J. C. Shepherdson*

The Reduction of Two-Way Automata To One-Way Automata

Rabin has proved[1,2] that two-way finite automata, which are allowed to move in both directions along their input tape, are equivalent to one-way automata as far as the classification of input tapes is concerned. Rabin's proof is rather complicated and consists in giving a method for the successive elimination of loops in the motion of the machine. The purpose of this note is to give a short, direct proof of the result.

The idea behind the proof is as follows: The only way an initial portion t of the input tape can influence the future behaviour of the two-way machine A when A is not actually scanning this portion of the tape is via the state transitions of A which it causes. The external effect of t is thus completely determined by the transition function, or "table," τ_t which gives (in addition to the state in which the machine originally exits from t), for each state s of A in which A might re-enter t, the corresponding state s' which A would be in when it left t again. This is all the information the machine can ever get about t however many times it comes back[3] to refer to t; so it is all the machine needs to remember about t. But there are only a finite number of different such transition tables (since the number of states of A is finite), so the machine has no need to use the input tape to supplement its own internal memory; a one-way machine $\bar{A}$ with a sufficiently large number of internal states could store the whole transition table τ_t of t as it moved forward, and would then have no need to reverse and refer back to t later. If we think of the different states which A could be in when it re-entered t as the different questions A could ask about t, and the corresponding states A would be in when it subsequently left A again, as the answers, then we can state the result more crudely and succinctly thus: A machine can spare itself the necessity of coming back to refer to a piece of tape t again, if, before it leaves t, it thinks of all the possible questions it might later come back and ask about t, answers these questions now and carries the table of question-answer combinations forward along the tape with it, altering the answers where necessary as it goes along.

In case this sketch of the proof leaves some readers unconvinced, we now give a detailed proof. First we must state the definitions and theorems from Ref. 2 which we require.

• *Definition 1*

A *one-way finite automaton* over a finite alphabet Σ is a system $A=(S, M, s_0, F)$, where S is a finite non-empty set (the internal states of A), M is a function from $\Sigma\times S$ into S (the table of moves of A), s_0 is an element of S (the initial state of A), and F is a subset of S (the designated final states of A). The class $T(A)$ of tapes *accepted* by A is the class of all finite sequences $\sigma_1 \ldots \sigma_n$ of symbols of Σ for which the sequence s_0 (initial state), $s_1, \ldots, s_n$ defined by $s_{i+1}=M(\sigma_{i+1}, s_i)\,(i=0, \ldots, n-1)$ satisfies $s_n \varepsilon F$. A set of tapes is said to be *definable* by a one-way automaton if it is equal to $T(A)$ for some A.

• *Definition 2*

A *two-way finite automaton* over Σ is a system $A=(S, M, s_0, F)$ as in Definition 1 with the difference that now M is a function from $\Sigma\times S$ into $L\times S$ where $L=\{-1, 0, 1\}$. A operates as follows: It starts on the leftmost square of the given tape in state s_0. When its internal state is s and it scans the symbol σ, then if $M(\sigma, s)=(p, s')$ it goes into the new state s' and moves one square to the right, stays where it is, or moves one square to the left according as $p=+1, 0$, or -1. The class $T(A)$ of tapes accepted by A is the class of those tapes t such that A eventually moves off the right-hand[4] edge of t in a state belonging to F.

• *Theorem 1*

A necessary and sufficient condition for a set $U\subset T_\Sigma$[5] *to be the set $T(A)$ of accepted tapes of some one-way finite automaton A is that T_Σ is partitioned into a finite number of classes by the equivalence relation U_R defined thus:*

$t_1\equiv t_2(\text{mod } U_R)\leftrightarrow(t)\,(t_1 t\varepsilon U\leftrightarrow t_2 t\varepsilon U)$.

Furthermore there is an effective procedure for constructing such[6] an automaton A from the following data: (i) a method of deciding for each t whether $t\varepsilon U$; (ii) an upper bound N for the number of elements of T_Σ/U_R.[7]

Now the theorem we prove in this note is:

• *Theorem 2*

For every two-way finite automaton A there exists a one-

*University of Bristol, England. At present, visitor at University of California, Berkeley.

way finite automaton $\overline{A}$ such that $T(A)=T(\overline{A})$. Furthermore $\overline{A}$ can be obtained effectively from A.

