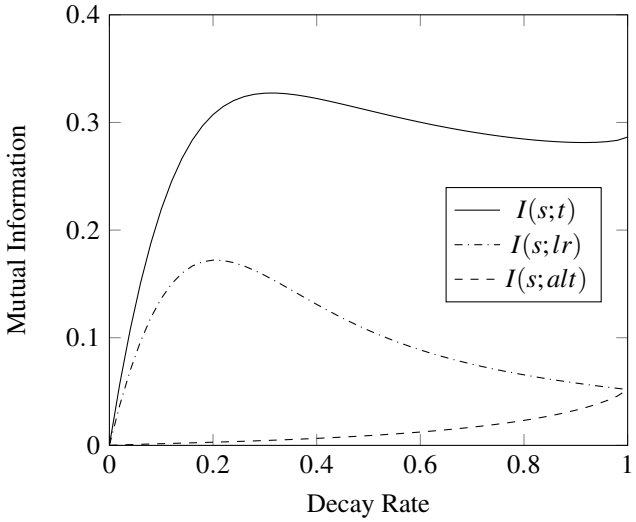


Figure 3.6: Drop Machine Information about $t$

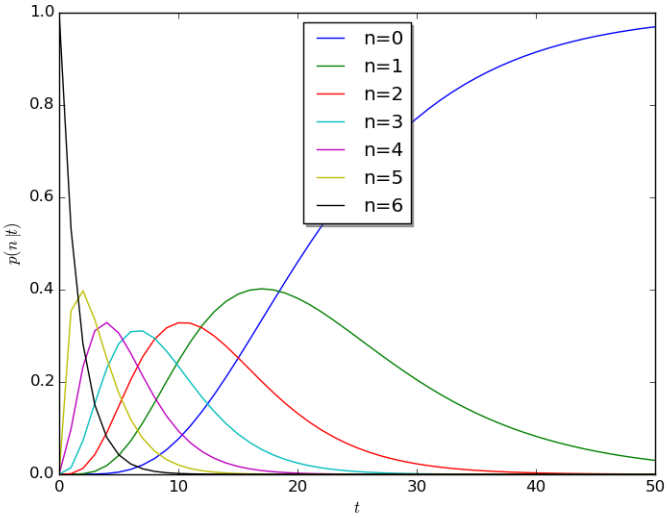


Figure 3.7: Measurements for the Parallel machine with $N = 6$

This experiment shows that even identical Drop machines, used in conjunction are able to tell more about time and give sharper distributions.

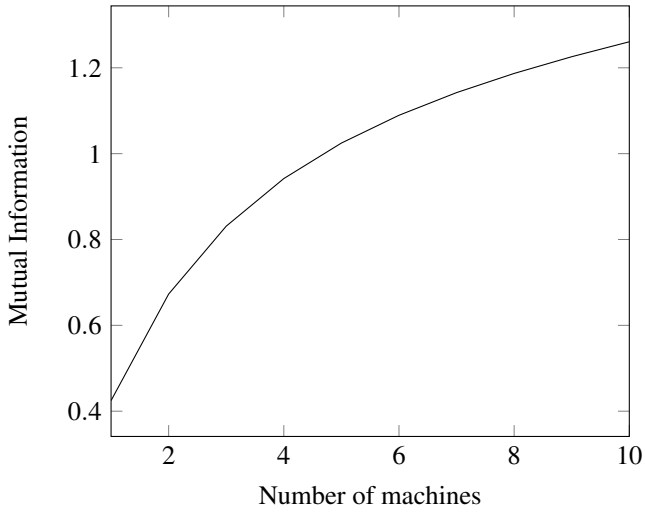


Figure 3.8: Amount of information for tuples of different sizes

Figure 3.8 shows how much information is gained by having a larger tuple of clocks.

## 3.4 Desynchronising Clock

The purpose of this experiment is to explore the behaviour of the Cyclic machine.

