Thermodynamics of sensing, correlations and computation

a thesis presented for the degree of

Doctor of Philosophy of Imperial College London

by

Rory A. Brittain

Department of Mathematics

Imperial College

180 Queen's Gate, London SW7 2BZ

September 2019

I certify that this thesis, and the research to which it refers, are the product of my own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline.

Copyright

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Thesis advisors: Nick S. Jones and Thomas E. Ouldridge

Thermodynamics of sensing, correlations and computation

Abstract

In this thesis we discuss several aspects of the thermodynamics of information. In some cases this is motivated by its application to biological organisms. First we examine the interpretation of a quantity proposed to measure sensors and find that it is not a general measure of sensor quality in the steady state. Then we propose a biochemical version of the Szilard engine and use it as the basis for a simple feeding model of a micro-organism. We find that more complex environments can support more complex inference processes. Finally, we use a similar set-up to the feeding model to model a reversible chemical Turing machine. We show that an additional memory tape is required but for a different reason than previously suggested. Also, in order to discount edge costs, many machines must be run simultaneously.

Acknowledgments

I would like to thank my supervisors Nick and Tom for all their help over the years. I could not have done it without them.

I would also like to thank my parents for helping me with writing up this thesis and giving me somewhere to live while I did it.

Contents

1 Introduction **1**

1.1 Sensing 2

1.2 Information theory 4

1.3 Thermodynamics and the non-equilibrium free energy 6

1.4 Dilute well mixed solutions 9

1.4.1 Catalysts 12

1.4.2 DNA strand displacement 12

1.5 Thermodynamics of simple information-processing tasks . . . 14

1.5.1 Mutual information and the second law 17

1.5.2 Explaining Szilard's engine with mutual information and the generalised free energy 18

1.6 Devices that extract work from information 20

1.7 Cost of computing 22

1.7.1 Models of computation 23

1.8 Thesis overview 24

1.9 Publications and Contributions 25

2 What we learn from the learning rate **26**

2.1 Abstract 26

2.2 Introduction 27

2.3 Set up: Sensing and thermodynamics in bipartite Markov chains 28

2.4 Results and Discussion 31

2.4.1 The learning rate reports on correlations in the simplest bipartite sensing system 31

2.4.2 Feedback from the downstream to the upstream system 35
2.4.3 A more complex signal process 42
2.4.4 Optimisation of sensors 45
2.5 Conclusions 49

3 Biochemical Szilard engines for memory-limited inference **52**
3.1 Abstract 52
3.2 Introduction 53
3.3 Materials and Methods 55
3.3.1 Molecular implementation of a measurement-and-feedback machine 56
3.3.2 Numerics 59
3.3.3 Example system and calculation: Extracting work from a biased environment 60
3.4 Results 64
3.4.1 A Biochemical Szilard engine 64
3.4.2 Exploiting a series of correlated bits 69
3.5 Discussion 93

4 Reversible chemical Turing machines **96**
4.1 Abstract 96
4.2 Introduction 97
4.3 System specifications 99
4.4 Model 100
4.4.1 Overview 100
4.4.2 Stepping 102
4.4.3 Update head 105
4.4.4 Halting 108

4.5 Discussion 110
4.6 Conclusions 112

5 Conclusion **114**

References **124**

Appendix A Supplementary material **125**

A.1 Proof that l_Y tends to zero as $N \to \infty$ for the system in section 2.4.3 125
A.2 The molecular hook 127
A.3 Work calculation for Biochemical Szilard engine 128
A.3.1 Measurement 128
A.3.2 Extraction 131
A.4 Quasistatic embeddability of Markov machine 132
A.5 Work calculation for Markov machine 134
A.5.1 Measurement 134
A.5.2 Extraction 136
A.6 DNA design of the Biochemical Szilard engine and Markov machine 136
A.7 Work calculation for Binary batch machine 138
A.7.1 Measurement 138
A.7.2 Extraction 139
A.8 DNA design of batch machine measurement 141
A.9 Work calculation for full batch machine 142
A.9.1 Measurement 142
A.9.2 Extraction 142
A.10 Alternative Molecular motor 144
A.11 Examples 145

A.11.1 2-state 3-symbol Machine 145
A.11.2 4-state 6-symbol Machine 151

1

Introduction

This thesis is concerned with fundamental questions about the thermodynamic limits to sensing, information processing and computation. In particular, we will focus on concrete chemical systems in part due to their applicability to biological systems. In this introduction we will first discuss sensing, which has been well studied, particularity in the context of sensing in biological cells. A key question is how to quantify accurate sensing. Answering this question brings us onto the mathematical basis for the theory of communication: i.e. information theory. Information theory is closly linked to statistical mechanics because the principle of maximum entropy in statistical mechanics is equivalent to maximising the uncertainty of a probability distribution subject to constraints[1].

We will next discuss the physical theory of modern non-equilibrium statistical mechanics and then its application to chemical systems. These deep connections mean that information and information processing embedded in physical systems have important thermodynamic consequences, so we will discuss more specifically the role of information in thermodynamics. One popular way to think about the relationship between information and thermodynamics is to design machines that exploit information registers to do

useful work, so we will review the literature on such machines. The inverse of a machine to exploit a series of bits is one that puts in work to produce information. Finally we will discuss the thermodynamics of computation; a more general class of information-processing operations.

1.1 Sensing

Sensing is required when an agent wants to measure some detail about its environment. It involves the state of a sensor becoming correlated with the state of the environment. The state of the sensor must then be read by the agent or the state of the sensor must somehow affect the future decisions or behaviour of the agent for it to be useful.

Sensing is only useful if the environment varies in some way. If the temperature was constant all of the time, then there would be no value to measuring it. It also must be the case that the state of the part of the environment that is measured is not totally independent of all other parts of the environment. If the measured part of the environment was uncorrelated with the rest of the environment then after measurement the sensor will not be correlated with the environment.

There are innumerable examples of sensors. A classic example is of a centrifugal governor in a steam engine. It consists of two weights attached to a rotating axle. As the speed of the axle increases the weights are pulled further from the axle by the centrifugal force and rise upwards. The higher of the weights is then correlated with the speed of the axle and can be used to mechanically control the engine. Alternatively, consider a central heating system. In older systems there is a bimetallic switch that changes its shape due to its temperature and in doing so turns the heater on if the temperature is below a certain temperature.

Sensors are also vital to biological life. Humans, for example, are often said to have five senses. These senses, such as sight, involve collecting a large amount of data that is used in complicated ways. Of course, humans actually have many more senses such as measuring temperature or internal water balance

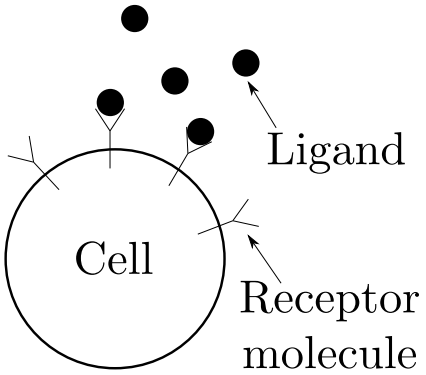


Figure 1.1: A cell can detect the presence of ligands in its environment using receptor molecules on the cell surface that bind to the ligands.

or the orientation sensors in the inner ears.

There are much simpler examples of sensing that are more amenable to analysis in unicellular organisms. A well studied example is *E. coli* chemotaxis. Bacteria detect the presence of certain molecules in their environment using receptor proteins on their surface that can bind to the ligands and in doing so pass the signal inside the cell. A simplified schematic of such a sensing system is shown in figure 1.1.

Organisms have limited resources so one would assume that natural selection has led those organisms to use those resources in a close to optimal way[2]. Of course this optimum must also be with respect to considerations of robustness as well as expected value. Knowledge of these limits is also useful to inform the engineering of synthetic sensors.

The first analysis of how the resources of a cell can be used for optimum sensing was by Berg and Purcell[3]. They showed how the accuracy of sensing ligand concentration depends on the number of receptor molecules on the cell surface. The literature on this model is reviewed by ten Wolde *et al.*[4]. Berg and Purcell assume a constant ligand concentration and diffusion limited binding to the receptor molecule but later Bialek and Setayesghar[5] extended the model to include a reaction-diffusion equation to account for the fact that when the receptor has just released a ligand there is then a higher concentration of the ligand close to the receptor.

Berg and Purcell showed that the accuracy of the sensor can be improved if it can integrate its signal over a period of time. It is not possible for an equilibrium system to do such an integration so a cell using this technique must make use of some non-equilibrium fuel. This shows that thermodynamic resources can be a limit to sensing. This is an idea that we will return to in this thesis.

1.2 Information theory

In order to make precise statements about fundamental limits, we need a mathematical framework for the information a sensor has about its environment. This mathematical framework is information theory; introduced by Claude Shannon in 1948[6] to quantify the limits on information transmission.

This thesis will only require knowlage of the basic information theoretical quantities defined on random variables[7]. For discrete random variables X, using the shorthand $P(X = x) = p(x)$, define the Shannon entropy as

$$H[X] = -\sum_x p(x) \ln p(x). \tag{1.1}$$

The entropy is a measure of the uncertainty of the random variable. It is the average of the quantity $-\ln p(x)$, which is a measure of how unlikely the outcome x is. If all outcomes are equally likely then the random variable is maximally uncertain and the entropy is maximum. If one outcome has probability one then the random variable has no uncertainty and the entropy is zero.

With a second discrete random variable Y define the conditional entropy as

$$H[X|Y] = -\sum_{xy} p(x, y) \ln p(x|y). \tag{1.2}$$

The conditional entropy quantifies the expected uncertainty there is in X

given knowledge of the state of Y. Also, define the mutual information

$$I[X;Y] = H[X] - H[X|Y] = \sum_{xy} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}. \tag{1.3}$$

The mutual information is symmetric in the two variables and quantifies how much knowledge of one variable reduces uncertainty in the other variable. An important open question is how to quantify the effectiveness of a learning system. The mutual information is often used as a measure of the information a sensors has if one variable is the sensor and the other is the signal. However, this is only a measure of the stationary correlation between two random variables.

There are other quantities for measuring information in time varying systems derived from the mutual information. One measure is the learning rate[8,9]. Given that the transitions in the sensor and signal systems are independent, the time derivative of the mutual information can be split into two parts

$$\frac{\mathrm{d}}{\mathrm{d}t} I[X;Y] = \dot{I}^X + \dot{I}^Y. \tag{1.4}$$

$\dot{I}^X$ is the change in the mutual information due to the transitions in the signal and $\dot{I}^Y$, which is also written as l_y and known as the learning rate, is the change in the mutual information due to the transitions in the sensor[10]. The meaning of this quantity will be explored further in chapter 2.

Another popular measure of the information gained by a sensor is the transfer entropy[11]. We mention it briefly here—though it is not used in the remainder of the thesis. It is an asymmetric measure of the transfer of information from the discrete time series of one random variable to another. It is the additional information about the current state of Y you gain from the history of X that is not already contained in the history of Y:

$$\begin{aligned} T_{X\to Y} &= I[Y_t; X_{0:t}|Y_{0:t}] \\ &= \sum p(y_t, x_{0:t}|y_{0:t}) \ln \frac{p(y_t, x_{0:t}|y_{0:t})}{p(y_t|y_{0:t})p(x_{0:t}|y_{0:t})} \end{aligned} \tag{1.5}$$

where Y_t is the random variable Y at time t, $Y_{0:t}$ is a shorthand for $Y_0, \ldots, Y_{t-1}$ and $I[Y_t; X_{0:t}|Y_{0:t}]$ is a conditional mutual information.

1.3 Thermodynamics and the non-equilibrium free energy

The term 'entropy' used in the previous section is suggestive of a link to thermodynamics. In fact, the Shannon entropy is precisely the non-equilibrium entropy of a physical system that is well described by a probability distribution over discrete states.

Consider a physical system with discrete microstates. When the system is in microstate x the system has energy $E(x)$. For a probability distribution over the states of the system, $p(x)$, there is an expected energy

$$\langle E(X) \rangle = \sum_x p(x) E(x). \tag{1.6}$$

As previously mentioned the entropy of this system is simply the Shannon entropy as given in equation 1.1. The generalised non-equilibrium free energy of the system is[12]

$$\mathcal{F}(X) = \langle E(X) \rangle - k_{\mathrm{B}} T H[X]. \tag{1.7}$$

If the system can exchange energy with a heat bath then the system relaxes to the equilibrium distribution which minimises this free energy. The equilibrium corresponds to maximising the entropy of the joint system of the system of interest and the heat bath. The joint system is isolated so the total energy is constant. Minimising the free energy of the system is a trade-off between minimising the energy and maximising the entropy. The temperature gives the relative weighing of the energy and entropy. If the temperature is very low then the system is very likely to be in the ground state. If the energy is very high then the differences between the energies of the states have only a small effect on the equilibrium distribution because they are small compared to the thermal energy, $k_{\mathrm{B}} T$, so all states are equally likely.

What if the microstates of the system can be partitioned into macrostates? A macrostate y is a set of microstates. Each microstate is in precisely one

macrostate. If the microstate x is in macrostate y we write $x \in y$. How is the free energy of a distribution over the macrostates calculated? For the expected energy the calculation is simply

$$\langle E(Y) \rangle = \sum_y p(y) E(y). \tag{1.8}$$

where $p(y) = \sum_{x \in y} p(x)$ is the macrostate probability and $E(y) = \sum_{x \in y} p(x|y) E(x)$ is the expected energy of the macrostate.

The entropy is not so simple because equation 1.1 is non-linear in the microstate probability. It is possible to rewrite the entropy as

$$H[X] = -\sum_y p(y) \sum_{x \in y} \frac{p(x)}{p(y)} \ln \frac{p(x)}{p(y)} - \sum_y p(y) \ln p(y). \tag{1.9}$$

This is simply the chain rule for joint entropy[7]. The chain rule is

$$\begin{aligned} H[X, Y] &= H[X] + H[Y|X] \\ &= H[Y] + H[X|Y]. \end{aligned} \tag{1.10}$$

The macrostate y is fully determined by the microstate so $H[Y|X] = 0$ and $H[X] = H[Y] + H[X|Y]$. The entropy can be split into two parts: $H[Y]$ is the entropy of the distribution over macrostates and $H[X|Y]$ is the expected entropy of the microstates given the macrostate.

Now, this expected entropy of the microstates given the macrostate can be used to define the macrostate free energy for macrostate y as

$$\begin{aligned} F(y) &= E(y) - k_\mathrm{B} T H[X|y] \\ &= \sum_{x \in y} p(x|y) E(x) + k_\mathrm{B} T \sum_{x \in y} \frac{p(x)}{p(y)} \ln \frac{p(x)}{p(y)} \end{aligned} \tag{1.11}$$

Therefore, the free energy of the system can be written in terms of the macrostate free energies and the distribution over the macrostates. The free energy is

$$\mathcal{F}(Y) = \langle F(Y) \rangle - k_\mathrm{B} T H[Y], \tag{1.12}$$

where the macrostate free energy functions similarly to the energy of a microstate but includes internal energetic and entropic contributions. Equation 1.12 gives a coarse-grained description of the system. The purpose of this course graining is that the free energy of the system can be calculated from the practically useful and measurable facts about the system and not the microscopic details. Macrostates are arbitrary but can be chosen such that they are relevant to the problem being solved.

The problem with this coarse-graining of the system is that to calculate the free energy of a macrostate y, $F(y)$, requires knowing the probability distribution over the microstates that make up the macrostate but this was precisely what we wanted to avoid by coarse-graining. A conventional assumption is that the distribution over the microstates is the Boltzmann distribution. This assumption is that the system is in equilibrium conditioned on the macrostate. The macrostates must be chosen so that this assumption is valid. Therefore, the macrostate free energy can be calculated for the Boltzmann distribution and then used as if it is an energy for the macrostate level description of the system.

To give an example, the Szilard engine, which will be discussed in full in the next section, consists of a box containing a single gas particle. There is a solid partition down the middle of the box so the particle is constrained to either the left or the right. The microstate of the system is the precise position and momentum of the particle but there are only two macrostates: the particle is on the left side of the partition or the right side of the partition. Partitioning the system into two macrostates means that this physical system can be analysed as an idealised bit without considering the specific dynamics of the motion of the particle in the box.

Another example is a dilute well mixed chemical system. This will also be discussed in more detail in a later section. Consider a single molecule of interest in a large bath of a solvent. The microstate of such a system is the specific position and momentum (and orientation if the molecules are not modelled as spheres) of the molecule of interest and all of the solvent molecules. The macrostate is the state of the molecule of interest and the states of all of the solvent molecules can be integrated over.

The partition function of a macrostate can be calculated

$$Z(y) = -k_{\mathrm{B}}T \ln F(y). \tag{1.13}$$

So, in equilibrium, the probability that the system is in a macrostate is

$$p(y) = \frac{Z(y)}{Z}. \tag{1.14}$$

1.4 Dilute well mixed solutions

This thesis will, in particular, analyse chemical systems so we must discuss how to apply this notion of non-equilibrium free energy to chemical systems. We will draw on a review by Ouldridge[13]. The chemical systems we will discuss are 'dilute' which means that there are a relatively small number of particles of interest in a solution of relatively many more solvent molecules (and often when we say 'molecule' we will be implicitly ignoring solvent molecules). This means that, except for in the specified chemical reactions, the non-solvent molecules do not interact with each other. The statistics/partition function of a molecule in the volume is as if there is only that molecule and the solvent. The solutions discussed are also well mixed, which means that the time-scales we consider are long compared to the time-scale of diffusion of the molecules throughout the volume. Therefore, the theory is not concerned with the spatial location of the molecules because they are assumed to be equally likely to be in a position in the volume.

As in the previous section we will use the Helmholtz free energy $\mathcal{F}$. In chemical systems it is more typical for the pressure to be constant rather than the volume so the Gibbs free energy should be used instead as it is the quantity minimised in equilibrium. In dilute systems the vast majority of the volume is taken up by the solute molecules so any change in the overall volume caused by internal chemical reactions is insignificant. Therefore, it makes very little difference if the Gibbs free energy or the Helmholtz free energy is used.

As mentioned in the previous section, the microstate of a volume of molecules

is a full description of the position, momentum and orientation of all of the molecules. We choose the macrostates to be the states that are defined by the numbers of the molecules of interest. We integrate over all of the states of the solvent molecules and assume that the molecules of interest have their equilibrium distribution of positions and momentums.

Therefore, using the equations in section 1.3, one can calculate the partition function of a single molecule in a certain volume at a certain temperature. If one wants to calculate this partition function for a specific case it depends on the size of the volume and the details and amount of the solvent molecules. Then, using assumption that the solution is dilute, the partition function of the solution of many molecules of different types is the product of the partition function of the individual molecules. We will not actually have to do this calculation. We only note that in principle the partition function can be calculated and then we will use the properties of the partition function.

If the partition function for one molecule of species j is z_σ^j where σ represents the parameters of the environment such as the volume and temperature then the partition function for the solution is

$$Z_\sigma(\{N\}) = \prod_j \frac{(z_\sigma^j)^{N_j}}{N_j!}, \tag{1.15}$$

where N_j is the number of molecules of species j and $\{N\}$ is a list of all of the N_j. The $N_j!$ is to prevent over counting of microstates that are equivalent due to exchanging identical particles.

The free energy of the solution is therefore

$$\begin{aligned}
\mathcal{F}_\sigma(\{N\}) &= -k_\mathrm{B} \ln Z_\sigma(\{N\}) \\
&= \sum_j \mathcal{F}_\sigma^j(N_j) \\
&= -k_\mathrm{B}T \sum_j \ln\left(\frac{(z_\sigma^j)^{N_j}}{N_j!}\right) \\
&= -k_\mathrm{B}T \sum_j (N_j \ln z_\sigma^j - N_j \ln N_j + N_j),
\end{aligned} \tag{1.16}$$

where we have applied Sterling's approximation in the last line. We have used the assumption that the N_j are large. The chemical potential of species j is therefore

$$\mu_\sigma^j = \frac{\partial \mathcal{F}_\sigma(\{N\})}{\partial N_j} = -k_\mathrm{B}T \ln z_\sigma^j + k_\mathrm{B}T \ln N_j. \tag{1.17}$$

This chemical potential gives the change in free energy of the solution if a single molecule of species j is added assuming that there is already a large number of species j in the solution. This allows us to analyse the change in free energy due to chemical reactions. If there is a chemical reaction

$$\mathrm{A} + \mathrm{B} \rightleftharpoons \mathrm{C} + \mathrm{D} \tag{1.18}$$

then the change in free energy if this reaction occurs once from left to right is

$$\Delta\mathcal{F} = \mu_\sigma^\mathrm{C} + \mu_\sigma^\mathrm{D} - \mu_\sigma^\mathrm{A} - \mu_\sigma^\mathrm{B}. \tag{1.19}$$

This free energy difference depends on the concentrations of the species involved in the reaction. If the system is allowed to relax to equilibrium then the chemical reaction will occur more often in one direction rather than the other until the free energy difference of the reaction is zero. At this point the free energy is minimum and equilibrium has been reached. This equilibrium state has a certain concentration of the different species.

A common situation is for some of the species to be 'chemostated' so they are fixed at a certain concentration[14]. In this case the other species that are not fixed will eventually reach a steady state concentration. However, this steady state is not necessarily in equilibrium. Consider that the concentration of species A and C are fixed in reaction 1.18. The steady state concentrations of B and D which, in this case, are an equilibrium, depend on the concentrations of A and C. If the concentration of A is increased then the reaction is biased to occur with a higher rate in the rightwards direction so equilibrium concentration of D is higher. In this manner, the concentrations of chemostated species can be used to control the concentration of non-chemostated species.

1.4.1 Catalysts

A Catalyst is a species that takes part in a reaction but is not used up by it. It appears in the same number on both sides of the reaction equation. For example, in the reaction

$$A + B \rightleftharpoons A + C, \tag{1.20}$$

A is a catalyst.

The concentration of a catalyst changes the speed of the reaction but does not change the equilibrium concentrations[13]. They are useful for controlling chemical reactions. The reaction in equation 1.20 can be switched off by the absences of A assuming that any leak reactions such as

$$B \rightleftharpoons C, \tag{1.21}$$

are slow.

We will see in chapter 3 that catalysts are particularly useful for implementing information processing tasks in chemistry. The presence or absence of a catalyst affects a downstream chemical reaction of different species. This can be interpreted as a copying of information[15]. Also the downstream species can also act as a catalyst so two species can simultaneously affect each other. This would be difficult in a non-chemical physical system.

1.4.2 DNA strand displacement

In order to construct models using chemical reaction networks it must be possible to physically implement such a network. If one constructs a chemical reaction network is it implementable? Using DNA strand displacement it is possible to implement any arbitrary chemical reaction network[16,17,18].

DNA consists of a long polymer made of four nucleotide bases: Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). Each base can bind via hydrogen bonds to its complementary base. A binds to T and C binds to G. This bonding enables strands of DNA to bind to each other to form double stranded DNA. The specific binding of the bases means that strands

of DNA can (approximately) only bind to strands with the complementary base sequence.

Bound complementary strands of length greater that approximately 20 base pairs are stable in typical conditions[19]. Individual base pairs bind and unbind spontaneously. Therefore, a short 'toehold' of approximately five base pairs can spontaneously bind and unbind. This allows the creation of a reaction from binding strands. Consider a double stranded DNA where one strand is longer than the other so there is a single stranded toehold at one end: the B species made of strands 2 and 3 in figure 1.2(a). Another single stranded DNA molecule, strand 1 in figure 1.2(a), has a region that is complementary to this exposed toehold so it can spontaneously bind as shown in figure 1.2(b). The base pairs on the end of the a/a′ region of strands 2 and 3 shown in red can unbind spontaneously and in doing so allow the invading strand, strand 1, to bind to those bases as it is also complementary to the a region. Figure 1.2(c) shows an intermediate state where some of the a region on strand 3 is bound to the a′ region of strand 1 and some is bound to the a′ region of strand 2. The position on strand 3 where there is the transition from binding to the original strand 2 to binding to the invading strand 1 does a random walk along the length of strand 3. Eventually, strand 1 has displaced strand 2 and strand 2 is only bound to strand 3 via the toehold region t_2 as shown in figure 1.2(d). This short toehold region can then spontaneously unbind to leave strands 1 and 3 bound and strand 2 separated as shown in figure 1.2(e). Therefore, the overall reaction

$$\mathrm{A} + \mathrm{B} \rightarrow \mathrm{C} + \mathrm{D} \tag{1.22}$$

has occurred.

Of course, in the state shown in figure 1.2(c) it is possible for the random walk of spontaneous binding and unbinding to cause the a′ region of strand 1 to detach and for the molecules to return to the state shown in figure 1.2(b). In fact, the whole reaction is reversible. If the state is initially as shown in figure 1.2(e) then strand 2 can spontaneously bind to strand 3 via the t_2 toehold and the whole reaction can proceed in the reverse direction. It would

be possible to implement an approximately irreversible reaction if the t_2 and t_2' toeholds were removed.

Although the overall reaction in equation 1.22 has occurred, the DNA strand displacement implementation is not exactly the same as the elementary reaction. The kinetics are different to the theoretical elementary reaction because there are many intermediate steps to the reaction. Another difference to the theoretical CRN is that the molecules are sequestered for a time before the reaction occurs.

1.5 Thermodynamics of simple information-processing tasks

The history of the connection between information and thermodynamics began with what is now known as the 'Maxwell's demon' thought experiment (although Maxwell did not use the term 'demon')[20]. In this thought experiment one imagines two volumes of gas at equilibrium and at the same temperature separated by a dividing wall containing a microscopic trapdoor. This is shown in figure 1.3. There is an agent (the demon) that can observe the position and velocity of all of the gas particles and can open and close the trapdoor at will. If the demon allows fast moving particles through the trapdoor in one direction and slow moving particles through the trapdoor in the other direction then one volume of the gas will become hotter and one will become colder. This is shown in figure 1.3 with the faster moving particles shown in red and the slower moving particles shown in blue. This temperature difference could then be used to drive a heat engine to convert heat into work. Opening and closing the frictionless trapdoor does not require any work so it seems that the demon has allowed the conversion of heat into work with no other side affects. The conversion of heat into work with no other side affects violates Kelvin's statement of the second law of thermodynamics. This lead to the assumption that measurement must have some thermodynamic cost but it was still unknown what form the cost took and how to quantify it.

More insight was found when the set-up of the Maxwell's demon was simplified to make the Szilard engine[21]. The Szilard engine uses only a single

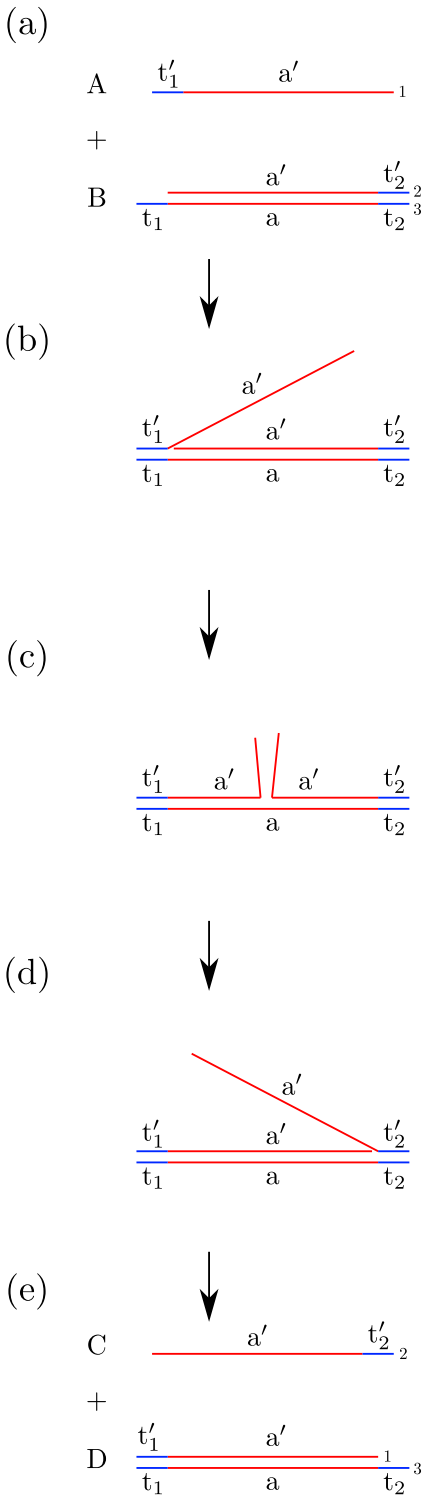


Figure 1.2: The steps of the DNA strand displacement implementation of the reaction A + B → C + D.

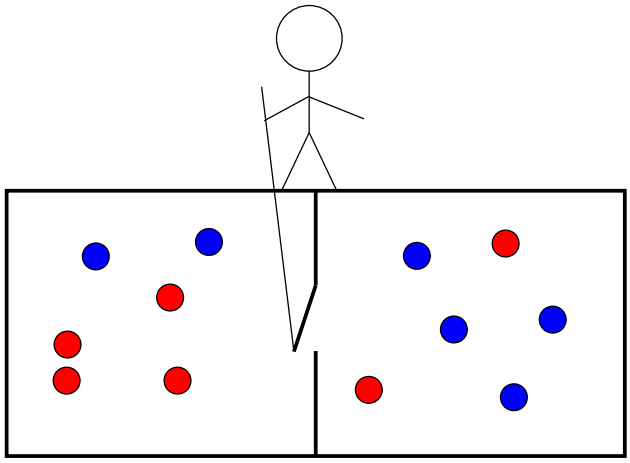


Figure 1.3: Starting with an equilibrium box of gas partitioned into two volumes Maxwell's demon can cause one volume to increase in temperature and the other side to decrease in temperature by measuring the speed of the particles and opening and closing a trapdoor to allow faster particles to move in one direction through the door and slower particles to move in the other direction.

gas particle that can exchange energy with a heat bath at temperature T. The demon measures what side of the partition the particle is on and then attaches a pulley to the partition on the correct side. A quasistatic expansion of the gas then does work by lifting a weight. This process is shown in figure 1.4. This shows that what goes on inside the demon's brain is not important. All that is required is that the side that the pulley is attached becomes correlated with the side the particle is on[21,22].

The Szilard engine shows the amount of work that can be extracted after

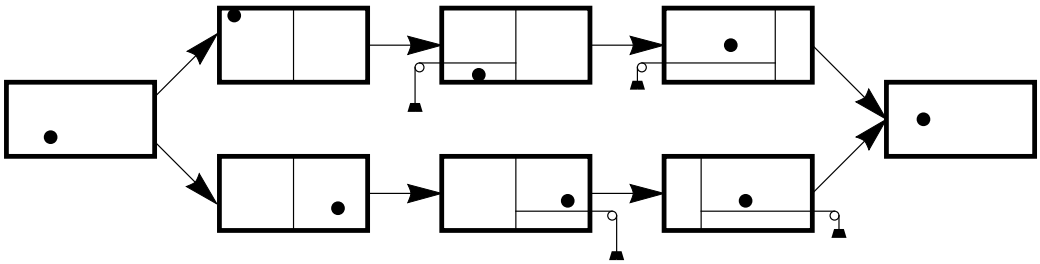


Figure 1.4: In the Szilard engine there is a box containing a single gas particle in equilibrium. A partition is inserted into the box separating it into two volumes: left and right. The demon inserts a weight attached to a pulley on the same side of the partition as the particle. Finally, the gas expands and lifts the weight.

doing a single binary measurement, so the measurement must have used the same amount of work to preserve the second law. It does not show precisely how that work is required.

1.5.1 Mutual information and the second law

In section 1.3 we have seen how to calculate the non-equilibrium free energy of a physical system. Szilard did not have the tools of mutual information and the generalised free energy. Can we apply these tools to analyse fully the Szilard engine? Can we find out where information is important and see when the work is required to be input to preserves the second law?

Consider a system consisting of two subsystems; the overall state of the system is the joint random variable (X, Y) where X and Y are the random variables that describe the individual subsystems. If we assume that the subsystems are not energetically coupled, so that it is possible to write the energy of any joint state as the sum of the energy of the states of the subsystems, then the free energy can be written[23]

$$\mathcal{F}_{\text{joint}}(X, Y) = \mathcal{F}_X(X) + \mathcal{F}_Y(Y) + \mathcal{I}(X; Y), \tag{1.23}$$

Eq. 1.23 shows that there is a real contribution of information to the free energy of a physical system. Fundamentally, correlation between two non-interacting subsystems means that the uncertainty in the state of the joint system is low without a compensating reduction in the energy—work is therefore available.

In terms of the non-equilibrium free energy, the second law of thermodynamics states that the free energy of an isolated system Z can never increase[12]:

$$\Delta\mathcal{F}(Z) = \mathcal{F}(Z, t + \tau) - \mathcal{F}(Z, t) \leq 0. \tag{1.24}$$

Let Z consist of two non-interacting subsystems X and Y, as in equation 1.23, and assume the mutual information between subsystems is zero at the time t. Then for any process between time t and $t + \tau$ that leaves X and Y

non-interacting in the final state,

$$\Delta\mathcal{F}_Y(Y) \leq -\Delta\mathcal{F}_X(X). \tag{1.25}$$

The reduction in the free energy of X can be used to increase the free energy of Y by an amount up to the magnitude of the change in free energy of X. In this thesis we will refer to this increase of free energy of Y as 'work' being performed on the physical system. We will also use work as a shorthand for the more formal term 'chemical work'[24,22]. Therefore, in a process that reduces the free energy of a subsystem X by $\Delta\mathcal{F}_X$ a work of $W \geq -\Delta\mathcal{F}_X$ can be done on another subsystem.

1.5.2 Explaining Szilard's engine with mutual information and the generalised free energy

Now we have this formalism we can apply it to the Szilard engine. The two subsystems are the position of the particle and the position of the pulley. As mentioned in the previous section, the relevant division of the position of the particle into macrostates is the state when the particle is on the left of the partition and the state when the particle is on the right of the partition. The volume on either side of the partition is the same so the free energy of each macrostate is the same. Similarly, there is no energetic difference between the two states of the pulley so only the entropies of the two subsystems change.

Initially, the particle and pulley are equally likely to be in either state so the (macrostate) entropy of each subsystem is $k_\mathrm{B}T\ln 2$. The two subsystems are independent so the mutual information is zero. The zero of the energy is arbitrary so we are free to set it however we want. It is convenient to set the zero of the energy such that the expected free energy of the macrostates is $k_\mathrm{B}T\ln 2$ so that in this initial equilibrium state the joint free energy is zero.

After the measurement each subsystem is equally likely to be in either state so the entropy, and therefore the free energy, of each subsystem has not changed. However, if the particle is on the left then the pulley will be on the left and if the particle is on the right the pulley will be on the right. Therefore, the

mutual information is $\ln 2$ so the free energy of the joint system is $k_{\mathrm{B}}T \ln 2$. This free energy increase must have been associated with a decrease in free energy of the same magnitude of some other system. In the conventional Szilard engine set-up this other system is not specified.

After the expansion of the gas the two subsystems are again uncorrelated so the mutual information is zero so the free energy is zero. The reduction in the free energy of the joint system was used to do work to lift the weight. It might seem mysterious that entropy has been used to increase the hight of a weight. The energy to lift the weight has been transferred from the heat bath.

The Szilard engine makes things clearer than in Maxwell's original thought experiment but there are still some problems. The most important is that it does not explain how the measurement is done and why it requires work to be done. The original question was whether an intelligent agent could break the second law of thermodynamics but instead of showing *how* this is impossible, it is simply assumed that the measurement must cost the precise work required to preserve the second law.

A second issue is that it is assumed that the maximal work can be extracted in a quasistatic expansion that lifts the weight, but the model assumes that a single weight is attached to the pulley. To actually do a quasistatic expansion (or to approximate one) many weights of varying mass must be attached to the pulley in turn so that the force applied to the string by the weight equals the force caused by the pressure of the gas on the partition. This is in principle possible (it does not break any laws of thermodynamics) but it is a considerably more complicated protocol.

A related problem is that of how the engine is controlled. The engine assumes that a quasistatic expansion can be done with no control cost in the same way as for a macroscopic gas in classic thermodynamics problems. Doing so requires an experimenter deciding when to start and stop the expansion. Forcing the engine to move in one direction rather than the other requires entropy production to break the time symmetry[25]. This is usually negligible compared to the magnitude of free energies involved in the expansion or

compression of a macroscopic gas. However, for a gas of a single molecule where the costs and work done are of order $k_{\mathrm{B}}T$ they are not negligible. The biochemical Szilard engine proposed in chapter 3 addresses some of these concerns.

1.6 Devices that extract work from information

The previous section showed that there is a fundamental link between information and thermodynamics. That means that information can create a high free energy resource. In the Szilard engine after the measurement step there is a correlation between the particle and the pulley and this correlation is then used to extract work from heat in a heat bath. What if there was some already existing correlations? Then a engine could use these correlations to do useful work. A schematic of such a machine is shown in figure 1.5.

Such a machine was proposed by Mandal and Jarzynski[26]. They proposed a three state engine that interacts with a string of bits and uses the information in the string to lift a weight. Their machine exploits the fact that the string of bits is initially biased towards zero rather than one. After the machine has interacted with a bit its bias is decreased so it is closer to being equally likely to be either a zero or a one. Because there is no structure in the string of bits and there is only an overall bias towards one state this machine is not really doing anything complicated with information. This biased string of bits is equivalent to a bath with a chemical potential or a pressure differential between two volumes of gas. Considering abstract bits rather than a physical implementation makes the fact that work can be done by randomising bits seem more mysterious than it actually is. Using a chemical potential to do work is already exploring an entropic/informational resource.

A chemical implementation of this machine was given Barato and Seifert[27]. Also, in this set-up the machine was allowed to interact with the bits in a random order. This makes it more clear that it is a chemical potential that is being exploited. Also, this version of the machine does not require an experimenter to sequentially connect the machine to each bit and so is autonomous.

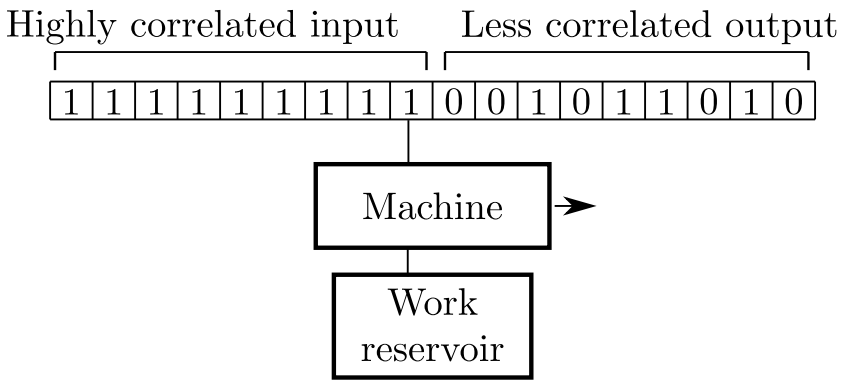


Figure 1.5: An information machine can extract work from a string of correlated bits.

McGrath *et al.*[28] proposed a chemical machine to exploit the correlations between two strings of bits instantiated as two polymers. This is perhaps more interesting but it still only exploited an overall bias.

Machines that exploit correlations between successive bits in a sequence but with no overall bias to zero or one have been explored by Boyd, Mandal and Crutchfield[29,30,31]. They showed that the maximum work that could be extracted per bit from a string of bits is the entropy rate. The entropy rate of a string of bits is

$$h = \lim_{n\to\infty} \frac{H[X_1, \ldots, X_n]}{n}, \tag{1.26}$$

where X_i is the random variable that describes the ith bit.

As has been highlighted by Stopnitzky et al[32], however, the machines in these works were designed without 'reversibly embeddable' dynamics—a necessity if the machines are to operate without external control, as was assumed. Stopnitzky et al did present systems with reversibly embeddable dynamics that extract positive work from a perfect sequence of alternating 1s and 0s, but the efficiency was very low. This is an example of how one can be mistaken about the thermodynamics of a system without a concrete model.

All of the machines discussed in this section run in finite time. Therefore, these machines do not extract all of the work available in the input bit sequences. It would be useful to analyse machines that reach the fundamental limits of the amount of work that can be extracted.

There is link between these machines that extract work from correlated inputs and the limits of cell sensing. A string of bits can be thought of as a very simple model of the environment of a micro-organism. Feeding models are convenient to consider because it is easy to compare the energy that sensing costs with the energy that can be extracted from food using the sensing. A feeding model of this kind is investigated in chapter 3.

1.7 Cost of computing

In the previous section we discussed machines that convert information in a string of bits to useful work. It would also be possible for a machine to do the reverse: to convert work into correlations within a sequence of bits. This is related to but subtly different from computation. One could say that a computer uses work from electricity to produce information. However, what we usually mean by computation is not the same as correlating some previously uncorrelated physical systems. Further, a simple lever converts the 'input' of the position of one end of the lever into an 'output' of the other end of the lever but most would not consider that a computation. However, in a fundamental way one could say that a classical computer is no different in that the state of one part of the computer is deterministically transformed into the state of a different part of the system.

We have seen in section 1.5.1 that the work required to transform a physical system from one state to another is equal to the change in non-equilibrium free energy between the initial and final states, which can include the mutual information between energetically uncoupled subsystems. Therefore, if a computation involves increasing the free energy of the computer then there is a lower bound on the work required for the computation. This is a generalisation of Landauer's principle[33]. In some discussions of Landauer's principle the requirement that work is input is mistakenly taken to mean that there is irreversible dissipation, i.e., entropy production. This is not correct. The work input is used to increase the free energy of the computer and it is in principle possible, after, say, the output is read, to reduce the free energy of the computer to the initial value while doing work.

However, this simple application of the second law to give a bound on the work is far from the whole story of the thermodynamics cost of computing. It assumes that one has full control over all parts of the computer. It requires knowledge about the state of the computer during the computation. A more interesting question is: what the costs are when the computer is constrained to be a certain model? We are also interested in what other resources are required. i.e. what complexity of control is needed or, as we will see, additional states are required? How does doing the computation in a series of small steps affect the cost?

