Weekly Proceedings of the Academy of Science, 257:2597–2600, Meeting of October 28, 1963.

MATHEMATICAL LOGIC. — *Reversible Turing machines. Recursive insolubility in $n \in \mathbf{N}$ of the equation $u = \theta^n u$, where θ is an "isomorphism of codes."* Note (*) by Mr. **Yves Lecerf**, presented by Mr. André Lichnerowicz.

We define "reversible Turing machines" and "isomorphisms of codes θ." Their properties make it possible to prove that the equation in $n \in \mathbf{N}$, $u = \theta^n u$ is recursively unsolvable. A second note will apply this to the demonstration of a conjecture of Schützenberger relating the Post correspondence problem to the problem of diagonalization of homomorphisms of free monoids.

1. ISOMORPHISMS OF CODES, EPIMORPHISMS OF CODES. — *a. A conjecture of Schützenberger.* — Given two nontrivial free monoids $\mathrm{A}^\dagger$ and $\mathrm{S}^\dagger$, and given two homomorphisms φ and ψ of $\mathrm{A}^\dagger$ into $\mathrm{S}^\dagger$, consider the problem of the search for nontrivial solutions $x \in \mathrm{A}^\dagger$ for the equation of diagonalization $\varphi x = \psi x$. A result of Post (6) is that this equation is recursively unsolvable in the case of φ and ψ being arbitrary homomorphisms. It is also so when one restricts φ to be a monomorphism; indeed, Chomsky and Schützenberger remarked (1) that this case can be reduced to Post's Tag-problem (5), itself recursively unsolvable according to a result of Minsky (3). Schützenberger conjectured that the equation $\varphi x = \psi x$ remains still recursively unsolvable when φ and ψ are both monomorphisms.

b. *Isomorphisms of codes.* — Instead of $\varphi x = \psi x$, it is equivalent to consider the equation $w = \theta w$, where $\theta = \psi\varphi^{-1}$ (this is shorthand notation for saying that θ is a bijection of $\varphi\mathrm{A}^\dagger$ into $\psi\mathrm{A}^\dagger$ defined by $\theta w = \psi x$ for $w = \varphi x$). For convenience, we will call the applications such as θ "isomorphisms of codes." The term recalls that $\theta(w_1 w_2) = \theta w_1\, \theta w_2$; and also that, $\mathrm{A} = \{\, a_i \,\}_{i\in\mathrm{I}}$ designating the alphabet (generators) of $\mathrm{A}^\dagger$, $\{\, \varphi a_i \,\}_{i\in\mathrm{I}}$ and $\{\, \psi a_i \,\}_{i\in\mathrm{I}}$ are called "codes" on $\mathrm{S}^\dagger$, because, for an arbitrary y in $\mathrm{S}^\dagger$, there exists a set of indices $\{\, i_1, i_2, \ldots, i_p \,\}$ such that $y = \varphi a_{i_1}\, \varphi a_{i_2} \ldots \varphi a_{i_p}$, and the same for ψ. In fact, it is especially the study of isomorphisms of codes to which will be devoted the present Note and the following one.

c. *Definitions of particular "isomorphisms of codes" using relation elements.* — With e_A and e_S designating the identity elements respectively of $\mathrm{A}^\dagger$ and $\mathrm{S}^\dagger$, it goes without saying implicitly for every θ that one has $e_\mathrm{S} = \theta e_\mathrm{S}$, with $\varphi e_\mathrm{A} = \psi e_\mathrm{A} = e_\mathrm{S}$ (whenceforth a trivial solution for $w = \theta w$ and for $w = \theta^n w$, with $n \in \mathbf{N}$). This being the case, each particular isomorphism of codes could be defined by a set of relation elements of the type $\{\, m_{i,\varphi} \to m_{i,\psi} \,\}_{i\in\mathbf{I}}$, provided that $\{\, m_{i,\varphi} \,\}_{i\in\mathrm{I}}$ and $\{\, m_{i,\psi} \,\}_{i\in\mathrm{I}}$ are "codes" and that the correspondence is bijective. Indeed, $\mathrm{A}^\dagger$ is implicitly defined by I, and $\mathrm{S}^\dagger$ by the symbols used to note the $m_{i,\varphi}$ and $m_{i,\psi}$; and one can interpret the relations like correspondences $\{\, \varphi a_i \to \psi a_i \,\}_{i\in\mathrm{I}}$.