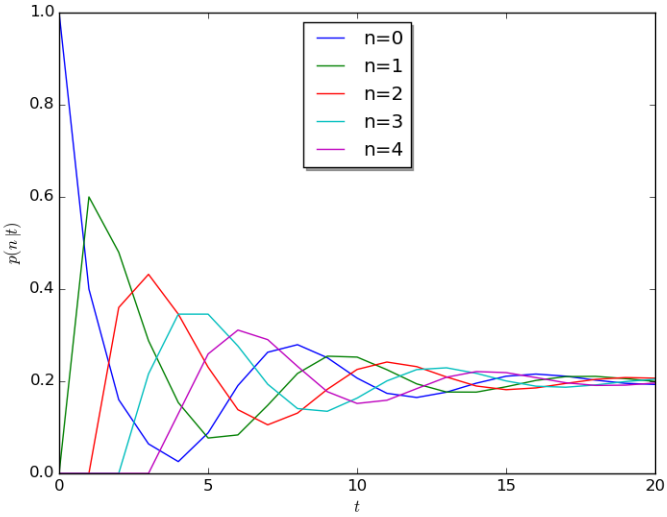


Figure 3.9: Clock loosing synchronisation

Figure 3.9 show a cyclic machine starting out in a clearly-defined state and gradually loosing sync as it ticks.

## 3.5 Deterministic Universe Bias

The purpose of this experiment is to show that predictions can be made about deterministic universes that start out in a well defined state, under minimal assumptions. The assumptions are that the universe starts

out in a known state and that all laws of physics are equally likely.

A computer program was created to simulate deterministic universes. The first plot in this experiment was created by running every possible universe of size 4 infinitely far in time and keeping track of how often each state was visited. As we only have finite resources, the universes were only simulated until their behaviour started to repeat, which implied their behaviour infinitely far in time. This simulation produced the probability distribution in Figure 3.10.

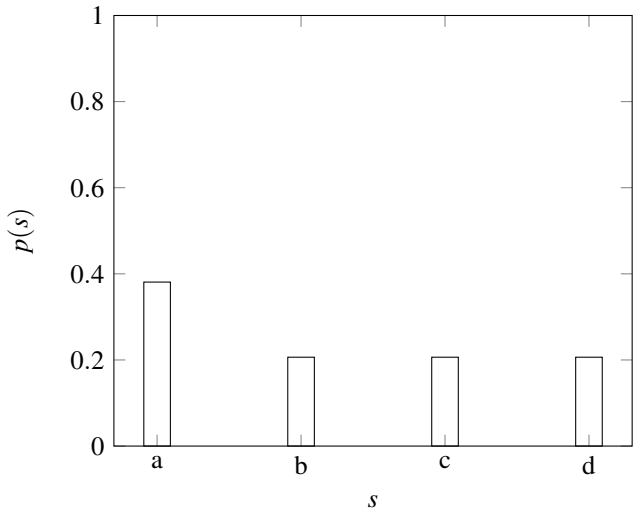


Figure 3.10: Probability distribution for a random universe at a random time

It is worth pausing here for an observation. When I refer to the *laws of physics*, I refer to the underlying mechanism of a universe: its transition function. There is a subtle distinction between my usage and the way it is commonly used in the field of Physics. Since it is philosophically impossible [9, p. 1] for us to know with absolute certainty the "actual" underlying mechanism for *our* universe, what physicists call *laws of physics* are actually patterns in the transition function.

Moving on to the next plot, if we simulate two universes simultaneously, we can plot the joint distribution of their states to produce Figure 3.11. This figure sets the scene for the next experiment on the mutual information between universes.