Proof: Suppose A is given in the form mentioned in Definition 2. For each $t \varepsilon T$ define a function $\tau_t:\{\bar{s}_0\}\cup S \rightarrow \{0\}\cup S$[11] as follows: For $s \varepsilon S$, $\tau_t(s)$ describes the ultimate result of the motion of A when started in state s on the rightmost symbol of t, that is to say, if with these initial conditions A ultimately leaves t from the right in internal state s' then $\tau_t(s)=s'$; if on the other hand A either leaves t from the left, or never leaves t then $\tau_t(s)=0$. $\tau_t(\bar{s}_0)$ similarly describes the result of the motion when A is started in the initial state s_0 on the *leftmost* symbol of t. Now it is easily seen that if $\tau_{t1}=\tau_{t2}$ then $t_1 \equiv t_2$ (mod. $T(A)_R$). But if n[13] is the number of internal states of A there are at most $(n+1)^{n+1}$ distinct τ_t. By Theorem 1 it follows that the set $T(A)$ is definable by a one-way automaton $\overline{A}$. We also have a bound, $(n+1)^{n+1}$, on the number of elements of $T_\Sigma/T(A)_R$ so to complete the proof that $\overline{A}$ can be obtained effectively from A we have only to show that (i) of Theorem 1 is satisfied, i.e., that there *is* an effective method of deciding for each t whether it belongs to $T(A)$, thus whether it is accepted by A. This is the case, for we have only to feed t into the machine and see what happens. If the machine does not move off t, either accepting or rejecting it, in less than $l(t)\times n$ units of time, then some combination of internal state and scanned square must have been repeated so that the machine will go into non-stop cyclic behavior and t will never be accepted. This completes the proof of Theorem 2.

• *Note 1*

As outlined in the preliminary informal proof, a direct construction of a (nonminimal) machine $\overline{A}$ such that $T(A)=T(\overline{A})$ can be given in terms of the above analysis. We have only to take the states of $\overline{A}$ to be the different functions τ_t which arise for $t \varepsilon T_\Sigma$, to define $M(\sigma,\tau_t)=\tau_{t\sigma}$ (which is permissible, since $\tau_t=\tau_{t'}$ implies $\tau_{t\sigma}=\tau_{t'\sigma}$), to define the initial state of $\overline{A}$ to be τ_Λ (where Λ is the null tape so that $\tau_\Lambda(s)=0$, $\tau_\Lambda(\bar{s}_0)=s_0$), and to take the designated final states of $\overline{A}$ to be those τ_t for which $\tau_t(\bar{s}_0)$ belongs to F. This avoids the use of Theorem 1, but to see that it is an effective construction of $\overline{A}$ we have to show (i) how to find τ_t given t; (ii) how to make a complete list of all the different τ_t. The solution to (i) is as above; to work out $\tau_t(s)$ simply feed the rightmost symbol of t into $\overline{A}$ in state s and see what happens; if the machine is still going after $l(t)\times n$ units of time then $\tau_t(s)=0$. The solution to (ii) is to make use of the argument used to prove (i) of Footnote 7; this tells us that all different τ_t's which occur will be obtained by considering all t of length less than $(n+1)^{n+1}$.

An even more direct construction of an $\overline{A}$ can be made by taking as states of $\overline{A}$ not merely those functions from $\{\bar{s}_0\}\cup S \rightarrow \{0\}\cup S$ which arise as τ_t's for some t, but all the $(n+1)^{n+1}$ possible such functions. The table of moves can then be defined in the obvious way, i.e., so that if the function f *is* equal to τ_t for some t then $M(\sigma,f)=\tau_{t\sigma}$. (However, the definition requires a little care due to the possibility of a large number of "shuttles" through the square marked σ.)

• *Note 2*

If $\tau_{t_1}=\tau_{t_2}$ then t_1, t_2 are only equivalent when regarded as initial tapes. If we want t_1, t_2 to be replaceable whenever they occur in a tape, i.e., if we want

$(t)(t')(tt_1t' \varepsilon T(A) \leftrightarrow tt_2t' \varepsilon T(A))$

then we must consider instead of τ_t a function τ_t' which registers the result of starting A in any state on either end of t. The function τ_t' will thus be a function from $\{-1,1\}\times S$ into $\{-1,1\}\times S\cup\{0\}$ such that $\tau_t'(p,s)$ gives the result of starting A in state s on the p-most symbol of t (-1 = left, $+1$ = right) and is equal to (p',s') if A leaves t in state s' from the p'-most symbol, is equal to 0 if A does not leave t. If $\tau_{t_1}'=\tau_{t_2}'$ then t_1, t_2 will be equivalent in this strong sense, i.e., will be truly indistinguishable by the machine.

