Citation for published version


DOI

Link to record in KAR

http://kar.kent.ac.uk/67363/

Document Version

Pre-print

Copyright & reuse
Content in the Kent Academic Repository is made available for research purposes. Unless otherwise stated all content is protected by copyright and in the absence of an open licence (eg Creative Commons), permissions for further reuse of content should be sought from the publisher, author or other copyright holder.

Versions of research
The version in the Kent Academic Repository may differ from the final published version. Users are advised to check http://kar.kent.ac.uk for the status of the paper. Users should always cite the published version of record.

Enquiries
For any further enquiries regarding the licence status of this document, please contact:
researchsupport@kent.ac.uk

If you believe this document infringes copyright then please contact the KAR admin team with the take-down information provided at http://kar.kent.ac.uk/contact.html
A thermodynamically consistent model of finite state machines

Dominique Chu
School of Computing
University of Kent
CT2 7NF, Canterbury
UK

Richard E. Spinney
Centre for Complex Systems
The University of Sydney
Sydney NSW 2006
Australia

June 14, 2018

Abstract

It is now a consensus opinion that there is no minimal thermodynamic cost to computation. However, truly
cost-free computation appears to require quasi-static processes. Practically useful computation, however, should
complete within a finite time and return the correct result with a probability close to 1. The cost of such finite-
time, quasi-deterministic remains an open question. In this contribution we address this by linking stochastic
thermodynamics with concepts from theoretical computer science. We propose a thermodynamically consistent
model of a finite state machines (FSM) modelled as a stochastic (Markovian) system. FSMs are non-universal
but practically and theoretically important model of computation. Unlike the better known Turing Machines,
FSMs are finite and can be modelled exhaustively. We calculate the entropy production of the machine, its error
probability, and the time required for a computation.

1 Introduction

Whenever a computation is done in the real world, some corresponding physical change to the world occurs (in
Landauer’s words: “all computation is physical”). This change can be that work was performed on a system, or
entropy produced or some heat generated. Attempts to understand the physical consequences of computation go
back to pioneering work in the 1980s. A key result that was established then is the celebrated Landauer limit. It
postulates that writing a bit to a memory device decreases the entropy of the memory by $k_B \ln(2)$ and thus requires
at least an amount of $k_B T \ln(2)$ work to be done. Yet, this Landauer limit does not fundamentally bound the cost of
computation. As was shown by Bennett [1,2] it is possible to conceive systems that compute with no lower limits on
the thermodynamic cost. The caveat is that all zero-cost computations proposed so-far either require infinite time
to complete or return a wrong results with a high probability. Both characteristics are unacceptable for real-world
computations. The cost of computations that complete within finite time and return the correct result with high
probability remains an open question.

Relatively recent frameworks that connect information and physics in a rigorous way are stochastic thermo-
dynamics [3] and information thermodynamics [4]. They have been applied successfully in theoretical studies on
information processing in biological systems [5–9] and also to probe the physical limits of “computational processes”
such as bit-rewriting or copying [10]. Intuitively, it appears clear that information processing in symbolic computers
must be limited by similar constraints as biological information processors. In reality, there are relatively few recent
attempts to apply statistical mechanics directly to concepts from computer science. Most attempts to do so [11,12]
have considered the zero-dissipation limit, thus not addressing how the cost of a computation is related to its execu-
tion time and accuracy. In the present contribution we now wish to probe the thermodynamic cost of computations
that complete within a finite time and return the correct result with a probability close to 1. More specifically, we will
not be content with considering proto-computations, or computational processes, but we will insist on staying close
to computational theory, so as to understand the resource requirements that are implied by models of computations
as they are formulated in computer science. Specifically, we will consider a particular model, namely finite state
machines (FSM).

FSMs and other models of computation are of great theoretical significance in computer science. These models
are mathematically rigorous descriptions of automated mechanisms of symbolic reasoning. There is not just one
such model, but many different ones. A fundamental result in theoretical computer science is that there is a class of
“Turing complete” or “universal” models that are (i) equivalent to one another and (ii) that are more general than
the models to which they are not equivalent. Equivalence means that anything that can be represented in one model can be translated into the other and vice versa. Turing complete models can represent non-Turing complete models as well, but the reverse is not true.