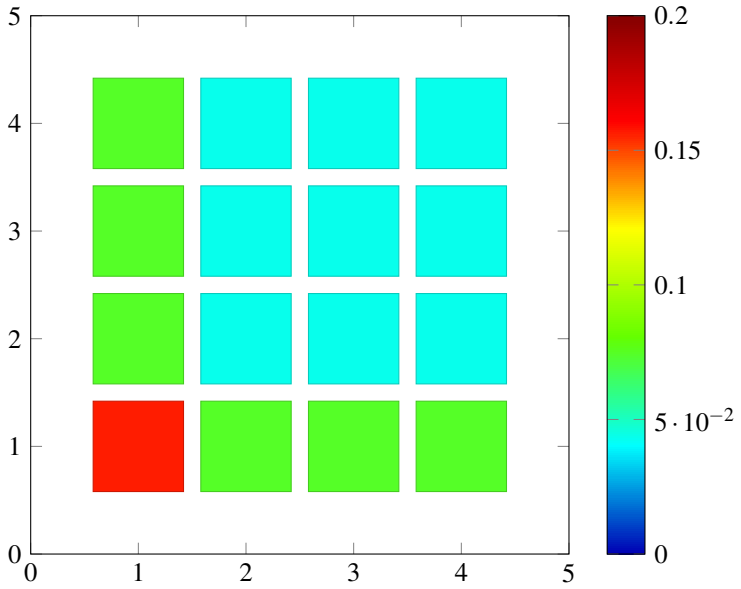


Figure 3.11: Joint Probability Distribution

## 3.6 Deterministic Mutual Information

The purpose of this experiment is to show that two random deterministic universes share some mutual information. The computer program from the previous section was used to generate the joint probability distribution for pairs of random universes of various sizes. Then, that joint distribution was used to calculate the amount of mutual information between them. A random universe of size 3, for example, shares 0.0019 bits of information with a random universe of size 4. This is generalised in Figure 3.12, that plots the amount of mutual information between two random universes of various sizes.

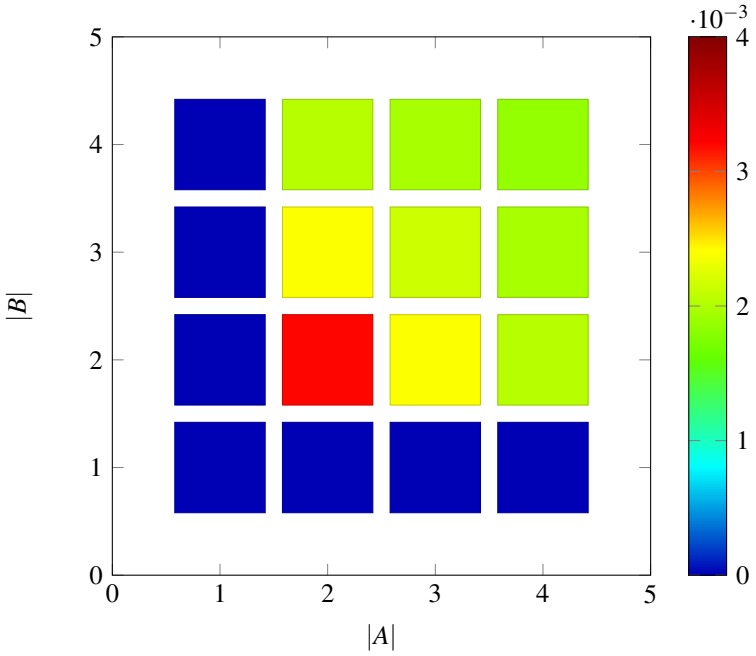


Figure 3.12: $I(A;B)$

Figure 3.12 shows that universes with only one state have no mutual information with the other universes. This is explained by the fact that no information can be had about a one state machine. The next interesting feature of Figure 3.12 is at point $(2,2)$, which corresponds to the pair of random universes of size two each. This pair shares the most mutual information, while bigger universes share less mutual information. My proposed explanation for why smaller universes share more mutual information is that larger universes have more complexity.

## 3.7 Probabilistic Universe Bias

The purpose of this experiment is to compare probabilistic universes against deterministic ones. We already know that we can have some expectations about a randomly chosen deterministic universe, from the *Deterministic Universe Bias* experiment in Section 3.5. This experiment investigates whether the probabilistic universes show a similar bias. Here, we use the *Asymmetrical Flip-Flop* to model probabilistic universes that have two states and a Big Bang.

To model a randomly chosen universe; we integrate the machine over its two parameters. I make the assumption that all the laws of physics for this universe are equally likely. My justification for this assumption is *Laplace's Principle of Insufficient Reason* and because of this assumption, we take a uniform distribution for all the parameters of this machine and integrate over that range. Because of the uniform distribution assumption and because the integration area is equal to 1, the normalisation constant cancels out to produce Expression (3.1).

$$\int_0^1 \int_0^1 \begin{bmatrix} 1-\alpha & \beta \\ \alpha & 1-\beta \end{bmatrix}^t d\alpha\, d\beta \tag{3.1}$$

We will integrate Expression (3.1) analytically for some values of $t$.

