I.4
Algorithmic methods

Throughout Chapters 14 and 15 we treat homogeneous one-dimensional independent operators. The number of states of a single automaton will be large but always finite. All the results of these chapters can be generalised trivially to the larger dimensions. All the results of these chapters have been published in [44,45] in slightly different formulations. One of these results has produced more interest than the others; it is the algorithmic unsolvability of the problem whether a homogeneous one-dimensional independent operator is ergodic, even if all the transitional probabilities are 0, 1, or $\frac{1}{2}$. We discuss it in Chapter 14. The other results are in Chapter 15. One of them is the possibility of recognition of any formal language with some specific one-dimensional homogeneous system.

All the proofs of these chapters use theoretical simulation of some Turing machines with our systems. A 'Turing machine' is a deterministic or stochastic (see [46]) one-tape machine.

Chapter 14
Algorithmic unsolvability of the ergodicity problem

The main result of this chapter, Theorem 14.1, can be formulated in various ways. We might take

$$U(h) = \{h-1,h,h+1\}$$

and any finite X_h. But it is more in line with our survey to take $X = \{0,1\}$ but any homogeneous neighbourhood system U. So we do.

Theorem 14.1 [44,45]. Consider the totality T of one-dimensional homogeneous independent operators with $X_h = \{0,1\}$,

$$U(h) = \{h_{-\ell},\ldots,h_{\ell}\},\ \ell \in \mathbb{Z}_+,$$

and transitional probabilities

$$\theta(x_h^{t+1}|x_{U(h)}^t) \in \{0,1,\tfrac{1}{2}\}.$$

There is no algorithm which would say whether any operator of T is ergodic or not.

Proof. Let K stand for the set of those Turing machines that never restore the void symbol λ on the tape and place special marks on the leftmost and rightmost cells where they have been. It is known that the problem of recognizing those Turing machines of K which ever stop, is unsolvable. We shall use this fact. For any Turing machine $M \in K$ we shall build in a constructive way some operator $P_M \in T$ which is ergodic if and only if M stops. This will prove our theorem.

So it remains to carry out the construction process. It starts with the construction of a deterministic operator D_M having

$$U(h) = \{h-1,h,h+1\}.$$

Let $\{\lambda,p_0,\ldots,p_n\}$ stand for M's set of symbols on the tape (where λ is the void symbol) and $\{q_0,q_1,\ldots,q_m\}$ stand for M's set of inner states (where q_0 is the initial state). Then the corresponding D_M has

$$X_0 = \{\lambda,p_0,\ldots,p_n,(\lambda,q_j),0\leqslant j\leqslant M,(p_i,q_j),0\leqslant i\leqslant n,0\leqslant j\leqslant m,+,-\}.$$

Here λ, p_0, ..., p_n imitate the tape cell without the head, (λ,q_j) and (p_j,q_j) imitate the tape cell with the head, and '+' and '−' are special extra symbols. The state '+' emerges if and only if the machine stops. This state is expansive: it remains and always turns its neighbours into pluses. In the absence of '+' the transitions of D_M imitate the work of M.

Note that the initial state may have many 'heads' (automata in the states imitating presence of a head). So the functioning of D_M imitates the work of many heads on the tape. But we take care against their interference: since the imitated Turing machine marks the leftmost and rightmost cells of its working zone, every head can register when it attempts to breach another head's zone. Having once done this, it blots out (that is, turns into λ) all the signs it has written (what the special sign '−' is used for), and after that it disappears.

We need another deterministic operator D'_M which has the same X and U but other transition function:

$$(xD'_M)_h = \begin{cases} + & \text{if there is a '+' among } x_{h-1},\, x_h,\, x_{h+1}, \\ (\lambda,q_0) & \text{if } \; x_{h-1} = x_h = x_{h+1} = \lambda, \\ \lambda & \text{in all other cases.} \end{cases}$$

Both deterministic operators D_M and D'_M may be considered as stochastic operators whose transitional probabilities equal 0 or 1 in the usual way. This allows us to introduce an operator W_M which has the same X and U and whose transitional probabilities are arithmetic means of those of D_M and D'_M. This W_M is ergodic if and only if M stops. Indeed, the measure concentrated in the state 'all pluses' is invariant for W_M.