If the computer has a discrete set of states then a computation that transforms an initial state into a final state can be described by a stochastic matrix. The question is: is it possible to apply such a transformation using a continuous time control and, if so, is it possible to do so quasisatically? This is called the embedding problem. It has been shown by Owen, Kolchinsky and Wolpert that any transformation can be applied quasistatically if additional 'hidden' states are appended to the states of the system[34]. They showed that the upper bound on the number of hidden states required is one less than the nonnegative rank of the stochastic matrix. The nonnegative rank is bounded by the dimension of the matrix so at most one additional hidden bit is required for any transformation to be implemented quasistatically.

1.7.1 Models of computation

An early theoretical model of computation is the Turing machine. The machine consists of a 'tape' containing evenly spaced symbols from a finite alphabet and a 'head' with a state from a finite set. The head reads the symbol at a certain position and then, depending on the symbol and the head's state, writes a new symbol at that position, moves either left or right along the tape and changes the head state[35]. Bennett showed how to modify this model using a second tape to record all of the transitions to make it thermodynamically reversible. In chapter 4 we will propose a chemical version of this model in order to investigate practical issues with this machine.

Another model of computation is a circuit. A circuit has the same computa-

tional power as a Turing machine if the number of gates is allowed to grow with the number of inputs in the sense that a circuit can do any computation that a Turing machine can do. An analysis of the cost of a circuit was done by Wolpert and Kolchinsky[36].

One element of the cost of a circuit is a 'modularity cost'[37]. As previously mentioned, if one has total control over all parts of a physical system then it is possible to quasistatically change the state so only input the work required to increase the free energy. However, if one only controls part of a system in a way that is independent of the state of the rest of the system then there is entropy production due to the reduction in mutual information between the controlled part of the system and the rest of the system. This entropy production is the modularity cost.

1.8 Thesis overview

In chapter 2 we investigate the interpretation of a quantity called the learning rate that has been proposed to measure the quality of sensors. We explore the behaviour of the learning rate for a number of simple model systems and we find that it is not generically a measure of the quality of a steady state sensor. Instead, the learning rate reflects the rate at which the downstream system must update its information in response to changes in the upstream system.

In chapter 3 we investigate the extraction of work from complex environments by minimal machines with finite capacity for memory and decision-making by designing and leveraging an explicit molecular realisation of a measurement-and-feedback-powered Szilard engine. We demonstrate that increasingly complex environments allow increasingly sophisticated inference strategies to extract more free energy than simpler alternatives. We argue that optimal design of such machines should also consider the free energy reserves required to ensure robustness against fluctuations due to mistakes.

In chapter 4, leverageing the ideas in the chemical Szilard engine, we propose a design of a chemical reversible Turing machine. Our explicit construction

gives an upper bound on the complexity of the set-up that is required to implement a Turing machine. A concrete design brings into focus some issues that are neglected in more abstract discussions. In particular, we argue that the controller must operate many machines simultaneously and therefore extra effort is required to detect when the machines have halted.

1.9 Publications and Contributions

R. A. Brittain, N. S. Jones and T. E. Ouldridge. What we learn from the learning rate **J. Stat. Mech.** 2017(6):063502 2017.
Appearance in thesis: Chapter 2.
Contribution: I did the research and wrote the paper.

R. A. Brittain, N. S. Jones and T. E. Ouldridge. Biochemical Szilard engines for memory-limited inference **New J. Phys.** 21(6):063022, 2019.
Appearance in thesis: Chapter 3.
Contribution: I did the research and wrote the paper.

R. A. Brittain, N. S. Jones and T. E. Ouldridge. Reversible chemical Turing machines. In preparation.
Appearance in thesis: Chapter 4.
Contribution: I conceived of the idea, did the research and wrote the paper.

2

What we learn from the learning rate

2.1 Abstract

The learning rate is an information-theoretical quantity for bipartite Markov chains describing two coupled subsystems. It is defined as the rate at which transitions in the downstream subsystem tend to increase the mutual information between the two subsystems, and is bounded by the dissipation arising from these transitions. Its physical interpretation, however, is unclear, although it has been used as a metric for the sensing performance of the downstream subsystem. In this chapter, we explore the behaviour of the learning rate for a number of simple model systems, establishing when and how its behaviour is distinct from the instantaneous mutual information between subsystems. In the simplest case, the two are almost equivalent. In more complex steady-state systems, the mutual information and the learning rate behave qualitatively distinctly, with the learning rate clearly now reflecting the rate at which the downstream system must update its information in response to changes in the upstream system. It is not clear whether this

quantity is the most natural measure for sensor performance, and, indeed, we provide an example in which optimising the learning rate over a region of parameter space of the downstream system yields an apparently sub-optimal sensor.

2.2 Introduction

As discussed in section 1.1 it is vital for living things to sense their environments and one would imagine that the sensing abilities of evolved organisms are close to the physical limits. In order to investigate what these limits are and measure how real biological sensors compare we must have a way to quantify the quality of a sensor.

When the states of the signal and sensor are described by random variables X and Y a natural measure of sensing quality is the mutual information, $I[X;Y]$, as discussed in section 1.2. A related quantity, l_Y, called the learning rate has been proposed as a metric for the performance a sensor when the signal/sensor system is modelled as a bipartite Markov chain[8,9]. The learning rate is bounded by the entropy production of the sensor so this appears to put an energetic bound on sensing quality. However, it is not clear that it is the most natural measure of sensing.

In this chapter, after defining the learning rate we will consider its behaviour for three simple steady-state systems, identifying and explaining differences between the physical content of l_Y and $I[X;Y]$. Typically, differences in behaviour arise because the learning rate quantifies the rate at which transitions in Y must act to restore information between X and Y, which is not necessarily closely related to the steady-state level of information or correlation. As a result, we are able to identify a fourth system in which optimizing over a parameter of the sensor at fixed input signal dynamics gives markedly different results when using the learning rate and the mutual information as metrics. In this case, optimising the mutual information provides the more intuitively reasonable optimal sensor.

2.3 Set up: Sensing and thermodynamics in bipartite Markov chains

Bipartite Markov chains[10,38] are a central tool in modelling Maxwell's demon-like behaviour and cellular sensing circuits. A bipartite Markov chain has states that are labelled with two variables x and y, which are generally taken to correspond to the states of two subsystems X and Y. Transitions only change one of the variables so the transition rate from (x, y) to (x', y') is

$$w_{yy'}^{xx'} = \begin{cases} w_y^{xx'} & \text{if } y = y' \text{ and } x \neq x', \\ w_{yy'}^{x} & \text{if } y \neq y' \text{ and } x = x', \\ 0 & \text{if } y \neq y' \text{ and } x \neq x'. \end{cases} \tag{2.1}$$

These rates have dimensions of inverse time. Throughout this chapter we will use arbitrary units. The systems we will consider have a discrete state space and continuous time.

The second law of thermodynamics constrains the entropy production of the system to be positive. We can use the bipartite structure to separate the entropy production, $\dot{S}_i$, into six parts[10]

$$\dot{S}_i = \mathrm{d}_t S^X + \dot{S}_r^X - \dot{I}^X + \mathrm{d}_t S^Y + \dot{S}_r^Y - \dot{I}^Y \geq 0 \tag{2.2}$$

where the rates of increase of entropy of the X and Y subsystems are

$$\begin{aligned} \mathrm{d}_t S^X &= \sum_{xx'y} w_y^{xx'} p(x, y) \log \frac{p(x)}{p(x')}, \\ \mathrm{d}_t S^Y &= \sum_{xyy'} w_{yy'}^{x} p(x, y) \log \frac{p(y)}{p(y')}, \end{aligned} \tag{2.3}$$

the rates of increase of the entropy of the environment due to transitions in

X and Y are

$$\begin{aligned}\dot{S}_r^X &= \sum_{xx'y} w_y^{xx'} p(x,y) \log \frac{w_y^{xx'}}{w_y^{x'x}}, \\ \dot{S}_r^Y &= \sum_{xyy'} w_{yy'}^{x} p(x,y) \log \frac{w_{yy'}^{x}}{w_{y'y}^{x}}\end{aligned} \tag{2.4}$$

and

$$\begin{aligned}\dot{I}^X &\equiv \sum_{xx'y} w_y^{xx'} p(x,y) \log \frac{p(y|x')}{p(y|x)}, \\ \dot{I}^Y &\equiv \sum_{xyy'} w_{yy'}^{x} p(x,y) \log \frac{p(x|y')}{p(x|y)}.\end{aligned} \tag{2.5}$$

Note that in Equation 2.4, if a single change of state of X, Y can be associated with multiple distinct reaction pathways, these separate pathways must be explicitly considered as separate contributions to the sum[39]. The flows $\dot{I}^X$ and $\dot{I}^Y$ are so called because they represent changes in the mutual information between X and Y due to transitions in X and Y respectively,

$$\frac{\mathrm{d}}{\mathrm{d}t} I[X;Y] = \dot{I}^X + \dot{I}^Y. \tag{2.6}$$

Importantly, the second law holds not just for Equation 2.2, but also for each subsytem individually[40,10,8]

$$\begin{aligned}\mathrm{d}_t S^X + \dot{S}_r^X - \dot{I}^X \geq 0, \\ \mathrm{d}_t S^Y + \dot{S}_r^Y - \dot{I}^Y \geq 0.\end{aligned} \tag{2.7}$$

This form of the second law has been extended by Horowitz[41] to systems with multiple subsystems that have independent noise, and has been used by Goldt and Seifert[42] to find the thermodynamic bound on the information learned by a feedforward neural network with Markovian dynamics.

We can identify $\sigma^X = \mathrm{d}_t S^X + \dot{S}_r^X$ and $\sigma^Y = \mathrm{d}_t S^Y + \dot{S}_r^Y$ as the entropy productions of X and Y, ignoring the correlations between the two subsystems. $\dot{I}^Y < 0$ then allows for $\sigma^Y < 0$—correlations between subsystems allow for

apparent violation of the second law if the subsystems are erroneously considered in isolation. If $\sigma^Y < 0$ without direct energy flow from X to Y, X can be interpreted as a demon acting on subsystem Y. In terms of sensing, Y is generally interpreted as a sensory network for the signal X.

In steady state the mutual information is constant and so

$$\dot{I}^X + \dot{I}^Y = 0. \tag{2.8}$$

In this case, transitions in one subsystem must reduce the mutual information as much as the transitions in the other subsystem increase it.

The quantities $\dot{I}^Y$ and $\dot{I}^X$ have been introduced in Equation 2.2 to separate the second law into two parts. But since $\dot{I}^Y$ represents the rate at which transitions in Y tend to increase information between X and Y, it has been named the learning rate $l_Y = \dot{I}^Y$ and proposed as a metric for the performance of Y as a sensor of X[8,9]. Identifying l_Y as a measure of sensory performance also sets a thermodynamic bound on sensor function, since from Equation 2.7, $l_Y \leq \sigma^Y$, the entropy generation due to transitions in Y. It is straightforward to show that one can equivalently write the learning rate as[9]

$$l_Y = \frac{\mathrm{d}}{\mathrm{d}\tau} I[X_t; Y_{t+\tau}]\Big|_{\tau=0}. \tag{2.9}$$

Intuitively, l_Y is the degree to which future values of Y are more predictive of the current value of X than the current value of Y is, due to Y continuing to learn about X.

Although the use of the learning rate to quantify sensor performance seems reasonable from its definition, its physical interpretation has not been extensively explored. Other authors, such as Das *et al.*[43], have preferred the instantaneous mutual information, $I[X;Y]$, over l_Y as a metric for performance when deriving thermodynamic constraints on sensor function. Is l_Y essentially reporting the degree of interdependence between X and Y, similar to quantities like $I[X;Y]$ or the covariance $\mathrm{Cov}(X,Y)$? Or does it highlight distinct features of the network? Is l_Y informative when trying to build optimal sensors for a given input signal? A further interpretational challenge

occurs in steady state (a common setting for sensors). In this case[8,9,43]

$$l_Y = -\sum_{xyy'} w^x_{yy'} p(x,y) \log \frac{p(x,y)}{p(x,y')}, \tag{2.10}$$

and

$$l_Y = -\frac{\mathrm{d}}{\mathrm{d}\tau} I[X_{t+\tau}; Y_t]\Big|_{\tau=0}. \tag{2.11}$$

Equation 2.11 suggests that learning rate also reflects 'nostalgia', a quantity defined for discrete-time systems by Still *et al.*[44]. In steady state, l_Y not only reflects how much more the sensor Y learns about the current value of X as time progresses incrementally (Equation 2.9), but also the degree to which Y has nostalgic information about the history of X that is irrelevant to future signals (Equation 2.11). From this 'nostalgia' perspective, it is unclear why large values of l_Y would correspond to effective sensing.

2.4 Results and Discussion

2.4.1 The learning rate reports on correlations in the simplest bipartite sensing system

We first consider a canonical steady-state sensing system involving a two-state signal X and a two-state sensor Y, in which transition propensities for X, $w^{xx\prime}_y$, are Y-independent[45,8]. This 2×2 system is the simplest possible sensing setup. In biophysical terms, this system (illustrated in figure 2.1) would correspond to a single receptor that can bind to ligand molecules, which is exposed to two discrete concentrations of ligands in its external environment[8]. The states $x = 0, 1$ correspond to low and high concentrations, and $y = 0, 1$ to unbound and bound receptors, respectively.

Let us, for convenience, assume that the high concentration is twice that of the low concentration. The rate at which the environment transitions from the low state to the high is $k_\uparrow$ and the reverse transition occurs at a rate $k_\downarrow$, independent of the receptor state. We only consider a single receptor molecule, which can be in two states: bound and unbound. The rate for a

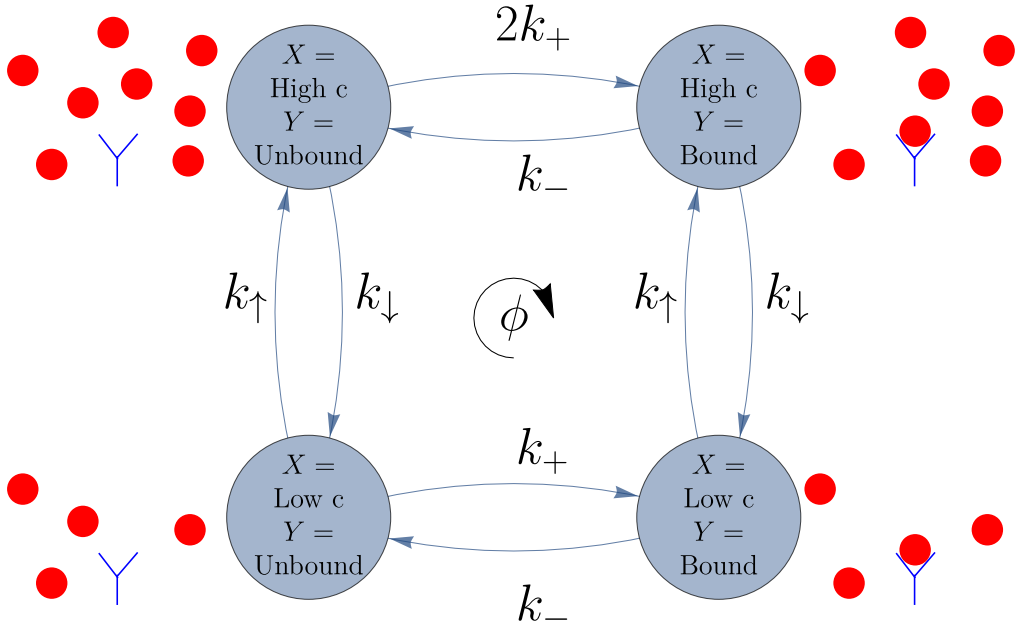


Figure 2.1: The states and transitions for a simple system involving a two-state signal and a two-state sensor, in which the signal transitions are independent of the sensor state. We illustrate a biophysical interpretation of such a system, corresponding to an environment that can switch between high and low concentrations of ligand molecules, and a sensor that can bind to and unbind from the ligands. The rate of binding is proportional to the ligand concentration and the high concentration state has twice the concentration of the low concentration state in this illustration. The ligand concentration is absorbed into the transition rates k_+ and $2k_+$. We illustrate the flux ϕ, the net flow of trajectories around the state space in steady state.

ligand molecule to bind to the receptor is proportional to the ligand concentration (i.e. mass-action kinetics and a well-mixed system are assumed). The concentrations are absorbed into the transition rates k_+ and $2k_+$ for ligand-binding in low and high concentration conditions, respectively. The rate of unbinding, k_-, is independent of the ligand concentration.

In figure 2.2, we plot the instantaneous mutual information $I[X;Y]$, the learning rate l_Y and the covariance $\mathrm{Cov(X,Y)}$ as a function of k_+ and k_-, at fixed $k_\uparrow$ and $k_\downarrow$ (in this case, $k_\uparrow = k_\downarrow$). Although the functions are not identical, the dependence on k_+ and k_- is very similar for all three. In particular, if $I[X;Y]$ is greater at (k_+, k_-) than at (k'_+, k'_-) then it is usually true that l_Y is greater at (k_+, k_-) than at (k'_+, k'_-). The metrics are therefore almost equivalent for the purpose of optimising this sensor design. Deviations

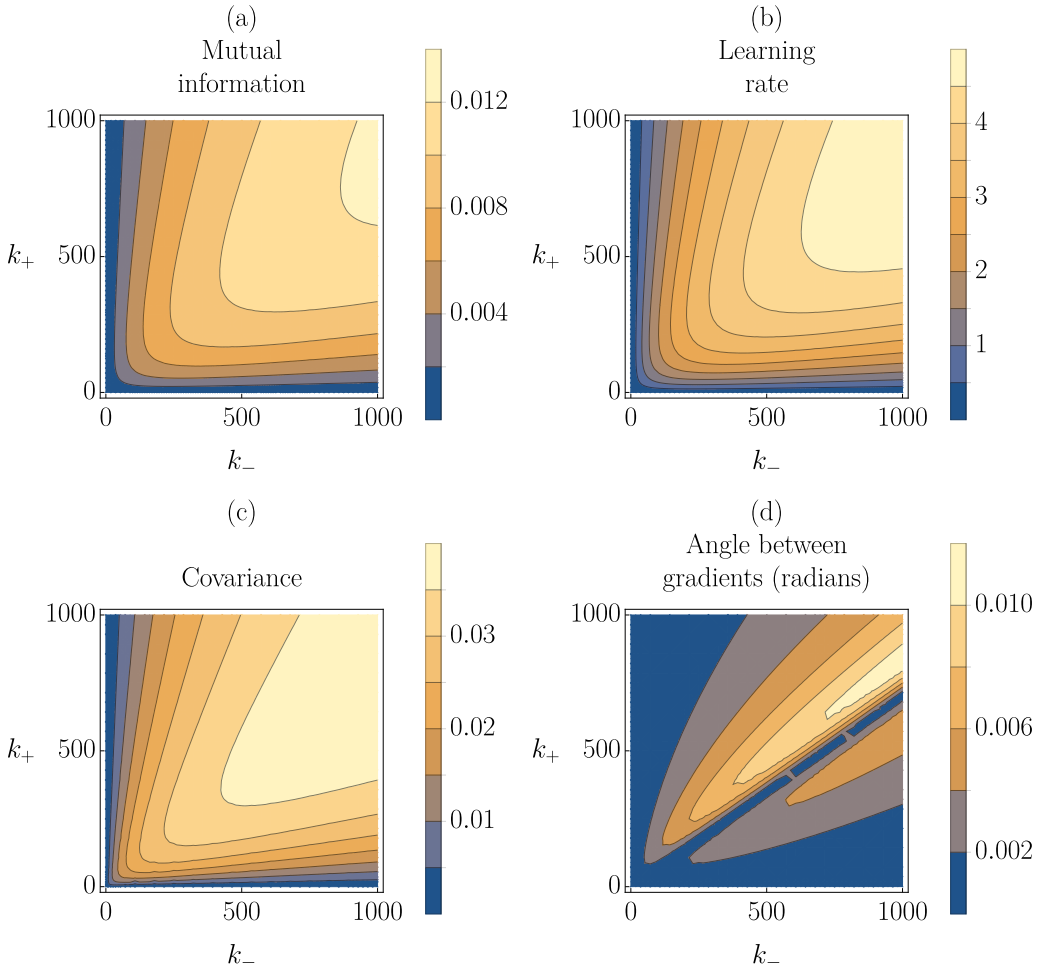


Figure 2.2: The mutual information, learning rate and covariance all show similar behaviour as a function of sensor parameters for the simplest possible sensing device, illustrated in figure 2.1. These quantities are plotted in (a), (b) and (c), respectively, as a function of k_+ and k_-, for $k_\uparrow = k_\downarrow = 100$. In (d), we plot the angle between $\nabla I[X;Y]$ and ∇l_Y, where ∇ is defined as differentiation with respect to (k_+, k_-).The small angles observed suggest that there is little difference between optimizing for l_Y and for $I[X;Y]$.

from this equivalence are manifest as non-parallel gradients of $I[X;Y]$ and l_Y with respect to k_+ and k_-; we plot the angle between gradients as a function of k_+ and k_- in figure 2.2 (d), showing that it is small.

It is possible to understand this similarity from the underlying definition of the learning rate. Using the notation $p(x,y) = p_{xy}$, then Equation 2.10 can be expanded as

$$\begin{aligned} l_Y = &- \left(w^0_{01}p_{00} - w^0_{10}p_{01}\right) \ln \frac{p_{00}}{p_{01}} \\ &- \left(w^1_{10}p_{11} - w^1_{01}p_{10}\right) \ln \frac{p_{11}}{p_{10}}. \end{aligned} \tag{2.12}$$

This simple system has one loop of states. In the steady state, any non-zero clockwise flux $\phi = w^0_{10}p_{01} - w^0_{01}p_{00}$ between the $(0,1)$ and $(0,0)$ states must be balanced by an equal clockwise flux for all other pairs of states around the loop, as illustrated in figure 2.1. In particular, $\phi = w^1_{01}p_{10} - w^1_{10}p_{11}$ so

$$l_Y = \phi \ln \frac{p_{00}p_{11}}{p_{01}p_{10}}. \tag{2.13}$$

The logarithm in Equation 2.13 is known as the 'information affinity'[10,39], and it reflects the informational driving force exerted by one subsystem on the other. Clearly, it is related to the correlation between X and Y. If X and Y are more likely to be in the same state than different states then the affinity is positive. If they are more likely to be in different states then the affinity is negative.

The flux ϕ is the conjugate current to the information affinity[10,39]. For this system the flux can be written

$$\begin{aligned} \phi &= p_{11}k_\downarrow - p_{01}k_\uparrow \\ &= p(Y=1|X=1)p(X=1)k_\downarrow \\ &\quad - p(Y=1|X=0)p(X=0)k_\uparrow. \end{aligned} \tag{2.14}$$

In this simple case we have assumed that X transitions are independent of Y, and thus $p(x)$ is determined solely by $k_\uparrow$ and $k_\downarrow$: $p(X=1) = k_\uparrow/(k_\uparrow + k_\downarrow)$

and $p(X=0) = k_\downarrow/(k_\uparrow + k_\downarrow)$. Thus

$$\phi = \frac{k_\uparrow k_\downarrow}{k_\uparrow + k_\downarrow}\Big(p(Y=1|X=1) - p(Y=1|X=0)\Big). \tag{2.15}$$

The quotient in Equation 2.15 is independent of k_+ and k_-. It reflects the overall timescale of the process, which is manifest in the learning rate since l_Y is a dimensional quantity, and the uncertainty or entropy in X. The second part is clearly related, again, to the correlation between X and Y: it is large and positive when they are correlated, and large and negative when anticorrelated, just like the information affinity.

Thus, for a 2×2 bipartite system, in which the value of Y does not influence the transitions in X, the learning rate l_Y essentially reports on the correlation between X and Y, and is thus closely related to the mutual information $I[X;Y]$ or covariance between X and Y. l_Y also incorporates a prefactor that reflects the overall timescale of the process. In sections 2.4.2 and 2.4.3, we shall violate the assumptions that lead to this conclusion and explore the consequences for the relationship between learning rate and mutual information.

2.4.2 Feedback from the downstream to the upstream system

The simplest way to violate the assumptions of section 2.4.1 is to allow the signal transition rates $w_y^{xx'}$ to depend on Y, whilst retaining the 2×2 structure of the overall system. To provide an intuition for such a setting, it is useful to consider the molecular system illustrated in figure 2.3, containing two bound molecules, X and Y. Both molecules have two states; for concreteness, we imagine that X can be phosphorylated (bound to an inorganic phosphate moiety) or unphosphorylated, corresponding to states of a random variable $X = 1$ and $X = 0$, respectively. Molecule Y, by contrast, can adopt two allosteric configurations, and an energetic coupling between X and Y favours states $X, Y = 1, 1$ and $0, 0$ relative to $1, 0$ and $0, 1$.

If these molecules are in contact only with a bath of phosphate, they will eventually reach a thermodynamic equilibrium described by a Gibbs distri-

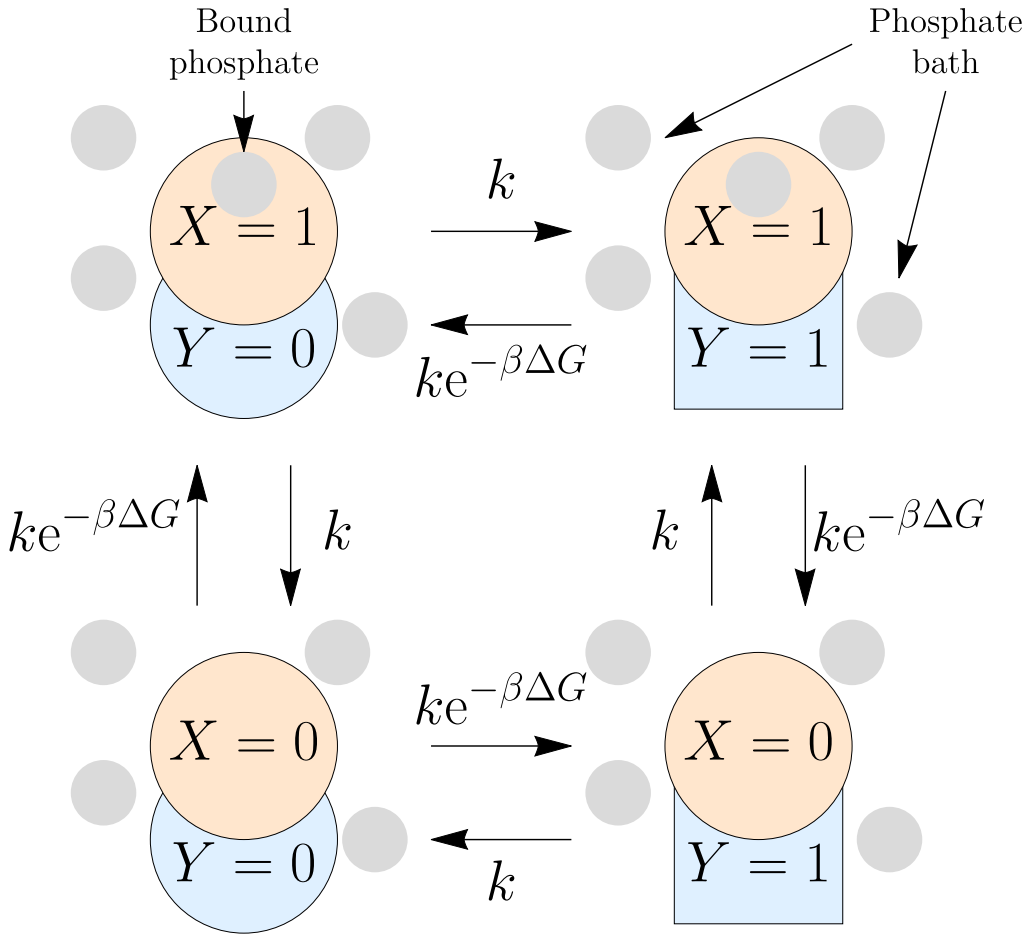


Figure 2.3: A simple equilibrium molecular system, in which the X and Y represent the configurations of two proteins, X and Y. X can be phosphate-bound or unbound, and Y has two allosteric conformations. Relative rates of forwards and backwards transitions are set by overall free-energy differences.

bution. Assuming for simplicity that the free energies of the 11 and 00 states are equal, and ΔG below the free energies of 10 and 01,

$$p_{10} = p_{01} = \frac{\mathrm{e}^{-\beta\Delta G}}{Z}$$
$$p_{00} = p_{11} = \frac{1}{Z}, \tag{2.16}$$

where $Z = 2 + 2\mathrm{e}^{-\beta\Delta G}$ is the partition function, and β is the inverse temperature $1/k_{\mathrm{B}}T$. At equilibrium, the system must obey detailed balance:

$$\frac{w_{01}^{0}}{w_{10}^{0}} = \frac{w_{10}^{1}}{w_{01}^{1}} = \frac{w_{0}^{01}}{w_{0}^{10}} = \frac{w_{1}^{10}}{w_{1}^{01}} = \mathrm{e}^{-\beta\Delta G}. \tag{2.17}$$

The simplest choice consistent with this requirement is $w_{01}^{0} = w_{10}^{1} = w_{0}^{01} = w_{1}^{10} = k\exp(-\beta\Delta G)$; $w_{10}^{0} = w_{01}^{1} = w_{0}^{10} = w_{1}^{01} = k$.

The transitions in X are thus strongly influenced by the state of Y—a natural and necessary feature in an equilibrium system involving coupling between X and Y[46]. One can still evaluate Equation 2.10 in this setting, obtaining a nominal $l_Y = 0$ for all parameter choices. Indeed, l_Y is always zero in thermodynamic equilibrium; the contributions arising from each transition in Equation 2.10 must cancel with those arising from the reverse transitions. Similarly, Equation 2.13 still holds, but the flux ϕ is necessarily zero due to detailed balance, even though $I[X;Y]$ and $\mathrm{Cov}(X,Y)$ can be large ($I[X;Y] = 0.56$ for $\beta\Delta G = 3.5$).

This equilibrium system and the driven system considered in section 2.4.1 are at opposite ends of a spectrum. In the equilibrium case, the influence of X on Y and Y on X are symmetric, in the sense that both experience the same biasing due to the interaction. Thus, there is no real sense in which Y is a sensor for X; it causes changes in X as much as it reacts to them. For the driven system in section 2.4.1, the influence is totally asymmetric; X influences Y but Y does not influence X at all, which is possible because the external process X is strongly driven. In this setting Y is a purely reactionary sensor.

One might argue that the 'learning rate' is no longer a meaningful term

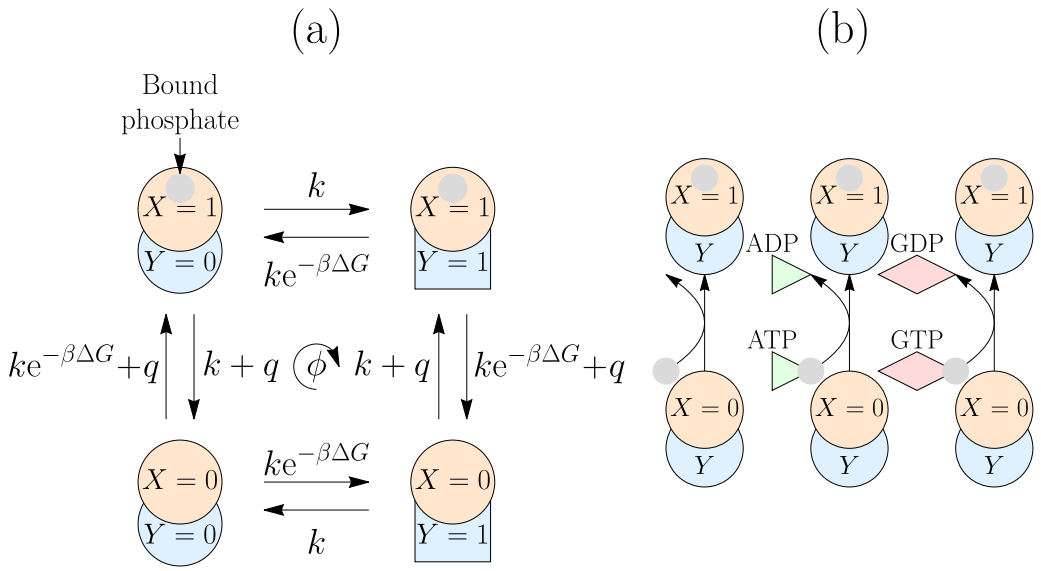


Figure 2.4: (a) An extension of the system in figure 2.3 in which it is possible to interpolate between the equilibrium limit and the limit where X is not influenced by Y by changing the ratio q/k. (b) This system could be implemented using three separate pathways for phosphorylation of X. The pathways are by binding phosphate from solution, or via donation of phosphate from ATP or GTP.

for the quantity defined by the second line of Equation 2.5 in the setting where Y influences X. However, l_Y still quantifies the degree to which Y systematically reacts to (or learns about) X. To see this in more detail, we now take the unusual step of interpolating between the two limits of equilibrium and strong driving. We do this by adding a finite-strength non-equilibrium driving term to the equilibrium system, to explore the response of the learning rate.

This interpolation takes the form of additional transitions of rate q between X and X*, independent of the state of molecule Y, alongside the original transitions of rate k and $k\exp(-\beta\Delta G)$. This system is illustrated in figure 2.4 (a). We can vary q/k to interpolate between equilibrium ($q/k \to 0$) and strongly-driven ($q/k \to \infty$) systems; in the second case, X is independent of Y.

A possible way to implement this scheme in a molecular setting is to drive the system by allowing phosphorylation of X to occur by two additional

pathways as well as exchange of phosphate with solution. These additional pathways proceed via exchange of phosphate with nucleotides supplied by large buffers or chemostats, as shown in figure 2.4 (b). With phosphate P and nucleotides ATP, ADP, GTP and GDP in solution, molecule X can change phosphorylation state by the following reactions

$$\begin{aligned} \mathrm{P} + \mathrm{X} &\rightleftharpoons \mathrm{X}^*, \\ \mathrm{ATP} + \mathrm{X} &\rightleftharpoons \mathrm{ADP} + \mathrm{X}^*, \\ \mathrm{GTP} + \mathrm{X} &\rightleftharpoons \mathrm{GDP} + \mathrm{X}^*. \end{aligned} \tag{2.18}$$

Here, X^* represents the molecule in the phosphorylated state. The concentrations of ATP, ADP, GTP and GDP contribute to the overall free energy changes of the second and third reactions; setting their concentrations to be incommensurate with the free energy change for $\mathrm{P} + \mathrm{X} \rightleftharpoons \mathrm{X}^*$ amounts to driving the system so that it cannot reach equilibrium. For example, if [ADP] and [GTP] are both kept low, X will frequently be converted into X^* through interactions with ATP, but will be converted from X^* to X by interactions with GDP. The relative rate of these two reactions is not constrained by the relative rate of the two reactions involved in direct exchange of phosphate with solution, $\mathrm{P} + \mathrm{X} \rightleftharpoons \mathrm{X}^*$, since the microscopic processes are distinct. Via this scheme it is therefore possible, at least in principle, to modify the stochastic process as shown in figure 2.4 (a). In particular, we could imagine that ATP/ADP coupling is only possible in the $Y = 0$ state, and GTP/GDP coupling is only possible when $Y = 1$. The concentrations could then be set so that the free energy changes of ATP/ADP and GTP/GDP interconversion exactly cancel with the $\pm\Delta G$ associated with the change of state of X, Y in these two cases, allowing phosphorylation/dephosphorylation by these pathways to have equal transition rates.

We perform the interpolation between equilibrium and driven systems in figure 2.5 (a), where we have plotted the mutual information and learning rate against q/k for $\beta\Delta G = 3.5$ at steady state. In this system the mutual information and learning rate behave quite differently. At $q/k = 0$, $I[X;Y]$ is large and its size is set by the energy gap; as $\beta\Delta G \to \infty$, $I[X;Y] \to \ln 2$,

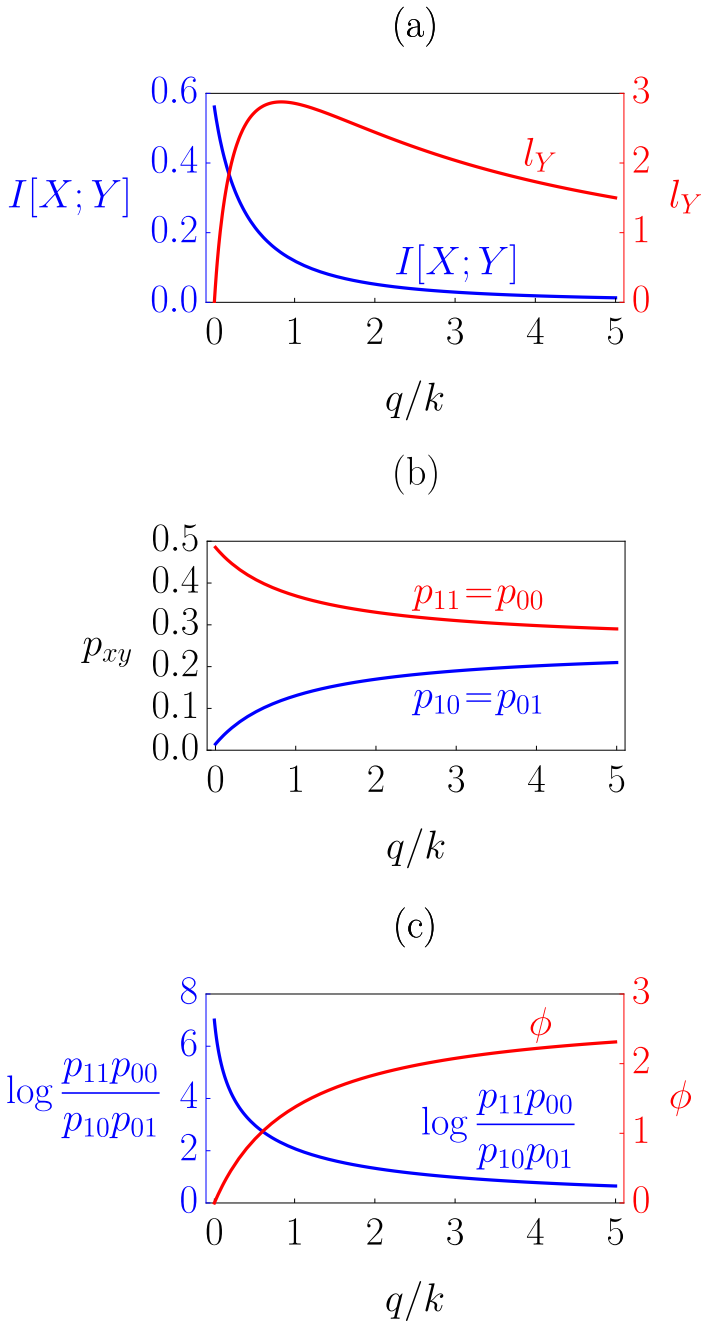


Figure 2.5: The behaviour of a 2×2 system that interpolates between equilibrium and driven limits, illustrated in figure 2.4 (a). Plots obtained with $\beta\Delta G = 3.5$ and $k = 11.5$. (a) The mutual information and learning rate behave distinctly. The learning rate is zero in the equilibrium limit ($q/k \to 0$) and then increases to a peak with increased driving q. The mutual information is highest at equilibrium and decreases as driving increases. (b) The non-equilibrium steady state probabilities. (c) Two competing contributions to l_Y; the flux, ϕ, and information affinity, $\ln(p_{11}p_{00}/p_{10}p_{01})$, show opposite dependencies on q/k, leading to a peak at finite q/k.

corresponding to perfect correlation. As q/k increases, X transitions start to become increasingly decoupled from the state of Y, and also much faster. Correlations are thus destroyed as shown in figure 2.5 (b). The X state changes randomly and too fast for Y to track it and consequently, $I[X;Y] \to 0$.

By contrast, the learning rate is zero for $q/k = 0$, has a peak at $q/k \sim 1$ and decays to zero as $q/k \to \infty$. To understand this behaviour, it is helpful to consider Equation 2.13. For a 2×2 state system the learning rate is equal to the flux ϕ around the states multiplied by the information affinity. As shown in figure 2.5 (c), the information affinity behaves very similarly to $I[X;Y]$. This is because, as we have discussed in section 2.4.1, the information affinity measures the correlation between X and Y and increasing q/k decreases the correlation. Therefore, the ratio $\frac{p_{11}p_{00}}{p_{01}p_{10}}$ becomes closer to 1 so the information affinity decreases.

In contrast to the information affinity, the flux ϕ increases with increased driving strength q/k. The reason for this is that as $q/k \to 0$, the system tends towards a detailed balanced equilibrium state, whereas as q/k increases the system is increasingly dominated by the non-equilibrium drive, which causes a current. The combination of the opposing behaviours of the information affinity and the flux is the peaked shape of l_Y. It may seem surprising that increasing q/k causes opposing behaviour of the information affinity and its conjugate current ϕ. However, in general affinities depend on relative (not absolute) rates, whereas fluxes are determined by absolute rates—similar behaviour therefore arises in general systems. In any case, now that the assumption of no feedback from Y to X is violated, it is no longer true that ϕ and hence l_Y reflect the strength of correlation, as in section 2.4.1.

What intuition, then, do we gain from the learning rate? The presence of a flux $\phi > 0$ is indicative of the fact that Y is responding to changes in X; systematically, X tends to transition from 0 to 1 before Y, which then tends to follow. The dependence of l_Y on ϕ thus reflects the fact that the learning rate measures the degree to which transitions in Y respond to transitions in X to maintain correlations, rather than the strength of correlations themselves. This is apparent from the definitions in Equations 2.9 and 2.11. In a

system with feedback from Y to X, these two metrics report on quite distinct properties. In the equilibrium limit, correlations are strong but transitions in Y are just as likely to precede transitions in X as to follow them, and hence the learning rate is zero.

2.4.3 A more complex signal process

In section 2.4.2, we relaxed the assumption that the downstream system Y did not influence the upstream system X. As a result, the tight connection between learning rate and correlation between X and Y, observed in section 2.4.1, was broken. We now explore the possibility of introducing more complex upstream signals X than assumed in section 2.4.1, whilst restoring the assumption that X is not influenced by Y. We still consider systems in the stationary state.