**For** $t = 0$

$$\int_0^1 \int_0^1 \begin{bmatrix} 1-\alpha & \beta \\ \alpha & 1-\beta \end{bmatrix}^0 d\alpha\, d\beta = \int_0^1 \int_0^1 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} d\alpha d\beta$$
$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

**For** $t = 1$

$$\int_0^1 \int_0^1 \begin{bmatrix} 1-\alpha & \beta \\ \alpha & 1-\beta \end{bmatrix}^1 d\alpha\, d\beta = \int_0^1 \begin{bmatrix} 0.5 & b \\ 0.5 & 1-b \end{bmatrix} d\beta$$
$$= \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

**For** $t = 2$

$$\int_0^1 \int_0^1 \begin{bmatrix} 1-\alpha & \beta \\ \alpha & 1-\beta \end{bmatrix}^2 d\alpha\, d\beta = \int_0^1 \int_0^1 \begin{bmatrix} (1-\alpha)^2+\alpha\beta & (1-\alpha)\beta+(1-\beta)\beta \\ (1-\alpha)\alpha+(1-\beta)\alpha & (1-\beta)^2+\alpha\beta \end{bmatrix} d\alpha d\beta$$
$$= \int_0^1 \begin{bmatrix} \frac{1}{6}(3\beta+2) & \frac{1}{2}(3-2\beta)\beta \\ \frac{1}{6}(4-3\beta) & \beta^2-\frac{3}{2}\beta+1 \end{bmatrix} d\beta$$
$$= \begin{bmatrix} \frac{7}{12} & \frac{5}{12} \\ \frac{5}{12} & \frac{7}{12} \end{bmatrix}$$

For all other values of $t$, we can complement the solution with numerical integration, to produce Figure 3.13

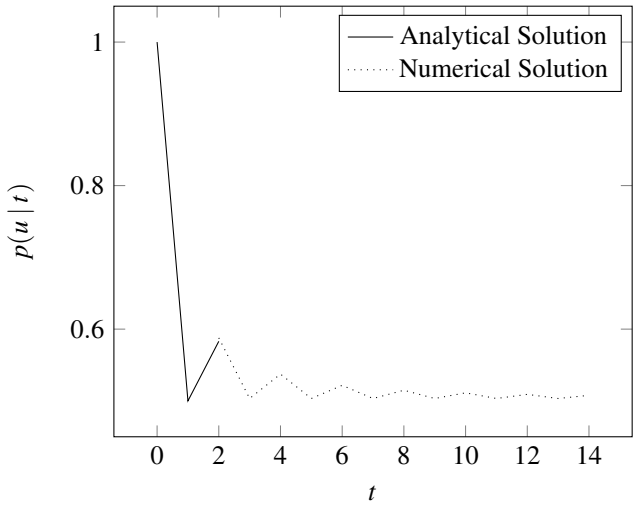


Figure 3.13: Average time evolution of random universes

Figure 3.13 shows that after marginalising over the possible transition functions, we can still make predictions about probabilistic universes and the analytical solution confirms that it's not just a numerical error.

As with the deterministic universes, the next step is to marginalise over time. Since I do not have an analytical solution to Equation (3.1), I cannot marginalise time altogether. I will instead marginalise for finite time-spans and gradually increase those time-spans, to get an improving approximation of marginalising to infinity. This produces Figure 3.14

Figure 3.14 suggests that once we fully marginalise over time, we lose any predictive capability that we had. More explicitly, Figure 3.14 suggests that Equation (3.2) is true.

$$E[S] = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \tag{3.2}$$

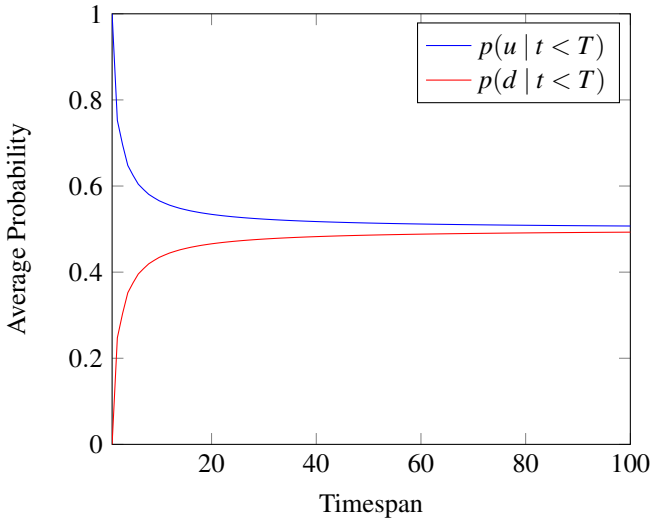


Figure 3.14: Average state of average universes measured within various time-spans

It is also interesting to note how Figure 3.14 says that we can make predictions about a random universe as long as we don't let it run for too long.