A well known example of a complete model of computation are Turing machines. A Turing machine is a formal construct that consists of a reading head. At any one time the reading head is in one of a finite number of possible states. It can also read and write symbols to a tape and move the tape. The allowed symbol set that the machine can read from and write to the tape is also finite. However, it is essential to the universality of the Turing machine that the tape to which it writes is unlimited, i.e. that the machine has an infinite amount of temporary memory at its disposal. This requirement for infinite size makes them rather bad models of necessarily finite real-world computers.

FSMs, which are the focus of this contribution, are a non-universal model of computation but they are finite. It is therefore possible to formulate thermodynamically consistent models of an entire FSM, calculate its energy dissipation and the time required to complete a computation, without assuming unlimited resources. Arguably, FSMs are more credible than Turing machines as models of actual computers, which are similarly limited in their computational power by their finiteness. There has also been a number of attempts to realise FSMs in biochemistry in order to understand how biological systems compute [13–15].

FSMs consist of an arbitrary but finite number of internal states and a finite input tape that delivers a sequence of symbols drawn from a finite alphabet [16]; see fig. 3 for a graphical depiction of a particular FSM. Formally, they can be defined as follows:

**Definition 1 (FSM).** A FSM is a tuple $(Q, \Sigma, \delta, I, F)$, where $Q$ is a finite set of internal states, $\Sigma$ represents a finite alphabet of $n$ different symbols, $\delta$ is a transition function $Q \times \Sigma \rightarrow Q$ and $I$ and $F$ are a set of initial states and accepting states respectively.

The operation of an FSM is as follows: At time $t = 0$ the machine is in a designated start state and in contact with the first symbol on the tape, which will be one of $n$ possible symbols that constitute the tape of length $L$. The transition function $\delta$ defines the next state for every state-symbol pair. State transitions always coincide with moving the tape ahead, such that the FSM is in contact with the next symbol on the tape when reaching the new state. The machine is updated until the end of the tape is reached when the computation halts. If the final state of the machine is a designated “accepting” state, then the input tape is deemed as “accepted,” otherwise it is not.

Computer scientists distinguish between deterministic and non-deterministic FSMs. A deterministic FSM is one where for each state there is at most one possible transition for each possible tape symbol, i.e. given a tape symbol the next state is determined. Non-deterministic FSMs, on the other hand, may have several possible transitions for each state/symbol pair. It is a well known result that deterministic and non-deterministic FSMs are computationally equivalent, in the sense that the two classes can solve the same problems. Henceforth, we will only consider deterministic FSMs.

One attraction of studying FSMs is that each state transition can be understood as a unit of computation, or an elementary cycle as we will refer to it. Once we know the thermodynamics of such an elementary unit, we know the thermodynamics of any conceivable FSM because any FSM is just a sequence of those elementary cycles. Furthermore, it will become evident that these elementary cycles can themselves be further broken down into more fundamental computational steps. We will find that only two types of computational steps are required to implement any FSM, namely bit-flips and a slight generalisation thereof, namely $n$-it setters.

The main results of this article are: (i) We show that an elementary cycle of a FSM can be implemented as a sequence of $n$-it setters. This implies that these $n$-it setters, even though they are extremely simple physical mechanisms, are sufficient for a very large class of computational problems. (ii) We present a general scheme to translate any specification of an FSM into a thermodynamically consistent model with a defined thermodynamic cost, and operating accuracy and execution time. (iii) We derive a relationship between the cost and error probability per state transition of the FSM and show that in the limit of high accuracy the resource consumption of an FSM when implemented as a Markovian system, becomes independent of the size of the tape alphabet.

## 2 Results

### 2.1 Stochastic information processors

Before describing a physically plausible model of FSMs, we will review briefly the thermodynamics of bit-flips, bit-setters and a slight generalisation thereof, namely $n$-it setters. These will be the fundamental building blocks from which arbitrary FSMs can be built.