Now let M stop. Note that D'_M generates heads spontaneously. These heads work and some of them will survive to stop with probability 1. Then + appears and expands to both sides, which means that the probability of + tends to 1.

Now let M never stop. Then the initial measure concentrated in the state 'all λ' will produce measures having no +. So, according to Corollary 2.8, there is an invariant measure in which the proportion of + is zero. So W_M has at least two different invariant measures. q.e.d.

But our theorem is not yet proved because we promised to restrict ourselves to the case $X_h = \{0;1\}$. So denote

$$\ell = \log_2|X_0| + 1.$$

We code all elements of X_0 with words $a_1, \ldots, a_\ell$ where $a_i \in \{0;1\}$. Every $a \in (X_0 \backslash \{+\})$ gets a code $\varphi(a)$ of the length $2\ell + 7$:

$$a_1 a_1 a_2 a_2 \ldots a_\ell a_\ell 0000010$$

where $a_1 a_2 \ldots a_\ell$ is the word corresponding to a. The code of + consists of $2\ell + 7$ zeros.

Now we define a mapping

$$\Phi\colon X^{\mathbb{Z}} \to \{0,1\}^{\mathbb{Z}}$$

by the rule

$$\Phi(\ldots,x_{-1},x_0,x_1,\ldots) = \ldots, \varphi(x_{-1}), \varphi(x_0), \varphi(x_1), \ldots.$$

Let Φ stand for the corresponding mapping of measures too.

The operator P_M will have

$$|U(h)| = 8\ell + 25.$$

We do not write down its transitional probabilities in full, but just claim the following:

(a) $\Phi(\mu W_M) = \Phi(\mu) P_M^2$
for any measure μ on $X_0^{\mathbb{Z}}$.
(b) If a state $y \in \{0,1\}^{\mathbb{Z}}$ has no $\Phi^{-1}(y)$, then P_M turns it into 'all zeros'.

Thus P_M^2 imitates W_M. If μ is invariant for W_M, then $\Phi(\mu)$ is invariant for P_M; hence non-ergodicity of W_M implies non-ergodicity of P_M. If W_M is ergodic, it converges to 'all +' and P_M converges to 'all 0'. Thus P_M is ergodic if and only if W_M is ergodic. q.e.d.

Note that the number 1/2 in the formulation of the theorem is not essential; it may be substituted by any pair α, $1 - \alpha$ where $0 < \alpha < \frac{1}{2}$.

Of course, many questions remain unsolved. These include:

(1) May 1/2 be omitted from the formulation of the theorem?
(2) Does a similar theorem hold for non-degenerate operators?

Chapter 15
Recognition of formal languages

As usual, an alphabet A is a finite non-empty set of symbols. We also have the void symbol λ, which is not an element of A. A word in A is a finite (perhaps, empty) sequence of symbols of A. Let A^* stand for the set of words in A. Any subset of A^* is a language in A.

Systems in this chapter are homogeneous in space as in the other chapters, but are not homogeneous in time. Transional probabilities here have a parameter which runs through the given alphabet A:

$$\theta(x_h^{t+1} | x_{U(h)}^t, a), \ a \in A. \tag{15.1}$$

Of course, the resulting operator P_a depends on $a \in A$ too and P_a with different $a \in A$ may be applied at different times $t = 1, 2, \ldots$. The initial measure is always concentrated in a certain initial state $x_h \equiv x_0$ where $x_0 \in X_0$. The resulting evolution measure $\tilde{\mu}$ is regarded as a function of the input sequence $a_1, a_2, a_3, \ldots$, members of which are those elements of A which served as parameters at times $t = 1, 2, 3, \ldots$. We say that a word $a_1, a_2 \ldots, a_m$ is fed into the system if the input sequence is $a_1, a_2 \ldots, a_m, \lambda, \lambda, \lambda, \ldots$. Elements of some proper subset $X_+ \subset X_0$ are termed yes-states, and a function $q\colon X_0 \to \{0,1\}$ is introduced:

$$q(x_0) = \begin{cases} 1 & \text{if } x_0 \in X_+ \\ 0 & \text{otherwise.} \end{cases}$$