As a physical model to provide intuition, we again consider a model of concentration sensing by a single receptor. In this case, we consider an oscillating concentration signal[47], as might be relevant to a cell experiencing roughly periodic, but noisy, fluctuations in its environment. Such a situation might be relevant to cells in animal intestines, for example. To construct an oscillating concentration, we consider a Markov process, X, with $2N$ discrete states, as shown in figure 2.6 (a). States of X with $x = 1...N$ have a concentration $C = xC_0/N$, and states $x = N + 1...2N$ have a concentration $C = (2N + 1 - x)C_0/N$. Thus there are N distinct concentrations, each present at two values of X. We consider transitions between only neighbouring X states, with $w^{x\,x+1} = Nk_\uparrow$ and $w^{x\,x-1} = Nk_\downarrow$.

This partitioning of X into two 'rows' allows the construction of a Markov process X that leads to noisy oscillations in the concentration C, between $\frac{C_0}{N}$ and C_0. Taking $Nk_\uparrow > Nk_\downarrow$, the system will tend to move to higher values of C when $1 \leq x \leq N$, and towards lower values when $N + 1 \leq x \leq 2N$ (clockwise in figure 2.6 (a)). Scaling rate constants with N means that the overall flux around the cycle is N-independent. The typical behaviour of $X(t)$ and $C(t)$ is shown in figure 2.6 (c). A similar Markov chain biased by a constant chemical potential to produce noisy oscillations was used by Barato

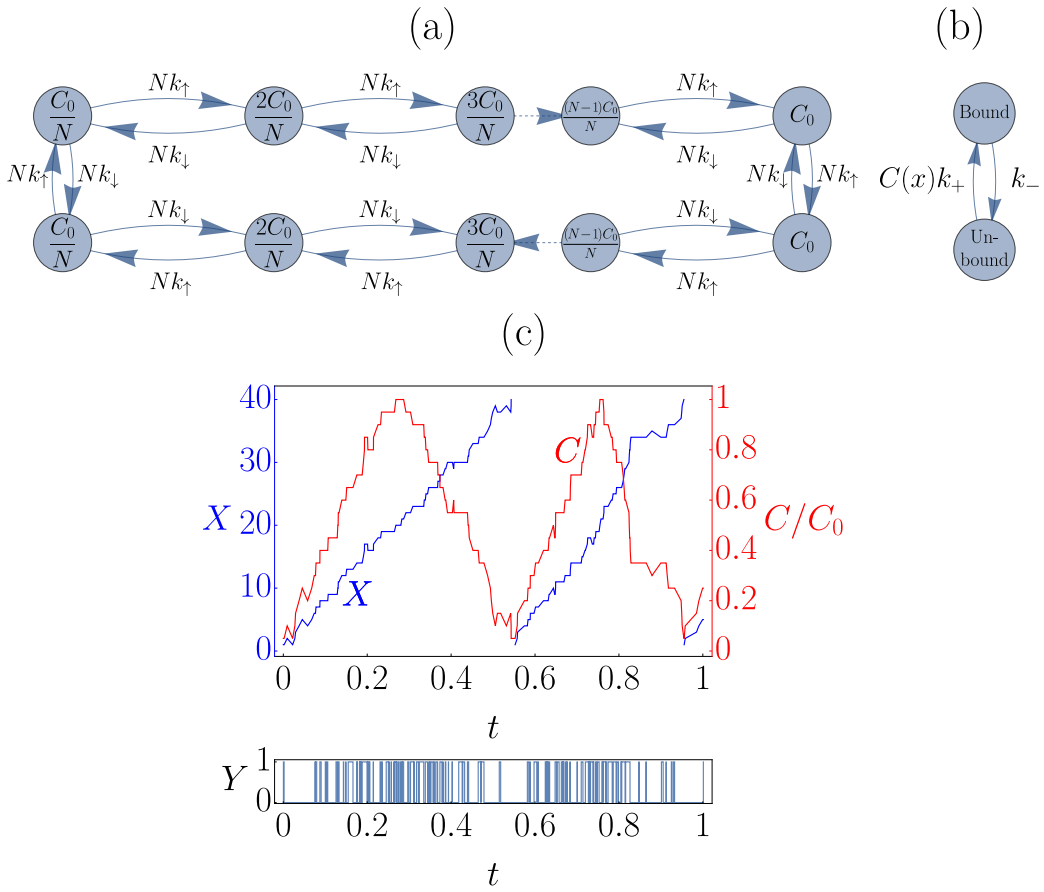


Figure 2.6: A bipartite system in which a receptor modelled by Y responds to an oscillating concentration determined by X. (a) A graphical representation of the process X, indicating the ligand concentration C for each X. X is biased to move clockwise around the loop so C is driven up and then down in an oscillating fashion. X is not influenced by Y. (b) For Y, the binding rate is proportional to the ligand concentration and the unbinding rate is constant, as in section 2.4.1. (c) Typical behaviour of X, C and Y over time (for $N = 20$, $C_0 = 1$, $k_\uparrow = 5$, $k_\downarrow = 0.5$ and $k_+ = k_- = 300$), showing stochastic oscillations in concentrations

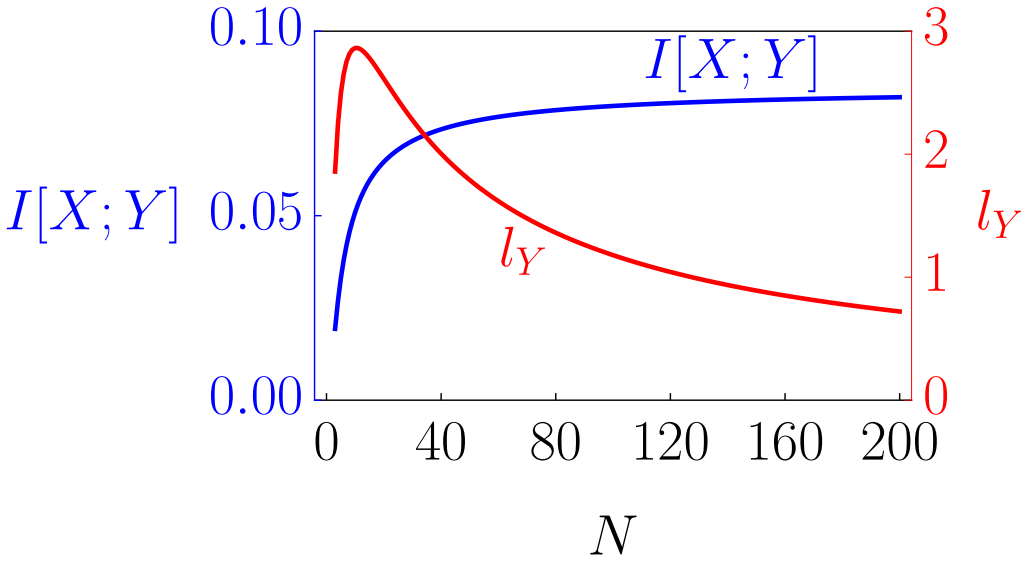


Figure 2.7: The learning rate and instantaneous mutual information show opposite behaviours as the number of external states N is varied for the cyclic system illustrated in figure 2.6 (a). The learning rate decays from a finite value to zero, whereas the instantaneous mutual information increases monotonically to a finite plateau. Data obtained for $k_\uparrow = 20$, $k_\downarrow = 10$, $k_+ = 10000$, $k_- = 2000$ and $C_0 = 1$.

and Seifert[48] to model a clock.

The receptor state defines the sub-process Y; states and transitions are shown in figure 2.6 (b). The receptor properties are identical to that considered for the simpler system in section 2.4.1. There is a constant rate of unbinding, $w^x_{10} = k_-$, and the binding rate $w^x_{01} = C(x)k_+$ is proportional to the ligand concentration.

From the perspective of the learning rate, it is illuminating to vary N (the number of discrete concentration states) whilst fixing all other parameters. As is evident from figure 2.7, $I[X;Y]$ increases with N up to a finite plateau, whereas the learning rate decreases to zero. The behaviour of $I[X,Y]$ is intuitive; as more states are added, it becomes easier to reliably distinguish values of X that correspond to high and low values of C. In particular, for low N, C changes in large jumps due to X transitions, meaning that Y is an inaccurate reporter for X immediately after the transition. For larger N, the change in C is smoother. Eventually, however, this effect saturates, and

$I[X;Y]$ reaches a finite plateau.

By contrast, the learning rate eventually falls with large N, tending to zero in the limit $N \to \infty$ (we present a general proof in Appendix A.1). To understand this difference in behaviour, note that our choice of $w^{x\,x+1} = Nk_\uparrow$ and $w^{x\,x-1} = Nk_\downarrow$ ensures that the overall average rate at which the concentration undergoes oscillations is fixed, regardless of N. The number of forward and backward steps of the process X in a time Δt are independently Poisson-distributed with means $Nk_\uparrow\Delta t$ and $Nk_\downarrow\Delta t$, respectively. Thus, the mean net number of forward steps in Δt is $\Delta tN(k_\uparrow - k_\downarrow)$, proportional to N, compensating for the fact there are more states covering the same concentration window. The standard deviation in the net number of forward steps in Δt is $\sqrt{\Delta tN(k_\uparrow + k_\downarrow)}$, however. Consequently, as N is increased, the relative uncertainty in the net number of forward steps during Δt falls as $1/\sqrt{N}$. Larger N therefore implies a smaller relative error in predicting the future value of X, given knowledge of its current value. Increasing the number of states N is thus effectively a method to interpolate between highly stochastic and quasi-deterministic behaviour of X. Note that it is the convergence on deterministic behaviour of the signal that in general takes $l_Y \to 0$; not $N \to \infty$.

As $N \to \infty$, therefore, the rate at which the current value of Y becomes ineffective in predicting future values of X tends towards zero. In other words, the 'nostalgia' of Y (Equation 2.11 and Reference[44]), and hence l_Y, tends towards zero. The learning rate quantifies the rate at which transitions in Y must respond to transitions in X to maintain a steady-state information; this rate is zero for large N despite $I[X;Y]$ increasing monotonically with N. Thus, using a more complex signal than in the simple 2×2 network in section 2.4.1 also serves to highlight the distinctions between the physical meaning of l_Y and $I[X;Y]$.

2.4.4 Optimisation of sensors

Due to its provenance as an informational quantity, l_Y has been proposed as a metric for sensor performance in steady-state sensing circuits[8,9]. In

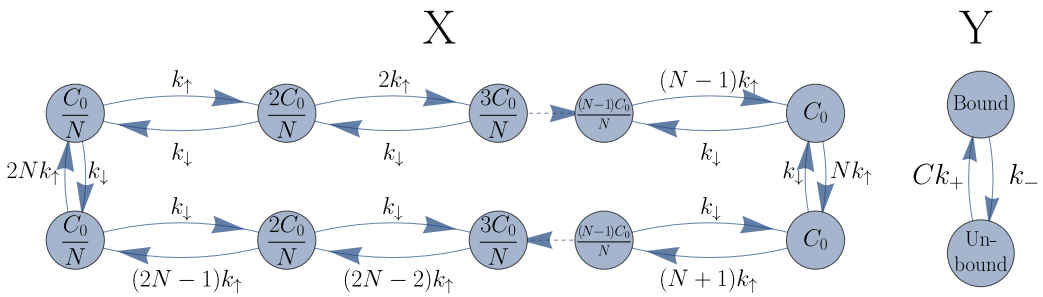


Figure 2.8: A bipartite system X, Y for which optimising the parameters of the downstream subsystem Y to maximise learning rate and mutual information give markedly different results. The system is essentially identical to that considered in figure 2.6, but with hopping rates for the upstream process X that depend on x.

sections 2.4.2 and 2.4.3, however, we have emphasised that l_Y reflects the rate at which transitions in the downstream subsystem Y respond to changes in X to maintain a steady-state information $I[X;Y]$. There is no reason *a priori* to suppose that maximising such a quantity is inherently optimal for a sensor—indeed, one might imagine that having to compensate for transitions in X at a lower rate would be preferable in reliable sensing. Further, our results in sections 2.4.2 and 2.4.3 show that the response of l_Y to parameter variation can be unrepresentative of the degree to which Y relates to signal X.

Sections 2.4.2 and 2.4.3, however, cannot reasonably be described as sensor optimisation. To optimise a sensor, it is natural to consider a setting in which the dynamics of signal X are fixed, and optimisation is performed over the parameters of the downstream network Y. In sections 2.4.2 and 2.4.3, however, we considered the response of l_Y to variations in the dynamics of X. Moreover, in section 2.4.2, we even considered a 'downstream' network Y which influences the 'upstream' network X. This is hardly a well-defined sensor. In the simple system of section 2.4.1, for which we did consider variation over the parameters of the Y sub-process, and X was not influenced by Y, optimising l_Y and $I[X;Y]$ over sensor parameters was essentially equivalent. It is therefore worth considering whether signal/sensor architectures do exist in which optimising over the sensor parameters for l_Y and $I[X;Y]$ give markedly different results.

To demonstrate this possibility, we consider a system identical to the os-

cillating concentration sensor considered in section 2.4.3, but for which the transition rates of the X sub-process are x-dependent (figure 2.8). The sensor Y, with transition rates $w^x_{10} = k_-$ and $w^x_{01} = C(x)k_+$, is unchanged.

We consider optimising over the relative rates of the X and Y subsystems at fixed number of X states N, fixed transition rates in the X subsystem, and fixed k_+/k_- within Y. In figure 2.9, we show that the instantaneous mutual information increases monotonically with the rates of the downstream subsystem (represented by k_-), from $I[X;Y] = 0$ up to a plateau. By contrast, the learning rate increases from $l_Y = 0$ to a peak at finite k_-, before decaying to a plateau. Maximising the instantaneous mutual information and maximising the learning rate give qualitatively different 'optimal sensors'.

We have considered optimising over this subspace with k_+/k_- fixed rather than the whole (k_+, k_-) space for ease of illustration. For a metric to be useful for sensor optimisation it must be able to meaningfully differentiate sensing quality between any two sets of parameters, and provide a useful optimisation within any subspace of the parameter space. We note that similar behaviour is observed for other values of k_+/k_-.

The 'optimal sensor' indicated by the instantaneous mutual information is intuitively reasonable. When the rates of the sensor are as large as possible, Y can keep up with X; otherwise it simply lags behind, reporting earlier values of X. Whilst it is possible that a slower-responding sensor could provide lower $I[X,Y]$, but increased predictive power at some specific future time τ, $I[X_{t+\tau}; Y_t]$[47], we find no evidence of this for our case (see figure 2.9). Therefore, the instantaneous mutual information provides a metric for sensor performance that is intuitively reasonable, if not necessarily unique. By contrast, we can find no intuitive justification of the peak of the learning rate in terms of optimal sensing. Indeed, the peak in the learning rate is reminiscent of the peak in the dissipation rate σ_Y that arises in such systems[49]. This peak is indicative of a subsystem Y that lags behind X sufficiently that the response to driving is highly irreversible, but is not so slow that it barely responds at all[49]. Similarly, it is plausible that the peak in l_Y reflects a delay in Y relative to X that its large enough to ensure that transitions have a strong tendency to increase $I[X;Y]$, but not so large that Y barely responds

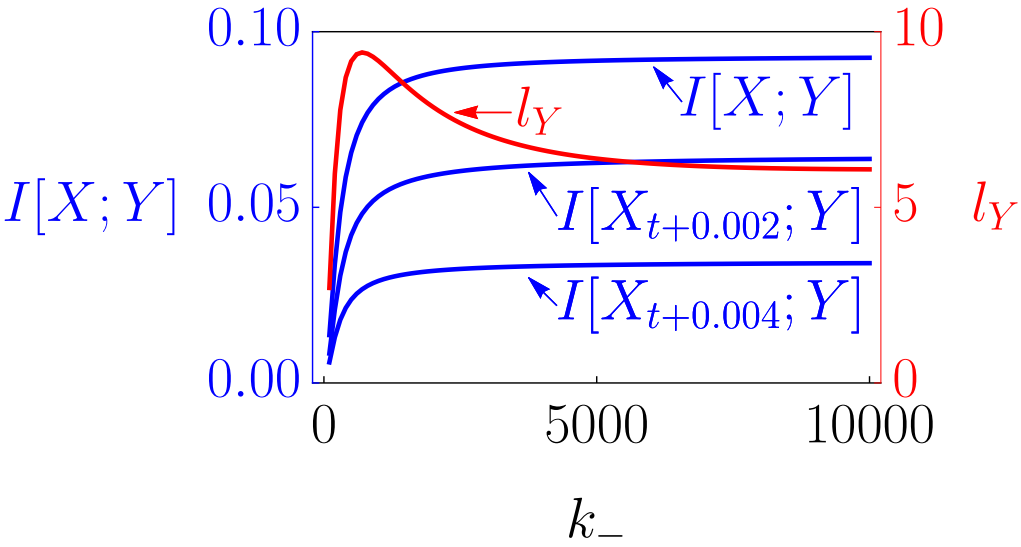


Figure 2.9: For a sensor detecting an oscillating system with a non-uniform hopping rate, $I[X;Y]$ increases monotonically with the relative response rate of the sensor, whereas l_Y shows a peak at a finite response rate. The two metrics thus imply distinct 'optimal' sensors. Also shown are some examples of $I[X_{t+\tau};Y_t]$, where τ is a positive time delay. These too are monotonic in response rate. Data obtained for $N = 20$, $k_\uparrow = 200$, $k_\downarrow = 10$, $k_+/k_- = 2$ and $C_0 = 1$

to changes in X at all. Regardless, since we are unable to explain the location of this peak through any optimal tracking perspective, it is difficult to see how l_Y is informative of quality of sensing.

2.5 Conclusions

We have explored the physical interpretation of the learning rate l_Y, a recently proposed statistical metric defined on bipartite stochastic systems[8,9,50]. l_Y quantifies the rate at which transitions in a downstream system Y act to increase the mutual information between Y and an upstream system X, or the rate at which future values of Y become more predictive of the current values of X than the current value of Y. In the steady state, a biophysically relevant scenario, l_Y is also equivalent to the rate at which transitions in X reduce the mutual information, or the 'nostalgia' rate at which information about the current value of X becomes irrelevant.

We have shown that, in the simplest steady-state bipartite system in which X is not influenced by Y, l_Y essentially reports on the correlation between subsystems. In other settings l_Y can show quantitatively and qualitatively different behaviour from measures of interdependence such as the mutual information and covariance. Moreover, we have demonstrated that this difference can be explained by the fact that l_Y quantifies the rate at which transitions in Y act to maintain a steady state $I[X;Y]$, rather than the magnitude of $I[X;Y]$. In general, there is no reason why these two properties should behave similarly.

Fundamentally, since l_Y represents the rate at which transitions in Y act to increase information between X and Y, the 'learning rate' is a reasonable description of this quantity. However, the rate of learning and quality of sensing are not directly related, and we do not find that l_Y is a good general metric for sensor performance, as has been proposed[8,9]. We have shown that, in at least one signal/sensor context, maximising the learning rate over a subset of sensor parameters gives a result that is hard to interpret as an optimal sensor. Without first identifying how the learning rate is reporting on sensor performance in a given context, it is difficult to see how l_Y can be

reliably used as a metric for sensing in general.

One can imagine two sensors, Y and Y', of the same signal X, which produce very different joint probability distributions $p(x, y)$ and $p(x, y')$ but the same instantaneous mutual information $I(X, Y) = I(X, Y')$. It is perfectly possible, for example, that Y predicts the future of X less well than Y', despite the instantaneous mutual information being equal, if the fluctuations in X coupled to by Y are less informative about system dynamics than those reported by Y'. In this case, to maintain equal information in the steady state, Y will need to refresh its information more rapidly, and hence we would have $l_Y > l_{Y'}$. However, it would seem unreasonable to classify Y as a better sensor than Y'; indeed, if anything, Y' seems more effective since it possesses the same information about the present state of X and more information about its future evolution. High l_Y may therefore be indicative of a sensor that is in some sense inefficient: constantly needing to update information that rapidly becomes worthless.

If l_Y were a direct measure of sensor performance then the fact that l_Y is bounded by the entropy production due to Y-transitions, $\sigma_Y > l_Y$, would imply that entropy production is limiting for sensing. l_Y is not a direct measure of sensor performance so this argument does not hold. However, we do not argue that entropy production is not limiting for sensing—only that such a justification via the learning rate would be flawed. As argued by Govern and ten Wolde[46], in any equilibrium system, there is no real distinction between 'signal' and 'readout'—both must necessarily influence each other, and therefore the construction of a true sensor is impossible. In Reference[43] a bound relating entropy generation to instantaneous mutual information is derived in the case where the sensor does not influence the signal.

Our work has considered only a small set of possible Markovian systems, and has not touched on issues such as performance of readout networks that are designed to integrate a signal over time[51,15] or perform more complex operations such as differentiation[47]. It is possible that the learning rate reflects an aspect of the performance in these circuits. Further, we have not investigated the behaviour of the learning rate in systems that are yet to

reach steady state. In such cases, l_Y is no longer equal to either the nostalgia rate, or the rate at which transitions in X tend to reduce $I[X;Y]$. Moreover, it is potentially non-zero even for passive (non-driven) systems. The meaning of the learning rate in such cases warrants further consideration.

Finally, the learning rate can provide insight into the functioning of systems even if it is not a metric for performance of sensing or similar functionality. For example, its properties were recently applied to derive results that do not depend on its interpretation in a simple model of a perceptron[42]. l_Y can also reasonably be interpreted as an indicator of the direction and magnitude of information flow in a steady-state bipartite network, as evidenced by its role in the refined second law (Equation 2.7). In the context of a model of an autonomous Maxwell's demon it is the flow of information between the 'demon' subsystem and the 'engine' subsystem[10].

One interpretation of the learning rate in steady state might be as a quantification of the effort put in by the downstream system to maintain a certain interdependence of X and Y, regardless of whether this interdependence is optimal in some sense. For example, if the magnitude of all rates is increased but their ratios kept constant, then the learning rate is increased although there is no change in the correlation between the sensor and signal. Indeed, this sense of l_Y as a measure of how hard the downstream system has to work is apparently reinforced by the relation $l_Y < \sigma_Y$, with σ_Y being the entropy generation due to transitions in the Y-subsystem. It should be noted, however, that although entropy is generated by Y transitions, this is not necessarily indicative of a consumption of resources held by the Y subsystem[8]. In the biomolecular examples given here, for example, the Y subsystems are totally passive, consuming no chemical fuel. The thermodynamic work is done by the driving of the external process X[8], which is particularly evident in the example of section 2.4.2, where the nucleotides fuelling the driving are explicitly considered. From the perspective of a sensor or a cell, entropy generation arising from environmental changes, rather than due to internal fuel consumption, is not obviously a limiting factor.

3

Biochemical Szilard engines for memory-limited inference

3.1 Abstract

By designing and leveraging an explicit molecular realisation of a measurement-and-feedback-powered Szilard engine, we investigate the extraction of work from complex environments by minimal machines with finite capacity for memory and decision-making. Living systems perform inference to exploit complex structure, or correlations, in their environment, but the physical limits and underlying cost/benefit trade-offs involved in doing so remain unclear. To probe these questions, we consider a minimal model for a structured environment—a correlated sequence of molecules—and explore mechanisms based on extended Szilard engines for extracting the work stored in these non-equilibrium correlations. We consider systems limited to a single bit of memory making binary 'choices' at each step. We demonstrate that increasingly complex environments allow increasingly sophisticated inference strategies to extract more free energy than simpler alternatives, and argue that optimal design of such machines should also consider the free energy

reserves required to ensure robustness against fluctuations due to mistakes.

3.2 Introduction

Living and human-made systems exploit out-of-equilibrium fuel supplies to do useful work. For example, if glucose is present in the environment in higher than equilibrium concentrations relative to carbon dioxide and water, bacteria can power themselves through respiration. Similarly, internal combustion engines use an out-of-equilibrium concentration of their fuel, i.e. petrol, and are powered by the conversion of fuel and oxygen to carbon dioxide and water.

The amount of work that can be done using the fuel is bounded by the non-equilibrium free energy of the fuel[23] as discussed in section 1.5.1. This free energy contains both energetic and entropic terms. As one would expect, if the fuel contains more energy, then, in general, the amount of work that can be done is higher. However, fuels are also more useful if they are in well-defined initial states, with limited microscopic uncertainty. This uncertainty is quantified by the entropy, which is why the entropy contributes to the free energy.

The idea of using high energy fuel is intuitive. If the fuel initially has greater energy than at equilibrium, then that extra energy can be transferred to somewhere else to do useful work as the fuel equilibrates. It is less obvious how to exploit low entropy fuel—nonetheless, entropy is an important component of the free energy stored in biochemical fuel molecules and cellular membrane potentials. For example, the free energy released by converting an ATP molecule to an ADP molecule in a cell is approximately 1.5 times the standard free energy difference between an ATP and an ADP molecule[52].

Spurred by a desire to understand the fundamental physics of computation and information processing, there has been significant recent interest in the exploitation of purely entropic resources as discussed in section 1.6.

Although inspired by the physics of computation, the question of how to exploit correlations is also of fundamental biological relevance. If organisms

existed in a homogeneous non-equilibrium environment, there would be no need to develop sophisticated information-processing machinery to survive. However, from the chemotaxis system of E. coli to the brains of humans, complex molecular and cellular networks have been evolved to exploit the fact that the environment exhibits correlated fluctuations. These systems rely on the fact that what is sensed at a certain point in space and time contains information about nearby points[53]. They have evolved even though they are costly to maintain, and despite the fact that the information obtained is limited by features such as the memory and processing power available[54]. However, the fundamental trade-offs that determine the sophistication of these systems are not fully explored.

In this chapter we take steps towards unifying these two perspectives on the exploitation of correlations. We first present a molecular design for a measurement-and-feedback device (a Szilard engine[21]) in which the mechanics of the feedback is explicit within the molecular system. We then leverage this construct to propose biomolecular machines that make repeated binary choices about how to act based on measurements of their environment (an array of 'molecular bits'). These machines use their single bit of memory to extract chemical work from correlated arrays, demonstrating that it is possible to design minimal biophysical systems that exploit minimal structured environments.

No memory at all is needed to extract all of the available work from an input consisting of an array of uncorrelated subsystems, and simple schemes with one-bit memories can extract all of the stored free energy from binary Markovian environments. If we increase the complexity of the environment further, by making it a hidden Markov process, 100% efficiency becomes impossible with a single-bit memory and some implicit inference of the hidden state is required. In this setting, schemes that perform batch averaging to obtain a better estimate of the hidden Markov state can become more efficient than the most direct approaches, at the expense of increased biochemical complexity. We are thus able to construct a minimal thermodynamic setting in which increasingly complex information-processing machinery becomes advantageous in increasingly complex environments.

We first, in section 3.3.1, introduce our model. Numerical methods are briefly discussed in section 3.3.2. In section 3.3.3 we demonstrate how work can be extracted from a single molecule in a non-equilibrium state by our setup. Next, in section 3.4.1, we discuss how to make a biochemical version of the Szilard engine, which forms the basis of our machines to extract work from correlations. Subsequently we find the maximum amount of work that a device with a persistent memory can extract from a series of correlated bits (section 3.4.2). We discuss a device based on the biomolecular Szilard engine that reaches this limit and can extract all of the work available from a Markovian input in section 3.4.2. In sections 3.4.2 and 3.4.2, we discuss the limitations of this machine when acting on an input produced by a hidden Markov model. We propose a different machine, in section 3.4.2, that averages over a batch of multiple input molecules that can extract more work in some cases. Finally, in section 3.4.2, we discuss the robustness of such devices to fluctuations in the input.

3.3 Materials and Methods

In this chapter, all physical systems are assumed to be well-described by discrete macrostates of molecules in dilute solution as discussed in section 1.4. The discussion in section 1.5.1 that mutual information contributes to the non-equilibrium free energy of a physical system and that the work can be done by a system is bounded by the change in its non-equilibrium free energy is vital to this chapter.

In this work we consider machines designed to extract work from a non-equilibrium series of bits with both the machines and the bits rendered as biomolecules. These devices exploit pre-existing information within the input via a series of measurement and feedback operations implemented through a 1-bit memory. A summary of prior work on theoretical constructs for the exploitation of information was given in section 1.6.

3.3.1 Molecular implementation of a measurement-and-feedback machine

We now present a general description of the devices considered in this work. Although our devices would be challenging to engineer, operate in ideal limits, and are not direct models of living systems, we nonetheless render the machines, and the input bits, as biomolecules. All operations, including the measurement and feedback, are driven by a concrete molecular mechanisms that are explicitly part of the devices themselves. By considering a concrete realisation, even in an idealised limit, we can explore the limits of what is thermodynamically possible in a positive sense, rather than simply exploring the space of systems that are not forbidden by a particular aspect of the second law[22]. Furthermore, we demonstrate the true complexity of systems required to instantiate efficient measurement-and-feedback systems like Szilard engines.

The schematic set-up of our devices is shown in figure 3.1. The model consists of an input, a reaction volume, a series of chemical buffers, and a molecular 'hook' that can bind to the input molecules independently of their state[55,56]. The input is a series of small boxes each containing a single input molecule. This molecule can be in one of two strongly metastable states, X and X^* so these input molecules represent a string of bits. This input is a minimal analogue of a fluctuating chemical environment, as experienced by single-celled organisms[57,58,59,60].

The rest of the system is our machine, a minimal analogue of an organism exploiting its environment. The machine functions by transferring molecules to and from its reaction volume via the molecular hook. Once in the reaction volume, input molecules undergo reactions with molecules that are internal to the system—for example, a molecule M encoding the memory. These reactions are coupled to large fuel buffers that collectively allow the machine to store the work extracted from the environment, similar to Refs.[27,53]. The buffers are the molecular analogue of a weight in a gravitational field that can be lifted by the system in the canonical Szilard engine[26,29,30,31]. To perform this role, the buffers should be sufficiently large that any reactions have a

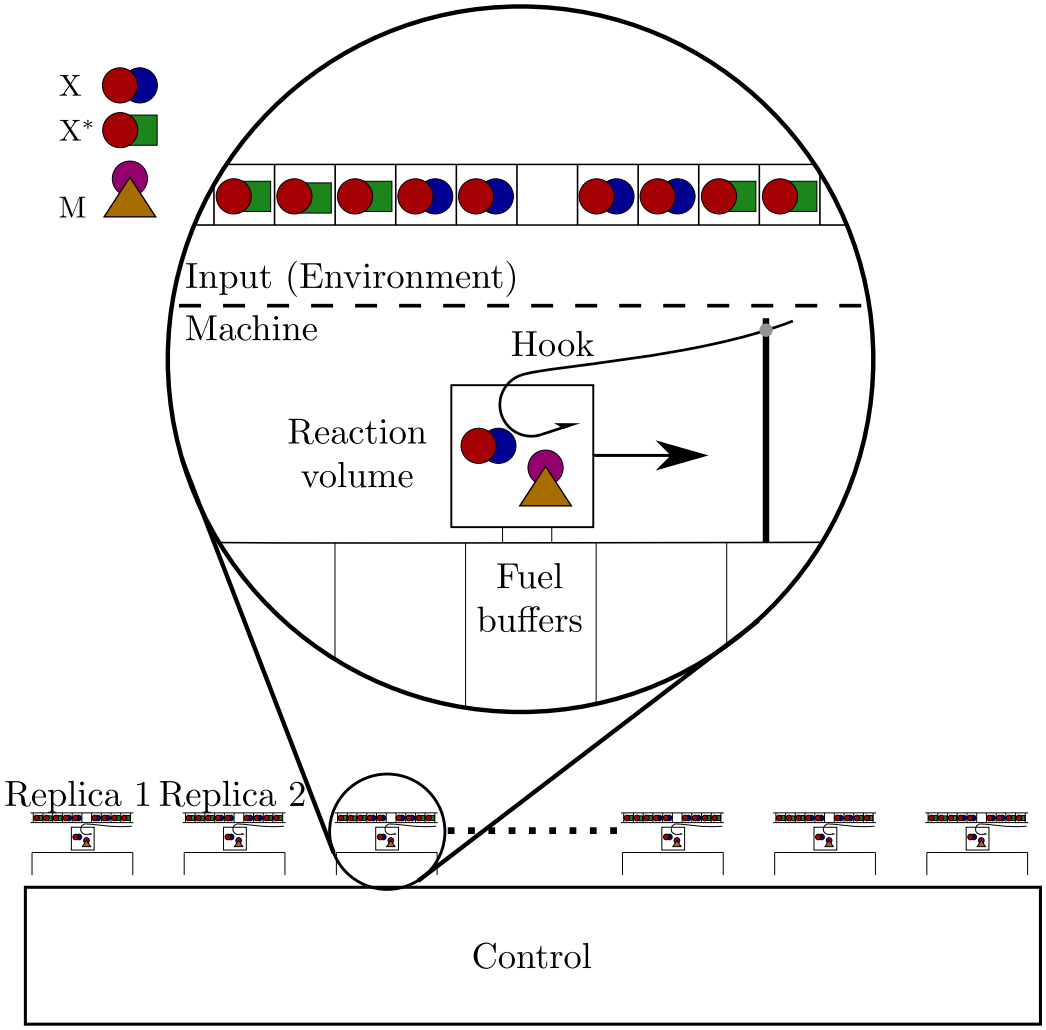


Figure 3.1: *General design of model systems considered in this work.* A series of input molecules each contained in a box (each one being an analogue of Szilard's box containing a gas molecule); a reaction volume; a series of buffers (acting as work reservoirs like the weight in the conventional Szilard engine); and a hook. One input molecule is moved from its box to the reaction volume by the hook. The reaction volume is then connected to a series of buffers in succession containing different concentrations of fuel molecules. These buffers exchange molecules with the reaction volume, driving the reaction in one direction or another. The net transfer of fuel to buffers of high chemical potential corresponds to the extraction of work from the input molecule. Then the hook moves the input molecule from the reaction volume back to its original box, and the next input molecule can be moved to the reaction volume. The dashed line separates the input/environment from the machine. The control can operate multiple replicas of the system simultaneously.

negligible effect on the probability distribution of bath macrostates[22]; in this limit, the buffer state is purely a deterministic concentration, whereas the input and other molecules within the reaction volume are represented through stochastic variables describing the fluctuating chemical macrostate. Recent experimental work from Joesaar *et al.* has demonstrated how molecules encapsulated within 'proteinosome' reaction volumes can be coupled to time-varying external buffer conditions, with the buffer molecules able to diffuse in and out of the reaction volumes and participate in reactions with the encapsulated species, as we require[61].

Details of how a molecular hook might operate are given in appendix A.2. Such a mechanism can transfer molecules to and from the reaction volume with no net expenditure of work, provided that the hook is controlled by a particular quasistatic protocol. The hook thus represents a work-free mechanism of ingesting and excreting molecules in a controlled manner. An alternative model without ingestion would have input molecules attached to consecutive sites on a polymer tape[53]; the machine would then interact with one or more of these molecules at any one time, based on proximity.

In this work we shall assume that buffer concentrations are manipulated by a well-defined protocol[55,13,14,62], as illustrated in figure 3.2. These protocols will not follow directly from the dynamics of degrees of freedom explicitly modelled—they are essentially externally imposed. Our system is then non-autonomous. We use externally applied protocols for two reasons. Firstly, because it allows us to design mechanisms in which first measurement of Y by X, and then feedback to exploit Y using X, are performed sequentially, as in Szilard's engine (see section 3.4.1); and secondly, because driving forces can be increased in a quasistatic manner (see section 3.3.3). Both features are essential if we are to maximise both the efficiency of the reactions, and the reliability of the implemented information-processing strategy: quasistatic manipulation allows us to push reactions implementing each step of the process to 100% completion, efficiently. We can then focus purely on the constraints on work extraction that arise from the fact that the implementable information-processing strategies are limited by the finite size of the device's memory.

Crucially, however, although our systems require external protocols, the protocols themselves require no decision-making intelligence: the same series of manipulations will be applied repeatedly, without feedback from the state of the system. All 'decisions' and feedback strategies must be made by the molecules that are explicitly represented. By avoiding protocols that require external decision-making dependent on the state of the system, we avoid implicit costs that have caused much of the confusion in the thermodynamics of computation, since the original thought experiment of Maxwell[20] as discussed in sections 1.5 and 1.6. In principle, the protocols we invoke could be applied in parallel to an arbitrarily large number of replicas (as shown in figure 3.1), rendering the marginal cost per machine of the external protocol negligible. Indeed, this is the assumption usually made with macroscopic thermodynamics. By contrast, if separate decisions had to be made for each replica, the economy of scale would not exist. Although the use of external control makes the individual devices a weaker analogy for single, autonomous organisms, the combined set-up of many devices and their controller is then an analogy for a single, albeit more complex, organism. We note in passing that the need for a quasistatic protocol to control the hook is equivalent to the need for a quasistatic protocol to deterministically advance the tape with which a machine interacts, as has previously been assumed in many bit-driven machines[26,29,30,31,32]—our physical instantiation makes this need clearer.

3.3.2 Numerics

All devices studied in this chapter produce a deterministic output of extracted work given a specific input sequence. Numerical results presented in this chapter are therefore obtained either by exhaustive summation of short input strings, or by sampling of long input strings by simulating the underlying generative model. The code and data to produce the figures in this chapter can be found at `https://doi.org/10.5281/zenodo.1976932`.

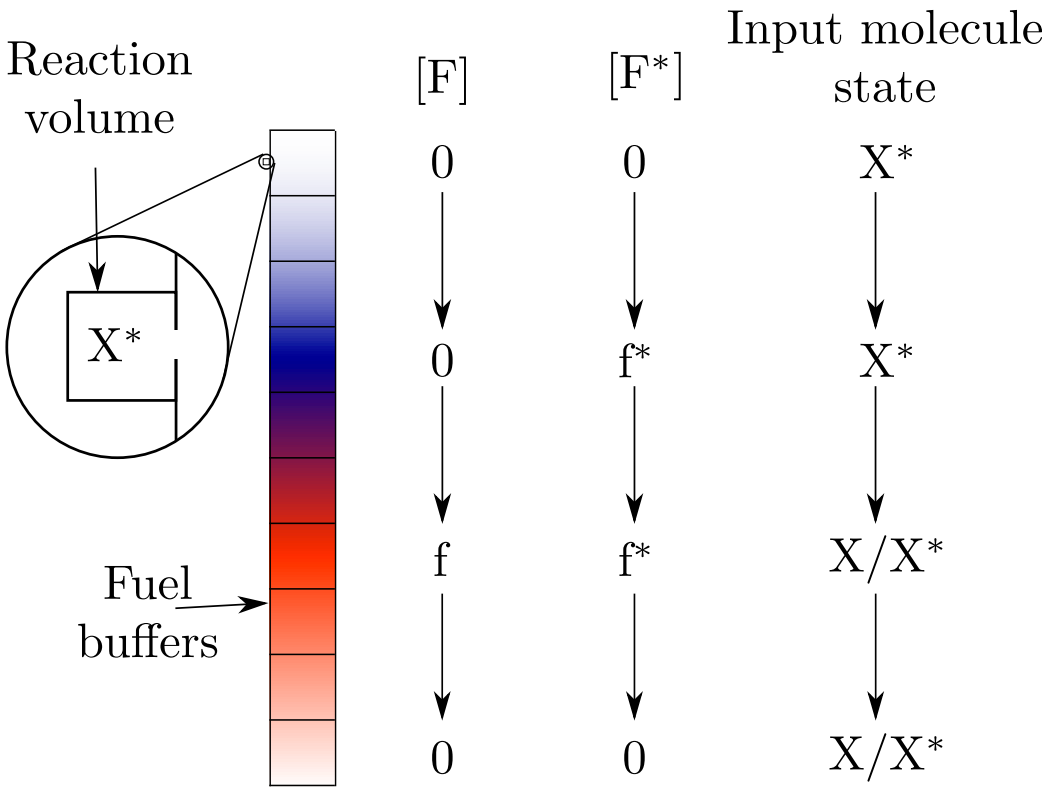


Figure 3.2: *Fuel manipulation protocol to extract maximal work for a known input state.* The concentration of the fuel molecules in the reaction volume is set by connecting the reaction volume with a chemical buffer. The concentrations can be gradually changed by connecting the reaction volume to a series of buffers with a small change in concentration between adjacent buffers. In this case, no reactions are initially possible since both fuel concentrations are zero. $[F^*]$ is first increased so that the presence of X^* and the absence of X is consistent with the equilibrium implied by the imbalance in $[F^*]/[F]$. $[F]$ is then increased, allowing X^* to convert to X in a quasistatic manner and extracting work from the process, as molecules of F at low chemical potential are converted to molecules of F^* at high chemical potential. Finally, both fuels are taken to zero concentration.

3.3.3 Example system and calculation: Extracting work from a biased environment

We illustrate the operation and analysis of the set-up outlined in section 3.3.1 by demonstrating it in the simplest possible context. We consider the reversible extraction of work from a low entropy input by increasing the free energy of a chemical buffer. In this setting, the input array consists of input molecules each initially in the state X^* with 100% probability. The X and X^* states of the input molecules have equal intrinsic free energy so

in equilibrium a single input molecule is equally likely to be in either state. Therefore, it is possible to extract a work of $k_{\mathrm{B}}T \ln 2$ per input molecule from the environment.

Each input molecule is transferred to and from the reaction volume by a hook with no net work expenditure, as outlined in appendix A.2. When the input is in the reaction volume, we extract work by increasing the free energy of a bath of fuel molecules F and F* with chemical potentials μ_{F} and μ_{F^*}. To do this we need a chemical reaction

$$\mathrm{X} + \mathrm{F}^* \rightleftharpoons \mathrm{X}^* + \mathrm{F}, \tag{3.1}$$

which couples the interconversion of X and X* to the interconversion of F and F*. The interconversion of X and X*, or F and F*, is assumed to be infinitely slow except via this reaction. No other molecules, such as those representing a memory, are necessary in this simple context. The central idea is that an excess of X* can be used to pump F into F* against a chemical potential difference, storing work in the buffers just as traditional heat engines store work by lifting a weight. We now consider the details of how this work might be done.