d. *Checking whether a given set of words is a code.* — Further, the following property will be often called upon: If C and K_r designate respectively a code and a right prefix-code on $S^\dagger$, and if α is a symbol (generator of $S^\dagger$) not appearing in C nor in K_r, then, the set $C \cup \alpha K_r$ is a code. In the same way, replacing K_r by a left prefix-code K_ℓ, the set $C \cup K_\ell \alpha$ is a code. Let us recall that any right prefix-code K_r is by definition ([4]) such that, if m_i, $m_j \in K_r$ and if, with $y \in S^\dagger$, one has $m_i = m_j y$, then $y = e_S$ (while for the left prefix-codes, it is $m_i = y m_j$ which imposes $y = e_S$).

e. *Epimorphisms of codes.* — One speaks about "epimorphisms of codes" τ in the case of relations $\{ m_{i,\varphi} \to m_{i,\psi} \}_{i \in I}$, where $\{ m_{i,\varphi} \}_{i \in I}$ is a complete code C_φ, but where $\{ m_{i,\psi} \}_{i \in I}$ is only constrained not to contain words other than those of a code C_ψ.

2. Reversible Turing machines. — Let MT be a Turing machine of which $\{ \varepsilon_p \}_{p \in P}$ and $\{ \sigma_q \}_{q \in Q}$ are the sets of states and symbols, and $\{ \delta_r \}_{r \in R}$ are tape displacements, which can be ± 1 or 0. One can define MT by a set of quintuples

$$\chi_{\mathrm{MT}} = \{ \varepsilon_{p_1(i)}; \sigma_{q_1(i)}; \varepsilon_{p_2(i)}; \sigma_{q_2(i)}; \delta_{r(i)} \}_{i \in I},$$

where the indices p_1, p_2, q_1, q_2, r are functions of index i. With each of the quintuples, let us decide to associate an "inverse image quintuple" ($\varepsilon^*_{p_2(i)}$; $\sigma_{q_2(i)}$; $\varepsilon^*_{p_1(i)}$; $\sigma_{q_1(i)}$; $-\delta_{r(i)}$). The set of those will generally not constitute a Turing machine; but when it does, we will say that MT is "reversible," and call the new machine the inverse image MT^* of MT. The ε^*_p will be known as the images of ε_p. The substituion of ε^*_p for ε_p in an instantaneous configuration U_k will be known as transformation of U_k to its image configuration U^*_k. The continuations of configurations of MT^* are images of those of MT, but MT^* traverses them in the opposite order. Now let us consider the machine R (MT), whose set of quintuples is

$$\chi_{\mathrm{R\,(MT)}} = \chi_{\mathrm{MT}} \cup \chi^*_{\mathrm{MT}} \cup \{ (\varepsilon_p; \sigma_q)_{\mathrm{halt}}; \varepsilon^*_p; \sigma_q; 0 \},$$

where $(\varepsilon_p; \sigma_q)_{\mathrm{halt}}$ designates any state-symbol pair for which MT halts. If one starts MT and R (MT) from the same instantaneous configuration U_0, they pass through the same configurations as long as MT does not halt (thus possibly indefinitely). When MT halts, R (MT) continues, traversing in the opposite order the image configurations of the traversed configurations, and passes by the image of the initial configuration. R (MT) will be known as the coupling of MT with its reverse image.