We assume here a model of the bit-flip as a two state system, where each state is associated with an energy level, $E = \{0, -\Delta E\}$ and $\Delta E > 0$. The system is also in contact with an external heat-bath that is kept at some
constant temperature $T$. If at time $t = 0$ the system is reset to state 1 (associated with energy $E_1 = 0$) and the energy of state 2 is $E_2 = -\Delta E$ then the system will eventually transition to state 2 and stay there with probability $\pi_2 = \exp(\Delta E/k_B T)/(1 + \exp(\Delta E/k_B T))$. We assume that there are two possible transition rates between states, namely $k_+$ and $k_-$ in the forward and backward direction respectively. We also postulate that the system obeys local detailed balance, i.e. $k_+/k_- = \exp(\Delta E/k T)$. The total entropy production of the bit-flip can then be expressed in terms of the transition rates. Assuming the bit was initialised at time $t = 0$ in the high energy state, then the total entropy change will be the average entropy change of the environment and the system:

$$\Delta S_{\text{tot}} = S_{\text{env}} + S_{\text{sys}} = \frac{k_+}{A} \ln \left( \frac{k_+}{k_-} \right) - \left[ \frac{k_+}{A} \ln \left( \frac{k_+}{A} \right) + \frac{k_-}{A} \ln \left( \frac{k_-}{A} \right) \right] = -\ln \left( \frac{k_-}{A} \right)$$

(1)

Here $A := k_+ + k_-$. In the case of $k_+ = k_-$, corresponding to $\Delta E = 0$, we find that, as expected, the entropy produced is $\Delta S = -\ln(1/2)$. In this case there is no entropy change to the environment, and all entropy production is due to randomisation of the bit, i.e. (Shannon) entropy change of the system. The opposite extreme of this simple system is the case where $k_- \ll k_+$ when the diverging entropy change of the environment dominates the entropy production. In this limit the entropy change of the system tends to zero.

The basic bit-flip can be extended to a $n$-it setter by adding $n$ additional states, such that the system contains an initial state (that we keep at energy $E = 0$) (state 1) from which it can access states 2, 3, 4, ..., $n + 2$. We choose state 2 to be the desired state, which will have energy $E_2 = -\Delta E$; states 3, 4, ..., $n + 2$ are distractor states with energy $E_D = \Delta E$. The only allowed transitions are from the initial state 1 to all other states and back; see fig. 1. The $n$-it setter can be interpreted as a device that starts in some fixed state, and then chooses the specific state 2 among $n + 1$ possible different states. How reliably it sets the state depends on the energy difference $\Delta E$ between states.

Mathematically, the $n$-it setter, as presented here, can be modelled as a continuous time Markov chains with $n + 2$ states. The simplest case is when we set $n = 1$. In this case state 3 is the only distractor state with energy $E_D = \Delta E$ and state 2 is the desired state with energy $E_2 = -\Delta E$. We assume that $\Delta E$ is chosen such that the rate from state 1 to state 2 is $k_+$; the reverse rate from state 2 to state 1 is $k_-$. Similarly, the rate from state 1 to 3 is $k_-$ and the reverse is $k_+$. In equilibrium this system obeys detailed balance $k_+ \pi_1 = k_- \pi_2$ and $k_- \pi_1 = k_+ \pi_3$, which leads to the steady-state probabilities,

$$\pi_1 = \frac{k_- k_+}{A}, \quad \pi_2 = \frac{k_+^2}{A}, \quad \pi_3 = \frac{k_-^2}{A}$$

(2)

Here $A := (k_-)^2 + k_- k_+ + (k_+)^2$.