It is possible to extract some work by connecting the X* molecule to a single bath of F and F* molecules with a high concentration of F*, so that $\mu_{\mathrm{F}^*} > \mu_{\mathrm{F}}$. Both the input and the bath are individually out of equilibrium, and tend to drive the reaction in Eq. 3.1 in opposite directions. In this case, the drive from the input is stronger and the reaction in Eq. 3.1 proceeds from right to left, with the input doing work on the bath. Over time, the bias of the input will decrease until the driving force of both contributions cancel; although the bath and the input are individually still out of equilibrium and store free energy, the input has reached a bias which is in equilibrium with driving force of the bath. At this point, the input will be in state X with probability $1/(1 + \mathrm{e}^{-\beta\Delta G})$ and in state X* with probability $\mathrm{e}^{-\beta\Delta G}/(1 + \mathrm{e}^{-\beta\Delta G})$ where $\beta = 1/(k_{\mathrm{B}}T)$, $\Delta G = \mu_{\mathrm{F}} - \mu_{\mathrm{F}^*} < 0$. During this relaxation to equilibrium, $1/(1 + \mathrm{e}^{-\beta\Delta G})$ molecules of F are converted to F* on average. Therefore, the free energy of the bath is changed by $-\Delta G/(1 + \mathrm{e}^{-\beta\Delta G})$—this is the work

extracted per input molecule.

Different choices of ΔG lead to different values of the work; however, $-\Delta G/(1+\mathrm{e}^{-\beta\Delta G})$ has a maximum of $\approx 0.28 k_{\mathrm{B}}T$, which is less than $k_{\mathrm{B}}T\ln 2$. This protocol has not extracted all of the work available; indeed the input molecule has not even reached its equilibrium distribution, so it is still a store of free energy. Thus the input molecule could be put in contact with a second bath with a lower concentration of F^* molecules but still with an excess of F^* above the equilibrium concentration and some more work could be extracted.

If the input molecule is connected to two successive baths with a non-infinitesimal difference in fuel concentrations, then the input molecule undergoes a thermodynamically irreversible relaxation, with some fraction of the free energy being wasted. However, if we take this idea of connecting the input molecule to successive baths with lower $\Delta G = \mu_{\mathrm{F}} - \mu_{\mathrm{F}^*}$ to the limit of a continuous change in ΔG we get a quasistatic process with no irreversible relaxations to equilibrium: the system is at equilibrium with the bath(s) at all points in time. This protocol is achieved by connecting the reaction volume to a large number of baths in succession for enough time to reach equilibrium with each bath as shown in figure 3.2. There is only a small change in concentration of fuel molecules between successive baths. Therefore, in the limit of infinite baths and infinitesimal changes in concentration the reaction volume experiences a quasistatic change in the concentrations of the fuel molecules.

The specific protocol of fuel molecule concentrations, illustrated in figure 3.2, is as follows. Initially $[\mathrm{F}] = [\mathrm{F}^*] = 0$ so reaction 3.1 cannot occur. Then $[\mathrm{F}^*]$ is slowly increased up to an appreciable value we name f^*. The reason the concentration must be increased slowly is so that fuel molecules are not irreversibly transferred between different buffers via the reaction volume. The reaction in equation 3.1 still cannot occur, since only X^* and F^* are present. Then, F is slowly increased. Now reaction 3.1 can occur; although, initially, the rate of converting X^* to X is much slower than the reverse so the input molecule is still in state X^* with high probability. [F] is increased to f, which is the concentration at which the free energy change in reaction 3.1 is $\Delta G = 0$, so the X and X^* states are equally likely.

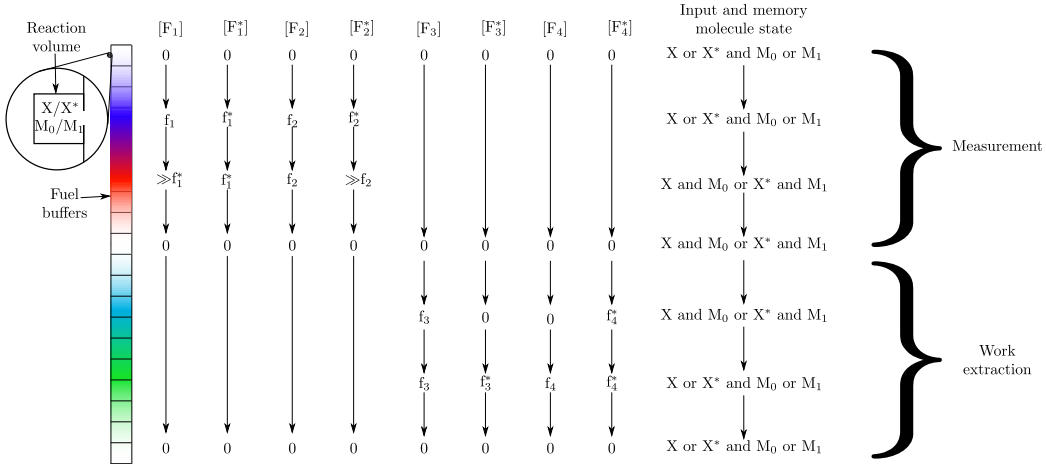


Figure 3.3: *Fuel manipulation protocol for the biochemical implementation of the Szilard engine.* In the first stage the concentrations of fuel molecules are changed to set the memory molecule to M_0 if the input molecule is X and M_1 if the input molecule is X^*. In the second stage work is extracted from the correlation between the input molecule and the memory molecule. The ability to first use X to set M via F_1 and F_2, and then couple M to the exploitation of X using F_3 and F_4, is the key to implementing Szilard's engine. In both stages, as in figure 3.2, the fuels are initially set to zero so no reaction can happen. They are then increased to finite values, but maintaining a ratio consistent with the equilibrium implied by the current state of the X and M molecules (right hand column), so that no net reactions happen. From this point, the concentrations of fuels are then manipulated to drive the X and M molecules towards the desired state. Finally, the fuel concentrations are set to zero again.

To calculate the average work extracted in this quasistatic process we consider the increase in free energy of the F/F* baths. Let the probability of the input molecule occupying state X*, when equilibrated with a buffer with a chemical potential difference of ΔG, be $p_{\Delta G}(\mathrm{X}^*)$. A change in chemical potential difference of $\Delta G - \delta\Delta G$ to ΔG is then associated with a probability change of $p_{\Delta G}(\mathrm{X}^*) - p_{\Delta G - \delta\Delta G}(\mathrm{X}^*) \approx \frac{\mathrm{d}p_{\Delta G}(\mathrm{X}^*)}{\mathrm{d}\Delta G}\delta\Delta G$. This change is also equal to the number of F* molecules that are on average converted to F molecules when the reaction volume is exposed to a new buffer. Therefore, the free energy of the bath increases by $\Delta G \frac{\mathrm{d}p_{\Delta G}(\mathrm{X}^*)}{\mathrm{d}\Delta G}\delta\Delta G$ on average.

Taking the limit of infinitely many baths, we integrate the total work done

$$\begin{aligned} W &= \int_{\infty}^{0} \Delta G \frac{\mathrm{d}p_{\Delta G}(\mathrm{X}^*)}{\mathrm{d}\Delta G}\mathrm{d}\Delta G \\ &= k_{\mathrm{B}}T\ln 2, \end{aligned} \tag{3.2}$$

where we integrate by parts and recall that $p_{\Delta G}(\mathrm{X}^*) = \mathrm{e}^{-\beta\Delta G}/(1+\mathrm{e}^{-\beta\Delta G})$. The quasistatic protocol is therefore able to recover all free energy stored in the initial low entropy state, $k_{\mathrm{B}}T\ln 2$, as work. In performing this calculation, we have ignored external costs associated with generating the quasistatic protocol, for reasons outlined in 3.3.1. The transfer of molecules between adjacent buffers, mediated by the reaction volume, has a cost that tends to zero as the concentration difference between buffers tends to zero. With the basic approach to set-up and analyse of our machines explained, we can discuss specific measurement and feedback processes.

3.4 Results

3.4.1 A Biochemical Szilard engine

Before analysing structured environments, we first present a measurement-and-feedback device that acts on a single binary input. This simpler setting allows us to illustrate the explicit measurement-and-feedback cycle that will underlie all the devices in this work. In particular, we demonstrate a mechanism by which the input is first able to influence the state of a memory, and

subsequently the influence is reversed so that the state of the memory affects how work is extracted from the input. This mechanism will be a biochemical version of the Szilard engine discussed in section 1.5.

The biochemical Szilard engine consists of an input molecule, a memory molecule, and chemical fuel buffers that are used to supply or recover chemical work; to implement a single cycle, we do not require a series of inputs, or a molecular hook. The input molecule is in one of two states: X or X^*. For simplicity, we assume the states have equivalent intrinsic free energy, and that the system is in equilibrium: the molecule is then found in each state with probability $\frac{1}{2}$. The memory molecule also has two states with equivalent intrinsic free energy, and is initially in state M_0 with probability $\frac{1}{2}$ and in state M_1 with probability $\frac{1}{2}$.

To 'measure' the state of the input means to set the state of the memory to M_0 if the input is X or to M_1 if the input is X^*: we correlate the states. This step follows the optimal copy protocol in[55] and can be done using the chemical reactions

$$\begin{aligned} M_0 + X + F_1^* &\rightleftharpoons M_1 + X + F_1, \\ M_0 + X^* + F_2^* &\rightleftharpoons M_1 + X^* + F_2, \end{aligned} \tag{3.3}$$

where F_1, F_1^*, F_2 and F_2^* are a fuel molecules that are present in excess, and X and X^* act as catalysts for the transformation of M between its states. Interconversions other than via the catalytic reactions in equation 3.3 are assumed to be so slow as to be negligible. The central idea is to drive the catalytic reactions in opposite directions through fuel imbalances, so that M can be set to M_0 in the presence of X, and to M_1 in the presence of X^*.

The selective catalysis in equation 3.3 is an approximation of the behaviour demonstrated by bi-functional kinases in cell signalling networks[63], and can also be engineered from nucleic acid networks (see appendix A.6 for details). The free energy changes of the reactions and the reaction rates can be controlled by the concentrations of the fuel molecules, as in the simple example in section 3.3.3.

It would be possible to set the memory molecule M to the correct state by directly coupling to a buffer with $[F_1] \gg [F_1^*]$ and $[F_2^*] \gg [F_2]$. As in section 3.3.3, however, the associated process would be thermodynamically irreversible, wasting the ability of the fuel buffer to do useful work. We therefore change the fuel concentrations quasistatically, as illustrated in figure 3.3, gradually forcing the memory to the M_0 state when in the presence of X, and to the M_1 state in the presence of X^*.

Initially, $[F_1]$, $[F_1^*]$, $[F_2]$ and $[F_2^*]$ are all set to zero. The reactions in equation 3.3 therefore cannot occur. Then, the concentrations are simultaneously increased at a fixed ratio of $[F_1]/[F_1^*]$ and $[F_2]/[F_2^*]$ that maintain an overall free energy change of zero for the reactions in equation 3.3. One of these interconversions (determined by whether there is an X or and X^* present) now occurs at an appreciable rate, but forwards reactions exactly balance reverse reactions so there is no overall change in the probability of observation of M_0 and M_1.

Next, $[F_1]$ and $[F_2^*]$ are increased while $[F_1^*]$ and $[F_2]$ are kept constant. As a result, the first reaction in equation 3.3 is pushed to the left and the second to the right. Consequently, if the input is X then the memory molecule is more likely to be M_0 and if the input is X^* the memory molecule is more likely to be M_1. Eventually, when $[F_1]$ and $[F_2^*]$ have been increased so that $[F_1] \gg [F_1^*]$ and $[F_2^*] \gg [F_2]$, the memory molecule M will be perfectly correlated with the input X. Next, $[F_1]$, $[F_1^*]$, $[F_2]$ and $[F_2^*]$ are decreased while maintaining $[F_1] \gg [F_1^*]$ and $[F_2^*] \gg [F_2]$ until $[F_1^*] = [F_2] = 0$. Finally $[F_1]$ and $[F_2^*]$ are decreased to zero. Now the reactions in equation 3.3, again, cannot occur so the memory molecule is fixed to be M_0 if the input is X and M_1 if the input is X^*.

In this correlated state the entropy of the combined (X, M) system is $k_B \ln 2$ because there are two equally likely states: $(X, M) = (X^*, M_1)$ or (X, M_0). Prior to measurement the entropy was $k_B \ln 4$ because the four combinations of X and M are equally likely. Thus the entropy of the system has decreased by $k_B \ln 2$ and so the free energy of the system has increased by $k_B T \ln 2$.

The increase in free energy of (X, M) is compensated by a decrease in the

free energy of the buffers. This decrease can be calculated as in section 3.3.3, except with the limits on the integral reversed and considering two equally likely possibilities: either the input molecule was X and the concentrations of F_1 and F_1^* are changed due to the first reaction in equation 3.3, or the input molecule was X^* and the concentrations of F_2 and F_2^* are changed due to the second reaction in equation 3.3. The result is that the free energy change of the buffers is $-k_B T \ln 2$, which exactly cancels the free energy increase of (X, M), as it should because the process is thermodynamically reversible (see appendix A.3.1 for more details on this calculation). This reduction in free energy of the buffers is the 'cost' to measurement that was recognised by Szilard as the resolution to the Maxwell's demon paradox[21].

We now consider the feedback step. The device extracts chemical work from the correlated state by allowing the input molecule to evolve in a manner that reflects the outcome of the measurement. The machine uses the reactions

$$\begin{aligned} M_0 + X + F_3^* &\rightleftharpoons M_0 + X^* + F_3, \\ M_1 + X + F_4^* &\rightleftharpoons M_1 + X^* + F_4, \end{aligned} \tag{3.4}$$

in which the F_3^*, F_4^*, F_3 and F_4 are further fuel molecules. Now, M_0 and M_1 act as catalysts for the transformation of X between its states; non-catalysed reactions are again assumed to be impossible. M and X must therefore be mutual bifunctional catalysts, which can be effectively switched on and off by modulating fuel concentrations. This explicit rendering demonstrates the complexity necessary in a minimal measurement-and-feedback device such as Szilard's engine, in which the memory and input must reverse their roles as the determinants of the dynamics. A design based on DNA strand displacement[16,17,18] is presented in appendix A.6.

As in the measurement step, the reaction rates are slowly manipulated by coupling to buffers with different concentrations of fuel molecules. Initially $[F_3] = [F_3^*] = [F_4] = [F_4^*] = 0$, along with the fuels used in the measurement process, and no reactions occur. Subsequently, $[F_3]$ and $[F_4^*]$ are increased. At this point the reactions in equation 3.4 do not occur since the right combination of fuels and substrates are not present.

Next, $[F_3^*]$ and $[F_4]$ are increased until the free-energy change of reactions in equation 3.4 is zero. As a result, the input molecule is slowly decorrelated from the memory, but the memory state determines which fuel buffer the input couples to during this process. Finally, $[F_3]$, $[F_3^*]$, $[F_4]$ and $[F_4^*]$ are decreased to zero while maintaining a free-energy change of zero for the reactions in equation 3.4.

As with the measurement, we can calculate the change in free energy of the input molecule and measurement molecule system and the chemical work done by the chemical fuel buffers. This extraction step is essentially the reverse of the measurement step, so the free energy of the input molecule and measurement molecule system decreases by $k_{\mathrm{B}}T\ln 2$ while simultaneously the free energy of the buffers increases by $k_{\mathrm{B}}T\ln 2$ (see appendix A.3.2 for more details on this calculation).

At the end of the cycle, both the memory molecule and the input have been returned to unbiased and statistically uncorrelated states. Chemical free energy has been transferred from the buffers 1 and 2, to buffers 3 and 4. The net chemical work extracted is then zero since the $k_{\mathrm{B}}T\ln 2$ cost of measurement balances the work extracted. This is, of course, expected—extracting work from the initially equilibrated input should be impossible. However, this basic design will underpin that of devices intended to exploit structured environments, and recover net positive work.

We note, in passing, three instructive features of our explicitly-described biochemical Szilard engine. Firstly, the measurement and feedback reactions can be implemented sequentially by coupling to buffers of first one, then another, type of fuel molecule. This ability to switch from having the input set the memory, to having the memory modulate the evolution of the input, is the key feature of our setup that allows us to represent the full cycle. Secondly, there is no need for an 'erase' step to reset the memory to a specific state[64]. Whilst it would be possible to include such a reset, it is not necessary, either for efficient operation or to preserve the second law of thermodynamics. The second law is preserved simply by the 'ominous' nature of the non-equilibrium correlations originally identified by Szilard. Thirdly, the measurement is simply the act of setting the engine into the correct state

to exploit the input (setting the memory to M_0 or M_1). There is no need for any other system, intelligent or otherwise, to record or be aware of the outcome of the measurement. In the context of the typical one-particle-gas description of Szilard's engine[21], the measurement is simply the correlation of the pulley and particle positions. Any additional recording of the particle position (for example in the brain of an intelligent being) corresponds to a useless extra correlation or measurement, with associated costs that must be carefully recovered at a later time to reach 100% efficiency.

3.4.2 Exploiting a series of correlated bits

Although the Szilard engine cannot extract useful work from its equilibrium input, it forms the basis of a device for exploiting a series of identical biochemical bits labelled with the index i, whose correlated states, described by the random variables $\{X_i\}$, are generated by a stationary stochastic process. The random variable X_i has the possible outcomes of X or X^*. We consider the series to be infinite in both directions. As with the Szilard engine in section 3.4.1, both states of the input bits are assumed to be equally intrinsically stable, and separate bits do not interact (they are in different boxes in the language of figure 3.1). The equilibrium distribution of the inputs is, then, for each molecule to be independently distributed uniformly between its two states.

Free energy is stored in the input array if either an initial bias towards X or X^* is present, and/or correlations exist between X_i and X_j for $i \neq j$. Since designing a system to exploit an intrinsic bias is simple, and requires no measurement or inference (see section 3.3.3), we focus exclusively on the case in which the marginalised probability of each bit occupying either state is 1/2.

Bounds on work extraction

The free energy per bit stored in such an array, and hence the available work per bit, is determined by the difference between the equilibrium Shannon

entropy per bit of $\ln 2$ and the entropy rate h[29]

$$W_{\text{available}} = k_{\text{B}}T\left(\ln 2 - h\right), \tag{3.5}$$

where

$$h = \lim_{n\to\infty} \frac{H(X_1, X_2, \ldots, X_n)}{n}, \tag{3.6}$$

as discussed in section 1.6.

An array of N bits has a state space of size 2^N. For an array with arbitrary correlations, an operation must be 'globally integrated' across all N bits to fully extract $W_{\text{available}}$[37]. Even if a system were able to achieve this integration by coupling to all bits in an array simultaneously, extracting the full available work would be highly non-trivial. In practice, the protocol would need to be tuned to the expected initial occupancy of each of the 2^N states to avoid losses.

The opposite limit to a device that is able to interact with the entire input at once is a device that interacts with each bit separately and in an independent manner. However, such a device can only extract the free energy stored in the state X_i, $\mathcal{F}_{\text{x}}(X_i)$, having marginalised over all other $X_{j\neq i}$. In our setting, $\mathcal{F}_{\text{x}}(X_i) = \mathcal{F}_{\text{x}}^{\text{eq}}$ and thus no work can be extracted. The correlations are wasted and a 'modularity cost' is incurred due to the fact that before the work extraction there is mutual information between X_i and later input states, but after the work extraction that mutual information is zero[37].

Let us consider a simple extension to the independent-bit device that is interpretable and offers the potential of extracting at least some of the stored work whilst retaining limited complexity. We still manipulate input bits individually, but allow for a memory that maintains its state when the device moves to the next subsystem. This memory permits some of the free energy stored in correlations between successive inputs to be exploited. We now derive a bound on work extraction by this method.

Consider two adjacent input bits labelled i and $i+1$, and the memory system. The initial state of the ith bit is the random variable X_i. X_i can take two values: X or X*. During the interaction of the memory system with the ith

bit, X_i is both measured and recorded in the memory as the state M_i, and work is extracted from the ith bit as it relaxes to a state X_i^{final}. We are now concerned with the work that can subsequently be extracted from the $i+1$th bit following the same procedure, given the correlations between M_i and X_{i+1} induced by the measurement.

Let $\mathcal{F}_{\text{joint}}(X_{i+1}, M_i)$ be the free energy of the joint system consisting of the $i+1$th bit and the memory system when in states X_{i+1} and M_i respectively. Before and after the coupling of the $i+1$th bit and the memory system, there is no direct interaction between the two subsystems, and hence the free energy can be written as the sum of individual contributions calculated using marginalised probabilities and an informational term arising from the correlation between X_{i+1} and M_i as in equation 1.23[23]. Prior to measurement, we have

$$\mathcal{F}_{\text{joint}}(X_{i+1}, M_i) = \mathcal{F}_{\text{X}}(X_{i+1}) + \mathcal{F}_{\text{M}}(M_i) + k_{\text{B}}T\mathcal{I}(X_{i+1}; M_i). \tag{3.7}$$

After the interaction window, we have

$$\mathcal{F}_{\text{joint}}(X_{i+1}^{\text{final}}, M_i) = \mathcal{F}_{\text{X}}(X_{i+1}^{\text{final}}) + \mathcal{F}_{\text{M}}(M_{i+1}) + k_{\text{B}}T\mathcal{I}(X_{i+1}^{\text{final}}; M_{i+1}). \tag{3.8}$$

The work extracted by any process operating between these start and end points is bounded by

$$\begin{aligned} W \leq \mathcal{F}_{\text{X}}(X_{i+1}) - \mathcal{F}_{\text{X}}(X_{i+1}^{\text{final}}) + \mathcal{F}_{\text{M}}(M_i) - \mathcal{F}_{\text{M}}(M_{i+1}) \\ + k_{\text{B}}T\Big(\mathcal{I}(X_{i+1}; M_i) - \mathcal{I}(X_{i+1}^{\text{final}}; M_{i+1})\Big). \end{aligned} \tag{3.9}$$

If the process that produces the inputs is stationary and the measurement protocol is the same each time, then $\mathcal{F}_{\text{M}}(M_i) = \mathcal{F}_{\text{M}}(M_{i+1})$. Moreover, by design, $\mathcal{F}_{\text{X}}(X_{i+1}) = \mathcal{F}_{\text{X}}^{\text{eq}}$ is minimal since the marginalised distribution of X_{i+1} is the equilibrium one. Invoking the positivity of the mutual information[7], we see that the available work is maximal when X_{i+1}^{final} also follows an equilibrium distribution, and the extraction process fully decorrelates the input from the memory ($\mathcal{I}(X_{i+1}^{\text{final}}; M_{i+1}) = 0$). Thus the work extracted per input

bit is bounded by

$$W \leq k_{\mathrm{B}} T \mathcal{I}(X_{i+1}; M_i). \tag{3.10}$$

A system that does not make use of a memory, such as the setup for directly exploiting biased inputs discussed in section 3.3.3, would therefore extract no work.

The value of the mutual information in equation 3.10 depends on the details of the measurement process. The state of the memory system, M_i, only depends on the state of the next input, X_{i+1}, through the previous input state, X_i, so by the data processing inequality the maximum work that can be extracted is

$$\begin{aligned} W_{\mathrm{single}}^{\mathrm{max}} &= k_{\mathrm{B}} T \mathcal{I}(X_{i+1}; X_i) \\ &= k_{\mathrm{B}} T (\ln 2 - H(X_{i+1}|X_i)). \end{aligned} \tag{3.11}$$

This work is, of course, not greater than the available work in the input. The input is stationary so it is possible to write the entropy rate as[7]

$$h = \lim_{n \to \infty} H(X_n | X_1, X_2, \ldots, X_{n-1}), \tag{3.12}$$

and if we use the fact that the conditional entropy is not increased when conditioning on additional variables then

$$\begin{aligned} W_{\mathrm{single}}^{\mathrm{max}} &= k_{\mathrm{B}} T (\ln 2 - H(X_{i+1}|X_i)) \\ &\leq k_{\mathrm{B}} T (\ln 2 - h) = W_{\mathrm{available}}. \end{aligned} \tag{3.13}$$

These results are a special case of the 'modularity cost' outlined in[37].

Single bit memory devices are therefore constrained by the amount of information they carry forward to the next bit in the chain. Note that carrying this information forward is not sufficient—it must also be used effectively during the interaction window. One might assume that there is an inherent trade-off between *updating* the memory to be the best possible predictor of the next bit, and *using* the memory to make the extraction of work from the current bit as efficient as possible. We will now explore this potential

trade-off, and these bounds on work extraction more generally, in the context of two distinct devices in two different types of environment.

Exploiting a Markovian input

We first consider the case in which the binary input is Markovian. That is, the probability distribution of the state of each input molecule only depends on the state of the previous molecule. Since we consider processes which have no bias to either state 0 or 1, this process is a one parameter model given by the probability of transitioning state from one input to the next. The entropy of a series of n Markovian random variables is

$$\begin{aligned} H(X_1, X_2, \ldots, X_n) &= \sum_{i=1}^{n} H(X_i|X_1, X_2, \ldots, X_{i-1}) \\ &= H(X_1) + \sum_{i=2}^{n} H(X_i|X_{i-1}), \end{aligned} \tag{3.14}$$

in which we have first used the chain rule for conditional entropies[7] and second the Markov property. Therefore, from Equation 3.5, the available work if the binary Markov chain is stationary is

$$W_{\text{available}}^{\text{Markov}} = k_{\text{B}}T(\ln 2 - H(X_{i+1}|X_i)). \tag{3.15}$$

Comparing Equations 3.11 and 3.15, we see that the maximum work for a single bit memory is equal to the full available work in a Markovian environment: $W_{\text{single}}^{\text{max}} = W_{\text{available}}^{\text{Markov}}$. We now outline a device that extracts all of this work, both achieving the required measurement accuracy $H(M_i, X_i) = 0$ and using this measurement to extract all of $W_{\text{single}}^{\text{max}}$ for each bit.

We first note that any update of the memory from M_{i-1} to M_i must occur before the ith bit is allowed to evolve. Thermodynamically efficient manipulation of the ith bit requires that any protocol is quasistatic, with the $\text{X} \rightleftharpoons \text{X}^*$ reactions reaching equilibrium with respect to the control faster than the control is updated. Thus, as soon as $\text{X} \rightleftharpoons \text{X}^*$ transitions are allowed by the control, all memory of the previous state is necessarily forgotten, and

subsequent updates of the memory using the initial value of X_i are impossible.

At first glance it might then seem impossible to extract all the work stored in this setting. We must apparently pay to update the memory from M_{i-1} to M_i using input X_i to carry information forward, before we are able to use the memory to exploit X_i. The recent result of[65] does not preclude the possibility of extracting all the stored work, but leaves open the possibility that a single additional 'hidden' state might be required to circumvent this apparent problem (see appendix A.4).

In fact, no additional states are required. The solution is to use the information carried forward from the measurement of the previous input, $\mathcal{I}(X_i; M_{i-1})$, to make a low work-cost but faithful measurement of X_i, (M_i) and then to use that measurement to extract $k_{\mathrm{B}}T\ln 2$ of work from the relaxation of the ith bit exactly as in Szilard's engine, see section 3.4.1. Here, the low work cost is measured relative to the $k_{\mathrm{B}}T\ln 2$ cost of a naïve measurement performed without information carried forward from the previous bit.

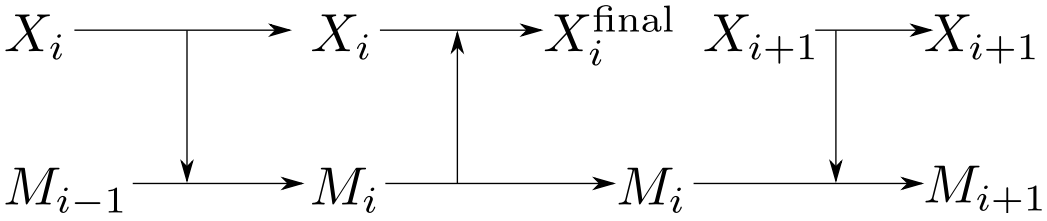


Figure 3.4: *Schematic diagram of an efficient measurement and feedback process.* The machine receives an input molecule characterised by the random variable X_i and the memory molecule is initially characterised by the random variable M_{i-1}. Then, the memory variable is set to M_i, which is a measurement of X_i. The state M_{i-1} does not affect the state M_i but the fact that M_{i-1} and X_i are correlated means that the measurement can be taken cheaply. Next the correlation between M_i and X_i is used to extract work when changing the state of the ith input molecule from X_i to X_i^{final}, which is in an equilibrium distribution. Then, that input molecule is returned to its box and the next input molecule, with state X_{i+1}, is moved to the reaction volume and the process can repeat.

An overview of this process is shown in figure 3.4. First the new input is copied to the memory. This copy is done using the same chemical reactions as in the measurement step of the biochemical Szilard engine in section 3.4.1,

repeated here for convenience

$$\begin{aligned} \mathrm{M}_0 + \mathrm{X} + \mathrm{F}_1^* &\rightleftharpoons \mathrm{M}_1 + \mathrm{X} + \mathrm{F}_1, \\ \mathrm{M}_0 + \mathrm{X}^* + \mathrm{F}_2^* &\rightleftharpoons \mathrm{M}_1 + \mathrm{X}^* + \mathrm{F}_2. \end{aligned} \tag{3.16}$$

The only difference from section 3.4.1 is that now the initial state of input and memory is different. It is still the case that the input molecule and memory molecules are each, when treated in isolation, equally likely to be in both of their states. Now, however, the states of the two molecules are correlated since the memory molecule has been set using the state of the previous input molecule, X_{i-1}. A different measurement protocol is therefore needed to make an optimal (reversible) measurement. Instead of starting from a chemical potential difference $\Delta G_1 = 0$ for the fuels, we must start with either F_1 and F_1^* in excess so that the equilibrium distribution dictated by this buffer matches the actual biased probability distribution of the memory molecule given that the input molecule is X. Similarly, either F_2 and F_2^* must be in excess so that the equilibrium distribution dictated by this buffer matches the biased probability distribution of the memory molecule given that the input molecule is X^*.

The ideal protocol therefore proceeds as follows. Initially, as in the biochemical Szilard engine in section 3.4.1, $[\mathrm{F}_1]$, $[\mathrm{F}_1^*]$, $[\mathrm{F}_2]$ and $[\mathrm{F}_2^*]$ are all set to zero so the reactions in equation 3.16 cannot occur. Then, the concentrations are simultaneously increased at a fixed ratio of $[\mathrm{F}_1]/[\mathrm{F}_1^*]$ and $[\mathrm{F}_2]/[\mathrm{F}_2^*]$ that maintains constant free-energy changes for the reactions of equation 3.16, $\Delta G_1 = \mu_{\mathrm{F}_1} - \mu_{\mathrm{F}_1^*}$ and $\Delta G_2 = \mu_{\mathrm{F}_2} - \mu_{\mathrm{F}_2^*}$, such that

$$\begin{aligned} \frac{\mathrm{e}^{-\beta\Delta G_1}}{1+\mathrm{e}^{-\beta\Delta G_1}} &= P(M_{i-1} = \mathrm{M}_1 | X_i = \mathrm{X}) \\ &= P(X_{i-1} = \mathrm{X}^* | X_i = \mathrm{X}), \\ \frac{\mathrm{e}^{-\beta\Delta G_2}}{1+\mathrm{e}^{-\beta\Delta G_2}} &= P(M_{i-1} = \mathrm{M}_1 | X_i = \mathrm{X}^*) \\ &= P(X_{i-1} = \mathrm{X}^* | X_i = \mathrm{X}^*). \end{aligned} \tag{3.17}$$

The reactions catalysed by whichever of X or X^* is present now occur at an

appreciable rate, but forwards reactions exactly balance reverse reactions so there is no overall change in the probability of observing of M_1 and M_2. If there is no overall bias towards X or X^* then $\Delta G_1 = -\Delta G_2 = \Delta G_{\text{offset}}$ by symmetry. We have used the term 'ΔG_{offset}' because the chemical potential difference has been 'offset' from zero, which is what it would be if the successive input molecules were uncorrelated.

The rest of the protocol is the same as for the measurement step of the biochemical Szilard engine in section 3.4.1. Next, $[F_1]$ and $[F_2^*]$ are increased while $[F_1^*]$ and $[F_2]$ are kept constant until $[F_1] \gg [F_1^*]$ and $[F_2^*] \gg [F_2]$. Then, $[F_1]$, $[F_1^*]$, $[F_2]$ and $[F_2^*]$ are decreased while maintaining $[F_1] \gg [F_1^*]$ and $[F_2^*] \gg [F_2]$ until $[F_1^*] = [F_2] = 0$. Finally $[F_1]$ and $[F_2^*]$ are decreased to zero. Now the reactions in equation 3.16, again, cannot occur so the memory molecule is fixed to be M_0 if the input is X and M_1 if the input is X^*.

The work done by the chemical fuel baths to make this measurement is once more calculated as in section 3.3.3 but with different limits on the integral due to the different ΔG_{offset}. As shown in appendix A.5.1, the work done is exactly $k_{\text{B}}TH(M_{i-1}|X_i) = k_{\text{B}}TH(X_{i-1}|X_i)$, as expected from the change in entropy of the input molecule and memory molecule joint system.

Now that the state of the memory molecule has been updated so that M_i perfectly reflects X_i, $k_{\text{B}}T\ln 2$ work is extracted in exactly the same way as in the biochemical Szilard engine. Thus the net work extracted per input molecule is

$$W^{\text{Markov}} = k_{\text{B}}T(\ln 2 - H(X_{i+1}|X_i)) = W^{\text{max}}_{\text{single}} = W^{\text{Markov}}_{\text{available}}, \tag{3.18}$$

which is all the available work in a stationary Markovian input as in equation 3.15. This machine has 100% efficiency and there is no irreversible dissipation.

It is therefore possible that a machine with a two-state memory that is well-calibrated for this Markovian environment—with the correct initial ΔG_{offset} in chemical potentials to reflect the nearest-neighbour correlations in the Markov chain—can extract all of the work available. Such a machine faces no trade-offs between exploiting and measuring X_i; the exact measurement

of X_i both carries the maximal information forward, and enables its full exploitation. How a machine might obtain the optimal offset parameter, either via design or some form of evolution (to effectively infer the one parameter specifying the Markov process), is beyond the scope of this chapter. We emphasise that calibration of ΔG_{offset} to the environment is not equivalent to being tuned to the specific fluctuations of one realisation of the environment, but rather to the overall statistical properties of the fluctuations. A poorly-chosen parameter would result in the 'mismatch costs' identified by Kolchinsky and Wolpert[66].

Exploiting a non-Markovian input

In a Markovian environment, if a machine measures the state of an input molecule it knows everything it could about the distribution of the next input. A more complex environment might have correlations that are not fully-described by those of adjacent inputs. In particular, we might imagine an environment with a hidden state S_i that influences the probability of X_i; as the hidden state changes, the device moves between regions in which the apparent environmental bias is different. The machine's challenge then becomes a more obvious inference task: to infer the overall state of the environment, and to accordingly exploit the inputs.

Specifically, we will consider a hidden state S_i with $s_i \in \{0, 1\}$. When moving from one input molecule to the next the hidden state has a probability k of changing. Conditioned on the hidden state, each input molecule is an independent Bernoulli random variable. The probability of an X* molecule is α if $S_i = 0$ and $1 - \alpha$ if $S_i = 1$. Some example sequences produced by this process are shown in figure 3.5. Due to the overall symmetry of the process, $X_i = \text{X}$ and $X_i = \text{X}^*$ are both equally likely having marginalised over all inputs $j \neq i$. Thus, as in section 3.4.2, no free energy is stored in the state of single molecules—only in the correlations between different molecules. The available work that can be extracted per input molecule is plotted against the parameters k and α in figure 3.6. Hidden states that either reliably persist ($k \to 0$) or switch ($k \to 1$), and which provide a predictable output

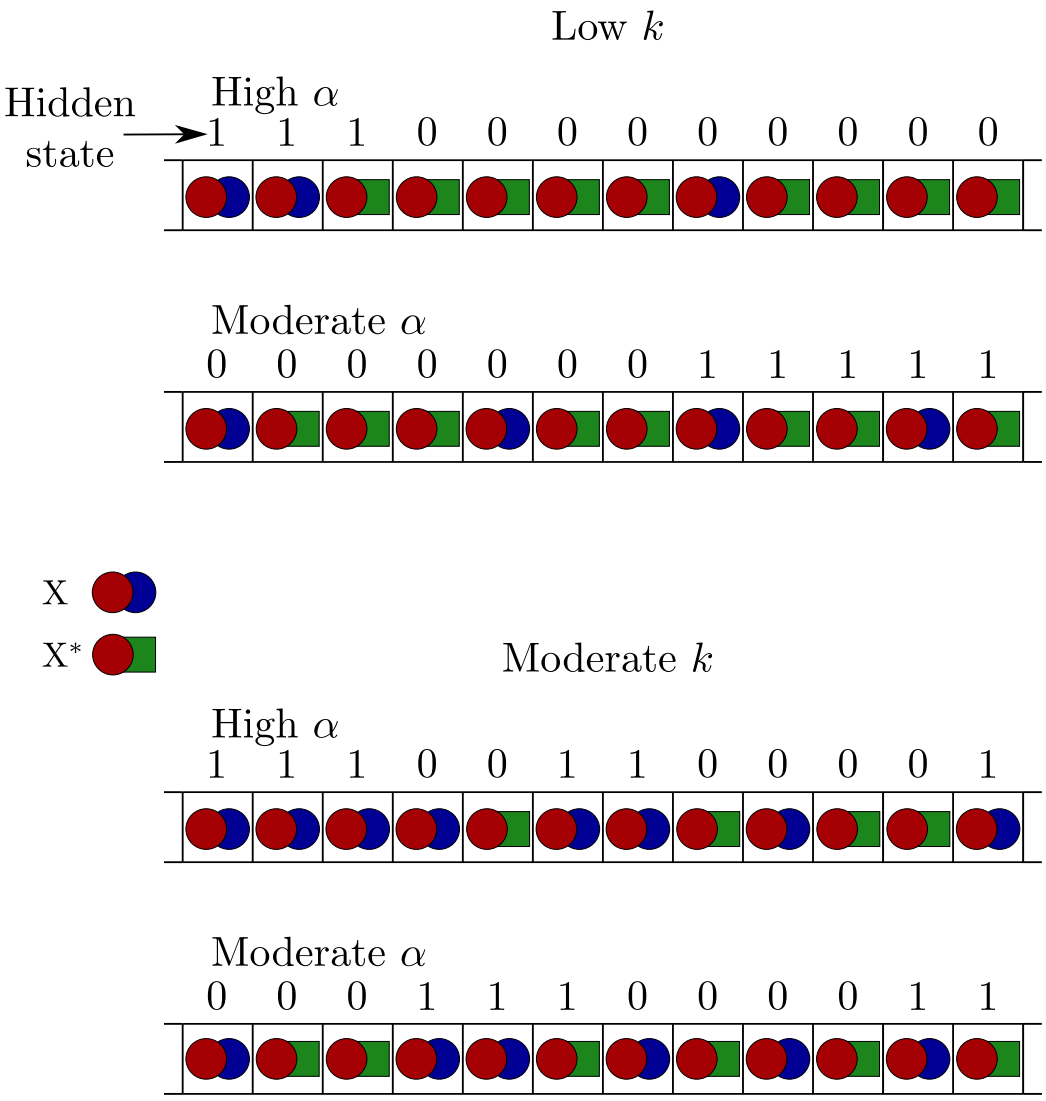


Figure 3.5: *Samples of inputs produced by the hidden Markov model for various values of the parameters.* When k is small, the hidden state has a low probability of changing at each step so there are long strings in which the system is biased towards either X or X^*. When k is higher then the hidden state changes more frequently, and the bias is more short-lived. When α is high then the state of the input molecule is strongly biased to match the hidden state, whereas when α is smaller, there is a weaker correlation between the hidden state and the state of the input molecule.

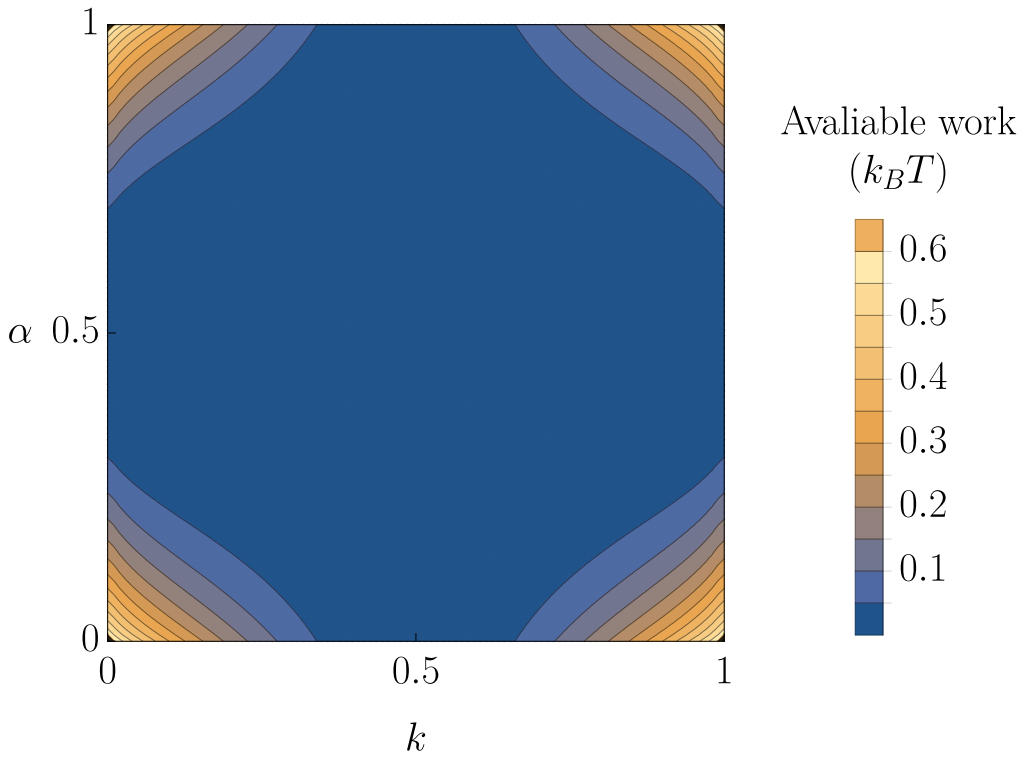


Figure 3.6: *Available work per molecule for the hidden Markov model input.* The available work is calculated as a function of bias α and persistence of the hidden state k by numerically evaluating equations 3.5 and 3.6 for $n = 20$ and is given in units of $k_{\mathrm{B}}T$.

$(\alpha \to 0, 1)$ lead to the most free energy stored in the environment.