## 3.8 Clock Evolution

The machine used in this experiment is the *Asymmetrical Flip-Flop* and the time-span of interest is of length 5. The purpose of this experiment is to mimic the biological evolution of a collection of clocks. The computer experiment is started without any clocks and the collection is built up iteratively, one clock at a time. Before being added to the collection, each clock is optimised to provide the most amount of information given the current circumstances and then permanently committed to the collection.

Figure 3.15 shows the first 7 machines that are collected. Notice how the first machine is a Flip-Flop and the others are Drop machines. The Flip-Flop is collected first because it is capable of providing 1 whole bit of information about the alternating relevancy variable. After the Flip-Flop has been collected no more information can be collected about the alternating variable. Therefore, the rest of the machines have no Flip-Flop component and instead focus on providing information about the Left-Right relevancy variable.

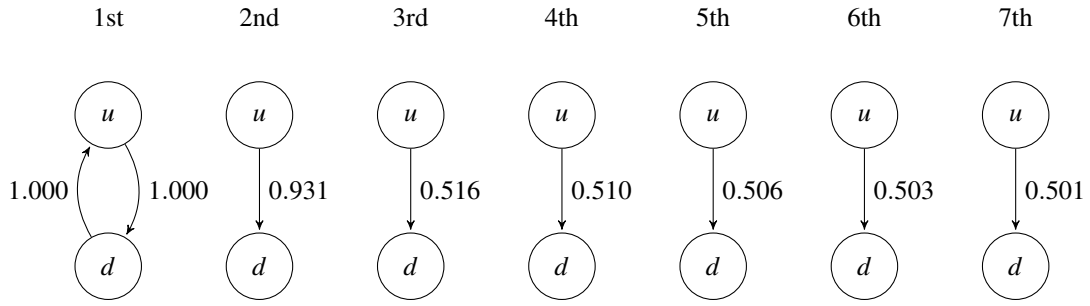


Figure 3.15: Reading from left to right, the first machines that are collected for a world size 5; probabilities have three decimals accuracy

To help us look for a trend in the collected decay rates, we can use the plot in Figure 3.16.

For a qualitative impression of the kind of information that these collected machines give, we plot the probability distributions from the measurements of the 2nd and the 3rd machines in Figure 3.17. It should be noted that; the 1st machine does not take part in the plot for clarity.

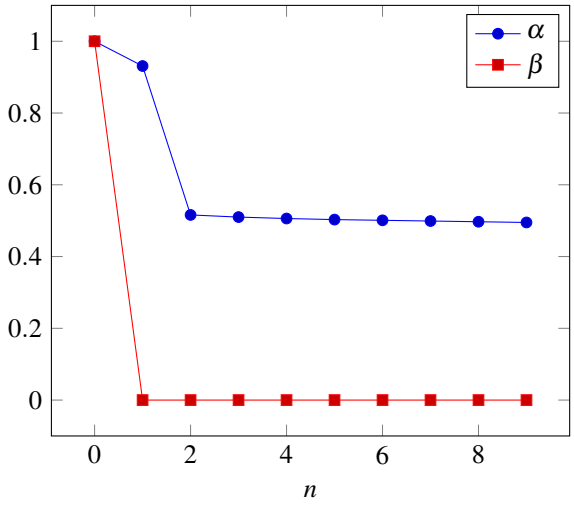


Figure 3.16: Decay rates for the first 10 machines that are found for world size 5