This system can be extended to the case where there are $n$ distractor states, i.e. states 3, ..., $(2 + n)$. In this case the detailed balance conditions are $k_+ \pi_1 = k_- \pi_2$ and $nk_- \pi_1 = k_+ \pi_D$, where $\pi_D$ is the probability to be in one of the distractor states (and the probability to be in any specific distractor state is $\pi_D/n$). A solution for this is again $\pi_D = n(k_-)^2$, but with a modified $A := n(k_-)^2 + k_- k_+ + (k_+)^2$. Assuming that the system was initialised in the first state, the change of the system entropy during the relaxation to equilibrium is $S_{\text{sys}} = -\pi_1 \ln(\pi_1) - \pi_2 \ln(\pi_2) - \pi_D \ln(\pi_D/n)$; note here the extra $n$ in the logarithm but not in the pre-factor. $S_{\text{sys}}$ takes a maximum for $k_+ = k_-$, in which case it is $-\ln(n)$. Note that this is (up to the sign) also the generalised Landauer limit for setting a system into 1 specific state out of $n$ possible states, i.e. the minimal work required to do this is $k_B T \ln(n)$.
For our system, however, running the n-it setter with high accuracy means that \( \pi_2 \approx 1 \) which is achieved when \( k_+ \gg k_- \). In this case the entropy change of the system \( S_{\text{sys}} \approx 0 \) whereas the entropy change of the environment dominates the total entropy production:

\[
S_{\text{env}} = \pi_2 \ln \left( \frac{k_+}{k_-} \right) + \pi_D \ln \left( \frac{k_-}{k_+} \right) = \frac{(k_+^2 - nk_-)}{(k_+^2 + k_+k_- + nk_-^2)} \ln \left( \frac{k_+}{k_-} \right) \tag{3}
\]

An interesting feature of this model is that in the high accuracy limit, i.e. when \( k_+ \gg nk_- \), both \( S_{\text{env}} \) and the probability that the n-it setter is in the intended state \( \pi_2 \) become independent of \( n \). So, setting a bit, or trit or any other n-it all has approximately the same cost for as long as we insist on high accuracy; see fig. 2.

Indeed, we now show that as \( n \) increases, the n-it setter operates increasingly close to the fundamental limit. To see this, we first consider again the case of \( n = 1 \), i.e. the bit-setter whose lower bound of operation is the (Landauer) cost of \( k_B \ln(2) \). In contrast, the Markovian system we consider here produces much more entropy than that. Indeed, in order to set a bit without error, the entropy production would be infinite. If one is prepared to accept that the bit is set incorrectly with a small probability, then the entropy produced is finite, but much higher than the Landauer limit. The same is true for \( n = 2 \), when the system sets a trit, i.e. a basic unit in the ternary numeral system. The generalised Landauer limit in this case is \( k_B \ln(3) \). Again, the Markov trit-setter will produce more entropy than that, at least if one insists that \( \pi_2 \approx 1 \).

We can now repeat this comparison for n-it setters with \( n > 2 \) and compare them to the generalised Landauer limit to set one of \( n + 1 \) states, which is \( k_B \ln(n+1) \). First, we set \( k_- = k_B = 1 \) (for simplicity) and determine the rate \( k_+ \) that would result in the n-it setter dissipating exactly the same amount of entropy as the generalised Landauer limit postulates, by solving

\[
S_{\text{sys}} + S_{\text{env}} = \ln(n+1) \tag{4}
\]

for \( k_+ \). This yields

\[
k_+^{\text{Lan}} = \frac{2n}{n + \sqrt{n^2 - 4n}}. \tag{5}
\]

Note that this is only meaningful for \( n \geq 2 \). The \( \ominus \) solution is the relevant one and gives the forward bias that would lead to an average entropy production that corresponds to the generalised Landauer limit. Substituting this into the expression for the probability that the n-it setter is in the desired state 2 then gives

\[
\pi_2|_{k_+ = k_+^{\text{Lan}}, k_- = 1} = \frac{2n}{(n+1)(n - \sqrt{(n-4)n})}. \tag{6}
\]

This can be seen to approach 1 for \( n \gg 1 \) and \( k_+ \gg n \). It means that a Markovian n-it setter can choose a state at a cost close to the generalised Landauer limit with increasing accuracy when \( n \) is increased — it operates increasingly close to the fundamental limit.

### 2.2 Time required to set a n-it

The model of the n-it setter can be solved for time, which yields a characteristic time \( \tau \) to reach the equilibrium given by

\[
\tau_{n-\text{it}}^{-1} = \frac{k_+ (n+1)}{2} - \sqrt{\frac{k_+ ((n-1)^2 k_+ + 4 k_-)}{2} + k_-}. \tag{7}
\]

In the high accuracy limit, \( k_+ \gg nk_- \), \( \tau \) scales like \( 1/k_+ \). This means that the n-setter operates fast when it operates accurately.