Given the history of the inputs $\{X_{j<i}\}$, the optimal statistical prediction of the next input X_i can be made via the forward algorithm[67]. A machine capable of both iterating the forward algorithm at each step, and using the previous value to optimally exploit the current input, would be able to extract the full $W_{\text{available}}$. However, implementing the forward algorithm is impossible for our machine with a single bit of memory that can make only a binary 'decision' during its feedback. For a hidden Markov process, the conditional probability distribution of the next input molecule given the entire history of the input is different for all possible states of the history. Equivalently, the process $\{X_i\}$ cannot be described by a finite state ϵ-machine[68], and thus the forward algorithm requires a memory that is a real number, and the exploitation step would need to have a continuous dependence on this real number.

It might be tempting to think that a simpler alternative to the forward algorithm, in which the two-bit memory variable M_i is set based on both the current input variable X_i and its previous value M_{i-1}, would give better predictions by allowing the machine to take in more historical information at each step. Such an approach would represent a trade-off, with a maximal information carried forward $\mathcal{I}(X_{i+1}; M_i)$ being obtained only at the expense of an increased uncertainty $H(X_i, M_i)$ in the state of the current input after the measurement. Whether or not the reduced measurement cost could compensate for the reduction in work obtained during the extraction step is moot, however, since such a strategy is impossible, at least in the quasistatic setting. One cannot update the memory from M_{i-1} to M_i quasistatically, in a way such that $\mathcal{I}(M_i; M_{i-1}) \neq 0$, without access to additional hidden memory states[69,65]. All information on initial conditions is necessarily lost immediately when a degree of freedom evolves under a quasistatic process. Thus in the quasistatic setting at least, our single bit memory cannot trade off the accuracy of measurement of the current input and information carried forward.

With the above limitations in mind, we first ask how well the *Markov machines* considered in section 3.4.2, that are limited to interact with one bit at a time, and carry only one bit of memory forward, function in the non-Markovian environment specified. For a perfect measurement of each bit, such that $H[X_i|M_i] = 0$, the expected work extracted per molecule for a quasistatically-operated device still follows from Equation 3.11

$$W_{\text{single}}^{\text{max}} = k_{\text{B}}T\Big(\ln 2 - H(X_{i+1}|X_i)\Big), \tag{3.19}$$

but now $W_{\text{single}}^{\text{max}} < W_{\text{available}}$ since there is additional information in long range correlations that is not taken into account by the information between nearest neighbours. Therefore, the machine has efficiency $\eta = W_{\text{single}}^{\text{max}}/W_{\text{available}} < 1$ and irreversibly generates entropy.

This efficiency, η, of Markov machines acting on a hidden Markov model input is plotted in figure 3.7(a). In making these plots, we first identify the optimal Markov machine offset parameter at each α and k, and then calculate the efficiency of that device—once again assuming that the machine's parameters are optimised to the statistical properties of its environment (perhaps through evolution). It is notable that the Markov machines perform reasonably well in these environments, except when $k \to 0$ or 1, and $\alpha \not\approx 0$ or 1. In these environments, the hidden state behaves predictably and so correlations are long-ranged, but X_i fluctuates considerably within the hidden state, effectively fooling the Markov machine that is only able to predict X_i based on X_{i-1}.

The behaviour of the Markov machine can be related to that of the Kalman filter[70], an algorithm for making real-time predictions of the state of a noisy dynamical system with noisy measurements of the system's state. The relative weight put on previous measurements versus the most recent input is a parameter that can be adjusted, and it is well known that high intrinsic noise implies that the current measurement should be weighted strongly, whereas high measurement noise calls for greater emphasis on the previous measurements. The Markov machine is effectively constrained to put all of

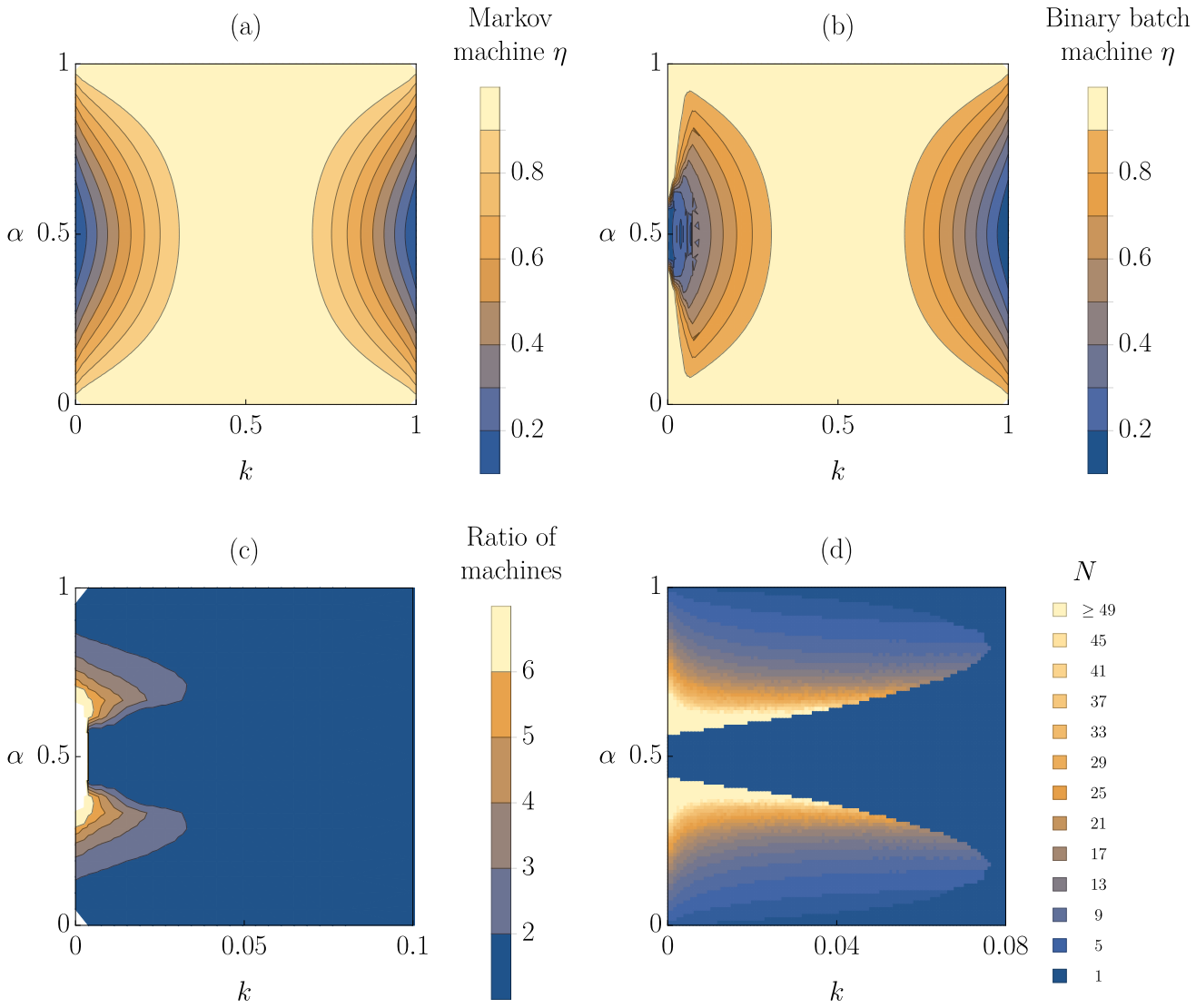


Figure 3.7: *More complex inference strategies can be beneficial in more complex environments.* (a) The efficiency η of the Markov machine when acting on an input produced by a hidden Markov model. The efficiency is low when the hidden state switching probability k is close to 0 or 1 but α, the bias implied by a given hidden state, is not close to 0 or 1 because there are long range correlations that are masked by short-term fluctuations. The available work is calculated as in figure 3.6 and we have an analytic expression for work extracted by the Markov machine from equation 3.18. (b) The efficiency of the binary batch machine. The efficiency is the same as the Markov machine except for the region close to $k = 0$, where it shows enhanced work extraction. The work extracted by the binary batch machine is calculated by simulating the hidden Markov model for 10,000,000 steps and calculating the work extracted per molecule for the binary batch machine with N from 1 to 25 and selecting the greatest work extracted. (c) The ratio between the work extracted by the binary batch machine and the Markov machine per molecule, clarifying the enhanced performance near to $k = 0$. (d) The batch size that gives the greatest expected work as a function of the input parameters. For $k \gtrsim 0.077$ it is always optimal to interact with one molecule at a time; larger batches are favourable for $k \to 0$ and values of α away from 0, 0.5 and 1. It is always the case that the optimum batch size is odd.

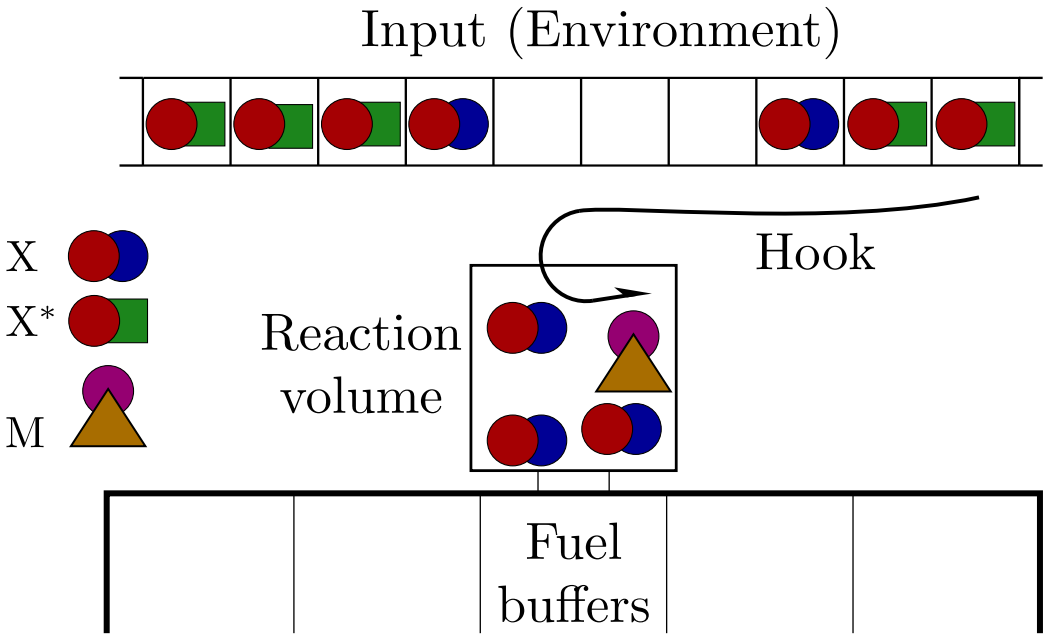


Figure 3.8: *Schematic of a batch machine for exploitation of non-Markovian environments.* The setup is similar to figure 3.1, so we have omitted the illustration of the control acting on multiple systems simultaneously for clarity. The difference is that now multiple input molecules are moved from their boxes to the reaction volume by the hook. In this figure we have illustrated the case of $N = 3$. The input molecules can freely diffuse inside the reaction volume, so their order in the input is lost.

its emphasis on the most recent measurement; it therefore functions better when the 'intrinsic' noise of the hidden state is relatively high ($k \sim 0.5$ and $\alpha \sim 0, 1$), and worse when the 'measurement' noise of the inputs is relatively large ($k \sim 0, 1$, $\alpha \sim 0.5$).

Batch averaging machines in non-Markovian environments

We now ask whether a more sophisticated strategy, still involving only a single memory molecule and a single binary decision, can overcome this weakness of the Markov machine. If we consider the region where $k \approx 0$, then it is likely that a run of multiple input molecules will be produced by the same hidden state. Inspired by our analogy with the Kalman filter, we look for a mechanism of somehow considering multiple input molecules to provide more reliable information about the hidden state, allowing more efficient work ex-

traction. Indeed, in the context of cellular sensing of the concentration of external ligands[54,3], it has been observed that averaging approaches can be beneficial when correlation times in the environment are long[71].

We therefore introduce the *batch machine*, illustrated in figure 3.8, which is similar to the Markov machine except that it interacts with (i.e. measures and exploits) a batch of multiple molecules simultaneously, rather than just one. An N-batch machine operates by: (a) transferring N inputs to the reaction volume (with no work cost—see appendix A.2); (b) performing an operation to set a memory based on these N inputs (for a low work cost because the state of the batch is correlated with the state of the memory, which is set based on the state of the previous batch); (c) exploiting the N inputs simultaneously using the memory; and (d) transferring the N inputs back to their array in a random order.

We will first consider a 'binary' machine that, like the Markov machine, has only two measurement reactions and two work extraction reactions. Let J_i be the random variable representing whether the number of X^* molecules in batch i is greater than half the batch-size, $N/2$ ($J_i = 1$ if true, 0 otherwise). The machine performs measurement of batch i by setting the memory molecule to $M_i = \mathrm{M}_0$ if $J_i = 0$, and to $M_i = \mathrm{M}_1$ if $J_i = 1$; we note that other binary measurement choices are possible, but this simple one serves to illustrate the possibilities of a more complex inference strategy. The machine then exploits the imbalance of inputs in the same way that the Markov machine exploits a measured $X_i = \mathrm{X}$ or $X_i = \mathrm{X}^*$, by allowing the inputs to relax to an unbiased distribution whilst transferring free energy to chemical buffers. For $N = 1$, the binary batch machine is identical to the Markov machine of section 3.4.2; for $N > 1$ the initial measurement essentially performs an average over N inputs to set its memory. In the limit $N \to \infty$, the batch machine interacts with all molecules at once. However, with only two possible measurement states (and hence two possible work extraction strategies), this limit is generally inefficient.

The measurement can be done with the reactions

$$\begin{aligned} \mathrm{M}_0 + \frac{N+1}{2}\mathrm{X} + \mathrm{F}_1^* &\rightleftharpoons \mathrm{M}_1 + \frac{N+1}{2}\mathrm{X} + \mathrm{F}_1, \\ \mathrm{M}_0 + \frac{N+1}{2}\mathrm{X}^* + \mathrm{F}_2^* &\rightleftharpoons \mathrm{M}_1 + \frac{N+1}{2}\mathrm{X}^* + \mathrm{F}_2, \end{aligned} \tag{3.20}$$

when N is odd and

$$\begin{aligned} \mathrm{M}_0 + \frac{N}{2}\mathrm{X} + \mathrm{F}_1^* &\rightleftharpoons \mathrm{M}_1 + \frac{N}{2}\mathrm{X} + \mathrm{F}_1, \\ \mathrm{M}_0 + \left(\frac{N}{2}+1\right)\mathrm{X}^* + \mathrm{F}_2^* &\rightleftharpoons \mathrm{M}_1 + \left(\frac{N}{2}+1\right)\mathrm{X}^* + \mathrm{F}_2, \end{aligned} \tag{3.21}$$

when N is even. There is an $N/2$ in one of the reactions in equation 3.21 and an $N/2+1$ in the other because for even N there is an arbitrary choice as whether to assign the state where there are N/x molecules of X^* in the batch to M_0 or M_1. We have chosen to assign the state with $N/2$ X^* molecules to M_0. Clearly, if there are N molecules of X and X^* in total, then only one of these reactions in equations 3.20 and 3.21 can occur at once. As a result, if the reactions are once more driven in opposite directions by fuel imbalances, an excess of X molecules can be used set the memory to M_0, and an excess of X^* can be use to set the memory to M_1. We immediately see the price for a more complicated strategy—our reactions now require $\sim N/2$ molecules to act as combined catalysts, rather than just a single input molecule (see appendix A.8 for a DNA strand displacement design for these reactions).

The protocol of changes to the fuel molecule concentrations required for measurement is very similar to that of the Markov machine. Initially, $[\mathrm{F}_1]$, $[\mathrm{F}_1^*]$, $[\mathrm{F}_2]$ and $[\mathrm{F}_2^*]$ are all set to zero so the reactions in equations 3.20 and 3.21 cannot occur. Then, the concentrations are simultaneously increased at a fixed ratio of $[\mathrm{F}_1]/[\mathrm{F}_1^*]$ and $[\mathrm{F}_2]/[\mathrm{F}_2^*]$ that maintain overall free energy

changes for the reactions, $\Delta G_1 = \mu_{\mathrm{F}_1} - \mu_{\mathrm{F}_1^*}$ and $\Delta G_2 = \mu_{\mathrm{F}_2} - \mu_{\mathrm{F}_2^*}$, such that

$$\begin{aligned}
\frac{\mathrm{e}^{-\beta\Delta G_1}}{1+\mathrm{e}^{-\beta\Delta G_1}} &= P(M_{i-1} = \mathrm{M}_1 | J_i = 0) \\
&= P(J_{i-1} = 1 | J_i = 0), \\
\frac{\mathrm{e}^{-\beta\Delta G_2}}{1+\mathrm{e}^{-\beta\Delta G_2}} &= P(M_{i-1} = \mathrm{M}_1 | J_i = 1), \\
&= P(J_{i-1} = 1 | J_i = 1).
\end{aligned} \tag{3.22}$$

In exactly the same way as in the biochemical Szilard engine and in the Markov machine, depending on whether there are more X or X^* molecules in the batch, one of the measurement reactions can now occur at an appreciable rate. The initial offsets ΔG_1 and ΔG_2 allow information between batches to be exploited, and are exactly analogous to the constant offsets introduced in section 3.4.2. Like in the Markov machine, if N is odd then $\Delta G_1 = -\Delta G_2 = \Delta G_{\mathrm{offset}}$ by symmetry. If N is even then $\Delta G_1 \neq -\Delta G_2$ because $P(J_i = 0) \neq P(J_i = 1)$.

Then, as in the previous machines, $[\mathrm{F}_1]$ and $[\mathrm{F}_2^*]$ are increased while $[\mathrm{F}_1^*]$ and $[\mathrm{F}_2]$ are kept constant until $[\mathrm{F}_1] \gg [\mathrm{F}_1^*]$ and $[\mathrm{F}_2^*] \gg [\mathrm{F}_2]$. Subsequently, $[\mathrm{F}_1]$, $[\mathrm{F}_1^*]$, $[\mathrm{F}_2]$ and $[\mathrm{F}_2^*]$ are decreased while maintaining $[\mathrm{F}_1] \gg [\mathrm{F}_1^*]$ and $[\mathrm{F}_2^*] \gg [\mathrm{F}_2]$ until $[\mathrm{F}_1^*] = [\mathrm{F}_2] = 0$. Finally $[\mathrm{F}_1]$ and $[\mathrm{F}_2^*]$ are decreased to zero. Now the reactions in equations 3.20 and 3.21, again, cannot occur and the memory molecule has been set to state M_0 if the batch contains more X molecules that X^* or equal number of X and X^* molecules, and to state M_1 if the batch contains more X^* molecules than X.

The cost of making the measurement is calculated in exactly the same way as for the Markov machine (see appendix A.7.1), and gives

$$W^{\mathrm{measure}} = -H(J_{i+1}|J_i). \tag{3.23}$$

The negative sign represents negative work extraction.

Subsequently, work is extracted from the correlated state of the measurement molecule and the batch. The binary batch machine uses the same reactions

as the biochemical Szilard engine and the Markov machine to extract work; they are repeated here for convenience:

$$\begin{aligned} \mathrm{M}_0 + \mathrm{X} + \mathrm{F}_3^* &\rightleftharpoons \mathrm{M}_0 + \mathrm{X}^* + \mathrm{F}_3, \\ \mathrm{M}_1 + \mathrm{X} + \mathrm{F}_4^* &\rightleftharpoons \mathrm{M}_1 + \mathrm{X}^* + \mathrm{F}_4. \end{aligned} \tag{3.24}$$

However, the protocol is modified, which is necessary because the state of the memory molecule does not report perfectly on the state of the inputs: any number of molecules in state X^* greater than $N/2$ correspond to $J_i = 1$ and hence $M_i = \mathrm{M}_1$, for example. Initially $[\mathrm{F}_3] = [\mathrm{F}_3^*] = [\mathrm{F}_4] = [\mathrm{F}_4^*] = 0$, along with the fuels used in the measurement process. Then, the concentrations are simultaneously increased at a fixed ratio of $[\mathrm{F}_3]/[\mathrm{F}_3^*]$ and $[\mathrm{F}_4]/[\mathrm{F}_4^*]$ that maintain an overall free energy changes for the reactions, $\Delta G_3 = \mu_{\mathrm{F}_3^*} - \mu_{\mathrm{F}_3}$ and $\Delta G_4 = \mu_{\mathrm{F}_4^*} - \mu_{\mathrm{F}_4}$, such that

$$\begin{aligned} \frac{\mathrm{e}^{-\beta\Delta G_3}}{1 + \mathrm{e}^{-\beta\Delta G_3}} &= \hat{p}_0, \\ \frac{\mathrm{e}^{-\beta\Delta G_4}}{1 + \mathrm{e}^{-\beta\Delta G_4}} &= \hat{p}_1, \end{aligned} \tag{3.25}$$

where $\hat{p}_0$ is the probability that an input molecule in the batch is in the state X^*, conditioned on $J_i = 0$, and $\hat{p}_1$ is the probability that an input molecule in the batch is in the state X^* conditioned on $J_i = 1$. It is clear that $[\mathrm{F}_3] > [\mathrm{F}_3^*]$ (assuming that the F_3 and F_3^* molecules have equal free energy) because $\hat{p}_0 < 1/2$ and $[\mathrm{F}_4] < [\mathrm{F}_4^*]$ because $\hat{p}_1 > 1/2$. On average, no chemical work is done at this stage. Similar to the measurement, if N is odd then $\Delta G_3 = -\Delta G_4$ by symmetry. If N is even then $\Delta G_3 \neq -\Delta G_4$ because $\hat{p}_0 \neq 1 - \hat{p}_1$.

Work can then be extracted from the batch exactly as for N independent molecules with a bias represented by $\hat{p}_0$ and $\hat{p}_1$. Therefore, $[\mathrm{F}_3^*]$ and $[\mathrm{F}_4]$ are increased quasistatically until the free energy differences, ΔG_3 and ΔG_4, are zero. Then $[\mathrm{F}_3]$, $[\mathrm{F}_3^*]$, $[\mathrm{F}_4]$ and $[\mathrm{F}_4^*]$ are decreased to zero while maintaining the same $[\mathrm{F}_3]/[\mathrm{F}_3^*]$ and $[\mathrm{F}_4]/[\mathrm{F}_4^*]$ ratios. When $[\mathrm{F}_3] = [\mathrm{F}_3^*] = [\mathrm{F}_4] = [\mathrm{F}_4^*] = 0$ again the reactions in equation 3.24 cannot occur. So, finally, the batch reaches an unbiased equilibrium, and during this process the free energy of

the buffers are increased. The work extracted in this step is simply N times the work extracted from one input molecule with a bias of $\hat{p}_0$ if $J_i = 0$ or $\hat{p}_1$ if the number of $J_i = 1$. It is therefore

$$\begin{aligned} W_{\text{batch}}^{\text{extract}} =& P(J_i = 0) k_{\text{B}} T N (\ln 2 + \hat{p}_0 \ln \hat{p}_0 + (1 - \hat{p}_0) \ln(1 - \hat{p}_0)) \\ &+ P(J_i = 1) k_{\text{B}} T N (\ln 2 + \hat{p}_1 \ln \hat{p}_1 + (1 - \hat{p}_1) \ln(1 - \hat{p}_1)). \end{aligned} \quad (3.26)$$

Therefore, the net work extracted by the binary batch machine from one batch is

$$\begin{aligned} W_{\text{batch}} = k_{\text{B}} T N \Big(& \ln 2 + P(J_i = 0) \big(\hat{p}_1 \ln \hat{p}_1 + (1 - \hat{p}_1) \ln(1 - \hat{p}_1) \big) \\ & + P(J_i = 1) \big(\hat{p}_2 \ln \hat{p}_2 + (1 - \hat{p}_2) \ln(1 - \hat{p}_2) \big) \Big) - k_{\text{B}} T H(J_{i+1} | J_i)]. \end{aligned} \quad (3.27)$$

As with the Markov machine, we can ask the question of how the optimal batch machine (with N and the free-energy offsets of the fuel baths optimally tuned to the environmental parameters k and α), would perform. Note that since the binary batch machine with $N = 1$ is a Markov machine, the optimal binary batch machine must perform at least as well as the optimal Markov machine.

The efficiency of the optimal binary batch machine is plotted for different values of α and k in figure 3.7(b), showing apparently higher efficiency than the optimal Markov machine for some values of k and α as $k \to 0$. To make this comparison clearer, in figure 3.7(c) we have plotted the work extracted by the binary batch machine per molecule divided by the work extracted by the Markov machine per molecule. We see that there are two regions where the binary batch machine extracts more work. Also, in figure 3.7(d) we have plotted the optimal batch size for the binary batch machine for the different values of the parameters. For $k > 0.08$ the optimum batch size is always 1 so the Markov machine and the binary batch machine are the same, but for smaller values of k larger batches are frequently favoured. It is always the case that the optimum batch size is odd, since the extraction reactions of the binary batch machine cannot extract work from a batch with equal numbers of X and X* molecules.

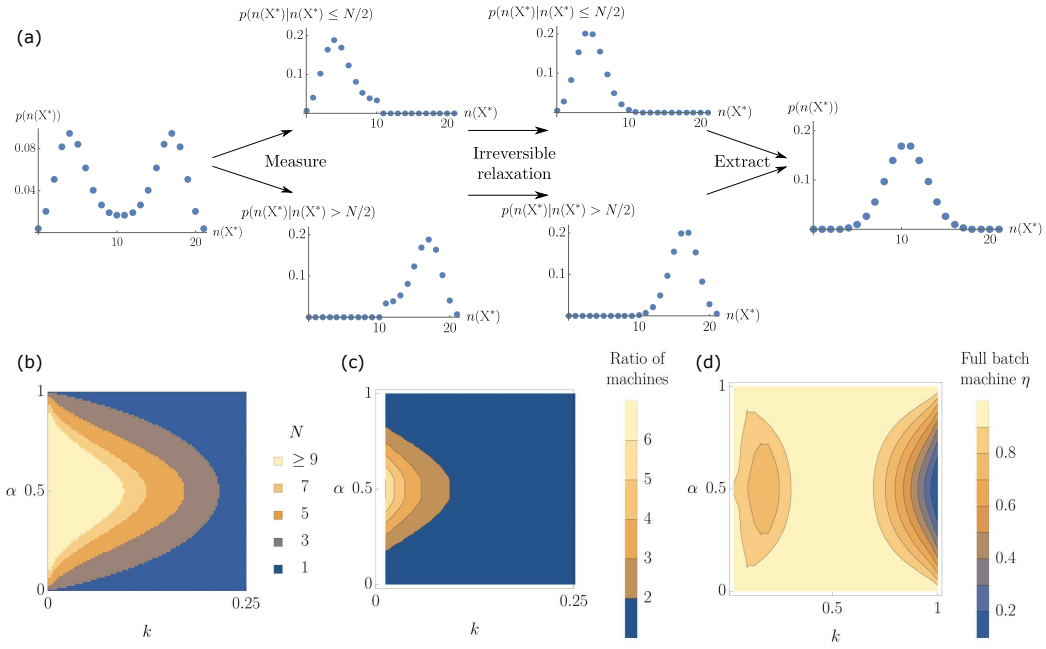


Figure 3.9: *Inefficiency in the operation of the batch machine arises from uncontrolled relaxations prior to work extraction.* (a) Evolution of the probability distribution of the number of molecules in the X^* state during the operation of the batch machine. These probabilities can be calculated exactly from the probabilities of each possible output from the hidden Markov model of length 20. Initially, there is a distribution produced by the hidden Markov model (in this case with $k = 0.8$ and $\alpha = 0.01$). Then, whether there are more than $N/2$ X^* molecules in the batch is measured, which produces two possibilities with associated conditional probabilities. Next, prior to the work extraction stage of the protocol, the reaction volume is put in contact with a buffer of fuel molecules with chemical potential differences as defined in equation 3.25, which has a ratio of fuel molecule concentrations consistent with the expected number of X and X^* molecules. The batch of input molecules irreversibly relaxes to equilibrium with the buffer, changing to a binomial distribution of $n(\mathrm{X}^*)$ from the post-measurement distribution. No work is extracted on average because both distributions have the same average of $n(\mathrm{X}^*)$. Finally, work is extracted by shifting the probability distribution to the equilibrium distribution. A putative 'full' batch machine that avoids the loss of this irreversible relaxation can out-perform the batch machine, as shown in (b)-(d). (b) The greatest expected work as a function of the input parameters for the full batch machine does not have an optimum batch size of 1 for $\alpha \approx 0.5$. The optimum N has been found by numerically calculating the work extracted for the values of N up to $N = 9$ using the equations in appendix A.9. (c) The ratio of the work extracted by the full batch machine and the work extracted by the Markov machine. In contrast to the binary batch machine this full batch machine can extract more work than the Markov machine when k is close to 0 and α is close to 0.5. (d) The efficiency η of the full batch machine.

The binary batch machine delivers, at least in part, on the prospect of improving work extraction from an environment with more complexity. It is unsurprising that a long hidden state life time, $k \to 0$, is necessary for this advantage to be manifest: the averaging strategy will clearly fare poorly when the hidden state switches rapidly. When α is close to 0 or 1 the state of the input molecule reflects the hidden state with a high probability so the string of input molecules is approximately Markovian, preventing the batch machine from finding a competitive advantage. The most subtle question, however, is why the binary batch machine does not extract more work than the Markov machine when $\alpha \approx 0.5$ and $k \to 0$. Naïvely, this regime would seem to be ideal for the batch machine to extract work from weak, but long-lived biases towards either X or X*. From the perspective of the analogy with Kalman filters, this regime should favour the approach that considers a wide range of inputs, rather than just the most recent. To understand why this intuition fails, we consider where the thermodynamic losses occur during the operation of the binary batch machine.

Several stages of the operation of the optimal binary batch machine are thermodynamically irreversible, resulting in efficiencies $\eta < 1$. They include the point at which the memory is updated without taking into account correlations between *non*-neighbour batches; the point at which the batch of N input molecules are mixed within the reaction volume (figure 3.8); and the point at which the work extraction begins using the measurement molecule. In the first process, a modularity cost is incurred. In the second, mixing causes the positional order within a batch to be lost, reducing our ability to extract work from the sequence of molecules within the batch. All that remains is a non-equilibrium distribution of the number of molecules in each state. In the third process, this non-equilibrium distribution relaxes further to a binomial distribution for the number of X* with parameter $\hat{p}_0$ if $J_i = 0$ or $\hat{p}_1$ if $J_i = 1$, with no work extracted on average during this relaxation, as shown in figure 3.9(a).

We can imagine a putative 'full batch machine' that could extract all of the work available from the unordered batch, after the initial mixing and measurement. Such a machine would require additional extraction processes

to which the memory could couple in each state. The optimal batch size for this full batch machine is plotted in figure 3.9(b). We can see that for this machine it is not the case that the optimal batch size is 1 when $\alpha \approx 0.5$. The contour plot for this machine is more similar to expectations: as $k \to 0$ optimal batch size increases for all values α. We have also plotted the ratio between the work extracted by the full batch machine and the Markov machine in figure 3.9(c), and see that the full batch machine extracts more work than the Markov machine when k is close to 0 and $\alpha \approx 0.5$. Thus the reason that the binary batch machine fails to provide an improvement in the vicinity of $\alpha = 0.5$ is at least in part because the free energy wasted during the simple binary work extraction mechanism is too large compared to the relatively low amounts of work available (as seen in figure 3.6).

Robustness

On average, all the machines can extract a positive amount of work from each input molecule or batch of input molecules. However, in a single realisation of the input produced by the stochastic process it is possible for the machines to extract a negative amount of work; i.e. lose free energy, since the prediction of the upcoming state is only probabilistic even in the best case.

Thus, the total work extracted by any machine is a biased random walk. If the machine is unlucky it can receive a fluctuation in the input and get many negative steps with few positive steps. If we are imagining that the machine needs to harvest enough work to power its decision-making, like a biological organism, a fluctuation in its environment where it loses all of its stored free energy would be disastrous. We therefore also consider fluctuations in the work extracted by the machines. If one protocol has a higher expected work extraction but a larger variance it might not be truly better.

The expected worst-case energy-loss—the infimum of the work extracted—can be thought of as the starting larder-size/fuel-reserves that such a reasoning machine requires. It also gives a minimum timescale that any machine would need to run before it could create a replica that is also robust to environmental fluctuations. This infimum of the total work extracted by the

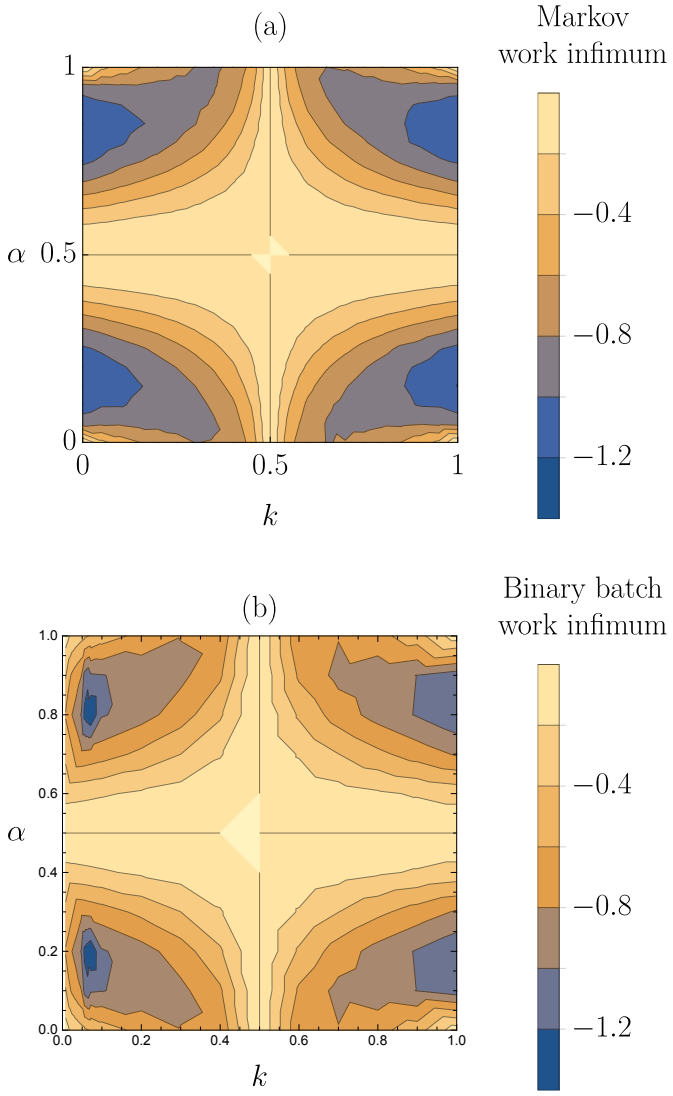


Figure 3.10: *The binary batch machine is more robust than the Markov machine in the regions in which it extracts more work.* (a) The mean infimum of work of the Markov machine in a run of 100 molecules averaged over 1000 trajectories. (b) The mean infimum of work of the binary batch machine in a run of 100 molecules averaged over 100000 trajectories. We see that in the region where the optimal batch size is greater than 1 (shown in figure 3.7(d)) the magnitude of the negative fluctuations is decreased compared to the Markov machine.

machines in a trajectory averaged over many simulated trajectories is plotted against the parameters of the input process in figure 3.10. When $k \approx 0.5$ or $\alpha \approx 0.5$, the work that is extracted by the machines is small so the size of the

negative fluctuations are also small for both machines. Comparing figures 3.10(a) and (b) shows that the binary batch machine exhibits reduced fluctuations in the regions where k is close to 0 and α is not close to 0, 0.5 or 1, where a batch size greater than 1 is favoured by the average work extracted. This fact is perhaps unsurprising, given that averaging over many inputs is inherently conservative.

3.5 Discussion

We have considered the question of how minimal molecular devices might be designed to exploit the free energy stored in simple non-equilibrium environments. Having outlined a concrete design for a biomolecular Szilard engine, we have shown how such a device can form the basis of machines for exploiting a correlated series of molecular bits, expanding on previous work that has only considered environments with a very particular structure[53,32,72].

Although our devices require externally-applied protocols to operate, all information-processing is performed by degrees of freedom that are explicitly represented as biomolecules undergoing reactions in dilute solution—there are no concealed degrees of freedom. As a result, the complexity of implementing minimal systems that exhibit efficient measurement and feedback is made clear, and ambiguities are eliminated. In particular, we have outlined a molecular mechanism for implementing sequential measurement and feedback in an explicit setting, providing clarity not only to the extended correlation-exploiting devices, but also to our representation of the canonical Szilard engine itself. The continuing debate surrounding such devices (see references in[13,23]) shows the importance of a concrete physical representation.

For an environment with no structure—without correlations between successively encountered molecular bits—there is no need to process information and all of the available free energy can be extracted as work without use of a memory or any decision making. For a Markovian array with non-zero correlations between consecutive bits, we show that a simple two-state memory that can select one of two work extraction protocols can extract all of the stored free energy in the environment. The two-state memory is sufficient

to carry all of the available information about the future of the environment forward, and we have identified a protocol that is simultaneously optimal for updating the memory according to the current input, and exploiting said input. For a more complex environment, involving a hidden variable that can only be inferred by the machine through noisy measurements, we argue that a machine with a finite memory cannot extract all of the available free energy as work. We demonstrate that in such a setting, a more complex strategy involving effectively averaging over a batch of molecules can be advantageous if correlations are long-ranged, but noise is substantial. This is similar to the result in[73] that a more complex predictive model is advantageous in a more complex environment, but, in this chapter, we give an explicit physical model for how our machines measure and exploit the environment. In our design, the complexity of the mechanism involved the ability to couple to multiple inputs simultaneously; we predict that alternatives (such as systems with larger memories and more possible decision) would also show the potential for improved performance.

A real living system must not only extract enough resources from its environment on average, but also over short intervals. In any fluctuating environment, an unlucky sequence of events might lead to starvation and death. We probe this situation in our minimal setting by considering the typical infimum (lowest point) of the work extracted by our devices, which represents the typical scale of negative fluctuations. We find that the more sophisticated inference strategy considered here also has smaller negative fluctuations when it is favourable on average, suggesting that it truly can be advantageous. In a minimal living system, reduced negative fluctuations would correspond to the need for a smaller reserve of energy, and the ability to produce viable offspring more quickly, since each offspring would need to be provided with the reserves to deal with typical negative fluctuations for a large fraction to survive.

The minimal devices we consider are clearly unnatural, and constitute only a first step towards understanding the physics of living or life-like systems that make simple decisions. A key feature of our design is that only single copies of some molecular species are present; it is an open question how to

design optimal systems in which the information-processing components, and indeed the inputs, fluctuate more widely. We do note, however, that much of the molecular decision making within cells occurs at the level of transcription factors binding to DNA—these transcription factor binding sites are present with a low and predictable copy number. Future work will focus on constructing minimal models in which the systems are autonomous, requiring no external control, and power their own information-processing tasks by the free energy harvested. A major challenge here would be to implement reliable and efficient measurement and feedback without an externally imposed clock that allows sequential operation, as in this work. More realistic environments of fluctuating chemical concentrations, rather than input molecules that arrive one-by-one, will need to be considered. In such descriptions, it will be necessary to construct a more detailed kinetic model of the underlying elementary reactions than we present here. A deeper question is whether we can design concrete systems that actually learn the statistics of their environment, evolving the parameters of their decision making process towards an optimal strategy, rather than having fixed, optimal parameters as in this work.

Despite the simplicity of our current approach, however, we believe that concrete lessons can be drawn for the physics of living or life-like systems making simple decisions. In our physical model, successively more complex, and potentially costly, information-processing architectures perform better in successively more complex environments. We would expect that the information-processing carried out by living organisms reflects a similar trade-off: more complex decision-making strategies are more worthwhile in environments that exhibit statistical structure over time scales that are long compared to the decision-making time, and large fluctuations that must not be misinterpreted. We also expect that true evolved strategies will not optimise exploitation of the environment on average in isolation; strategies should also be designed to hedge against the risk of negative short-term fluctuations, to a degree that depends upon the cost of storing resources that compensate for these fluctuations.

4

Reversible chemical Turing machines

4.1 Abstract

We show how to construct a reversible Turing machine in the style of Bennett using chemical reactions. Like Bennett's, our design uses a memory tape to make the transitions logically reversible. Logical reversibility is not, in general, necessary for thermodynamic reversibility but for a logically irreversible transition it is necessary to know the initial distribution over states, which in a computation is not known. Our explicit construction gives an upper bound on the number of components required to implement a Turing machine. There are additional considerations that are not taken into account in Bennett's original work. To be able to control a system reversibly the same protocol must be applied to many similar systems simultaneously so that the control costs per machine are negligible. If the controller controls many machines simultaneously then one also has to consider how the controller knows when the machines have halted.

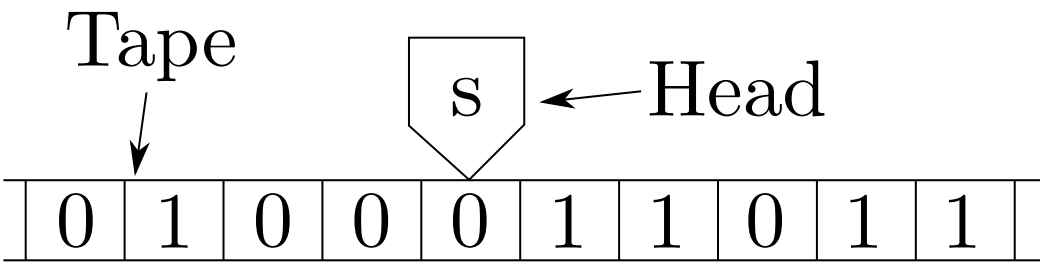


Figure 4.1: A schematic of the Turing machine model of computation. The tape consists of a series of cells each with a symbol. In this example the symbols are either 0 or 1. The head has state s.