3. Representation of Turing machines by epimorphisms (or isomorphismes) of codes. — Let us be given an arbitrary MT. With each quintuple having movement +1, that is to say for example $(\varepsilon_g, \sigma_h, \varepsilon_j, \sigma_k, 1)$, we associate three relation elements, namely: $\{ \alpha_g \sigma_h \to \sigma_k \alpha_j; \omega_g \sigma_h \to \sigma_k \alpha_j; \sigma_h \beta_g \to \sigma_k \alpha_j \}$. With $(\varepsilon_g, \sigma_h, \varepsilon_j, \sigma_k, 0)$, we associate: $\{ \alpha_g \sigma_h \to \omega_j \sigma_k; \omega_g \sigma_h \to \omega_j \sigma_k; \sigma_h \beta_g \to \omega_j \sigma_k \}$. With $(\varepsilon_g, \sigma_h, \varepsilon_j, \sigma_k, -1)$, we associate $\{ \alpha_g \sigma_h \to \beta_j \sigma_k; \omega_g \sigma_h \to \beta_j \sigma_k; \sigma_h \beta_g \to \beta_j \sigma_k \}$. Finally, with any symbol σ_q of MT, we associate $\sigma_q \to \sigma_q$. One can check, by the process given in paragraph 1*d*, that the set of these relations defines an epimorphism of codes. With $\tau_{\max}$ being this set, $\tau_{\max}$ is a representation of MT, because it defines its alphabet and quintuples. We can, in addition, find, for the instantaneous configurations of MT, notations such that for any pair of successive configurations u_i, u_{i+1}

we have $u_{i+1} = \tau_{\max} u_i$. For that, a configuration will be composed of a succession of symbols σ (the string on the tape) into which one will intercalate one of the letters α, ω or β, with an index p equal to that of the state ε_p of the machine, and indicating, not only the position π_1 of the next symbol to read, but also the position π_2 of the symbol previously written (with a particular convention for the initial configuration). An α_p signifies that π_1 is the first symbol to its right, π_2 the first on its left. A β_p, vice-versa. An ω_p means that π_1 and π_2 are both the first symbol to the right of the ω_p. We have then achieved that $u_{i+1} = \tau_{max} u_i$. So certain states ε_p can appear under only two or one of the forms α_p, ω_p, β_p, and $\tau_{\min}$ is obtained by removing from $\tau_{\max}$ all the relation elements containing the forms which never appear, so $\tau_{\min}$ is still such that $u_{i+1} = \tau_{\min} u_i$. If $\tau_{\min}$ is an isomorphism of codes, MT is reversible.

4. Simulation of arbitrary MT on reversible MT′. Application to isomorphisms of codes. — *a. Properties.* — One can simulate an arbitrary Turing machine MT (with configurations v_i) on a reversible Turing machine MT_ρ (with configurations $u_{i,j}$) so that: (1) when MT passes from v_i to v_{i+1}, MT_ρ passes from $u_{i,0}$ to $u_{i+1,0}$ via the intermediary of a finite number of configurations $u_{i,1}; u_{i,2}; \ldots;$ (2) we pass from one v_i to the next via an epimorphism of codes τ, and from one $u_{i,j}$ to the next via an isomorphism of codes θ; (3) if the initial configurations are v_0 for MT and $u_{0,0}$ for MT_ρ, with $u_{0,0} = \lambda v_0 \mu \nu$, then for any i, one has $u_{i,0} = \lambda v_i \mu w_i \nu$, where w_i is a string, and where λ, μ, ν are three symbols which appear neither in v_i nor in w_i, so that knowing $u_{i,0}$ gives v_i and w_i; (4) there are symbols r_k of which each one represents a relation element of τ other than that of identity; a blank symbol b; and for any i we have $w_i = b^2 r_{k_1} r_{k_2} \ldots r_{k_i} b$, where r_{k_p} is the relation invoked by $v_p = \tau v_{p-1}$. Thus, w_i represents the history of the computation of MT until time i; (5) MT_ρ halts on the $u_{i,0}$ corresponding to the halting of MT, and them only; (6) the machine $\mathrm{R}(\mathrm{MT}_\rho)$, coupling MT_ρ with its reverse image, starting from $u_{0,0} = \lambda v_0 \mu \nu$ passes through the image configuration $\lambda v_0^* \mu \nu$ if and only if MT, starting from v_0, halts; (7) there exists for $\mathrm{R}(\mathrm{MT}_\rho)$ certain instantaneous configurations $u_{s,t}$ such that, when started at $\lambda v_0 \mu \nu$, $\mathrm{R}(\mathrm{MT}_\rho)$ cannot reach those configurations other than by passing through $\lambda v_0^* \mu \nu$ (*i.e.*, if MT, starting from v_0, halts). One can thus arrange that the return of of $\mathrm{R}(\mathrm{MT}_\rho)$ to $\lambda v_0 \mu \nu$ (or the passage through u_{st} framed by $\lambda' \nu'$ instead of $\lambda \nu$) is conditional on the halting of MT.