The characteristic time for n-it setters to reach their equilibrium will determine the operating time of the FSM. In a strict sense, the system only reaches its equilibrium after an infinite time. Yet, equilibrium is approached exponentially fast, thus, in practice it will be sufficient to wait for a few multiples of \( \tau \) in order to be so close to the true equilibrium that for all practical purposes the probabilities of the system are indistinguishable from an equilibrium system. If we decide to let the system relax for given multiples of \( \tau \), say 10, before the system is measured or further manipulated, then the Markov n-it setter is operated in a defined finite time.
For high values of $n$ the $n$-it setter produces a large amount of system entropy when $k_+ \approx k_-$, resulting in a low probability of the system to return the correct result. As the value of $k_+$ increases, the system produces heat and the accuracy of the computation increases.

2.3 A physical model of FSMs

We will now turn to constructing a thermodynamically consistent model of FSMs. We will show that arbitrary FSMs can be constructed using a series of bit-flips followed by an $(n-1)$-it setter. FSMs are time-inhomogeneous Markov chains, in that the transition rates between 2 states depend on the current tape-symbol. Therefore, any physical implementation of an FSM also requires an imposed protocol that implements the rates. Before describing the model in section 2.4, we will motivate our modelling choices in a step by step manner.

We start by re-interpreting the above defined logical model of the FSM as an inhomogeneous, continuous time Markov chain $M$. We will then find that this naive translation of the conceptual FSM is not sufficient to implement the required mechanisms as a credible physical system. Such a system requires additional states.

We first translate the FSM directly into a Markov chain model $M$:

1. The states and transitions of $M$ are the same as the states and transitions of the FSM.
2. Every transition $t_{ij}$ from some state $i$ to some state $j$ is (conceptually) associated with a symbol $\alpha$ of the input alphabet.
3. Each time an internal state transition happens, the external tape moves forward and the machine is in contact with the next tape element which contains a (possibly different) symbol $\alpha$.
4. When the current tape symbol is $\alpha$ then the transition rates are $k_{ij}\alpha' = \delta_{\alpha'\alpha}$, i.e. they are 0 except for those associated with the current tape symbol $\alpha$, which are 1.

We now further re-interpret this model as a microscopic system in contact with some heat-bath. Again postulating local detailed balance, $k_{ij}/k_{ji} = \exp((E_i - E_j)/k_BT)$, we obtain a relationship between the transition rates from state $i$ and $j$ and the respective energies $E_i$ and $E_j$. Note, that this entails that if the forward transition $i \rightarrow j$ happens, then the reverse transition $j \rightarrow i$ may also take place, albeit with a possibly very small rate. In the original FSM model reverse transitions were not allowed. Introducing this micro-reversibility is an immediate consequence of thermodynamic consistency.

With thermodynamic consistency restored, the model still remains incomplete. One feature of the conceptual model of FSMs is that the internal state transitions are synchronised with moving the tape ahead. In the context of our interpretation here, this is no longer possible. The transition rates are determined by energy differences between states. In order to achieve correct transition rates, energy differences must be changed before a transition happens. This means that it is not possible that the tape moves at the same time as the transition happens. We will therefore require that the tape moves before the relevant state transition is taking place, but after the previous transition is completed. To achieve this, we need to add extra states to the model. Specifically, we duplicate each logical state of the FSM into two physical states, which we will call the label state and the transition state respectively. The transition from the label state to the transition state serves only one purpose, namely to move the tape ahead.

Before we arrive at a complete model of the FSM further states need to be introduced: Transitions between states need to be forced by energy differences between the states, i.e. desired next states should have a low energy and
all other states a high energy. In general it is not possible to use this scheme to transition from transition states to label states in this way. To see this consider the specific transition from state $i$ to state $j$. Assume that for symbol $\alpha$ this transition should happen at a low rate $k_{-}$, i.e. $\alpha$ is associated with a transition from $i$ to a different state $l$. Consequently, the energy of state $j$ should be high.