4.2 Introduction

A Turing machine is an idealised model of a computer invented by Alan Turing in 1936[35] in order to investigate questions about the nature of computation, such as the halting problem. The machine consists of a 'tape' containing the working memory of the machine in a sequence of cells each containing a symbol from a finite alphabet. The machine has a 'head' also with a state from a finite set that reads the symbol at a position on the tape and then, based on a set of transition rules for that particular machine, writes a new symbol on the tape at that position then moves either left or right to an adjacent position on the tape and changes the head state. There is no unique definition of a Turing machine. For example, some authors allow for the head to stay at the same position during a step of the computation as well as moving left or right. Machines with sufficient numbers of states and the appropriate set of transition rules can be 'Turing complete', which means that it can, in principle, do any computation that any other computer can do. Although, in practice, encoding a real computation in a Turing machine is complicated and it will be slow to run.

If we define A as the initial state of the head and T as the initial symbol at the position on the tape where the head is then we can write a transition rule as

$$AT \to T'\sigma A', \tag{4.1}$$

where T' is the symbol written to the tape, σ is the direction the head moves and A' is the final state of the head.

As Turing machines are a simple model capable of complex computation they are a useful model for investigating questions about the thermodynamics of computation. It has been known since Szilard that computational operations can have thermodynamic costs so there is interest in whether it is possible to do useful computation without thermodynamic cost.

Bennett proposed a scheme to make a thermodynamically reversible Turing machine[74]. In his view, the reason the standard Turing machine could not be thermodynamically reversible was because the steps were not logically reversible. The steps are not logically reversible because the transitions of the state of the machine do not depend on the direction the head moved in the previous transition. Also, in the set of transitions there can be multiple initial symbols and states that give the same final symbol and state and direction to step. To make the transitions logically reversible, Bennett added a second tape to the machine; this stored the whole history of the transition of the machine in the computation. He proposed copying the state of the main tape to a third tape after the computation so that then the computation can be reversed for no cost. This work on reversible Turing machines has been very influential both in computer science and physical approaches to computation[64,75].

There has been prevous work to constuct Turing machines or equivalent devices using macromolecules as the building blocks. One class of such devices uses macroscopic quantites of the molecules to make a single computer. The steps of the computation are carried out by the experimentor acting with macroscopic experimental techiniques on the solution containing the computer molecules. These started with Adleman[76] who invented a way to solve Hamiltonian path problems using DNA and experimentally implemented it. Beaver[77] and Rothemund[78] both designed methods for implementing Turing machines using DNA. Given the substantial intervention by an experimentalist these systems had an unquantified energetic cost.

In this chapter we use a more concrete chemical set-up to show what would be required to practically implement a reversible Turing machine similar to Bennett's original design. Due to recent advances in the theory of the thermodynamics of computation we give a different justification for using

logically reversible steps in the computation.

The reversibility of our design relies on an 'open loop' operator applying the same controls to many machines simultaneously in a manner entirely uncoupled to the details of each individual computation. This means there are additional considerations, such as how to know when the machine has halted, that are not present in the more abstract design.

In Bennett's scheme, he first modifies the transition rules of the Turing machine so that they either read and write to the tapes while possibly changing the head state or they make a step along the tapes. In our scheme we separate each computational step into a stage where the head steps along the tape and a stage where the head modifies the state of the symbol at that tape position and its own state. Bennett's approach meant he had to introduce a separate reverse of each transition whereas we avoid this, because if the control is reversed all transitions of our machine are reversed. We also reduce the number of memory states required to only those absolutely necessary to determine the history of the tape.

In this chapter we first specify the constraints under which we design our model in section 4.3 and the components of our Turing machine. Then in section 4.4.1 we give an overview of the model of a chemical Turing machine. In sections 4.4.2 and 4.4.3 we explain in detail how the head steps from one position on the tape to the next and how the transitions in the state of the head and the symbol on the tape are changed in one step of the calculation. In section 4.4.4 we discuss some issues related to the halting of the chemical Turing machine. Finally in section 4.5 we discuss the implications of and lessons we can draw from this concrete model.

4.3 System specifications

Before we introduce our scheme for implementing a reversible chemical Turing machine we must discuss what we are assuming is possible and what the constraints are.

The tape and head will be made from macromolecules: the tape could be a

strand of DNA and the head could be DNA origami. The different symbols are represented by molecules attached to the tape at regular intervals that can be in multiple quasistable states. The state of the head is also represented by molecules with multiple quasistable states. We are assuming that we can implement any arbitrary chemical reaction network (CRN) which has no leak reactions and we are not constrained by number of reactants and products.

We are assuming that is is possible to implement a truly quasistatic change of concentrations of molecules. This means that each step of the computation takes, in principle, infinitely long. Of course in practice it is possible to approximate a quasistatic protocol with a finite time protocol where the changes are sufficiently slow compared to the equilibration rate of the system of interest.

As is implicitly the case for other implementations of reversible Turing machines we are not considering the cost of setting up the initial state of the machines. We assume that at the start of a computation the head is in the correct starting state and is at the correct point on the tape. The tape has been set to the input for the program that is being run. We are, also, not considering the cost of reading the output. We consider 'the computation' to be what happens from when the tape is set to the input state and the head is in the initial state attached to the initial position to when the head has reached a halting state. We will not actually specify how to do the computation backwards we only note that it is in principle possible to do so.

4.4 Model

4.4.1 Overview

It is important to our design that many similar machines can be driven in parallel by a single external operator. Figure 4.2 shows an overview of the set-up. There is a reaction volume that contains the machines. The operator drives each step of the computation by applying a protocol of fuel concentrations to the reaction volume. It does this by connecting the reaction volume to a succession of fuel buffers at different concentrations.

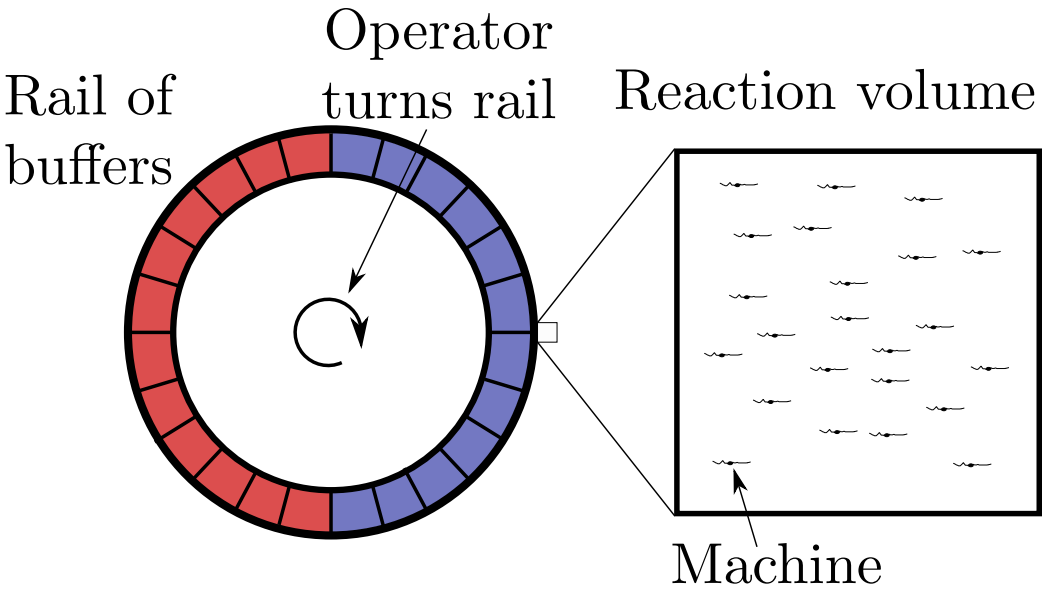


Figure 4.2: The reaction volume contains many chemical Turing machines. There is an external operator that sets the initial state of the machines and reads the outputs. The machines can all be running calculations for distinct problems (non-identical programs): there is no requirement that each machine has the same inputs. The operator drives the reactions by applying a protocol of fuel concentrations to the reaction volume. It does this by connecting the reaction volume to a succession of fuel buffers at different concentrations.The rail of buffers is shown as circular because the computation consists of many steps that are each driven by one turn of the rail. The two different colours of the rail are to represent the stepping stage and the head update stage of each computational step: one rotation is one computational step.

The operator must be able to drive the computations by applying the same protocol to all machines simultaneously. As an analogy, consider the reversible isothermal compression of a gas in a piston. If the piston is moved slowly enough then the reduction in entropy of the gas is exactly matched by an increase in entropy of the heat bath. What is not usually considered in this analysis is the controller who starts compressing at one predefined time and ends at another and who biases the piston to move in one direction rather than another (entropy production is required to break time symmetry)[25]. It is reasonable to discount these costs if the amount of gas is large because these costs will be small compared to the size of the heat flow from the gas to the heat reservoir. However, if the gas contained a single molecule, as in a Szilard engine, then the costs of the controller would be of comparable magnitude to the energies involved in the gas (on the order of $k_{\mathrm{B}}T$). For a similar reason, if a controller controlled an individual chemical Turing machine then the edge costs due to the specifics of how the controller worked would swamp the costs intrinsic to the chemical Turing machine. However, if one controller can control an arbitrary number of chemical Turing machines (operating on distinct inputs) then these edge costs can be discounted.

In order to reverse the computation the buffers are simply connected to the reaction volume in the opposite order. i.e. the rail is spun in the opposite direction. As noted above, we are not concerned with how the machines are initially set up or how the outcomes of the calculations are copied or read out. What matters is that the calculation could in principle be reversed.

4.4.2 Stepping

The head needs to be able to step to the next symbol on the tape in either direction and the direction must be determined by the state of the head and not by an external driver. Therefore, the head must have a 'direction molecule' that can either be in the 'left', L, or the 'right', R, state. This direction molecule should also have an H state that represents the halting state of the machine and does not catalyse stepping on the main tape. Alternatively, instead of a separate directional molecule, the molecular motor

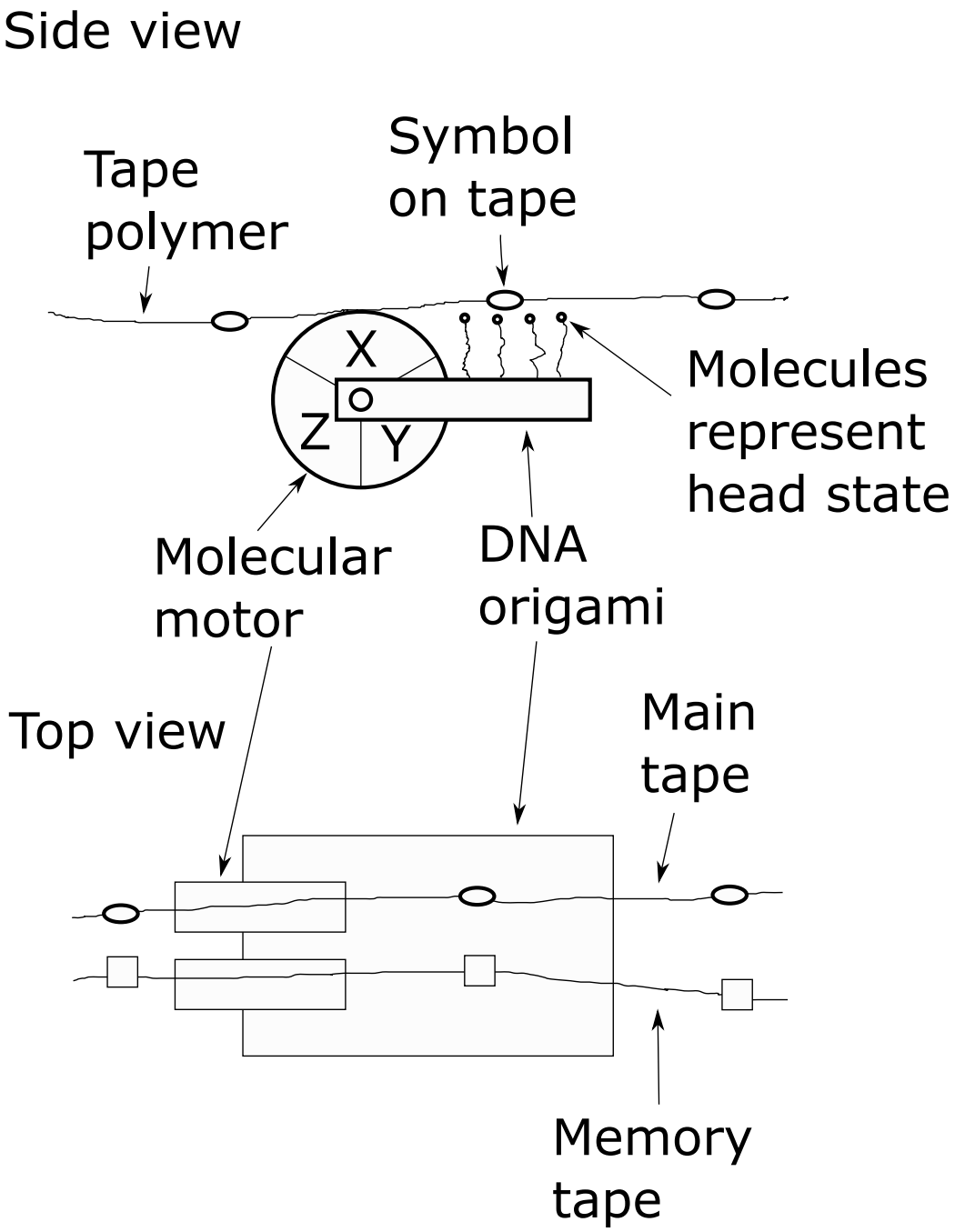


Figure 4.3: The overview of a single chemical Turing machine. The tape consists of a polymer and the symbols on the tape are evenly spaced molecules attached to the polymer. The head is made of some macromolecule such as a DNA origami structure. The state of the head is encoded in the state of molecules attached to the DNA origami structure. The head is attached to the tape by the molecular motor that moves the head from one symbol to the next.

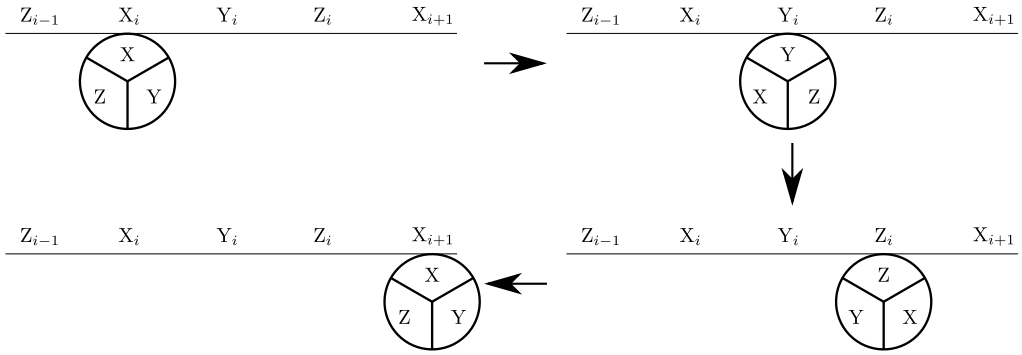


Figure 4.4: The stepping of the molecular motor.

could have left and right states or the number of states of the head state molecule could be increased so that it also represents left and right.

We assume that the head changing position is an elementary transition that can happen in a chemical reaction: in figure 4.3 we assume that the molecular motor can rotate a third of a step while simultaneously switching which segment of the motor is bound to the polymer in a single step without the polymer and motor becoming unbound at any point. This is a strong simplifying assumption so a version of stepping without this assumption is given in appendix A.10.

The tape has a repeating structure of sites where the molecular motor can bind. There are three different types of site: X, Y and Z that can, respectively, bind to the X, Y and Z regions of the molecular motor. This is shown in figure 4.4. The X sites are the positions on the tape where the symbol molecule on the tape lines up with the tethered head molecules (these are the only positions that correspond to positions in the abstract view of the machine) and the Y and Z are intermediate positions to which the head can attach.

To move right the reactions

$$\begin{aligned} \mathrm{F}_1^* + \mathrm{R} + \mathrm{X}_i &\rightleftharpoons \mathrm{R} + \mathrm{Y}_i + \mathrm{F}_1, \\ \mathrm{F}_2^* + \mathrm{R} + \mathrm{Y}_i &\rightleftharpoons \mathrm{R} + \mathrm{Z}_i + \mathrm{F}_2, \\ \mathrm{F}_3^* + \mathrm{R} + \mathrm{Z}_i &\rightleftharpoons \mathrm{R} + \mathrm{X}_{i+1} + \mathrm{F}_3, \end{aligned} \tag{4.2}$$

are used and to move left the reactions

$$\begin{aligned} \mathrm{F}_1^* + \mathrm{L} + \mathrm{X}_i &\rightleftharpoons \mathrm{L} + \mathrm{Z}_{i-1} + \mathrm{F}_1, \\ \mathrm{F}_2^* + \mathrm{L} + \mathrm{Z}_{i-1} &\rightleftharpoons \mathrm{L} + \mathrm{Y}_{i-1} + \mathrm{F}_2, \\ \mathrm{F}_3^* + \mathrm{L} + \mathrm{Y}_{i-1} &\rightleftharpoons \mathrm{L} + \mathrm{X}_{i-1} + \mathrm{F}_3, \end{aligned} \tag{4.3}$$

are used. If the head contains the direction molecule in the R state then the head can be quasistatically moved from X_i to Y_i to Z_i to X_{i+1} and if the direction molecule is in the L state then the head can be quasistatically moved from X_i to Z_{i-1} to Y_i to X_{i-1}. This is done using the protocol for three separate bit flips starting from a definite state.

For the left motion we could, alternatively, use three different pairs of fuel molecules, F_4, F_5 and F_6, from the right motion. This could make the mechanism easier to understand because the mechanisms for the two directions are independent. We adopt the approach in equations 4.3 and 4.2 as reusing the same species reduces the total number of fuel molecule required.

The protocol of concentrations for the fuel molecules is shown in figure 4.5. The work extracted from the buffers is the change in free energy of the head when it is attached to the different positions. We assume that the free energy is the same when the head is attached to any of the X_i. The free energy when attached to either of the intermediate states could be different but that is unimportant since the work done to move from X_i to X_{i+1} or X_{i-1} is zero.

4.4.3 Update head

In the second stage of a single computational step, the state of the head and the symbol at the head position on the tape are modified based on the initial state and symbol.

In Bennett's set-up he emphasises that the head must first write the new symbol, then move to the next position and then change its state. Our machine works differently in that the rules show how to update the directional state of the head and then there is a separate step where the head actually

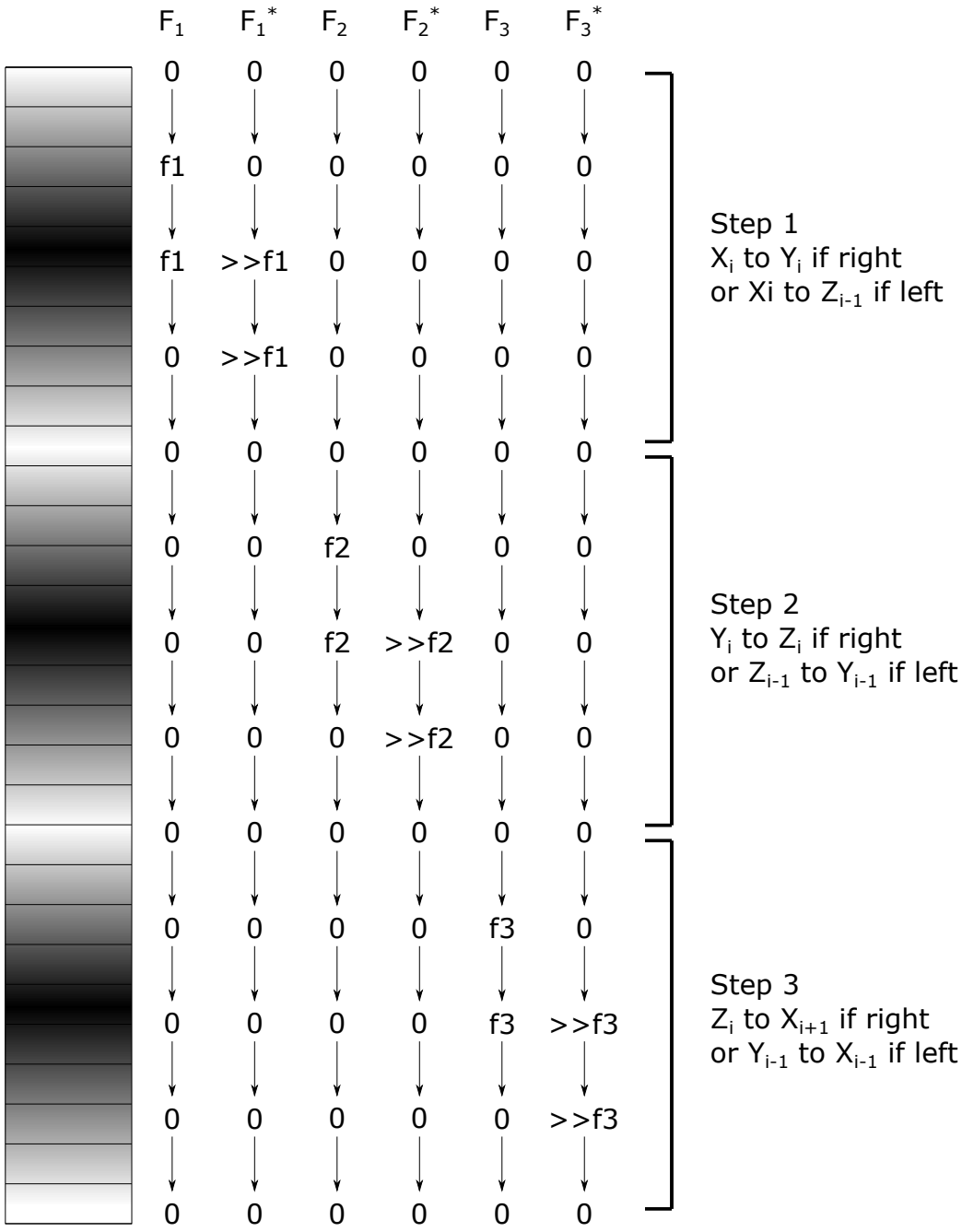


Figure 4.5: The protocol of concentrations of the fuel molecules to implement one step, which is left or right depending on the state of the direction molecule. While apparently complicated in fact the same approach is repeated three times. f1, f2 and f3 need only be a large concentration and in fact the same label could have been used in all cases.

moves. This means that we can have a single transition that simultaneously changes the head state, head direction and tape symbol.

The application of all of the rules to the combined state of the head and the molecule on the main tape at the head position gives a deterministic map. We are able to analyse this map from the perspective discussed in section 1.7[34,69]. Because the Turing machine rules give the same transition if the directional molecule was initially in the L or R state, (i.e. in each step of the computation it does not matter in which direction the head moved in the previous step) the map is non-invertible. It can also be the case that multiple initial symbols and states are mapped to the same symbol, state and next direction.

A single-valued non-invertible map can be reversibly embedded using no hidden states[65]. However, the specific protocol to do so depends on the initial probability distribution over the states. The problem is that this distribution is not known for intermediate steps of a computation. For an invertible map the reversible protocol is the same no matter what the input distribution is. Therefore, to make a reversible Turing machine one must, as in Bennett's design[74], make the map invertible by adding another 'memory' tape that records the previous state of each transition.

An invertible map requires one hidden state to implement. However, a hidden state is simply a state with zero initial probability, so if the system in question already has a state with zero initial probability, no extra hidden state is required. If the memory molecule is initially in a definite state then there are many states of the composite system of the two tape molecules and the head molecules that have zero initial probability.

The basic way to implement any set of transition rules is for all of the molecules on the memory tape to be initially in the state M_0 and for each rule to use the reactions

$$\begin{aligned} A + L + T + M_0 + G^* &\rightleftharpoons A' + \sigma + T' + M_{2i-1} + G \\ A + R + T + M_0 + G^* &\rightleftharpoons A' + \sigma + T' + M_{2i} + G \end{aligned} \tag{4.4}$$

where i numbers each transition rule, A and A' represent any state of the head state molecule, T and T' are states of the symbol molecule and σ is either L or R. The protocol

$$\begin{array}{lccccccccc} [\mathrm{G}]: & 0 & \to & g & \to & g & \to & 0 & \to & 0 \\ [\mathrm{G}^*]: & 0 & \to & 0 & \to & \gg g & \to & \gg g & \to & 0. \end{array} \tag{4.5}$$

is used. Appendix A.11 gives the reactions used to implement some examples of Turing machines.

4.4.4 Halting

In Bennett's scheme the controller knows when the Turing machine has reached its halting state and then the controller can copy the output and reverse the computation. In our scheme we need to analyse how the external controller detects that a machine has reached the halting state. Also, there are additional considerations due to the fact that the operator controls multiple Turing machines simultaneously.

If one machine has halted but others have not and the operator continues to drive the computation forward then the halted machine should not act irreversibly and it should not alter the output on its main tape. This is taken care of if the halting state of the directional molecule H does not catalyse stepping on the main tape but the head continues to step on the memory tape while not changing the memory molecules from their initial state of M_0.

One option for the overall protocol to take account of the fact that the different machines can halt at different times is for the operator to simply run the computation for a fixed number of steps. When the answer is read/copied whether each machine was in its halt state can also be copied. This has the benefit of being simple. In practice there will always be some limit to the maximum number of steps that can be taken, similar to the constraint that in reality an infinite tape is impossible. The downside is that some of the machines may not have halted. Alternately, if the number of steps is set much larger that the maximum required by any of the machines then a lot

of time is wasted.

Alternately, the operator could run the machine for a fixed number of steps and then check if all machines have stopped. If all machines have stopped the operator can move to reading the output. Otherwise, the operator does another set of a fixed number of steps. To do this the operator could insert a molecule, All-halted, into the reaction volume on a probe and use the reactions

$$\begin{aligned} \text{All-halted} + \text{L} + \text{F}^* &\rightleftharpoons \text{Not-all-halted} + \text{L} + \text{F} \\ \text{All-halted} + \text{R} + \text{F}^* &\rightleftharpoons \text{Not-all-halted} + \text{R} + \text{F} \end{aligned} \tag{4.6}$$

and apply the protocol

$$\begin{array}{lccccccccc} [\text{F}]: & 0 & \rightarrow & f & \rightarrow & f & \rightarrow & 0 & \rightarrow & 0 \\ [\text{F}^*]: & 0 & \rightarrow & 0 & \rightarrow & \gg f & \rightarrow & \gg f & \rightarrow & 0. \end{array} \tag{4.7}$$

We assume that the All-halted molecule can diffuse around inside the reaction volume so that it can interact with the L or R molecule of all of the machines. If any machines are not halted then the molecule will be in the Not-all-halted state so the operator should decide to continue the computation for another fixed number of steps.

The costs to doing these measurements are a fixed overhead whatever the number of machines. However, the All-halted/Not-all-halted molecule must interact with each machine in order to take the measurement. Of course, with a quasistatic protocol we are assuming that the protocol takes an infinite time, but in practice the rate of equilibration will be relevant. For the reactions to make this measurement the rate of equilibration will be slower the larger the number of machines so the protocol will take longer if there are more machines. Also, this will only work if we know beforehand that all machines will halt.

4.5 Discussion

Creating a more concrete model of a reversible Turing machine has meant that we have had to consider some aspects of the machine that are not relevant to a more abstract set-up.

In order for a thermodynamic system to undergo a reversible process (that is not simply remaining in a static equilibrium) it must be externally controlled. An isolated system simply evolves to equilibrium and when in equilibrium it makes a random walk through its microstates. To be able to apply a protocol a controller must force a parameter to change in a biased direction. This requires some entropy production. In the classical example of an isothermal compression of a gas in a cylinder the number of gas molecules is macroscopic so the heat transfers involved are of a much greater magnitude than the necessary entropy production of operating the piston (which are on the order of $k_{\mathrm{B}}T$, so they can be safely ignored). Conversely, in the case of microscopic systems, the entropy production of the operator will be large compared to the entropy flow.

Therefore, the operator costs of microscopic systems can be made negligible if the operator simultaneously controls many similar copies of the system. Of course, it is not the case that the cost of many different processes is smaller just because they are done by the same person; the systems must be controlled through the same parameter.

If all of the chemical Turing machines being operated at once were doing the same computation then operating on multiple machines would not be of any benefit because even though the cost per machine would be negligible, the cost per computation would not be. Fortunately, it is possible for all of the machines to be doing different computations but controlled by the same protocol of chemical fuels.

The operator could set up each main tape with a different initial state. If the Turing machine is universal then these different inputs can encode any computation. It is not required that all of the chemical Turing machines in the reaction volume be the same in the sense of having the same states,

symbols and transitions. Each step of the head transitions works on the same change in the concentrations of only one pair of fuel molecules.

The fact that many machines must be simultaneously controlled means there are some additional considerations that are not in Bennett's original paper.

In more abstract theoretical descriptions of Turing machines the machine 'knows' when it is in the halting state and then stops. In fact, the head does not need an explicit halting state (although there can be acceptor states) it simply has some initial states that cause no next step. This is natural if one considers e.g. the head to be controlled by a microcontroller or the Turing machine to be simulated by a person with pen and paper who can only follow the simple algorithm. Bennett follows this conception and assumes that the machine knows when to stop. In our set-up Bennett's approach is not possible due to the multiple machines. Therefore, we need the machine to act correctly, i.e. reversibly and without altering the main tape, if it continues to be driven past the point where it has halted. It is not possible to reversibly and autonomous decouple the machine from the driving. Consider that, if that were possible, how would the machine know, when the driving is reversed, to recouple? Therefore, we have to add explicit halt states to the machine and we have to consider either explicitly measuring the machines to see if they have halted or simply running for a long time. When the answer is being read whether the machine has halted can also be read; we are not concerned with the cost of this step. Measuring if the machines have halted necessarily involves sampling all machines.

We have seen that, as Bennett originaly said, to implement a reversible Turing machine the steps must be made logically reversible. He suggests that Laundauer's principle means logically reversible steps require heat dissipation and heat dissipation means irreversibility. If the computational system is connected to a heat bath then there is flow of heat from the system to the bath if the entropy of the system is reduced (in chemical systems there is no need for heat flow) but this does not imply thermodynamic reversibility. Consider again the isothermal compression of an ideal gas. There has been some confusion on this topic that erroneously conflates logical and thermodynamics reversibility[22]. It is not the case that one implies the other. However, to

implement a logically reversible map reversibly the protocol of the potential changes depends on the initial distribution; otherwise, there is a 'mismatch cost'[36]. If one knows the distribution over the inputs to a computation then it is possible to do any arbitrary computation reversibly. Although, if this computation is reversed only the correct distribution over the inputs will be output rather than the specific state. However, for the case of a Turing machine that only acts locally on part of the computation at any one time, even if the initial state of the tape was known it would not be possible to be thermodynamically reversible without being logicically reversible. There is a modularity cost[37]. Also, even if there was no modularity cost each step of the computation would require a different protocol that depends on the output of the computation so the machine would already have to know the output of the computation to set the correct protocol.

4.6 Conclusions

We have demonstrated it is possible to implement a version of Bennett's reversible Turing machine using chemical reactions. Stepping the head in either direction on the main tape and the transitions that change the head's state and write to the tapes are autonomous in the sense that the only external control that is applied is the same for each step of the computation so there is not feedback from the state of the machine.

Our design is somewhat simplified compared to Bennett's model. In the reverse stage the direction of the whole control is reversed so additional head states and transitions are not required for the reverse. The trade-off is perhaps the separate stepping section of the protocol for one step of the computation. Also, we have shown that usually fewer states of the molecules on the memory tape are required compared to Bennett's. There is a trade-off between the number of distinguishable states of a single molecule required in the scheme and the number of different molecules, each with fewer states. Using more molecules leads to the reactions involving more molecules simultaneously.

Our concrete set-up forces us to consider how the machine 'knows' that it has

halted and must proceed to the copying and reverse stages of the reversible computation. Our set-up allows the machine to be driven forwards reversibly when it is in its halting state without changing the state of the main tape. The time to check that all machines have halted scales with the number of machines. If one of the machines never halts then the cost of computation will be correspondingly divergent. Our set-up requires an explicit halt state, which is not always required in abstract definitions of Turing machines. Some Turing machines do not have an explicit halt state but instead to have no transition for some states of the machines head and tape symbol.

Finally, this set-up gives us a framework that could be used to investigate questions such as the trade-off between speed and entropy production if the protocols of changing fuel concentrations are not quasistatic.

5

Conclusion

This thesis has examined questions related to the thermodynamic limits to sensing, information processing and computation. Part of the motivation for these questions is the relevance to living organisms for which sensing is vital. In particular, the thermodynamics of information is relevant to single-celled organisms because of their small scale. We make use of recent developments in stochastic thermodynamics that have allowed for a more precise theory of the relationship between information and thermodynamics.

An important step in finding the limits to sensing is to be able to quantify sensing so in chapter 2 we examined a measure that has been used to quantify sensor quality to find if it is generally applicable. We compared this quantity (the learning rate) to the mutual information between the signal and the sensor in some example models and found that there were cases where varying the parameters of the model to maximise the learning rate did not give an apparently optimal sensor. In steady state the learning rate is not a general measure of the quality of a sensor. It instead measures how much the sensor must change to keep up with the signal.

One problem with previous work on the thermodynamics of information is that abstracted models can lead to neglect of relevant physical effects. To

remedy this problem, we introduced a chemical version of the classic Szilard engine in chapter 3. This enabled us to see concretely how work must be done on the engine to produce correlations. It is this work input that produces the high free energy state that is exploited in the decorrelation step to do work. This demonstrates the manner in which the second law of thermodynamics is not violated.

Also, in chapter 3 we used the biochemical Szilard engine as a basis for a very simplified model of a micro-organism feeding using temporal correlations in an environment with no overall chemical potential. It is useful to study feeding because it is simple to compare the energetic costs of sensing with the energy gained from the food. We found that in environments with long-ranged non-Markovian correlations it is beneficial for an organism to take complicated measurements. Of course, the specific model is unrealistic and could not be directly applied to any real organism but the results suggest a general principle as, less formally, suggested by Seoane and Solé[73], that more complicated environments require more complicated inference strategies.

The information thermodynamics of small scale systems is intimately related to the thermodynamic limits of computation. Rather than attempting to analyse the thermodynamics of real computers we can examine idealised models of computation of which there are several that are equivalent. Therefore, in chapter 4 we proposed a design for a reversible chemical Turing machine. This machine can, in principle, do computations with no entropy production. We found, like Bennett, that it is required to make the machine logically reversible and that to do so we must add an additional tape that records each step of the computation but, due to recent results in implementing stochastic maps reversibly, we have different reasoning to Bennett. It is important to note that to actually implement a reversible computation many computations must be done simultaneously so that edge costs associated with driving are zero. We found that there is a close link between the halting problem and the practical cost of reversible computation.

In future work it would be useful to examine more realistic models of cell sensing and inference for the purpose of feeding. There are several way in which the model could be made more realistic: non-quasistatic protocols

could be examined, the food molecules could be allowed to freely diffuse rather than being bound to a tape and the food and waste molecules could be made asymmetric. This would move us closer to finding the real energetic limits for cell sensing. The model of a chemical Turing machine proposed in chapter 4 could also be modified to examine possible trade-offs between speed of computation and entropy production.

References

[1] Jaynes, E. T., Information theory and statistical mechanics. *Phys. Rev.*, 106(4):620, 1957.

[2] Govern, C. C. and ten Wolde, P. R., How biochemical resources determine fundamental limits in cellular sensing. *arXiv preprint arXiv:1308.1449*, 2013.

[3] Berg, H. C. and Purcell, E. M., Physics of chemoreception. *Biophys. J.*, 20(2):193–219, 1977.

[4] ten Wolde, P. R., Becker, N. B., Ouldridge, T. E., and Mugler, A., Fundamental limits to cellular sensing. *J. Stat. Phys.*, 162(5):1395–1424, 2016.

[5] Bialek, W. and Setayeshgar, S., Physical limits to biochemical signaling. *Proc. Natl. Acad. Sci. U.S.A.*, 102(29):10040–10045, 2005.

[6] Shannon, C. E., A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948.

[7] Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. Wiley-Interscience, 2006.

[8] Barato, A. C., Hartich, D., and Seifert, U., Efficiency of cellular information processing. *New J. Phys.*, 16(10):103024, 2014.

[9] Hartich, D., Barato, A. C., and Seifert, U., Sensory capacity: An information theoretical measure of the performance of a sensor. *Phys. Rev. E*, 93:022116, 2016.

[10] Horowitz, J. M. and Esposito, M., Thermodynamics with continuous information flow. *Phys. Rev. X*, 4:031015, 2014.

[11] Schreiber, T., Measuring information transfer. *Phys. Rev. Lett.*, 85(2):461, 2000.

[12] Esposito, M. and Van den Broeck, C., Second law and landauer principle far from equilibrium. *EPL*, 95(4):40004, 2011.

[13] Ouldridge, T. E., The importance of thermodynamics for molecular systems, and the importance of molecular systems for thermodynamics. *Nat. Comput.*, 17(1):3–29, 2018.

[14] Rao, R. and Esposito, M., Nonequilibrium thermodynamics of chemical reaction networks: Wisdom from stochastic thermodynamics. *Phys. Rev. X*, 6(4):041064, 2016.

[15] Ouldridge, T. E., Govern, C. C., and ten Wolde, P. R., Thermodynamics of computational copying in biochemical systems. *Phys. Rev. X*, 7(2):021004, 2017.

[16] Soloveichik, D., Seelig, G., and Winfree, E., DNA as a universal substrate for chemical kinetics. *Proc. Natl. Acad. Sci. U.S.A.*, 107(12):5393–5398, 2010.

[17] Cardelli, L., Two-domain DNA strand displacement. *Math. Structures Comput. Sci.*, 23(2):247–271, 2013.

[18] Chen, Y.-J., Dalchau, N., Srinivas, N., Phillips, A., Cardelli, L., Soloveichik, D., and Seelig, G., Programmable chemical controllers made from DNA. *Nat. Nanotechnol.*, 8(10):755, 2013.

[19] SantaLucia Jr., J. and Hicks, D., The thermodynamics of DNA structural motifs. *Annu. Rev. Biophys. Biomol. Struct.*, 33:415–440, 2004.

[20] Maxwell, J. C. *Theory of Heat*. Longmans, Green, 1871.

[21] Szilard, L., Über die entropieverminderung in einem thermodynamischen system bei eingriffen intelligenter wesen (On the decrease of entropy in a thermodynamic system by the intervention of intelligent beings). *Z. Phys.*, 53(11-12):840–856, 1929.

[22] Ouldridge, T. E., Brittain, R. A., and ten Wolde, P. R. The power of being explicit: demystifying work, heat, and free energy in the physics of computation. In Wolpert, D. H., Kempes, C. P., Stadler, P., and Grochow, J., editors, *The Interplay of Thermodynamics and Computation in Natural and Artificial Systems*. SFI Press, 2018.

[23] Parrondo, J. M., Horowitz, J. M., and Sagawa, T., Thermodynamics of information. *Nat. Phys.*, 11(2):131, 2015.

[24] Van den Broeck, C. and Esposito, M., Ensemble and trajectory thermodynamics: A brief introduction. *Physica A*, 418:6–16, 2015.

[25] Crooks, G. E., Entropy production fluctuation theorem and the nonequilibrium work relation for free energy differences. *Phys. Rev. E*, 60:2721–2726, 1999.

[26] Mandal, D. and Jarzynski, C., Work and information processing in a solvable model of maxwell's demon. *Proc. Natl. Acad. Sci. U.S.A.*, 109(29):11641–11645, 2012.

[27] Barato, A. C. and Seifert, U., An autonomous and reversible maxwell's demon. *EPL*, 101(6):60001, 2013.

[28] McGrath, T., Jones, N. S., ten Wolde, P. R., and Ouldridge, T. E., Biochemical machines for the interconversion of mutual information and work. *Phys. Rev. Lett.*, 118:028101, 2017.

[29] Boyd, A. B., Mandal, D., and Crutchfield, J. P., Identifying functional thermodynamics in autonomous maxwellian ratchets. *New J. Phys.*, 18(2):023049, 2016.

[30] Boyd, A. B., Mandal, D., and Crutchfield, J. P., Correlation-powered information engines and the thermodynamics of self-correction. *Phys. Rev. E*, 95(1):012152, 2017.

[31] Boyd, A. B., Mandal, D., and Crutchfield, J. P., Leveraging environmental correlations: The thermodynamics of requisite variety. *J. Stat. Phys.*, 167(6):1555–1585, 2017.

[32] Stopnitzky, E., Still, S., Ouldridge, T. E., and Altenberg, L. Physical limitations of work extraction from temporal correlations. In Wolpert, D. H., Kempes, C. P., Stadler, P., and Grochow, J., editors, *The Interplay of Thermodynamics and Computation in Natural and Artificial Systems*. SFI Press, 2018.

[33] Landauer, R., Irreversibility and heat generation in the computing process. *IBM J. Res. Dev.*, 5(3):183–191, 1961.

[34] Owen, J. A., Kolchinsky, A., and Wolpert, D. H., Number of hidden states needed to physically implement a given conditional distribution. *New J. Phys.*, 2018.

[35] Turing, A. M., On computable numbers, with an application to the entscheidungsproblem. *Proc. London Math. Soc.*, 2(1):230–265, 1937.

[36] Wolpert, D. H. and Kolchinsky, A., Exact, complete expressions for the thermodynamic costs of circuits. *arXiv preprint arXiv:1806.04103*, 2018.

[37] Boyd, A. B., Mandal, D., and Crutchfield, J. P., Thermodynamics of modularity: Structural costs beyond the landauer bound. *Phys. Rev. X*, 8(3):031036, 2018.

[38] Hartich, D., Barato, A. C., and Seifert, U., Stochastic thermodynamics of bipartite systems: Transfer entropy inequalities and a Maxwell's demon interpretation. *J. Stat. Mech.*, 2014(2):P02016, 2014.

[39] Yamamoto, S., Ito, S., Shiraishi, N., and Sagawa, T., Linear irreversible thermodynamics and Onsager reciprocity for information-driven engines. *Phys. Rev. E*, 94:052121, 2016.

[40] Allahverdyan, A. E., Janzing, D., and Mahler, G., Thermodynamic efficiency of information and heat flow. *J. Stat. Mech.*, 2009(09):P09011, 2009.