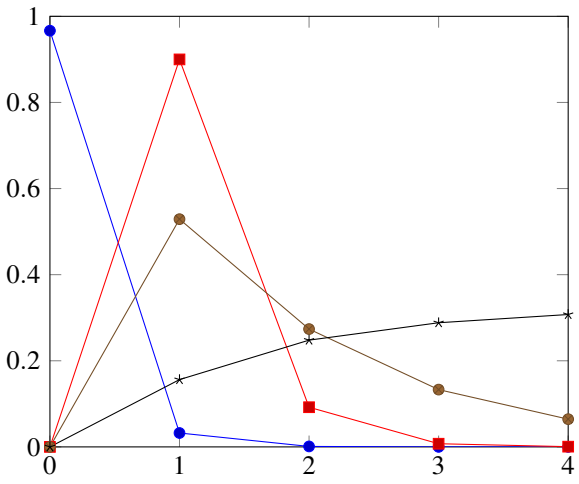


Figure 3.17: Measurements for 2nd and 3rd machines

## 3.9 Decomposing Universes

This experiment is motivated by the *Comprehensible Pac-Man* scenario in Section 1.2.3. This experiment will show how the behaviour of a machine can be understood as if the machine was composed of smaller, independent machines running in parallel.

This experiment will study the machine from Figure 3.18.

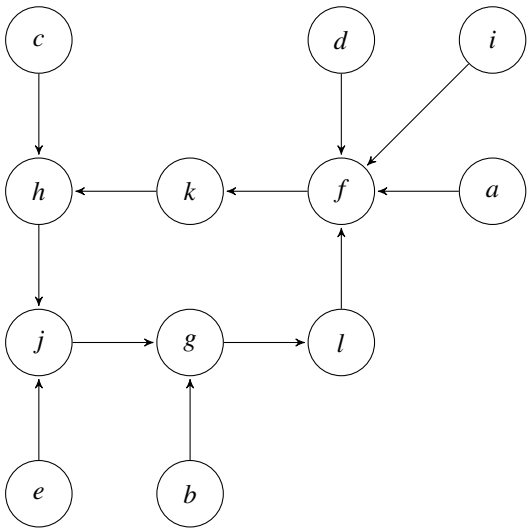


Figure 3.18: Decomposable Meta-Machine

Alternatively, we can describe this machine by its transition matrix, $M$:

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

As I mentioned in Section 1.4.1, in this report, I use the tensor product to compose machines. Therefore, in order to decompose a machine, we have to factorise the tensor product.

$$M = A \otimes B$$

The reason why I picked this particular machine is because I already knew its factorisation, whereas arbitrary tensors are a challenge for me to factorise. Here are the tensor factors of $M$:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix};$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Now that we have obtained the transition matrices for the component machines, we can return to the graph notation to produce Figure 3.20 and Figure 3.19 and notice that the component machines are arguably simpler.

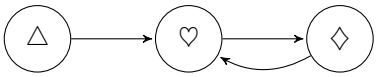


Figure 3.19: Component machine A

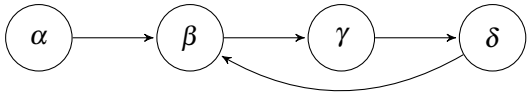


Figure 3.20: Component machine B

Notice how states of the meta-machine (the original machine) encode tuples of states of the component machines, which implies that the states of each of the component machines map to subsets of the states of the meta-machine. This leads to the observation that component machines say something about the meta-machine. Another observation is that together, the complete ensemble of component machines fully describes the behaviour of the meta-machine.

# Chapter 4

# Discussion

## 4.1 Conclusion

This report started with the motivation to study understating and predictions. For this, I have built models and proved results about their behaviour which I used in computer experiments.

I have shown how universes can be interpreted as being composed of smaller machines and individually, any of the smaller machines can be interpreted as a simplified version of the meta-universe, for the purpose of making partial predictions.

I have shown that when optimising a prediction mechanism, the relevancy of information should to be considered, instead of merely the bulk amount.

I have shown that more can be known about random deterministic universes than about random probabilistic universes, even under fewer assumptions.

The results found here suggest that Markov chains interpreted as clocks or as universes show rich behaviour and potential for fruitful future research.

## 4.2 Future Work

This project does not have a formal measure of the complexity of clocks with respect to the demands and constraints of their physical systems. Although many metrics for complexity exist, their applicability has not been investigated in this context.

This report only used an artificial example of universe decomposition. An automatic way to decompose universes could be researched in the future.