At the same time, it may be the case that the tape symbol $\alpha$ is associated with the transition from a different state $k \neq i$ to $j$, i.e. this transition should happen with a rate $k_{+}$. Accordingly, the energy of $j$ should be low. In general, it is therefore not possible to achieve consistent transition rates by adjusting the energy levels of label states. This can be overcome by introducing a new state for each transition. We call it the dynamic state. Altogether we thus add another $n$ states for each label state; see fig. 3.

We summarise the procedure to transform an FSM into a physically plausible Markov model:

1. Split every internal state of the FSM into two states, designating one of them as a label state and the other as the transition state.

2. Make a connection from every label state to its corresponding transition state.

3. For each transition state, introduce another $n$ states — the dynamic states. Each is associated with a different symbol of the tape alphabet.

4. Introduce possible transitions between labelled states and transition states via a dynamic state. The transitions should be as specified by the FSM model: If the FSM specifies that there is a transition from state $i$ to state $j$ for a tape symbol $\alpha$, then the model should contain reversible transitions from the label state $i$ to the transition state $i$ to a dynamic state $i\alpha$ to the label state $j$.

Figure 3 illustrates the procedure with a specific example.

2.4 Detailed description of the model

In order to implement any FSM it is sufficient to postulate three energy levels, corresponding to $E = \{\pm \Delta E, 0\}$. We choose that the transitions states always remain at $E = 0$. Label states are moved between $E = 0$ and $E = \Delta E$, and the dynamic states are at $E = \pm \Delta E$. An elementary cycle is the procedure that needs to be executed in order to implement a single state transition of a FSM. This elementary cycle is generic in that it can be used in any network topology and for any number of tape symbols; fig 4 illustrates an elementary cycle for a tape consisting of 2 symbols.
2.4.1 Outline of the elementary cycle

The elementary cycle consists of altogether 3 atomic computational steps, namely 2 bit-flips followed by the execution of an \((n - 1)-it\) setter, where \(n\) is the size of the tape alphabet. Firstly, an initial bit-flip moves the machine from a dynamic state from the previous elementary cycle into the label state of the current cycle; this first bit-flip corresponds to step 1 in fig 4. Next, the machine transitions from a label state to the associated transition state while moving the tape ahead and adjusting the energy landscape that modifies the transition rates. This is the second bit-flip. Finally, this is followed by a jump from the transition state to a dynamic state, which corresponds to setting a \((n - 1)-it\). These latter two atomic computations are combined during step 3 in fig 4.

The machine operates over three fixed energy levels. Continuous operation is therefore only possible if, through input of work, the energy levels of states are lifted to drive the computations. We will now describe a generic protocol so as to realise an arbitrary FSM. Note that the execution of this protocol does not depend on knowledge of the state of the machine at a particular time.

2.4.2 Step by step procedure

Assume that the machine starts at a point of the cycle where the label and transition states have an associated energy \(E = 0\). The machine is in a dynamic state, with associated energy \(E = -\Delta E\); see fig. 4. In addition to the energy differences, there are also energy barriers between transition states and dynamic states. We assume that the heights of these barriers are effectively infinite, such that over relevant time scales there will be no transition over these barriers.

**Step 1** Lift all dynamic states to energy \(E = +\Delta E\). This will now implement the first bit-flip. The intended state of the machine after the first step is the next label state. Wait for a chosen multiple of \(\tau_{hit} = (k_+ + k_-)^{-1}\) before proceeding to the next step. **Step 2** Existing energy barriers are removed and instead new energy barriers between dynamic and label states are erected. **Step 3** All dynamic states are lifted to energy \(E = +\Delta E\). At this point the machine state proceeds to the transition state; this is coupled to moving the tape ahead and results in an instantaneous modification of the energy levels of the dynamic states in accordance with the new tape symbol. After this modification, the dynamic states associated with the current tape symbol are lowered to the energy \(E = -\Delta E\). Altogether, this step corresponds to a bit-flip followed by setting a \((n - 1)-it\). Wait for a given multiple of \(\tau_{hit} + \tau_{n-it}\) before proceeding to the next step. The time \(\tau_{n-it}\) is given in eq. 7. **Step 4** Existing barriers are removed and new barriers are erected between the label states and the dynamic states. **Step 5** The energy of the label states are lowered to \(E = 0\). This takes the machine back to the beginning of a new elementary cycle.