[41] Horowitz, J. M., Multipartite information flow for multiple Maxwell demons. *J. Stat. Mech.*, 2015(3):P03006, 2015.

[42] Goldt, S. and Seifert, U., Stochastic thermodynamics of learning. *Phys. Rev. Lett.*, 118:010601, Jan 2017.

[43] Das, S. G., Rao, M., and Iyengar, G., Universal lower bound on the free-energy cost of molecular measurements. *Phys. Rev. E*, 95(6):062410, 2017.

[44] Still, S., Sivak, D. A., Bell, A. J., and Crooks, G. E., Thermodynamics of prediction. *Phys. Rev. Lett.*, 109:120604, 2012.

[45] Barato, A. C., Hartich, D., and Seifert, U., Information-theoretic versus thermodynamic entropy production in autonomous sensory networks. *Phys. Rev. E*, 87(4):042104, 2013.

[46] Govern, C. C. and ten Wolde, P. R., Energy dissipation and noise correlations in biochemical sensing. *Phys. Rev. Lett.*, 113:258102, 2014.

[47] Becker, N. B., Mugler, A., and ten Wolde, P. R., Optimal prediction by cellular signaling networks. *Phys. Rev. Lett.*, 115:258103, 2015.

[48] Barato, A. C. and Seifert, U., Cost and precision of Brownian clocks. *Phys. Rev. X*, 6:041053, 2016.

[49] Becker, N. B., Mugler, A., and ten Wolde, P. R., Prediction and dissipation in biochemical sensing. *arXiv preprint arXiv:1312.5625*, 2013.

[50] Zhang, Y. and Barato, A. C., Critical behavior of entropy production and learning rate: Ising model with an oscillating field. *J. Stat. Mech.*, 2016(11):113207, 2016.

[51] Govern, C. C. and ten Wolde, P. R., Optimal resource allocation in cellular sensing systems. *Proc. Nat. Acad. Sci. USA*, 111:17486–17491, 2014.

[52] Alberty, R. A. and Goldberg, R. N., Standard thermodynamic formation properties for the adenosine 5'-triphosphate series. *Biochemistry*, 31(43):10610–10615, 1992.

[53] McGrath, T., Jones, N. S., ten Wolde, P. R., and Ouldridge, T. E., Biochemical machines for the interconversion of mutual information and work. *Phys. Rev. Lett.*, 118(2):028101, 2017.

[54] Govern, C. C. and ten Wolde, P. R., Optimal resource allocation in cellular sensing systems. *Proc. Natl. Acad. Sci. U.S.A.*, 111(49):17486–17491, 2014.

[55] Ouldridge, T. E., Govern, C. C., and ten Wolde, P. R., Thermodynamics of computational copying in biochemical systems. *Phys. Rev. X*, 7(2):021004, 2017.

[56] Ouldridge, T. E. and ten Wolde, P. R., Fundamental costs in the production and destruction of persistent polymer copies. *Phys. Rev. Lett.*, 118(15):158103, 2017.

[57] Micali, G. and Endres, R. G., Bacterial chemotaxis: information processing, thermodynamics, and behavior. *Curr. Opin. Microbiol.*, 30:8–15, 2016.

[58] Becker, N. B., Mugler, A., and ten Wolde, P. R., Optimal prediction by cellular signaling networks. *Phys. Rev. Lett.*, 115(25):258103, 2015.

[59] Parkinson, J. S., Hazelbauer, G. L., and Falke, J. J., Signaling and sensory adaptation in escherichia coli chemoreceptors: 2015 update. *Trends Microbiol.*, 23(5):257–266, 2015.

[60] Mitchell, A., Romano, G. H., Groisman, B., Yona, A., Dekel, E., Kupiec, M., Dahan, O., and Pilpel, Y., Adaptive prediction of environmental changes by microorganisms. *Nature*, 460(7252):220, 2009.

[61] Joesaar, A., Yang, S., Bögels, B., van der Linden, A., Pieters, P., Kumar, B. V. V. S. P., Dalchau, N., Phillips, A., Mann, S., and de Greef, T. F. A., DNA-based communication in populations of synthetic protocells. *Nat. Nanotechnol.*, 14:369–378, 2019.

[62] Schmiedl, T. and Seifert, U., Stochastic thermodynamics of chemical reaction networks. *J. Chem. Phys.*, 126(4):044101, 2007.

[63] Stock, A. M., Robinson, V. L., and Goudreau, P. N., Two-component signal transduction. *Annu. Rev. Biochem.*, 69(1):183–215, 2000.

[64] Bennett, C. H., The thermodynamics of computation——a review. *Int. J. Theor. Phys.*, 21(12):905–940, 1982.

[65] Owen, J. A., Kolchinsky, A., and Wolpert, D. H., Number of hidden states needed to physically implement a given conditional distribution. *arXiv preprint arXiv:1709.00765*, 2017.

[66] Kolchinsky, A. and Wolpert, D. H., Dependence of dissipation on the initial distribution over states. *J. Stat. Mech. Theory Exp.*, 2017:083202, 2017.

[67] Stratonovich, R. L., Conditional Markov processes. *Theory Probab. Its Appl.*, 5(2):156–178, 1960.

[68] Shalizi, C. R. and Crutchfield, J. P., Computational mechanics: Pattern and prediction, structure and simplicity. *J. Stat. Phys.*, 104(3-4):817–879, 2001.

[69] Wolpert, D. H., Kolchinsky, A., and Owen, J. A., The minimal hidden computer needed to implement a visible computation. *arXiv preprint arXiv:1708.08494*, 2017.

[70] Kalman, R. E., A new approach to linear filtering and prediction problems. *J. Basic. Eng.*, 82(1):35–45, 1960.

[71] Malaguti, G. and ten Wolde, P. R., Theory for the optimal detection of time-varying signals in cellular sensing systems. *arXiv preprint arXiv:1902.09332*, 2019.

[72] Chapman, A. and Miyake, A., How an autonomous quantum maxwell demon can harness correlated information. *Phys. Rev. E*, 92(6):062125, 2015.

[73] Seoane, L. F. and Solé, R. V., Information theory, predictability and the emergence of complex life. *Royal Soc. Open Sci.*, 5(2):172221, 2018.

[74] Bennett, C. H., Logical reversibility of computation. *IBM J. Res. Dev.*, 17(6):525–532, 1973.

[75] Toffoli, T. and Margolus, N. H., Invertible cellular automata: a review. *Physica D*, 45(1-3):229–253, 1990.

[76] Adleman, L. M., Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021–1024, 1994.

[77] Beaver, D. A universal molecular computer. In Landweber, L. F. and Baum, E. B., editors, *DNA Based Computers*, pages 29–36. American Mathematical Society, 1995.

[78] Rothemund, P. W. K. A DNA and restriction enzyme implementation of turing machines. In Landweber, L. F. and Baum, E. B., editors, *DNA Based Computers*, pages 75–119. American Mathematical Society, 1995.

[79] Smith, A., Universality of wolfram's 2, 3 turing machine. *Complex Syst.*, 2007.

[80] Woods, D. and Neary, T., The complexity of small universal turing machines: A survey. *Theor. Comput. Sci.*, 410(4-5):443–450, 2009.

[81] Rogozhin, Y., Small universal turing machines. *Theor. Comput. Sci.*, 168(2):215–240, 1996.

A

Supplementary material

A.1 Proof that l_Y tends to zero as $N \to \infty$ for the system in section 2.4.3

Using Equations 2.5 and 2.8, in the steady state

$$l_Y = -\dot{I}^X = -\sum_{xx'y} w^{xx'} p(x,y) \log \frac{p(y|x')}{p(y|x)} = \sum_{xx'y} w^{xx'} p(x,y) \big(\log p(y|x) - \log p(y|x') \big).$$

Relabelling dummy variables,

$$l_Y = \sum_{xx'y} \Big(p(x,y) w^{xx'} - p(x',y) w^{x'x} \Big) \log p(y|x). \tag{A.1}$$

In this system, $p(x) = \frac{1}{2N}$ for all x by symmetry. There are only transitions between adjacent X states and so

$$l_Y(N) = \frac{1}{2N} \sum_{xy} \Big(p_N(y|x) w^{xx^+} - p_N(y|x^+) w^{x^+x} + p_N(y|x) w^{xx^-} - p_N(y|x^-) w^{x^-x} \Big) \log p_N(y|x) \tag{A.2}$$

where x^+ is the next X state in the cycle and x^- is the previous state, and we have introduced an explicit dependence of probabilities on N. In this system

$$\begin{aligned} w^{xx^+} &= w^{x^-x} = Nk_\uparrow, \\ w^{x^+x} &= w^{xx^-} = Nk_\downarrow, \end{aligned} \tag{A.3}$$

and therefore

$$l_Y(N) = \frac{1}{2}\sum_{xy}\Big(k_\uparrow\big(p_N(y|x)-p_N(y|x^-)\big)+k_\downarrow\big(p_N(y|x)-p_N(y|x^+)\big)\Big)\log p_N(y|x). \tag{A.4}$$

Defining $z = \frac{x}{2N}$ we have

$$l_Y = \frac{1}{2}\sum_{zy}\Big(k_\uparrow\big(p_N(y|z)-p_N(y|z-\tfrac{1}{2N})\big)+k_\downarrow\big(p_N(y|z)-p_N(y|z+\tfrac{1}{2N})\big)\Big)\log p_N(y|z). \tag{A.5}$$

We now assume that for large N, $p_N(y|z)$ tends smoothly towards $p_\infty(y|z)$, where $p_\infty(y|z)$ is a smooth function of z with $p_\infty(y|0) = p_\infty(y|1)$ due to the boundary conditions that connect $z = 1$ ($x = 2N$) to $z = 0$ ($x = 0$). For sufficiently large N,

$$p_N(y|z) = p_\infty(y|z) + \frac{1}{N}\left.\frac{\mathrm{d}p_N(y|z)}{\mathrm{d}1/N}\right|_{N\to\infty} + \mathcal{O}\left(\frac{1}{N^2}\right). \tag{A.6}$$

Thus

$$p_N(y|z \pm \tfrac{1}{2N}) = p_\infty(y|z) \pm \frac{1}{2N}\frac{\mathrm{d}p_\infty(y|z)}{\mathrm{d}z} + \frac{1}{N}\left.\frac{\mathrm{d}p_N(y|z)}{\mathrm{d}1/N}\right|_{N\to\infty} + \mathcal{O}\left(\frac{1}{N^2}\right)$$

and the sum can be approximated by an integral

$$\sum_z \tfrac{1}{2N} f(z) \to \int_0^1 \mathrm{d}z f(z) + \mathcal{O}\left(\frac{1}{N}\right). \tag{A.7}$$

So for large N,

$$l_Y(N) = \frac{1}{2}\sum_y \int_0^1 \mathrm{d}z \left((k_\uparrow - k_\downarrow)\frac{\mathrm{d}p_\infty(y|z)}{\mathrm{d}z}\log p_\infty(y|z)\right) + \mathcal{O}\left(\frac{1}{N}\right). \tag{A.8}$$

Performing the integral within the sum gives

$$\int_0^1 \mathrm{d}z \frac{\mathrm{d}p_\infty(y|z)}{\mathrm{d}z} \log p_\infty(y|z) = \Big[p_\infty(y|z)\big(\log p_\infty(y|z) - 1\big)\Big]_0^1 \\ = 0, \tag{A.9}$$

where the second equality follows from $p_\infty(y|0) = p_\infty(y|1)$. Therefore, $l_Y(N) \sim \mathcal{O}\left(\frac{1}{N}\right)$ and thus tends to zero in the limit of $N \to \infty$.

A.2 The molecular hook

The input molecules can be moved reversibly between their boxes and the reaction volume. Hooks to reversibly move molecules between different volumes has been previously discussed in the supplementary material of[56]. First the input molecule must be attached to the hook. This can be done, for example, using the chemical reactions

$$\begin{aligned} \mathrm{H_B} + \mathrm{X} + \mathrm{F_5^*} &\rightleftharpoons \mathrm{H_BX} + \mathrm{F_5}, \\ \mathrm{H_B} + \mathrm{X^*} + \mathrm{F_5^*} &\rightleftharpoons \mathrm{H_BX^*} + \mathrm{F_5}, \end{aligned} \tag{A.10}$$

where $\mathrm{H_B}$ is the hook with no molecule attached inside the input molecule box and $\mathrm{H_BX}$ and $\mathrm{H_BX^*}$ are the hook with an input molecule attached. The binding of the hook is insensitive to the state of the input molecule.

The $\mathrm{X/X^*}$ molecule can be transferred to the reaction volume in the following steps. Initially, $[\mathrm{F_5}] = [\mathrm{F_5^*}] = 0$. After the hook is introduced to the box $[\mathrm{F_5}]$ is increased up to a value of f_5. Then, $[\mathrm{F_5^*}]$ is increased to a value $f_5^* \gg f_5$. Now the $\mathrm{X/X^*}$ molecule is for certain attached to the hook. Next, the hook is transferred from the input box to the reaction volume. This transfer can be done either chemically using a conformational change in the hook molecule mediated by another pair of fuel molecules as in the equations:

$$\begin{aligned} \mathrm{H_BX} + \mathrm{F_6^*} &\rightleftharpoons \mathrm{H_{RV}X} + \mathrm{F_6}, \\ \mathrm{H_BX^*} + \mathrm{F_6^*} &\rightleftharpoons \mathrm{H_{RV}X^*} + \mathrm{F_6}, \end{aligned} \tag{A.11}$$

or, alternatively, by a purely mechanical quasistatic process.

Then, to release the input molecule from the hook inside the reaction volume the reverse of the protocol the attach the input molecule is used. Initially $[F_5] = f_5$ and $[F_5^*] = f_5^*$ with $f_5^* \gg f_5$. Then, $[F_5^*]$ is decreased to zero and, subsequently, $[F_5]$ is decreased to zero. Now the input molecule is for certain released from the hook inside the reaction volume.

Subsequently, the measurement and extraction protocols can be carried out. Afterwards, the input molecule, which could have changed state, is moved back to its box. This is done by attaching the input molecule to the hook, moving the hook and detaching the input molecule from the hook in the exact reverse of the protocol to transfer the input molecule to the reaction volume.

The free energy of the input molecule is the same whether it is in the X or X^* state. In general it is not the case that the free energy of the input molecule when it is in its box, $\mathcal{F}_{\text{box}}$, is the same as the free energy of the input when it is in the reaction volume, $\mathcal{F}_{\text{reaction volume}}$. Therefore, if the input molecule is transferred reversibly from its box to the reaction volume then the control must have done a work of $\mathcal{F}_{\text{reaction volume}} - \mathcal{F}_{\text{box}}$ on the input molecule. If the molecule is moved back to the box reversibly then this work is recovered.

Clearly, this principle can be extended to moving multiple molecules into the reaction volume as required by the batch machines, and then back again to the reaction volumes. Again, the whole process requires no net work. The initial transfer is not reversible for a structured environment, however, since the order of molecules is randomised within the reaction volumes.

A.3 Work calculation for Biochemical Szilard engine

A.3.1 Measurement

The measurement follows the optimal copy protocol in[55]. Initially, the memory molecule is in states M_0 and M_1 with equal probability and the input molecule is independently in states X and X^* with equal probability. The

state of the input is measured using the equations

$$\begin{aligned} \mathrm{M}_0 + \mathrm{X} + \mathrm{F}_1^* &\rightleftharpoons \mathrm{M}_1 + \mathrm{X} + \mathrm{F}_1, \\ \mathrm{M}_0 + \mathrm{X}^* + \mathrm{F}_2^* &\rightleftharpoons \mathrm{M}_1 + \mathrm{X}^* + \mathrm{F}_2. \end{aligned} \tag{A.12}$$

The free energy difference $\Delta G_1 = \mu_{\mathrm{F}_1} - \mu_{\mathrm{F}_1^*} = \Delta G_1^0 + \ln \frac{[\mathrm{F}_1]}{[\mathrm{F}_1^*]}$, where ΔG_1^0 depends on the intrinsic nature of the F_1 and F_1^* molecules and the reaction volume but not their concentrations, is quasistatically changed from 0 to ∞ and $\Delta G_2 = \mu_{\mathrm{F}_2} - \mu_{[2]} = \Delta G_2^0 + \ln \frac{[\mathrm{F}_2]}{[\mathrm{F}_2^*]}$ is quasistatically changed from 0 to $-\infty$.

Firstly, let us assume that the input molecule is in state X, which occurs with prob of 1/2. In this case, only the first reaction in equation A.12 can occur. At any point in the process there is a probability $p(\mathrm{M}_1)$ that the memory molecule is in state M_1. This probability only changes with a corresponding change in the number of F_1 and F_1^* in the buffer. If $p(\mathrm{M}_1)$ changes by a small amount $\mathrm{d}p(\mathrm{M}_1)$ then $\mathrm{d}p(\mathrm{M}_1)$ F_1^* are converted into F_1 so a work of $\mathrm{d}p(\mathrm{M}_1)\Delta G_1$ is done on the buffer connected to the reaction volume. Therefore, in a process a work is done on the buffers of

$$W = \int_{t_i}^{t_f} \mathrm{d}t \frac{\mathrm{d}p(\mathrm{M}_1)}{\mathrm{d}t} \Delta G_1(t). \tag{A.13}$$

Because the change in concentration of the fuels is quasistatic, at all times in the process the memory molecule is in equilibrium with the fuel buffer the reaction volume is connected to. Therefore,

$$p(\mathrm{M}_1) = \frac{\mathrm{e}^{-\beta \Delta G_1}}{1 + \mathrm{e}^{-\beta \Delta G_1}}, \tag{A.14}$$

where $\beta = \frac{1}{k_{\mathrm{B}} T}$.

The fact that only dependence $p(\mathrm{M}_1)$ has on time is through ΔG_1 means that equation A.13 can be converted into an integral over ΔG_1 instead. Because the change is quasistatic the particular function of time that ΔG_1 is does not

matter. Only the change in ΔG_1 matters. Therefore,

$$W = \int_0^\infty \mathrm{d}\Delta G_1 \frac{\mathrm{d}p(\mathrm{M}_1)}{\mathrm{d}\Delta G_1} \Delta G_1. \tag{A.15}$$

Now to get the work we simply have to use equation A.14 and evaluate the integral. It is convenient to first integrate by parts to get

$$W = [p(\mathrm{M}_1)\Delta G_1]_0^\infty - \int_0^\infty \mathrm{d}\Delta G_1 p(\mathrm{M}_1), \tag{A.16}$$

and then exploit equation A.14 to get

$$\begin{aligned} W &= \left[\frac{\mathrm{e}^{-\beta\Delta G_1}}{1+\mathrm{e}^{-\beta\Delta G_1}}\Delta G_1\right]_0^\infty - \int_0^\infty \mathrm{d}\Delta G_1 \frac{\mathrm{e}^{-\beta\Delta G_1}}{1+\mathrm{e}^{-\beta\Delta G_1}} \\ &= -\left[-\frac{1}{\beta}\ln\left(1+\mathrm{e}^{-\beta\Delta G_1}\right)\right]_0^\infty \\ &= -k_\mathrm{B}T\ln 2, \end{aligned} \tag{A.17}$$

using l'Hôpital's rule for the $\Delta G_1 \to \infty$ limit in the first line. A negative work corresponds to a decrease in free energy of the buffers.

Alternatively, there is a probability of 1/2 that the input molecule is X^* so only the second reaction in equation A.12 can occur. In this case

$$p(\mathrm{M}_1) = \frac{\mathrm{e}^{-\beta\Delta G_2}}{1+\mathrm{e}^{-\beta\Delta G_2}}. \tag{A.18}$$

where $\Delta G_2 = \mu_{\mathrm{F}_2} - \mu_{\mathrm{F}_2^*} = \Delta G_2^0 + \ln\frac{[\mathrm{F}_2]}{[\mathrm{F}_2^*]}$ and the work done on the buffers is

$$\begin{aligned} W &= \int_0^{-\infty} \mathrm{d}\Delta G_2 \frac{\mathrm{d}p(\mathrm{M}_1)}{\mathrm{d}\Delta G_2}\Delta G_2 \\ &= \left[\frac{\mathrm{e}^{-\beta\Delta G_2}}{1+\mathrm{e}^{-\beta\Delta G_2}}\Delta G_2\right]_0^{-\infty} - \int_0^{-\infty} \mathrm{d}\Delta G_2 \frac{\mathrm{e}^{-\beta\Delta G_2}}{1+\mathrm{e}^{-\beta\Delta G_2}} \\ &= \left[\frac{\mathrm{e}^{-\beta\Delta G_2}}{1+\mathrm{e}^{-\beta\Delta G_2}}\Delta G_2 + \frac{1}{\beta}\ln\left(1+\mathrm{e}^{-\beta\Delta G_2}\right)\right]_0^{-\infty}. \end{aligned} \tag{A.19}$$

To evaluate the upper limit it is convenient to substitute in equation A.18

$$\begin{aligned} W &= \Big[-p(\mathrm{M}_1)\ln p(\mathrm{M}_1) - \big(1-p(\mathrm{M}_1)\big)\ln\big(1-p(\mathrm{M}_1)\big)\Big]_{1/2}^{1} \\ &= -k_\mathrm{B}T\ln 2. \end{aligned} \tag{A.20}$$

Each of these possibilities is equally likely so the expected work is

$$W = -k_\mathrm{B}T\ln 2. \tag{A.21}$$

A.3.2 Extraction

Now the system is either in state $(\mathrm{X},\mathrm{M}_0)$ or $(\mathrm{X}^*,\mathrm{M}_1)$. Work is extracted from this high free energy state using the reactions

$$\begin{aligned} \mathrm{M}_0 + \mathrm{X} + \mathrm{F}_3^* &\rightleftharpoons \mathrm{M}_0 + \mathrm{X}^* + \mathrm{F}_3, \\ \mathrm{M}_1 + \mathrm{X} + \mathrm{F}_4^* &\rightleftharpoons \mathrm{M}_1 + \mathrm{X}^* + \mathrm{F}_4. \end{aligned} \tag{A.22}$$

If the system is in state $(\mathrm{X},\mathrm{M}_0)$ then only the first reaction in equation A.22 can occur. In this case the probability of the input molecule being in the X^* state is:

$$p(\mathrm{X}^*) = \frac{\mathrm{e}^{-\beta\Delta G_3}}{1+\mathrm{e}^{-\beta\Delta G_3}}, \tag{A.23}$$

where $\Delta G_3 = \mu_{\mathrm{F}_3^*} - \mu_{\mathrm{F}_3} = \Delta G_3^0 + \ln\frac{[\mathrm{F}_3^*]}{[\mathrm{F}_3]}$. As ΔG_3 is changed from ∞ to 0, we obtain

$$\begin{aligned} W &= \int_\infty^0 \mathrm{d}\Delta G_3 \frac{\mathrm{d}p(\mathrm{X}^*)}{\mathrm{d}\Delta G_3}\Delta G_3 \\ &= \left[\frac{\mathrm{e}^{-\beta\Delta G_3}}{1+\mathrm{e}^{-\beta\Delta G_3}}\Delta G_3\right]_\infty^0 - \int_\infty^0 \mathrm{d}\Delta G_3 \frac{\mathrm{e}^{-\beta\Delta G_3}}{1+\mathrm{e}^{-\beta\Delta G_3}} \\ &= -\left[-\frac{1}{\beta}\ln\left(1+\mathrm{e}^{-\beta\Delta G_3}\right)\right]_\infty^0 \\ &= k_\mathrm{B}T\ln 2. \end{aligned} \tag{A.24}$$

This is exactly the same calculation as equation A.17. The sign is positive

because the free energy of the fuel molecule buffers is now increased.

If the system is in state $(\mathrm{X}^*, \mathrm{M}_1)$ then only the second reaction in equation A.22 can occur. In this case the probability of the input molecule being in the X^* state is:

$$p(\mathrm{X}^*) = \frac{\mathrm{e}^{-\beta\Delta G_4}}{1+\mathrm{e}^{-\beta\Delta G_4}}, \tag{A.25}$$

where $\Delta G_4 = \mu_{\mathrm{F}_4^*} - \mu_{\mathrm{F}_4} = \Delta G_4^0 + \ln\frac{[\mathrm{F}_4^*]}{[\mathrm{F}_4]}$. As ΔG_4 is changed from $-\infty$ to 0, we obtain

$$\begin{aligned} W &= \int_{-\infty}^{0} \mathrm{d}\Delta G_4 \frac{\mathrm{d}p(\mathrm{X}^*)}{\mathrm{d}\Delta G_4}\Delta G_4 \\ &= \left[\frac{\mathrm{e}^{-\beta\Delta G_4}}{1+\mathrm{e}^{-\beta\Delta G_4}}\Delta G_4\right]_{-\infty}^{0} - \int_{-\infty}^{0}\mathrm{d}\Delta G_4 \frac{\mathrm{e}^{-\beta\Delta G_4}}{1+\mathrm{e}^{-\beta\Delta G_4}} \\ &= \left[\frac{\mathrm{e}^{-\beta\Delta G_4}}{1+\mathrm{e}^{-\beta\Delta G_4}}\Delta G_4 + \frac{1}{\beta}\ln\left(1+\mathrm{e}^{-\beta\Delta G_4}\right)\right]_{-\infty}^{0} \\ &= k_{\mathrm{B}}T\ln 2. \end{aligned} \tag{A.26}$$

This is exactly the same calculation as equations A.19 and A.20.

Each of these possibilities is equally likely so the expected work is

$$W = k_{\mathrm{B}}T\ln 2. \tag{A.27}$$

Therefore, in a measure and extract cycle the net work done by the fuel molecule buffers is zero.

A.4 Quasistatic embeddability of Markov machine

Any transformation of a probability distribution over discrete states can be represented by a stochastic matrix. A quasistatic embedding is a non-homogeneous continuous time Markov chain that produces such a transformation with no entropy production[65]. It is not possible to find such an embedding for all stochastic matrices. For some stochastic matrices the state-space must be extended with additional 'hidden' states before a quasistatic

embedding can be found. Owen *et al.*[65] have found bounds on the number of hidden state required.

We can apply the results of[65] to the Markov machine. The joint system of the input molecule and memory molecule has four states. We order them $(\mathrm{XM}_0, \mathrm{XM}_1, \mathrm{X}^*\mathrm{M}_0, \mathrm{X}^*\mathrm{M}_1)$. The transformation that measures the state of the input molecule to the state of the memory molecule and takes the input molecule to its equilibrium distribution is then

$$P = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \end{pmatrix}. \tag{A.28}$$

The determinant of P is zero so according to[65] the lower bound on the number of additional hidden states required for a quasistatic embedding is zero.

The upper found on the number of hidden states required is $r^+(P) - 1$ where $r^+(P)$ is the nonnegative rank of P. For an $n \times n$ stochastic matrix, M, the nonnegative rank is the smallest m such that M can be written $M = RS$ where R is a $n \times m$ stochastic matrix and S is a $m \times n$ stochastic matrix.

P can be written as

$$P = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \\ \frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \tag{A.29}$$

so the nonnegative rank of this matrix is two and the upper bound on the number of additional hidden states required is one.

In the main text we see that the actual number of hidden states required is zero because we construct the protocol explicitly.

A.5 Work calculation for Markov machine

A.5.1 Measurement

The measurement follows the optimal copy protocol in[55]. Instead of the input and memory molecules initially being independent, as they were for the Szilard engine, the molecules now start off correlated.

The reactions

$$\begin{aligned} \mathrm{M}_0 + \mathrm{X} + \mathrm{F}_1^* &\rightleftharpoons \mathrm{M}_1 + \mathrm{X} + \mathrm{F}_1, \\ \mathrm{M}_0 + \mathrm{X}^* + \mathrm{F}_2^* &\rightleftharpoons \mathrm{M}_1 + \mathrm{X}^* + \mathrm{F}_2. \end{aligned} \tag{A.30}$$

are, again, used. If the input is X then only the first reaction in equation A.30 can occur. Now $p(\mathrm{M}_1|\mathrm{X})$ is not $1/2$, it is set by the input process. To set the memory molecule to M_0 using the least work $\Delta G_1 = \mu_{\mathrm{F}_1} - \mu_{\mathrm{F}_1^*} = \Delta G_1^0 + \ln \frac{[\mathrm{F}_1]}{[\mathrm{F}_1^*]}$ must be initially set to $\Delta G_1 = \Delta G_{\mathrm{offset}}$ such that

$$p(\mathrm{M}_1|\mathrm{X}) = \frac{\mathrm{e}^{-\beta \Delta G_{\mathrm{offset}}}}{1 + \mathrm{e}^{-\beta \Delta G_{\mathrm{offset}}}}, \tag{A.31}$$

and then quasistatically changed to $\Delta G_1 = \infty$. Therefore, the work done is

$$\begin{aligned} W &= \int_{\Delta G_{\mathrm{offset}}}^{\infty} \mathrm{d}\Delta G_1 \frac{\mathrm{d}p(\mathrm{M}_1)}{\mathrm{d}\Delta G_1} \Delta G_1 \\ &= \left[\frac{\mathrm{e}^{-\beta \Delta G_1}}{1 + \mathrm{e}^{-\beta \Delta G_1}} \Delta G_1 \right]_{\Delta G_{\mathrm{offset}}}^{\infty} - \int_{\Delta G_{\mathrm{offset}}}^{\infty} \mathrm{d}\Delta G_1 \frac{\mathrm{e}^{-\beta \Delta G_1}}{1 + \mathrm{e}^{-\beta \Delta G_1}} \\ &= -\frac{\mathrm{e}^{-\beta \Delta G_{\mathrm{offset}}}}{1 + \mathrm{e}^{-\beta \Delta G_{\mathrm{offset}}}} \Delta G_{\mathrm{offset}} - \left[-\frac{1}{\beta} \ln \left(1 + \mathrm{e}^{-\beta \Delta G_1} \right) \right]_{\Delta G_{\mathrm{offset}}}^{\infty} \\ &= -\frac{\mathrm{e}^{-\beta \Delta G_{\mathrm{offset}}}}{1 + \mathrm{e}^{-\beta \Delta G_{\mathrm{offset}}}} \Delta G_{\mathrm{offset}} - \frac{1}{\beta} \ln \left(1 + \mathrm{e}^{-\beta \Delta G_{\mathrm{offset}}} \right) \\ &= \frac{1}{\beta} \Big(p(\mathrm{M}_1|\mathrm{X}) \ln p(\mathrm{M}_1|\mathrm{X}) + \big(1 - p(\mathrm{M}_1|\mathrm{X})\big) \ln \big(1 - p(\mathrm{M}_1|\mathrm{X})\big) \Big). \end{aligned} \tag{A.32}$$

Similarly, if the input molecule is X^* then $\Delta G_2 = \mu_{\mathrm{F}_2} - \mu_{\mathrm{F}_2^*} = \Delta G_2^0 + \ln \frac{[\mathrm{F}_2]}{[\mathrm{F}_2^*]}$

must be initially set to$\Delta G_2 = -\Delta G_{\text{offset}}$ such that

$$p(\mathrm{M}_1|\mathrm{X}^*) = \frac{\mathrm{e}^{\beta\Delta G_{\text{offset}}}}{1+\mathrm{e}^{\beta\Delta G_{\text{offset}}}}, \tag{A.33}$$

and then quasistatically changed to $\Delta G_1 = -\infty$. Therefore, the work done is

$$\begin{aligned}
W &= \int_{-\Delta G_{\text{offset}}}^{-\infty} \mathrm{d}\Delta G_2 \frac{\mathrm{d}p(\mathrm{M}_1)}{\mathrm{d}\Delta G_2}\Delta G_2 \\
&= \left[\frac{\mathrm{e}^{-\beta\Delta G_2}}{1+\mathrm{e}^{-\beta\Delta G_2}}\Delta G_2\right]_{-\Delta G_{\text{offset}}}^{-\infty} - \int_{-\Delta G_{\text{offset}}}^{-\infty} \mathrm{d}\Delta G_2 \frac{\mathrm{e}^{-\beta\Delta G_2}}{1+\mathrm{e}^{-\beta\Delta G_2}} \\
&= \left[\frac{\mathrm{e}^{-\beta\Delta G_2}}{1+\mathrm{e}^{-\beta\Delta G_2}}\Delta G_2 + \frac{1}{\beta}\ln\left(1+\mathrm{e}^{-\beta\Delta G_2}\right)\right]_{-\Delta G_{\text{offset}}}^{-\infty} \\
&= \frac{\mathrm{e}^{\beta\Delta G_{\text{offset}}}}{1+\mathrm{e}^{\beta\Delta G_{\text{offset}}}}\Delta G_{\text{offset}} - \frac{1}{\beta}\ln\left(1+\mathrm{e}^{\beta\Delta G_{\text{offset}}}\right) \\
&= \frac{1}{\beta}\Big(p(\mathrm{M}_1|\mathrm{X}^*)\ln p(\mathrm{M}_1|\mathrm{X}^*) \\
&+ \big(1-p(\mathrm{M}_1|\mathrm{X}^*)\big)\ln\big(1-p(\mathrm{M}_1|\mathrm{X}^*)\big)\Big).
\end{aligned} \tag{A.34}$$

The first case occurs with probability $p(\mathrm{X})$ and the second occurs with probability $p(\mathrm{X}^*)$ so the expected work is

$$\begin{aligned}
W =& p(\mathrm{X})\frac{1}{\beta}\Big(p(\mathrm{M}_1|\mathrm{X})\ln p(\mathrm{M}_1|\mathrm{X}) \\
&+ \big(1-p(\mathrm{M}_1|\mathrm{X})\big)\ln\big(1-p(\mathrm{M}_1|\mathrm{X})\big)\Big) \\
&+ p(\mathrm{X}^*)\frac{1}{\beta}\Big(p(\mathrm{M}_1|\mathrm{X}^*)\ln p(\mathrm{M}_1|\mathrm{X}^*) \\
&+ \big(1-p(\mathrm{M}_1|\mathrm{X}^*)\big)\ln\big(1-p(\mathrm{M}_1|\mathrm{X}^*)\big)\Big) \\
=& -\frac{1}{\beta}H[M_i|X_{i+1}].
\end{aligned} \tag{A.35}$$

The random variable M_i is an exact copy of X_i so

$$\begin{aligned} W &= -\frac{1}{\beta} H[M_i|X_{i+1}] \\ &= -\frac{1}{\beta} H[X_i|X_{i+1}]. \end{aligned} \tag{A.36}$$

The input process is stationary so $H[X_i] = H[X_{i+1}]$ so[7]

$$\begin{aligned} W &= -\frac{1}{\beta} H[X_i|X_{i+1}] \\ &= -\frac{1}{\beta} \Big(H[X_{i+1}|X_i] + H[X_i] - H[X_{i+1}] \Big) \\ &= -\frac{1}{\beta} H[X_{i+1}|X_i]. \end{aligned} \tag{A.37}$$

$H[X_{i+1}|X_i] \leq \ln 2$ so the fact that the memory molecule and input molecule are initially correlated means that the measurement requires less work to be done on the system by the fuel molecule buffers.

A.5.2 Extraction

The extraction process is exactly the same as for the biochemical Szilard engine. Therefore, the work done on the fuel molecule buffers is $k_{\mathrm{B}}T \ln 2$ so the net work per input molecule is

$$W = k_{\mathrm{B}}T \Big(\ln 2 - H[X_{i+1}|X_i] \Big). \tag{A.38}$$

A.6 DNA design of the Biochemical Szilard engine and Markov machine

In this section we present a domain level DNA-based design to implement the measurement and work extraction reactions of the Biochemical Szilard engine and Markov machine using DNA strand displacement. The design is shown in figure A.1. Our designs leverage the general construction of[16].

The nature of DNA strand displacement reactions means that additional auxiliary strands, labelled A_1 to A_{12}, are required. We assume that these strands are always present in the reaction volume in excess.

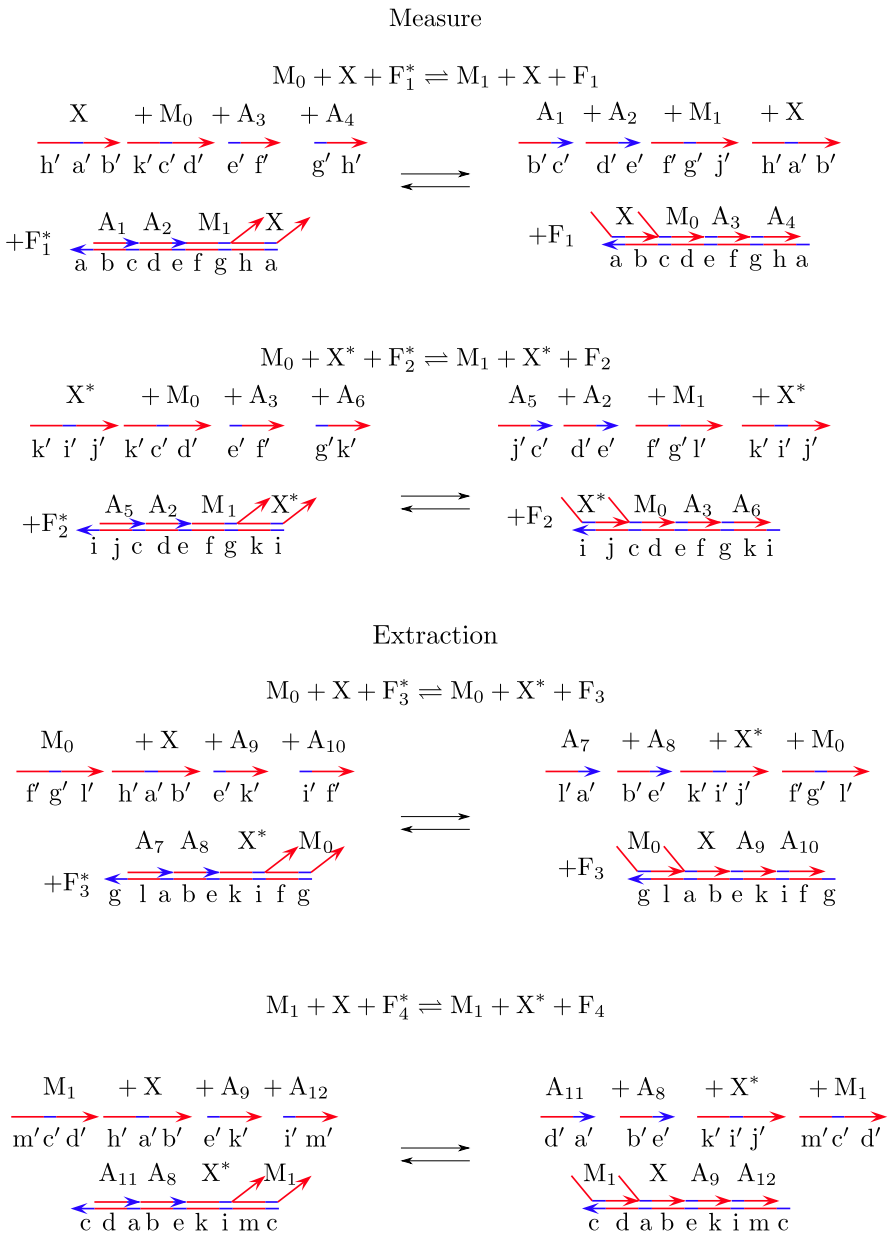


Figure A.1: *A domain level DNA-based design to implement the measurement and work extraction reactions of the Biochemical Szilard engine.* The arrowheads show the 3′ end of the strands. The label 'a' represents a sequence of bases (domain) and 'a′' represents the complimentary sequence.

A.7 Work calculation for Binary batch machine

A.7.1 Measurement

The measurement follows the optimal copy protocol in[55]. The measurement is done using the reactions

$$\begin{aligned} \mathrm{M}_0 + \frac{N+1}{2}\mathrm{X} + \mathrm{F}_1^* &\rightleftharpoons \mathrm{M}_1 + \frac{N+1}{2}\mathrm{X} + \mathrm{F}_1, \\ \mathrm{M}_0 + \frac{N+1}{2}\mathrm{X}^* + \mathrm{F}_2^* &\rightleftharpoons \mathrm{M}_1 + \frac{N+1}{2}\mathrm{X}^* + \mathrm{F}_2, \end{aligned} \tag{A.39}$$

when N is odd and

$$\begin{aligned} \mathrm{M}_0 + \frac{N}{2}\mathrm{X} + \mathrm{F}_1^* &\rightleftharpoons \mathrm{M}_1 + \frac{N}{2}\mathrm{X} + \mathrm{F}_1, \\ \mathrm{M}_0 + \left(\frac{N}{2}+1\right)\mathrm{X}^* + \mathrm{F}_2^* &\rightleftharpoons \mathrm{M}_1 + \left(\frac{N}{2}+1\right)\mathrm{X}^* + \mathrm{F}_2, \end{aligned} \tag{A.40}$$

when N is even.

Let J_i be the random variable representing whether the number of X^* molecules in batch i is greater than $N/2$ ($J_i = 1$ if true, 0 otherwise). The measurement process is exactly the same as for the Markov machine except that the chemical potential differences, $\Delta G_1 = \mu_{\mathrm{F}_1} - \mu_{\mathrm{F}_1^*} = \Delta G_1^0 + \ln \frac{[\mathrm{F}_1]}{[\mathrm{F}_1^*]}$ and $\Delta G_2 = \mu_{\mathrm{F}_2} - \mu_{\mathrm{F}_2^*} = \Delta G_2^0 + \ln \frac{[\mathrm{F}_2]}{[\mathrm{F}_2^*]}$, are initially set to $\Delta G_1 = \Delta G_{\mathrm{offset}}^1$ and $\Delta G_2 = \Delta G_{\mathrm{offset}}^2$ such that

$$p(\mathrm{M}_1|J_i = 0) = \frac{\mathrm{e}^{-\beta \Delta G_{\mathrm{offset}}^1}}{1 + \mathrm{e}^{-\beta \Delta G_{\mathrm{offset}}^1}}, \tag{A.41}$$

and

$$p(\mathrm{M}_1|J_i = 1) = \frac{\mathrm{e}^{-\beta \Delta G_{\mathrm{offset}}^2}}{1 + \mathrm{e}^{-\beta \Delta G_{\mathrm{offset}}^2}}. \tag{A.42}$$

Therefore, the work done is

$$W = -k_{\mathrm{B}} T H[M_i|J_{i+1}], \tag{A.43}$$

and, similar to the Markov machine, M_i is an exact copy of J_i and $H[J_{i+1}] =$

$H[J_i]$ so

$$W = -k_{\mathrm{B}} T H[J_{i+1}|J_i]. \tag{A.44}$$

A.7.2 Extraction

The extraction step uses the same reactions as the biochemical Szilard engine and the Markov machine:

$$\begin{aligned} \mathrm{M}_0 + \mathrm{X} + \mathrm{F}_3^* &\rightleftharpoons \mathrm{M}_0 + \mathrm{X}^* + \mathrm{F}_3, \\ \mathrm{M}_1 + \mathrm{X} + \mathrm{F}_4^* &\rightleftharpoons \mathrm{M}_1 + \mathrm{X}^* + \mathrm{F}_4. \end{aligned} \tag{A.45}$$

However, the protocol of the chemical potential differences must be different. In the biochemical Szilard engine and Markov machine, if the memory molecule was in the state M_0 then the input molecule would be for certain in the state X. However, in the binary batch machine if the memory molecule is in the state M_0 then there is a nonzero probability that some of the input molecules in the batch are in state X^*.