Let

$$x = (x_h^t), \ h \in \mathbb{Z}, \ t \in \mathbb{Z}_+$$

stand for a realisation (or trajectory) of the system at all times. Introduce the yes-function

$$Q(t) = \lim_{n\to\infty} \sum_{h=-n}^{n} q(x_i^t).$$

It is easy to prove that this limit exists in our cases and equals $\tilde{\mu}(x_h^t \in X_+)$ which does not depend on $h \in \mathbb{Z}$. We are interested in dependence of $Q(t)$ on the input word W. So we shall write

$$Q(t) = Q_W(t).$$

We shall speak of language recognition. First take a finite automaton R with input that has input alphabet $A \cup \{\lambda\}$ with the fixed initial state and another fixed 'signal' state. Any word $a_1 \ldots a_m$ of A^* is said to be fed into R if the input state sequence is $a_1, \ldots a_m, \lambda, \lambda, \lambda, \ldots$. A language $L \subset A^*$ is said to be recognized by R if R gets into the signal state in those and only those cases when a word of L is fed into it. A language $L \subset A^*$ is said to be finite-automatic if some R recognizes L.

The following definition is an application of the same idea to the multicomponent case.

Definition 15.1 A system recognizes some language $L \subseteq A^*$ with parameters α and β, where $0 \leqslant \alpha \leqslant \beta \leqslant 1$ if the two following conditions hold:

(1) For any word W of L there is such t that $Q_W(t) > \beta$ or $Q_W(t) = 1$.
(2) For any word W in A which is not of L

$$Q_W(t) \leqslant \alpha \quad \text{and} \quad Q_W(t) < 1$$

at all times.

We term a language L α/β-recognizable if there is such X_0 and such transitional probabilities (15.1) that the resulting system recognizes L.

Proposition 15.2 There is a system which 0/1-recognizes the given language L and has all the transitional probabilities (15.1) equal to 0 or 1, if and only if the given language L is a finite-automatic language.

Proof is evident because in this deterministic case the automata cannot be in different states at one time, so the functioning of the system boils down to the functioning of one finite automaton with input. What we really want to say is stronger:

Theorem 15.3 A language is 0/1-recognizable if and only if it is a finite-automatic language [44,45]

Proof is based on Proposition 15.2.

The two following theorems use the notion of enumerability of a language, which is known in the theory of algorithms. Briefly, a language is termed enumerable if it is recursively enumerable when considered as a set, or if it is recognizable by some deterministic Turing machine. A Turing machine recognizes a language $L \subseteq A^*$ if, having started with any input word $W \in A^*$, it stops if and only if $W \in L$.

Theorem 15.4 A language is $0/\beta$-recognizable if and only if it is enumerable [44,45].

Sketch of a proof. For any enumerable language L we build a system which imitates the work of infinitely many copies of that Turing machine which recognizes L (like we built D in proving Theorem 14.1). The points imitating heads are chosen at random at $t = 0$. Those heads that do not have enough place to work destroy both their writings and themselves (if an emulation of a Turing machine does not stop, every head will ultimately destroy itself with probability 1). The only source of randomness in the system is the initial state. But the randomness of the initial state is essential.

For any $0/\beta$-recognizable language we prove its enumerability by looking through all possible (that is, having positive probability) variants of work of a segment of the system of length $2T + 1$ during T first steps of time, which is done for all natural T.