2.5 Energetics of the FSM

The computation is driven by lifting the energy levels of the machine during steps 1 and 3. This requires work. During step 1 the dynamic state is lifted from \(E = -\Delta E\) to \(E = \Delta E\). Recalling that \(\Delta E = k_B\ln(k_+/k_-)\) the amount of work required to perform this step can be expressed in terms of the transition rates between the dynamic state and the index state.

\[
\Delta W_1 = 2k_B\ln\left(\frac{k_+}{k_-}\right)
\]  

(8)

The average amount of work during this step is given by \(\langle \Delta W_1 \rangle = \pi_d \Delta W_1\), where \(\pi_d\) is the equilibrium probability to find the machine in the dynamic state before the energy has been lifted.

\[
\pi_d = \frac{k_+}{2k_- + k_+}
\]  

(9)

The second contribution to the work arises during the third step when the label states are lifted. In this case the work required when the machine is in a label state is

\[
\Delta W_2 = k_B\ln\left(\frac{k_+}{k_-}\right)
\]  

(10)

The average amount of work during this step is given by \(\langle \Delta W_2 \rangle = \pi_l \Delta W_2\), where \(\pi_l\) is , the equilibrium probability to find the machine in the label state before the energy has been lifted.

\[
\pi_l = \frac{k_+}{nk_- + 2k_+}
\]  

(11)

Here \(n\) is the size of the alphabet.
Figure 4: Schematic outline of an elementary cycle; see main text for an explanation for each step. The procedure is followed along the large grey arrows. Black circles indicate labelled states, grey circles transition states and grey squares are dynamic states. The dashed horizontal lines indicate the three energy levels of the states, which are \{0, \pm \Delta E\}. Transitions are only annotated in the first diagram (for clarity). In each of the six diagrams, the solid arrows indicate the intended path of computation, whereas the dashed arrows are paths that should not be followed. These are curtailed for clarity. The energy barriers are symbolised by double vertical short lines. The input tape is shown with the current symbol being marked in red. Similarly, the current intended state of the FSM is marked in red.

Altogether, the average work performed during one elementary cycle is \( \langle W \rangle = \langle \Delta W_1 \rangle + \langle \Delta W_2 \rangle \).

\[
\langle W \rangle = \frac{k_+ (2nk_+ + 2k_+ + 5k_+)}{(2k_- + k_+)(nk_+ + 2k_+)} k_B T \ln \left( \frac{k_+}{k_-} \right)
\]

The second line contains an approximation, which is good for \( k_+ \gg nk_- \).

\[ k_- \approx \frac{5}{2} k_B T \ln \left( \frac{k_+}{k_-} \right) \] (12)

The second line contains an approximation, which is good for \( k_+ \gg nk_- \).

### 2.6 Error probabilities

The errors probability of the computation depends on the energy differences \( \Delta E \) between the states. The larger the difference, the smaller the error. However, the errors to the computation occur when the energy barriers are re-distributed. The re-distribution steps may fix the system in incorrect states and thus affects the reliability of the computation. When analysing the error probability of the machine, we only need to consider steps 2 and 4 where the barriers are re-distributed.

Errors arise during the second step. Before the re-distribution, the machine could be in the high energy dynamic state rather than the intended label state. This will then result in an error. The probability that the machine is in the high energy state and that an error occurs is:

\[ p_{err1} = \frac{k_-}{k_- + 2k_+} \] (13)
2.7 Time required for the computation

The time required for an elementary cycle to complete is spent while waiting for the system to approach equilibrium after Step 1 and Step 3. This time is $M(2\pi_{\text{bit}} + \pi_{n\text{-bit}})$, where $M$ is the chosen multiple of the relaxation time; see


3 Discussion

3.1 Energy dissipation of a FSM

We find for the FSM in the high accuracy regime that the probability of an error scales like $k_+^{-1}$ (see eq. 16), while the cost in terms of the work required to implement the cycle only increases logarithmically with $k_+$ (see eq. 12 and fig. 5).