The chemical potential differences, $\Delta G_3 = \mu_{\mathrm{F}_3^*} - \mu_{\mathrm{F}_3} = \Delta G_3^0 + \ln \frac{[\mathrm{F}_3^*]}{[\mathrm{F}_3]}$ and $\Delta G_4 = \mu_{\mathrm{F}_4^*} - \mu_{\mathrm{F}_4} = \Delta G_4^0 + \ln \frac{[\mathrm{F}_4^*]}{[\mathrm{F}_4]}$, are started at $\Delta G_3 = \Delta G_{\mathrm{offset}}^3$ and $\Delta G_4 = \Delta G_{\mathrm{offset}}^4$ where $\Delta G_{\mathrm{offset}}^3$ and $\Delta G_{\mathrm{offset}}^4$ are set so that

$$\hat{p}_0 = \frac{\mathrm{e}^{-\beta \Delta G_{\mathrm{offset}}^3}}{1 + \mathrm{e}^{-\beta \Delta G_{\mathrm{offset}}^3}}, \tag{A.46}$$

and

$$\hat{p}_1 = \frac{\mathrm{e}^{-\beta \Delta G_{\mathrm{offset}}^4}}{1 + \mathrm{e}^{-\beta \Delta G_{\mathrm{offset}}^4}}, \tag{A.47}$$

First, there is an irreversible relaxation in the batch from the initial input distribution, which depends on the input stochastic process, to a binomial distribution over the number of X^* molecules with a mean of $N\hat{p}_0$ or $N\hat{p}_1$. If the memory molecule is is M_0, the work extracted in this relaxation is

$$W_{\mathrm{relax}} = \Delta G_{\mathrm{offset}}^3 \left(N\hat{p}_0 - \langle \mathrm{X}^* \rangle_{\mathrm{M}_0}^{\mathrm{initial}} \right), \tag{A.48}$$

where $\langle \mathrm{X}^* \rangle_{\mathrm{M}_0}^{\mathrm{initial}}$ is the expected number of X^* in the batch initially. If the

memory molecule is is M_1, the work extracted in this relaxation is

$$W_{\mathrm{relax}} = \Delta G^4_{\mathrm{offset}} \left(N\hat{p}_1 - \langle \mathrm{X}^* \rangle^{\mathrm{initial}}_{\mathrm{M}_1} \right), \tag{A.49}$$

where $\langle \mathrm{X}^* \rangle^{\mathrm{initial}}_{\mathrm{M}_1}$ is the expected number of X^* in the batch initially. Then, ΔG_3 and ΔG_4 are quasistatically changed to zero. If the memory molecule is in state M_0 the work that is done in this quasistatic step is

$$\begin{aligned}
W_{\mathrm{q}} =& N \int_{\Delta G^3_{\mathrm{offset}}}^{0} \mathrm{d}\Delta G_3 \frac{\mathrm{d}p(\mathrm{X}^*)}{\mathrm{d}\Delta G_3} \Delta G_3 \\
=& N \left(\left[\frac{\mathrm{e}^{-\beta \Delta G_3}}{1 + \mathrm{e}^{-\beta \Delta G_3}} \Delta G_3 \right]^{0}_{\Delta G^3_{\mathrm{offset}}} - \int_{\Delta G^3_{\mathrm{offset}}}^{0} \mathrm{d}\Delta G_3 \frac{\mathrm{e}^{-\beta \Delta G_3}}{1 + \mathrm{e}^{-\beta \Delta G_3}} \right) \\
=& N \left[\frac{\mathrm{e}^{-\beta \Delta G_3}}{1 + \mathrm{e}^{-\beta \Delta G_3}} \Delta G_3 + \frac{1}{\beta} \ln \left(1 + \mathrm{e}^{-\beta \Delta G_3} \right) \right]^{0}_{\Delta G^3_{\mathrm{offset}}} \\
=& N \left(\frac{1}{\beta} \ln 2 - \frac{\mathrm{e}^{-\beta \Delta G^3_{\mathrm{offset}}}}{1 + \mathrm{e}^{-\beta \Delta G^3_{\mathrm{offset}}}} \Delta G^3_{\mathrm{offset}} - \frac{1}{\beta} \ln \left(1 + \mathrm{e}^{-\beta \Delta G^3_{\mathrm{offset}}} \right) \right) \\
=& N \frac{1}{\beta} \Big(\ln 2 + \hat{p}_0 \ln \hat{p}_0 + (1 - \hat{p}_0) \ln (1 - \hat{p}_0) \Big).
\end{aligned} \tag{A.50}$$

Similarly, if the memory molecule is in state M_1 the work that is done in this quasistatic step is

$$W_{\mathrm{q}} = N \frac{1}{\beta} \Big(\ln 2 + \hat{p}_1 \ln \hat{p}_1 + (1 - \hat{p}_1) \ln (1 - \hat{p}_1) \Big). \tag{A.51}$$

Therefore, if the memory molecule is in state M_0 the total work that is done in the irreversible relaxation and quasistatic steps is

$$N \frac{1}{\beta} \Big(\ln 2 + \langle \mathrm{X}^* \rangle^{\mathrm{initial}}_{\mathrm{M}_0} / N \ln \hat{p}_0 + (1 - \langle \mathrm{X}^* \rangle^{\mathrm{initial}}_{\mathrm{M}_0} / N) \ln (1 - \hat{p}_0) \Big). \tag{A.52}$$

This is maximised if $\hat{p}_0 = \langle \mathrm{X}^* \rangle^{\mathrm{initial}}_{\mathrm{M}_0} / N$. Similarly, if the memory molecule is in state M_0 the work is maximised by setting $\hat{p}_1 = \langle \mathrm{X}^* \rangle^{\mathrm{initial}}_{\mathrm{M}_1} / N$. i.e. $\hat{p}_0$ is the probability that an input molecule in the batch is X^* if $J_i = 0$ and $\hat{p}_1$ is the probability that an input molecule in the batch is X^* if $J_i = 1$. This means that initially no work is done on the fuel molecule buffers during the

irreversible relaxation because on average there is no net change of number of X^* in the batch.

Therefore, the expected work done in the extraction step is

$$W = k_B T N \Big(\ln 2 + p(J_i = 0)\big(\hat{p}_0 \ln \hat{p}_0 + (1-\hat{p}_0)\ln(1-\hat{p}_0)\big) + p(J_i = 1)\big(\hat{p}_1 \ln \hat{p}_1 + (1-\hat{p}_1)\ln(1-\hat{p}_1)\big)\Big), \tag{A.53}$$

and the net work for the measure and extract cycle is

$$W = k_B T N \Big(\ln 2 + p(J_i = 0)\big(\hat{p}_0 \ln \hat{p}_0 + (1-\hat{p}_0)\ln(1-\hat{p}_0)\big) + p(J_i = 1)\big(\hat{p}_1 \ln \hat{p}_1 + (1-\hat{p}_1)\ln(1-\hat{p}_1)\big)\Big) - k_B T H[J_i|J_{i-1}]. \tag{A.54}$$

A.8 DNA design of batch machine measurement

In this section we present a domain level DNA-based design to implement the measurement reactions of the batch machine using DNA strand displacement. The design is shown in figure A.2 for the case when $N = 5$. Our designs leverage the general construction of[16]. This design is the same as the measurement reactions for the Biochemical Szilard engine and Markov machine except for that the gates are extended so that three X or X^* strands must bind for the reaction to occur. In principle, the mechanism could be generalised to an arbitrary number of inputs—although this may prove challenging in practice.

The nature of DNA strand displacement reactions means that additional auxiliary strands, labelled A_1 to A_{16}, are required. We assume that these strands are always present in the reaction volume in excess.

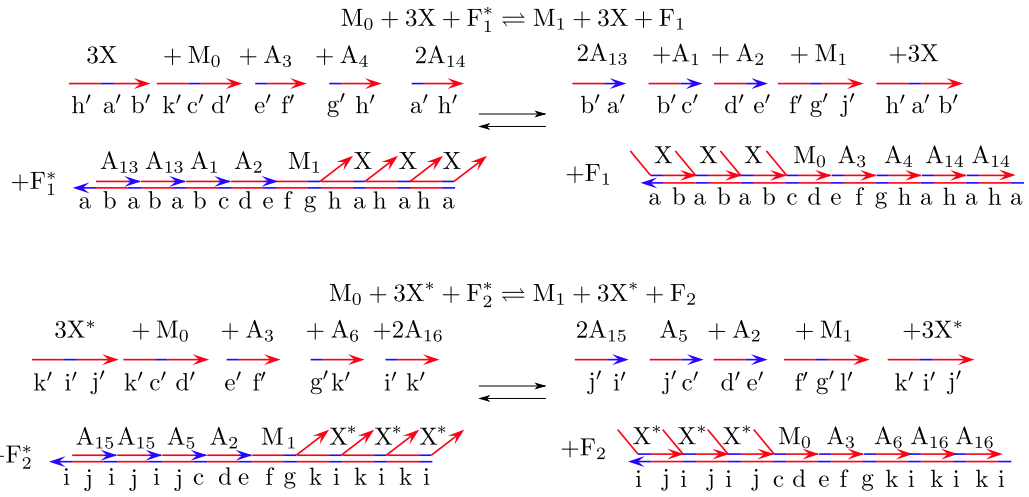


Figure A.2: *A domain level DNA-based design to implement the measurement reactions of the batch machines.* The arrowheads show the 3′ end of the strands. The label 'a' represents a sequence of bases (domain) and 'a′' represents the complimentary sequence. The extraction reactions are the same as for the biochemical Szilard engine.

A.9 Work calculation for full batch machine

A.9.1 Measurement

The measurement is exactly the same as for the binary batch machine so the work is

$$W_{\text{measure}} = -k_{\text{B}}TH[J_{i+1}|J_i]. \tag{A.55}$$

A.9.2 Extraction

In this section we will not give an explicit chemical scheme to extract all of the work from an unordered batch of input molecules. We will simply calculate the available work. In equilibrium the number of X^* molecules in the batch, $n(\text{X}^*)$, is described by a random variable B^{eq}, which is distributed as

$$p\big(B^{\text{eq}} = n(\text{X}^*)\big) = \frac{1}{2^N}\frac{N!}{n(\text{X}^*)!(N - n(\text{X}^*))!}. \tag{A.56}$$

If we define the free energy of each state of the unordered batch as

$$\mathcal{F}(n(\mathrm{X}^*)) = -k_B T \ln p\big(B^{\mathrm{eq}} = n(\mathrm{X}^*)\big), \tag{A.57}$$

then the equilibrium free energy is zero. Initially the number of X^* molecules in the batch is described by a random variable B^{initial} The free energy of the batch is initially

$$\mathcal{F}_{\mathrm{B}}(B^{\mathrm{initial}}) = \sum_{n(\mathrm{X}^*)=0}^{N} p_{\mathrm{in}}(n(\mathrm{X}^*)\mathcal{F}(n(\mathrm{X}^*)) + k_{\mathrm{B}}T \sum_{n(\mathrm{X}^*)=0}^{N} p_{\mathrm{in}}(n(\mathrm{X}^*) \ln p_{\mathrm{in}}(n(\mathrm{X}^*), \tag{A.58}$$

where $p_{\mathrm{in}}(n(\mathrm{X}^*)$ is the initial distribution over the number of X^* in the batch and B^{initial} is the random variable that describes the initial state of the batch. Therefore, using equation 1.23, the free energy of the joint system of the batch and the memory molecule is

$$\mathcal{F}_{\mathrm{joint}}(B^{\mathrm{initial}}, M) = \mathcal{F}_{\mathrm{B}}(B^{\mathrm{initial}}) + \mathcal{F}_{\mathrm{M}}(M) + k_{\mathrm{B}}T\mathcal{I}[B^{\mathrm{initial}}; M], \tag{A.59}$$

where M is the state of the memory molecule after the batch has been measured. After the work extraction the free energy of the joint system of the batch and the memory molecule is

$$\mathcal{F}(B^{\mathrm{eq}}, M) = \mathcal{F}(B^{\mathrm{eq}}) + \mathcal{F}(M) + k_{\mathrm{B}}T\mathcal{I}[B^{\mathrm{eq}}; M]. \tag{A.60}$$

As previously mention, $\mathcal{F}(B^{\mathrm{eq}}) = 0$. It is also the case that $\mathcal{I}[B^{\mathrm{eq}}; M] = 0$. Therefore, the maximum work that can be extracted is

$$\begin{aligned} W_{\mathrm{extract}} &= \mathcal{F}_{\mathrm{joint}}(B^{\mathrm{initial}}, M) - \mathcal{F}_{\mathrm{joint}}(B^{\mathrm{eq}}, M) \\ &= \mathcal{F}_{\mathrm{B}}(B^{\mathrm{initial}}) + k_{\mathrm{B}}T\mathcal{I}[B^{\mathrm{initial}}; M] \\ &= \mathcal{F}_{\mathrm{B}}(B^{\mathrm{initial}}) + k_{\mathrm{B}}TH[M], \end{aligned} \tag{A.61}$$

where the last line follows because knowing the state of B^{initial}gives you exact knowledge of the state of M so $H[M|B^{\mathrm{initial}}] = 0$. Therefore, the net work

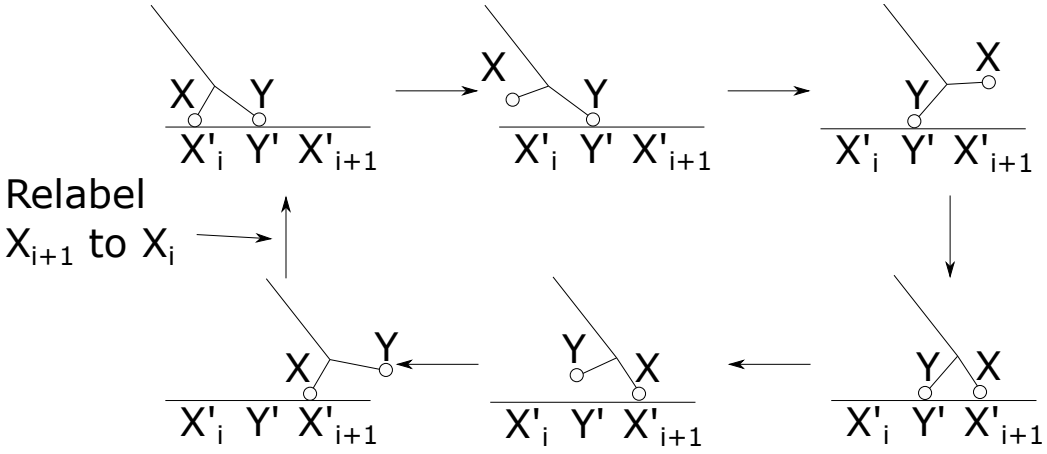


Figure A.3: Overview of more realistic molecular motor. The arrows represent the direction of the quasistatic change when taking a rightwards step. Of course, each step is microscopically reversible and when moving left the arrows are all in the opposite directions. The X′ is meant to show that that is a place where the X foot can bind.

extracted per batch is

$$W = \mathcal{F}(B^{\text{initial}}) + k_{\text{B}}T\mathcal{I}[J_{i+1}, J_i]. \tag{A.62}$$

A.10 Alternative Molecular motor

It is more realistic if the motor can detach from the tape. Consider an actin/myosin type thing where the motor has two feet. An overview is shown in figure A.3.

This is more complicated than the previous set-up because the motor has more possible states but there are only 5 different reactions required. To move right there are the reactions

$$\begin{aligned} \mathrm{F}_1^* + \mathrm{R} + \mathrm{X}_{\text{attached}} &\rightleftharpoons \mathrm{R} + \mathrm{X}_{\text{unattched}} + \mathrm{F}_1 \\ \mathrm{F}_2^* + \mathrm{R} + \mathrm{Y}_{\text{attached}} &\rightleftharpoons \mathrm{R} + \mathrm{Y}_{\text{unattached}} + \mathrm{F}_2 \\ \mathrm{F}_3^* + \mathrm{R} + \mathrm{XY} &\rightleftharpoons \mathrm{R} + \mathrm{YX} + \mathrm{F}_3 \end{aligned} \tag{A.63}$$

where XY and YX represent the configurations where either the X foot is on the left of the motor or the Y foot is on the left of the motor. To move

	A	B
0	B1R	A2L
1	A2L	B2R
2	A1L	A0R

Table A.1: The rules for Wolfram's 2 state 3 symbol machine.

left the L state catalyses the same equations but the fuels that cause the attaching and detaching of the X and Y are swapped and the same reaction to change the conformation of the walker is used

$$\begin{aligned} \mathrm{F}_2^* + \mathrm{L} + \mathrm{X}_{\text{attached}} &\rightleftharpoons \mathrm{L} + \mathrm{X}_{\text{unattched}} + \mathrm{F}_2 \\ \mathrm{F}_1^* + \mathrm{L} + \mathrm{Y}_{\text{attached}} &\rightleftharpoons \mathrm{L} + \mathrm{Y}_{\text{unattached}} + \mathrm{F}_1 \end{aligned} \tag{A.64}$$

However, a more complicated protocol of fuel concentrations is required. This protocol is shown in figure A.4. This protocol requires twice as many steps as the previous one.

A.11 Examples

A.11.1 2-state 3-symbol Machine

This machine is claimed to be the smallest possible Turing machine[79] but it is only Turing complete if the tape initially has a complex encoding of an infinite string. The usual definition of a Turing machine requires that the tape is initialised with an infinitely repeating symbol containing only a finite string of different symbols[80].

The molecules on the tape can be in three states: 0, 1 and 2. The 'state molecule' of the head (like the 'memory molecule') has two states: A and B. The transitions for this machine are shown in table A.1.

As mentioned in section 4.4.3, the most straightforward way to convert this list of rules to an invertible map is for the memory molecule to be initially in the state M_0 and to convert it to a state labelled by the number of the rule

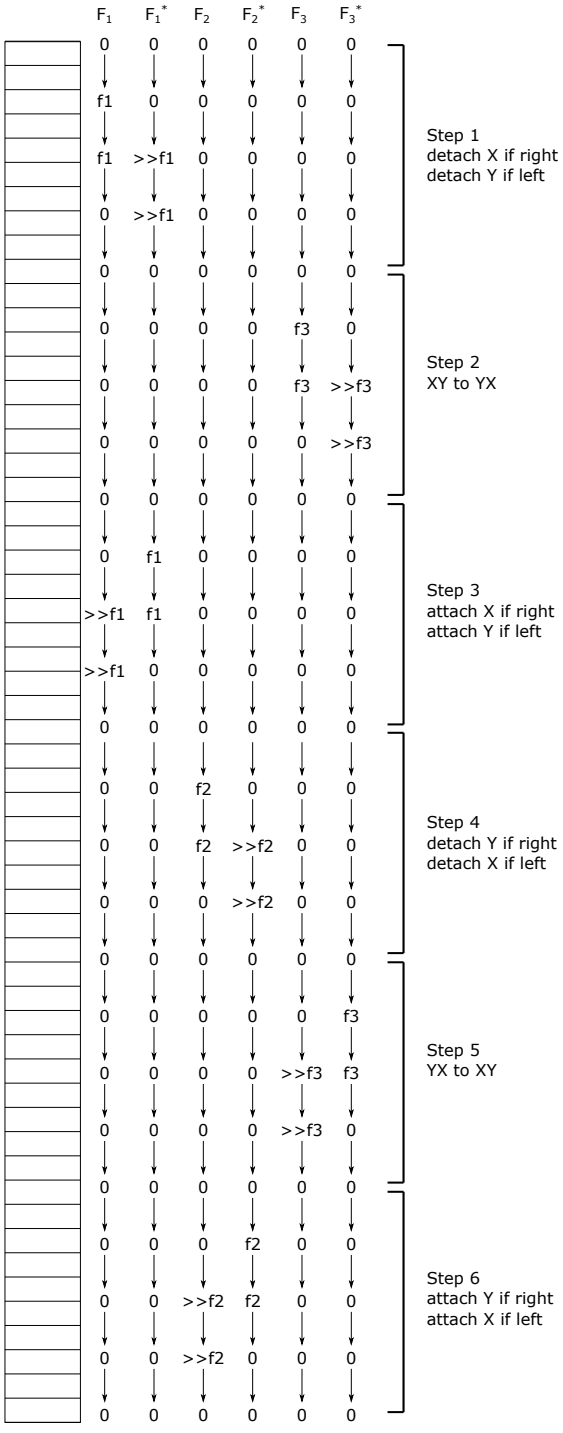


Figure A.4: The protocol of fuel concentrations for the molecular motor that is inspired by actin myosin.

that is used as below:

$$
\begin{aligned}
\mathrm{A} + \mathrm{L} + 0 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{B} + \mathrm{R} + 1 + \mathrm{M}_1 + \mathrm{G} \\
\mathrm{A} + \mathrm{R} + 0 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{B} + \mathrm{R} + 1 + \mathrm{M}_2 + \mathrm{G} \\
\mathrm{A} + \mathrm{L} + 1 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 2 + \mathrm{M}_3 + \mathrm{G} \\
\mathrm{A} + \mathrm{R} + 1 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 2 + \mathrm{M}_4 + \mathrm{G} \\
\mathrm{A} + \mathrm{L} + 2 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 1 + \mathrm{M}_5 + \mathrm{G} \\
\mathrm{A} + \mathrm{R} + 2 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 1 + \mathrm{M}_6 + \mathrm{G} \\
\mathrm{B} + \mathrm{L} + 0 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 2 + \mathrm{M}_7 + \mathrm{G} \\
\mathrm{B} + \mathrm{R} + 0 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 2 + \mathrm{M}_8 + \mathrm{G} \\
\mathrm{B} + \mathrm{L} + 1 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{B} + \mathrm{R} + 2 + \mathrm{M}_9 + \mathrm{G} \\
\mathrm{B} + \mathrm{R} + 1 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{B} + \mathrm{R} + 2 + \mathrm{M}_{10} + \mathrm{G} \\
\mathrm{B} + \mathrm{L} + 2 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{R} + 0 + \mathrm{M}_{11} + \mathrm{G} \\
\mathrm{B} + \mathrm{R} + 2 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{R} + 0 + \mathrm{M}_{12} + \mathrm{G}.
\end{aligned}
\tag{A.65}
$$

We only need one pair of fuel molecules because each reaction is otherwise unique. The protocol is to start with $[\mathrm{G}] = [\mathrm{G}^*] = 0$, then increase [G] up to some value then increase $[\mathrm{G}^*]$ up to a value much greater than [G] then decrease [G] to zero, then decrease $[\mathrm{G}^*]$ to zero.

There are no reactions that convert the directional molecule to the H state in equation A.65 because this machine does not have a halting state. This lack of a halting state is another way in which this machine differs from the standard definition of a Turing machine.

This scheme uses a state molecule with two states, a directional molecule with two states, a symbol molecule for each position on the main tape with three states, and a memory molecule at each position on the memory tape with 13 states. One pair of fuel molecule species is also needed (not including the stepping).

This naive scheme uses more memory states than necessary. Figure A.5 shows that the maximal in degree of any state is four so only M_1 to M_4 are needed to distinguish the previous state of the machine.

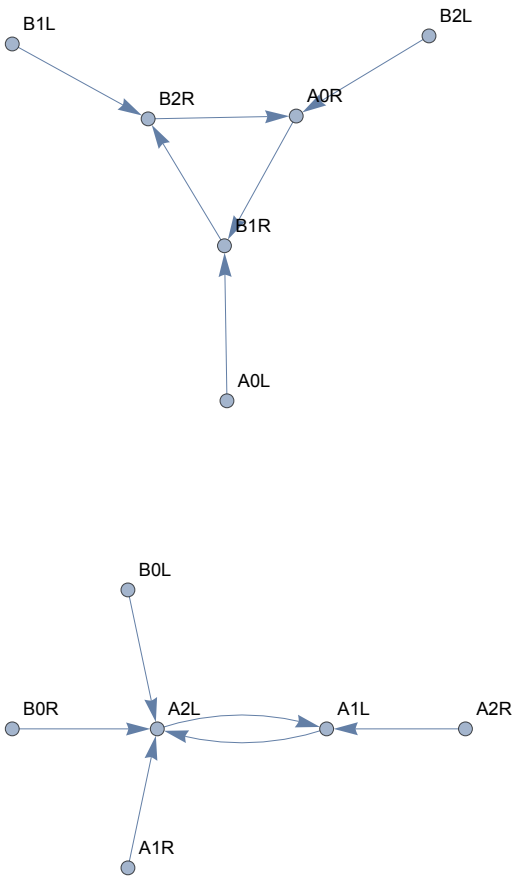


Figure A.5: The transitions in a single step of the 2-state 3-symbol machine. The largest in degree is four.

Therefore, we only need five states for the memory molecule and we can use the reactions:

$$\begin{aligned}
\mathrm{A} + \mathrm{L} + 0 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{B} + \mathrm{R} + 1 + \mathrm{M}_1 + \mathrm{G} \\
\mathrm{A} + \mathrm{R} + 0 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{B} + \mathrm{R} + 1 + \mathrm{M}_2 + \mathrm{G} \\
\mathrm{A} + \mathrm{L} + 1 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 2 + \mathrm{M}_1 + \mathrm{G} \\
\mathrm{A} + \mathrm{R} + 1 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 2 + \mathrm{M}_2 + \mathrm{G} \\
\mathrm{A} + \mathrm{L} + 2 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 1 + \mathrm{M}_1 + \mathrm{G} \\
\mathrm{A} + \mathrm{R} + 2 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 1 + \mathrm{M}_2 + \mathrm{G} \\
\mathrm{B} + \mathrm{L} + 0 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 2 + \mathrm{M}_3 + \mathrm{G} \\
\mathrm{B} + \mathrm{R} + 0 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 2 + \mathrm{M}_4 + \mathrm{G} \\
\mathrm{B} + \mathrm{L} + 1 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{B} + \mathrm{R} + 2 + \mathrm{M}_1 + \mathrm{G} \\
\mathrm{B} + \mathrm{R} + 1 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{B} + \mathrm{R} + 2 + \mathrm{M}_2 + \mathrm{G} \\
\mathrm{B} + \mathrm{L} + 2 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{R} + 0 + \mathrm{M}_1 + \mathrm{G} \\
\mathrm{B} + \mathrm{R} + 2 + \mathrm{M}_0 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{R} + 0 + \mathrm{M}_2 + \mathrm{G}.
\end{aligned} \tag{A.66}$$

This scheme uses a state molecule with two states, a directional molecule with two states, a symbol molecule for each position on the main tape with three states, a memory molecule at each position on the memory tape with five states, and one pair of fuel molecule species.

Instead of a single molecule with five states we can use three molecules each with two states. Of course, this leaves multiple unused states. Use one molecule for direction of previous step ($\mathrm{M_L}/\mathrm{M_R}$), one to distinguish the 2-to-1 rules ($\mathrm{M}_1/\mathrm{M}_2$) and one to give reaction direction/take place of hidden states ($\mathrm{M}_i/\mathrm{M}_f$). We are free to choose the initial state of the ($\mathrm{M_L}/\mathrm{M_R}$) and ($\mathrm{M}_1/\mathrm{M}_2$) molecules so we arbitrarily choose $\mathrm{M_L}$ and M_1.

Therefore, reactions become:

$$
\begin{aligned}
&\mathrm{A}+\mathrm{L}+0+\mathrm{M}_i+\mathrm{M}_\mathrm{L}+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{B}+\mathrm{R}+1+\mathrm{M}_f+\mathrm{M}_\mathrm{L}+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{A}+\mathrm{R}+0+\mathrm{M}_i+\mathrm{M}_\mathrm{L}+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{B}+\mathrm{R}+1+\mathrm{M}_f+\mathrm{M}_\mathrm{R}+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{A}+\mathrm{L}+1+\mathrm{M}_i+\mathrm{M}_\mathrm{L}+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{A}+\mathrm{L}+2+\mathrm{M}_f+\mathrm{M}_\mathrm{L}+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{A}+\mathrm{R}+1+\mathrm{M}_i+\mathrm{M}_\mathrm{L}+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{A}+\mathrm{L}+2+\mathrm{M}_f+\mathrm{M}_\mathrm{R}+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{A}+\mathrm{L}+2+\mathrm{M}_i+\mathrm{M}_\mathrm{L}+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{A}+\mathrm{L}+1+\mathrm{M}_f+\mathrm{M}_\mathrm{L}+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{A}+\mathrm{R}+2+\mathrm{M}_i+\mathrm{M}_\mathrm{L}+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{A}+\mathrm{L}+1+\mathrm{M}_f+\mathrm{M}_\mathrm{R}+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{B}+\mathrm{L}+0+\mathrm{M}_i+\mathrm{M}_\mathrm{L}+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{A}+\mathrm{L}+2+\mathrm{M}_f+\mathrm{M}_\mathrm{L}+\mathrm{M}_2+\mathrm{G}\\
&\mathrm{B}+\mathrm{R}+0+\mathrm{M}_i+\mathrm{M}_\mathrm{L}+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{A}+\mathrm{L}+2+\mathrm{M}_f+\mathrm{M}_\mathrm{R}+\mathrm{M}_2+\mathrm{G}\\
&\mathrm{B}+\mathrm{L}+1+\mathrm{M}_i+\mathrm{M}_\mathrm{L}+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{B}+\mathrm{R}+2+\mathrm{M}_f+\mathrm{M}_\mathrm{L}+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{B}+\mathrm{R}+1+\mathrm{M}_i+\mathrm{M}_\mathrm{L}+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{B}+\mathrm{R}+2+\mathrm{M}_f+\mathrm{M}_\mathrm{R}+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{B}+\mathrm{L}+2+\mathrm{M}_i+\mathrm{M}_\mathrm{L}+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{A}+\mathrm{R}+0+\mathrm{M}_f+\mathrm{M}_\mathrm{L}+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{B}+\mathrm{R}+2+\mathrm{M}_i+\mathrm{M}_\mathrm{L}+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{A}+\mathrm{R}+0+\mathrm{M}_f+\mathrm{M}_\mathrm{R}+\mathrm{M}_1+\mathrm{G}
\end{aligned}
\tag{A.67}
$$

Clearly, our trade-off for reducing the number of states is to have reactions of six molecules.

The situation can be improved because we do not need all of the molecules in all of the reactions. We only need the $\mathrm{M}_1/\mathrm{M}_2$ reaction in the rule that

needs it. Therefore, the reactions are:

$$\begin{aligned}
\mathrm{A} + \mathrm{L} + 0 + \mathrm{M}_i + \mathrm{M}_\mathrm{L} + \mathrm{G}^* &\rightleftharpoons \mathrm{B} + \mathrm{R} + 1 + \mathrm{M}_f + \mathrm{M}_\mathrm{L} + \mathrm{G} \\
\mathrm{A} + \mathrm{R} + 0 + \mathrm{M}_i + \mathrm{M}_\mathrm{L} + \mathrm{G}^* &\rightleftharpoons \mathrm{B} + \mathrm{R} + 1 + \mathrm{M}_f + \mathrm{M}_\mathrm{R} + \mathrm{G} \\
\mathrm{A} + \mathrm{L} + 1 + \mathrm{M}_i + \mathrm{M}_\mathrm{L} + \mathrm{M}_1 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 2 + \mathrm{M}_f + \mathrm{M}_\mathrm{L} + \mathrm{M}_1 + \mathrm{G} \\
\mathrm{A} + \mathrm{R} + 1 + \mathrm{M}_i + \mathrm{M}_\mathrm{L} + \mathrm{M}_1 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 2 + \mathrm{M}_f + \mathrm{M}_\mathrm{R} + \mathrm{M}_1 + \mathrm{G} \\
\mathrm{A} + \mathrm{L} + 2 + \mathrm{M}_i + \mathrm{M}_\mathrm{L} + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 1 + \mathrm{M}_f + \mathrm{M}_\mathrm{L} + \mathrm{G} \\
\mathrm{A} + \mathrm{R} + 2 + \mathrm{M}_i + \mathrm{M}_\mathrm{L} + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 1 + \mathrm{M}_f + \mathrm{M}_\mathrm{R} + \mathrm{G} \\
\mathrm{B} + \mathrm{L} + 0 + \mathrm{M}_i + \mathrm{M}_\mathrm{L} + \mathrm{M}_1 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 2 + \mathrm{M}_f + \mathrm{M}_\mathrm{L} + \mathrm{M}_2 + \mathrm{G} \\
\mathrm{B} + \mathrm{R} + 0 + \mathrm{M}_i + \mathrm{M}_\mathrm{L} + \mathrm{M}_1 + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{L} + 2 + \mathrm{M}_f + \mathrm{M}_\mathrm{R} + \mathrm{M}_2 + \mathrm{G} \\
\mathrm{B} + \mathrm{L} + 1 + \mathrm{M}_i + \mathrm{M}_\mathrm{L} + \mathrm{G}^* &\rightleftharpoons \mathrm{B} + \mathrm{R} + 2 + \mathrm{M}_f + \mathrm{M}_\mathrm{L} + \mathrm{G} \\
\mathrm{B} + \mathrm{R} + 1 + \mathrm{M}_i + \mathrm{M}_\mathrm{L} + \mathrm{G}^* &\rightleftharpoons \mathrm{B} + \mathrm{R} + 2 + \mathrm{M}_f + \mathrm{M}_\mathrm{R} + \mathrm{G} \\
\mathrm{B} + \mathrm{L} + 2 + \mathrm{M}_i + \mathrm{M}_\mathrm{L} + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{R} + 0 + \mathrm{M}_f + \mathrm{M}_\mathrm{L} + \mathrm{G} \\
\mathrm{B} + \mathrm{R} + 2 + \mathrm{M}_i + \mathrm{M}_\mathrm{L} + \mathrm{G}^* &\rightleftharpoons \mathrm{A} + \mathrm{R} + 0 + \mathrm{M}_f + \mathrm{M}_\mathrm{R} + \mathrm{G}. \quad \text{(A.68)}
\end{aligned}$$

A.11.2 4-state 6-symbol Machine

There is a tradeoff between the number of head states and the number of tape symbols[80]. The machine with the smallest number of different transiton rules, which correspond to chemical reactions, is the four state six symbol machine from Rogozhin[81]. The transitions for this machine are shown in table A.2

	A	B	C	D
0	A3L	B4R	C0R	D4R
1	A2R	C2L	D3R	B5L
2	A1L	B3R	C1R	D3R
3	A4R	B2L	H	H
4	A3L	B0L	A5R	B5L
5	D4R	B1R	A0R	D1R

Table A.2: The rules for Rogozhin's universal 4 state 6 symbol machine (with relabelled states and symbols).The letters are the states and the numbers are the symbols.

We see in figure A.6 that the largest in degree is 4 so only 5 memory states

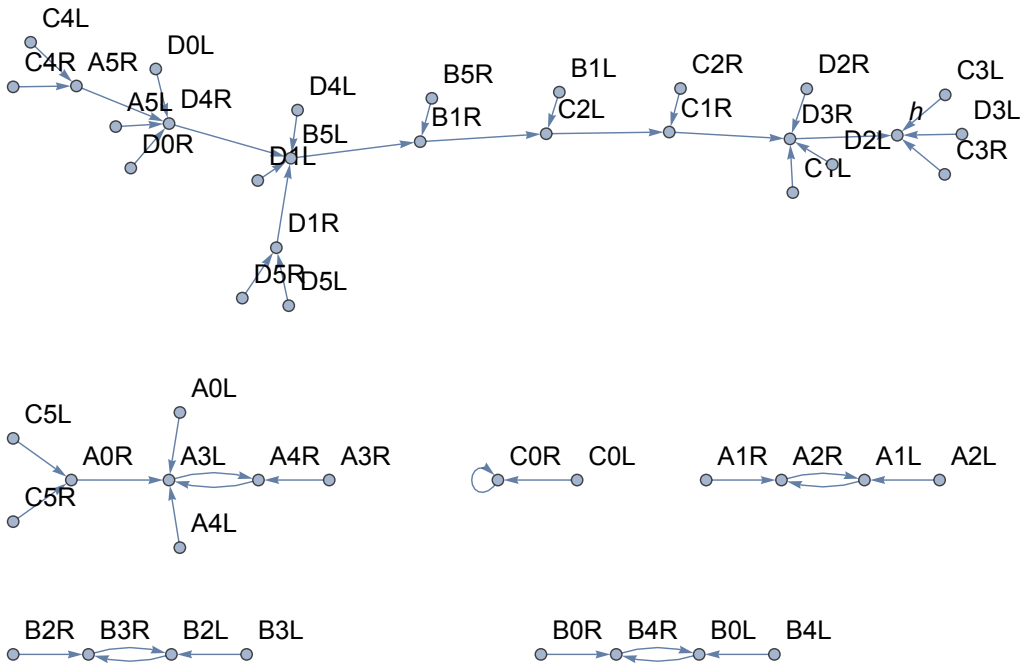


Figure A.6: Graph showing one step of Rogozhin's 4-state 6-symbol machine. There are 5 nodes with an in degree of 4.

are required. The reactions are:

$$
\begin{aligned}
&\mathrm{A}+0+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{A}+3+L+\mathrm{M}_1+\mathrm{G} && \mathrm{A}+3+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{A}+4+R+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{A}+0+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{A}+3+L+\mathrm{M}_2+\mathrm{G} && \mathrm{A}+3+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{A}+4+R+\mathrm{M}_2+\mathrm{G}\\
&\mathrm{B}+0+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{B}+4+R+\mathrm{M}_1+\mathrm{G} && \mathrm{B}+3+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{B}+2+L+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{B}+0+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{B}+4+R+\mathrm{M}_2+\mathrm{G} && \mathrm{B}+3+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{B}+2+L+\mathrm{M}_2+\mathrm{G}\\
&\mathrm{C}+0+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{C}+0+R+\mathrm{M}_1+\mathrm{G} && \mathrm{C}+3+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{C}+3+H+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{C}+0+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{C}+0+R+\mathrm{M}_2+\mathrm{G} && \mathrm{C}+3+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{C}+3+H+\mathrm{M}_2+\mathrm{G}\\
&\mathrm{D}+0+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{D}+4+R+\mathrm{M}_1+\mathrm{G} && \mathrm{D}+3+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{D}+3+H+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{D}+0+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{D}+4+R+\mathrm{M}_2+\mathrm{G} && \mathrm{D}+3+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{D}+3+H+\mathrm{M}_2+\mathrm{G}\\
&\mathrm{A}+1+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{A}+2+R+\mathrm{M}_1+\mathrm{G} && \mathrm{A}+4+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{A}+3+L+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{A}+1+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{A}+2+R+\mathrm{M}_2+\mathrm{G} && \mathrm{A}+4+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{A}+3+L+\mathrm{M}_2+\mathrm{G}\\
&\mathrm{B}+1+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{C}+2+L+\mathrm{M}_1+\mathrm{G} && \mathrm{B}+4+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{B}+0+L+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{B}+1+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{C}+2+L+\mathrm{M}_2+\mathrm{G} && \mathrm{B}+4+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{B}+0+L+\mathrm{M}_2+\mathrm{G}\\
&\mathrm{C}+1+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{D}+3+R+\mathrm{M}_1+\mathrm{G} && \mathrm{C}+4+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{A}+5+R+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{C}+1+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{D}+3+R+\mathrm{M}_2+\mathrm{G} && \mathrm{C}+4+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{A}+5+R+\mathrm{M}_2+\mathrm{G}\\
&\mathrm{D}+1+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{B}+5+L+\mathrm{M}_1+\mathrm{G} && \mathrm{D}+4+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{B}+5+L+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{D}+1+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{B}+5+L+\mathrm{M}_2+\mathrm{G} && \mathrm{D}+4+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{B}+5+L+\mathrm{M}_2+\mathrm{G}\\
&\mathrm{A}+2+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{A}+1+L+\mathrm{M}_1+\mathrm{G} && \mathrm{A}+5+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{D}+4+R+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{A}+2+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{A}+1+L+\mathrm{M}_2+\mathrm{G} && \mathrm{A}+5+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{D}+4+R+\mathrm{M}_2+\mathrm{G}\\
&\mathrm{B}+2+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{B}+3+R+\mathrm{M}_1+\mathrm{G} && \mathrm{B}+5+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{B}+1+R+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{B}+2+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{B}+3+R+\mathrm{M}_2+\mathrm{G} && \mathrm{B}+5+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{B}+1+R+\mathrm{M}_2+\mathrm{G}\\
&\mathrm{C}+2+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{C}+1+R+\mathrm{M}_1+\mathrm{G} && \mathrm{C}+5+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{A}+0+R+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{C}+2+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{C}+1+R+\mathrm{M}_2+\mathrm{G} && \mathrm{C}+5+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{A}+0+R+\mathrm{M}_2+\mathrm{G}\\
&\mathrm{D}+2+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{D}+3+R+\mathrm{M}_1+\mathrm{G} && \mathrm{D}+5+\mathrm{L}+\mathrm{M}_0+\mathrm{M}_1+\mathrm{G}^* \rightleftharpoons \mathrm{D}+1+R+\mathrm{M}_1+\mathrm{G}\\
&\mathrm{D}+2+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{D}+3+R+\mathrm{M}_2+\mathrm{G} && \mathrm{D}+5+\mathrm{R}+\mathrm{M}_0+\mathrm{M}_2+\mathrm{G}^* \rightleftharpoons \mathrm{D}+1+R+\mathrm{M}_2+\mathrm{G}
\end{aligned}
\tag{A.69}
$$