Proof. — It is shown how to proceed from τ, presumed to be given by a set of relation elements $\{ \mathrm{I}_{k,\tau} \}_{k \in \mathrm{K}_\tau}$ to the set of relation elements $\{ \mathrm{I}_{j,\theta} \}_{j \in \mathrm{J}_\theta}$ defining θ and MT_ρ. We delimit the principles of this construction, by showing how to simulate an $\mathrm{I}_{k_i,\tau}$ of the form $\alpha_p \sigma_q \to \sigma_f \alpha_g$. With this, we associate: an instruction $\alpha_p \sigma_q \to \sigma_{f,g,\alpha}\, \varepsilon_{\alpha,\alpha,p,q,f,g,\sigma}$, where the symbol $\sigma_{fg\alpha}$ marks the place where one must modify v_i, and the nature of the modification; instructions allowing control to be lead to the left from ν through a state $\varepsilon_{\alpha\alpha pqfg\nu}$; an instruction $b \varepsilon_{\alpha\alpha pqfg\nu} \to \varepsilon_s r_{k_i}$, where r_{k_i} represents $I_{k_i,\tau}$, supplementing w_i; working instructions moving ν and possibly also λ, μ and the entire w_i, to restore the necessary blanks in $u_{i,0}$ and then defer control in $\sigma_{f,g,\alpha}$ with a state ε_σ; an instruction $\sigma_{f,g,\alpha}\, \varepsilon_\sigma \to \sigma_f \alpha_g$ which supplements v_i in $u_{i,0}$.

b. Theorem 1. — *The halting problem for a general reversible Turing machine is undecidable. Similarly for the problem of returning to the initial configuration, and* *that of the passage through a given configuration other than the initial configuration.*

c. THEOREM 2. — *The equation* $w = \theta^n w$, *where* θ *is an isomorphism of codes, with* $n \in \mathbf{N}$ *is recursively unsolvable in* n *given arbitrary* w, θ. *The equation* $w_1 = \theta^n w_2$, *with* $w_1 \neq w_2$, *is also recursively unsolvable in* n.

(*) Meeting of October 21, 1963.

(1) N. CHOMSKY et M. P. SCHÜTZENBERGER, *Computer Programming and Formal Systems*, Hirschberg and Braffort, North-Holland Publ. Co., Amsterdam, 1963, p. 118–161.

(2) H. WANG, *Mathematische Annalen*, **152**, 1963, p. 65–74.

(3) M. MINSKY, *Annals of Math.*, **74-3**, 1961.

(4) M. P. SCHÜTZENBERGER, *I. R. E. Trans. Inf. Theory*, IT-2, 1956, p. 47-60.

(5) E. POST, *Amer. J. Math.*, **65**, 1943, p. 196–215.

(6) E. POST, *Bull. Amer. Math. Soc.*, **52**, 1946, p. 264–268.

(*Euratom, 51, rue Belliard, Bruxelles.*)