Expressed in terms of the energy differences between states $\Delta E$, we find that the average work scales (expectedly) linear with $\Delta E$ but the error falls exponentially. This benign scaling echoes previous results on different models of computation (c.f. [17]). It tells us that perfect accuracy can only be reached in the limit of infinite energy dissipation, but quasi-deterministic computation that has practically negligible error probabilities can be done at a finite, even modest energy expense.

At the same time, it is clear that — especially for high accuracies — the model we proposed here dissipates well above the theoretical limit. The information processing of the FSM here is achieved by a series of $n$-it setters. As discussed in section 2.1, the Landauer cost of those systems scales with logarithmically with the symbol size. However, the Markovian $n$-it setter as presented here incurs a negligible extra cost when compared to the simple bit-setter. In the high accuracy limit, therefore the size of the input alphabet no longer matters to the cost of the computation; see fig. 3. However, per computational step more information can be processed when larger alphabets are considered, suggesting that it is more efficient to run Markovian computers with large alphabets than with small ones.

3.2 Programmability and entropy production

Another property of our model is that it entails a thermodynamic cost of programmability. The model proposed above requires 5 interventions per cycle, namely lifting/lowering of state-energies and erecting barriers. The procedure can be simplified when the model of the FSM is made less general, i.e. more assumptions are made about the type of computation (c.f. [17]). Computationally, nothing is gained by the procedure from, say, state $A$ to state $C$ when the current tape element is $\alpha$, then in the unrolled graph, this would entail a transition from $A_i^T$ to $A_i^{D,\alpha}$ to $C_{i+1}$. All states $A_i^{D,\alpha'}$ with $\alpha' \neq \alpha$ would be at a higher energy than $A_i^{D,\alpha}$. This unrolled transition graph would have many more states than the original model, but the total number of states and the total number of transitions to be completed for one computation are known. It is therefore possible to arrange the states over 3$L$ energy levels and it is no longer necessary to erect energy barriers, nor to lift the label states, which simplifies the computing protocol from 5 to 2 steps per elementary cycle.

As a further simplification consider the case where the precise symbol sequence of the tape, and not only its length are known. Then transition and dynamic states are dispensable and the machine can be reduced to state transitions between label states only. This is because once the input tape is known, the sequence of intended state transitions is known, and it is possible to reduce the transition graph to $L$ states arranged over $L$ energy levels. At this point a further reduction is possible. Computationally, nothing is gained by the $L - 2$ states between the initial and final state. Directly connecting the initial and final state leads to a logically equivalent chain. Hence, once the actual input tape is known in advance, it is possible to reduce the computation to a simple bit-flip. The cost of this operation is independent of the length of the tape.

3.3 Logical reversibility and the FSM

Physical limits on the energy usage of computations have now also caught the interest of computer scientists and computer engineers who try to tackle the immense practical problem of ever increasing power consumption of computer-related activities. In particular, hopes are placed on reversible architectures and programming languages [18,19] that
they will lead to more energy efficient computation. We suspect that these hopes will be disappointed. It is now well known that logical reversibility is not the same as thermodynamic reversibility [20]. This is also highlighted by our model. In FSMs logical reversibility is straightforward to ensure. All that is required for an FSM to be logically reversible is that it is deterministic and that from each state there is at most one (outgoing) transition for each symbol of the tape. It is a known result in computer science that for every FSM there is an equivalent logically reversible FSM [21].

At the same time, whether or not the FSM is logically reversible has little bearing on its energy consumption at least in our implementation of FSMs. Independently of the logic of the FSMs, the components on which our implementation relies, namely bit-flips and n-bit setters, are logically irreversible in the sense that they produce system entropy during their operation. Indeed, work needs to be invested in order to prevent the production of systems entropy during the operation of the n-bit setter. Logical reversibility is relevant for energy usage if one operates near the theoretical minimum. Real-world computers will normally operate in finite time and thus far away from the theoretical minimum. The Landauer limit is therefore not normally informative for their efficiency.

4 Acknowledgments

We thank Andre Barato for reading and commenting draft versions of this manuscript.

References