Theorem 15.5 If $\alpha > 0$, a language is α/β-recognizable if and only if it is enumerable [44,45].

Proof is like the proof of Theorem 15.4.

Of all the results of this chapter the following one was the most unexpected.

Theorem 15.6 There is a natural K (which is an absolute constant not exceeding a million) such that for any α and β, where $0 < \alpha \leqslant \beta < 1$ and for any alphabet A any language $L \subseteq A^*$ is α/β-recognized by some system in which every single automaton has K states [44,45].

Proof uses some ideas of [46]. It is cumbersome and we only sketch it here. It includes simulation by our systems of probabilistic Turing machines (about which see [46] in particular). Since $|A|$ may exceed K, the input word is remembered by the system in a statistical sense.

Elements of A have as counterparts not elements of K but different probabilities of writing 1 which moves in a certain direction (say, right) with speed 1. Thus the input word W is 'remembered' by a sequence of zeros and ones of the same length. Of course, one such sequence reminds of the input word only very slightly. But, since the word is 'written' in infinitely many places, the system as a whole 'knows' the word with probability 1. The language is 'known' by the system in the statistical sense too, and checking whether one belongs to the other is done in a statistical way.

Theorem 15.6 is possible because the segment (0,1) has the same continual cardinality as the set of all languages in a given alphabet. This is highlighted by the following.

Proposition 15.7 Let $0 < \beta - \alpha < 1$. A language L is α/β-recognized by some system which has rational transitional probabilities (15.1) if and only if L is enumerable.

Proof. Let $\beta - \alpha < 1$ and L be enumerable. Then its being α/β-recognized by some system with rational (15.1) parameters follows from Theorems 15.4 and 15.5.

Now let L be α/β-recognized by some system with rational (15.1) with $\alpha < 1$. In this system any $\tilde{\mu}(x_h^t = c)$ is a rational number which is defined as soon as the input word W is given.

Remember that a language L is called enumerable if there is such a Turing machine with input that stops if and only if the word fed into it belongs to L.

Now assume that a language L is α/β-recognizable. So there is a system with rational transitional probabilities which α/β-recognizes L. Take any rational $r \in (\alpha,\beta)$. The probability that some element of the system is in the yes-state at some moment, is a rational number which can be computed. Let us arrange such a Turing machine that computes this probability for all moments of time and stops if and only if this probability ever exceeds r. This Turing machine recognizes L.

Note that the functioning of the system constructed in the proof of Theorem 15.6 depends heavily on the exact values of its parameters and gets spoiled by arbitrarily small changing of them. To highlight this fragility we introduce the following definition and theorem.

Definition 15.8 A system α/β-recognizes a language L in the stable way if it α/β-recognizes it and still does so even if those of its transitional probabilities (15.1) that differ from 0 and 1 are modified within some non-zero range.

Theorem 15.9 Let $0 < \beta - \alpha < 1$. A language is α/β-recognized by some system in the stable way if and only if it is enumerable [44,45].

If $0 < \beta - \alpha < 1$ and a language is enumerable, its being α/β-recognizable in the stable way can be proved along with our proofs of

Theorems 15.4 and 15.5. The inverse statement follows from Proposition 15.7 and the fact that rational numbers are dense on (0,1).

Of course, our conceptions are not the only possible ones. Some other definitions of a language's being recognized by a system seem interesting too. For example, the value or existence of $\lim_{t \to \infty} \tilde{\mu}(x^t_h \in X_+)$ in the place of the inequalities of Definition 15.1 leads to something analogous to limit computability [20].

It seems appropriate to investigate the amount of time needed by a system to recognize a language in a stable way as depending on the language's complexity (see [79]). Shnirman has suggested making states of automata dependent on some parameter of $\tilde{\mu}$ at time t [72]. The very idea of combining the local interdependence, which is the main theme of this survey, with all elements' dependence on some global input has analogues in computer science, polymer chemistry, and even sociology.