In this report, I have only treated the universe decomposition which produces perfectly independent components. I consider this to be a shortcoming since many interesting phenomena from the natural world, like the operation of a space-shuttle or the mechanism of a clock, can only be understood as the composition of interacting machines.

## 4.3 Personal Reflection

This was an ambitious project for me. I encountered many challenges, from inadequate mathematical abilities, to running out of steam, but in the end I still managed to make something that I'm personally proud of. I say this in spite of all the shortcomings of my project, including my poor literature review, my sloppy proofs and my incomplete interpretation of my results.

# References


[1] Carl Anderson. “Telling time without a clock: Scandinavian daymarks”. In: *Harvard. edu* (1998).

[2] Howard C Berg, Douglas A Brown, et al. “Chemotaxis in Escherichia coli analysed by three-dimensional tracking”. In: *Nature* 239.5374 (1972), pp. 500–504.

[3] Ted J Biggerstaff, Bharat G Mitbander, and Dallas Webster. “The concept assignment problem in program understanding”. In: *Proceedings of the 15th international conference on Software Engineering*. IEEE Computer Society Press. 1993, pp. 482–498.

[4] Guy Bloch et al. “Animal activity around the clock with no overt circadian rhythms: patterns, mechanisms and adaptive value”. In: *Proceedings of the Royal Society of London B: Biological Sciences* 280.1765 (2013), p. 20130019.

[5] Timothy M Brown and Hugh D Piggins. “Electrophysiology of the suprachiasmatic circadian clock”. In: *Progress in neurobiology* 82.5 (2007), pp. 229–255.

[6] FY Edgeworth. “A Treatise on Probability, by John Maynard Keynes”. In: *Journal of the Royal Statistical Society* 85 (1922), pp. 107–13.

[8] Esther Inglis-Arkell. *The world that only formerly-blind people can see*. 2013. URL: http://io9.com/the-world-that-only-formerly-blind-people-can-see-476400679.

[9] John Randolph Lucas. “Space, time and causality”. In: (1984).

[11] Daniel Polani Martin Biehl. “One way to see two in one”. In: *European Conference on Artificial Life* (2013).

[12] Masato Nakajima et al. “Reconstitution of circadian oscillation of cyanobacterial KaiC phosphorylation in vitro”. In: *Science* 308.5720 (2005), pp. 414–415.

[13] NIH. *Circadian Rhythms Fact Sheet*. http://www.nigms.nih.gov/Education/Pages/Factsheet_CircadianRhythms.aspx/. [Online; accessed 22-April-2015]. 2012.

[14] Riley T Perry. *The temple of quantum computing*. 2006.

[16] Christoph Salge and Daniel Polani. “Digested information as an information theoretic motivation for social interaction”. In: *Journal of Artificial Societies and Social Simulation* 14.1 (2011), p. 5.

[17] Theodore Christian Schneirla et al. “A unique case of circular milling in ants, considered in relation to trail following and the general problem of orientation. American Museum novitates; no. 1253”. In: (1944).

[18] Naftali Tishby, Fernando C Pereira, and William Bialek. “The information bottleneck method”. In: *arXiv preprint physics/0004057* (2000).

[20] Mitchell M Waldrop. *Complexity: The emerging science at the edge of order and chaos*. Simon and Schuster, 1993.

[21] Alex AR Webb. “The physiology of circadian rhythms in plants”. In: *New Phytologist* 160.2 (2003), pp. 281–303.

# Bibliography


[7] *Entropy (information theory)*. Last visited 28 April 2015. URL: http://en.wikipedia.org/wiki/Entropy_%28information_theory%29.

[10] David JC MacKay. *Information theory, inference, and learning algorithms*. Vol. 7. Citeseer, 2003.

[15] J-C Pomerol and Patrick Brezillon. "About some relationships between knowledge and context". In: *Modeling and Using Context*. Springer, 2001, pp. 461–464.

[19] Koustubh M Vaze and Vijay Kumar Sharma. "On the adaptive significance of circadian clocks for their owners". In: *Chronobiology international* 30.4 (2013), pp. 413–433.

[22] Daniel Wilkerson. Last visited 28 April 2015. URL: http://daniel-wilkerson.appspot.com/entropy.html.