• *Note 3*

Not only the set $T(A)$ of tapes which are accepted by A, but also the set $T_1(A)$ of tapes which are definitely rejected (i.e., machine ultimately leaves the tape but not from the right-hand side, or not in a designated state), and the set $T_2(A)$ of tapes which give rise to non-stop behavior of A, are definable by a one-way automaton. The easiest way to see this is to use the construction of Note 1 but with the "full" functions τ_t' of Note 2 instead of τ_t; to get an $\overline{A}_1$ defining $T_1(A)$ we take as designated final states those τ_t' with $\tau_t'(-1,s_0)=(p',s')$ where $p'=-1$, or $p'=1$ and $s' \bar{\varepsilon} F$; to get an $\overline{A}_2$ defining $T_2(A)$ take as designated final states those τ_t' with $\tau_t'(-1,s_0)=0$.

• *Note 4*

Theorem 2 is the reduction theorem for two-way automata to one-way automata in the case of automata which classify tapes into just two classes. The theorem obviously applies also to the case of automata which classify tapes into any finite number of classes, where the class to which a tape belongs depends only on the final state of the machine. But it does not apply to the case of automata which produce an output tape, one symbol per unit of time, as the machine moves over the input tape. For a two-way machine could be constructed which would give an output tape of 0^n [12] when presented with an input tape of 0^n and an output tape of 0^n10^n when presented with a tape of the form 0^n1t (t being any tape on the alphabet $\{0,1\}$)—all it has to do is to go along to the first 1 putting out 0's, then put out a 1, reverse and go off the tape to the left, putting out 0's as it goes. And it is easy to see that no one-way output automaton could do this.

• *Note 5*

At first sight it would appear that with a two-way automaton more general sets of tapes could be defined if all tapes were provided with special marker symbols b, e (not in Σ) at the beginning and end, respectively. For then it would be possible, e.g., to design a two-way machine which, under certain conditions, would go back to the beginning of a tape, checking for a certain property, and then return again. This would appear to be impossible for an unmarked tape because of the danger of inadvertently going off the left-hand edge of the tape in the middle of the computation. In fact, the machine has no way of telling when it has returned to the first symbol of the tape. However, Theorem 2 implies that this is not so; that the addition of markers does not make any further classes of tapes definable by two-way automata. For if the set $\{b\}U\{e\}$(of all tapes of the form bte for $t \varepsilon U$) is definable by a two-way automaton then it is definable by a one-way automaton; and it is easy to prove that $\{b\}U\{e\}$ is definable by a one-way automaton if and only if U is.

Footnotes and References

1. M. O. Rabin, "Two-Way Finite Automata," *Proc. Summer Institute of Symbolic Logic,* 1957 at Cornell, pp. 366-369.
2. M. O. Rabin and D. Scott, "Finite Automata and Their Decision Problems," *IBM Journal* **3,** 114 (1959).

3. It is crucial in this argument that the automata we are considering are strictly finite automata which are not allowed to alter their input tape; hence whenever the machine returns to t it finds t unchanged.
4. The classes of tapes definable by such machines are easily seen to be unaffected by the exact convention adopted here, e.g., we might prefer to allow a machine to accept a tape by moving off either edge or by stopping in the middle of the tape, either absolutely or by a "loop stop."
5. T_Σ denotes the set of tapes on the alphabet Σ, i.e., the set of all finite (or null) sequences of elements of Σ. The variables t, t_1, t_2 range over T_Σ.
6. In fact, the one with the least number of states.
7. The present statement of this theorem arose out of correspondence with Rabin. But closely related results are to be found in Moore (Ref. 8, pp. 142-145), Myhill (Ref. 9, p. 133), and Nerode (Ref. 10, p. 543). The effectiveness of the construction follows from two facts: (i) The shortest tapes in each of the classes of T_Σ/U_R are of length $<N$. (ii) In the definition $t_1 \equiv t_2$ the unbounded universal quantifier (t) can be replaced by $(t)l(t)<N\ldots$, where $l(t)$ denotes the length of t. (This bound is due to Moore, *op. cit.*, p. 145; a bound of 2^N, which is enough for our purposes, can be established by an even simpler argument.)

8. E. F. Moore, "Gedanken Experiments On Sequential Machines," *Automata Studies,* Princeton 1956, pp. 129-153.
9. J. Myhill, "Finite Automata and the Representation of Events," *WADC Technical Report* 57-624 (1957), pp. 112-137.
10. A. Nerode, "Linear Automata Transformations," *Proc. Amer. Math. Soc.* **9,** 541-544 (1958).

11. The introduction of the new symbol $\bar{s}_0$ is necessitated by the fact that we are interested both in the result of starting A in state s_0 on the leftmost symbol of t and also on the rightmost symbol of t.
12. 0^n stands for $00\ldots(n \text{ times})0$.
13. Not to be confused with the variable n in Definition 1 which ranges over all natural numbers.

Received October 10, 1958