Comptes Rendus Hebdomadaires des Séances de L'académie des Sciences,
257: 2597–2600, Séance du 28 Octobre 1963.

Logique Mathématique. — *Machines de Turing réversibles. Récursive insolubilité en $n \in \mathbf{N}$ de l'équation $u = \theta^n u$, où θ est un « isomorphisme de codes ».* Note (*) de M. **Yves Lecerf**, présentée par M. André Lichnerowicz.

On définit des « machines de Turing réversibles » et des « isomorphismes de codes θ ». Leurs propriétés permettent de prouver que l'equation en $n \in \mathbf{N}$, $u = \theta^n u$ est récursivement insoluble. Une second Note appliquera ceci à la démonstration d'une conjecture de Schützenberger rattachant le problème des correspondances de Post à des problèmes de diagonalisation d'homomorphismes de monoïdes libres.

1. Isomorphismes de codes, épimorphismes de codes. — *a. Une conjecture de Schützenberger.* — Soient deux monoïdes libres non triviaux $A^\dagger$ et $S^\dagger$, soient deux homomorphismes φ et ψ de $A^\dagger$ dans $S^\dagger$, et soit le problème de la recherche de solutions non triviales $x \in A^\dagger$ pour l'équation de diagonalisation $\varphi x = \psi x$. Un résultat de Post ([6]) est que cette équation est récursivement insoluble dans le cas de φ et ψ homomorphismes quelconques. Elle l'est également lorsqu'on astreint φ à être un monomorphisme; Chomsky et Schützenberger ont fait observer, en effet ([1]), que ce cas peut se ramener au Tag-problem de Post ([5]) lui-même récursivement insoluble d'après un résultat de Minsky ([3]). Schützenberger a conjecturé que l'équation $\varphi x = \psi x$ reste encore récursivement insoluble quand à la fois φ et ψ sont deux monomorphismes.

b. *Isomorphismes de codes.* — Au lieu de $\varphi x = \psi x$, il revient au même de considérer l'équation $w = \theta w$, où $\theta = \psi\varphi^{-1}$ (ce qui est une notation abrégée pour dire que θ est une bijection de $\varphi A^\dagger$ dans $\psi A^\dagger$ définie par $\theta w = \psi x$ pour $w = \varphi x$). Par facilité de langage, on appellera « isomorphismes de codes » les applications telles que θ. Ce terme rappelle que $\theta(w_1 w_2) = \theta w_1\, \theta w_2$; et aussi que, $A = \{ a_i \}_{i\in I}$ désignant l'alphabet (les générateurs) de $A^\dagger$, $\{ \varphi a_i \}_{i\in I}$ et $\{ \psi a_i \}_{i\in I}$ sont appelés des « codes » sur $S^\dagger$, car, pour y quelconque de $S^\dagger$, il existe au plus un ensemble d'indices $\{ i_1, i_2, \dots, i_p \}$ tels que $y = \varphi a_{i_1}\, \varphi a_{i_2} \dots \varphi a_{i_p}$, et de même pour ψ. En fait, c'est surtout à l'étude des isomorphismes de code que seront consacrées la présente Note et la suivante.

c. *Définitions d'« isomorphismes de codes » particuliers par donnée de relations.* — e_A et e_S désignant les éléments neutres respectivement de $A^\dagger$ et de $S^\dagger$, il va de soi implicitement pour tout θ que l'on a $e_S = \theta e_S$, avec $\varphi e_A = \psi e_A = e_S$ (d'où, d'ailleurs, une solution triviale pour $w = \theta w$ et pour $w = \theta^n w$, avec $n \in \mathbf{N}$). Ceci étant, chaque isomorphisme de codes particulier pourra être défini par la donnée d'un ensemble de relations du type $\{ m_{i,\varphi} \to m_{i,\psi} \}_{i\in\mathbf{I}}$, à condition que $\{ m_{i,\varphi} \}_{i\in I}$ et $\{ m_{i,\psi} \}_{i\in I}$ soient des « codes » et que la correspondance soit bijective. En effet, $A^\dagger$ est implicitement défini par I, $S^\dagger$ l'est par les symboles utilisés pour noter les $m_{i,\varphi}$ et $m_{i,\psi}$; et l'on peut interpréter les relations comme des correspondances $\{ \varphi a_i \to \psi a_i \}_{i\in I}$.

d. *Vérification qu'un ensemble de mots donnés est un code.* — Plus loin, on invoquera souvent la propriété suivante : Si C et K_d désignent respectivement un code et un préfix-code droit sur $\mathrm{S}^\dagger$, et si α est un symbole (générateur de $\mathrm{S}^\dagger$) ne figurant pas dans C ni dans K_d, alors, l'ensemble $\mathrm{C} \cup \alpha\mathrm{K}_d$ est un code. De même, en remplaçant K_d par un préfix-code gauche K_g, l'ensemble $\mathrm{C} \cup \mathrm{K}_g\alpha$ est un code. Rappelons que tout préfix-code droit K_d est par définition ([4]) tel que, si $m_i,\ m_j \in \mathrm{K}_d$ et si, avec $y \in \mathrm{S}^\dagger$, on a $m_i = m_j y$, alors, $y = e_\mathrm{S}$ (tandis que pour les préfix-codes gauches, c'est $m_i = y m_j$ qui impose $y = e_\mathrm{S}$).

e. *Épimorphismes de codes.* — On parlera d' « épimorphisme de codes » τ dans le cas de relations $\{\, m_{i,\varphi} \to m_{i,\psi} \,\}_{i\in\mathrm{I}}$, où $\{\, m_{i,\varphi} \,\}_{i\in\mathrm{I}}$ est bien un code C_φ, mais où $\{\, m_{i,\psi} \,\}_{i\in\mathrm{I}}$ est astreint seulement à ne pas contenir d'autres mots que ceux d'un code C_ψ.

2. Machines de Turing réversibles. — Soit une machine de Turing MT dont $\{\, \varepsilon_p \,\}_{p\in\mathrm{P}}$ et $\{\, \sigma_q \,\}_{q\in Q}$ sont les ensembles d'états et de symboles, et $\{\, \delta_r \,\}_{r\in R}$ les déplacements de ruban, qui peuvent être ± 1 ou 0. On peut définir MT par un ensemble de quintuples

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

où les indices p_1, p_2, q_1, q_2, r sont fonctions de l'indice i. A chacun de ces quintuples, décidons d'associer un « quintuple image inverse » $(\varepsilon^*_{p_2(i)};\ \sigma_{q_2(i)};\ \varepsilon^*_{p_1(i)};\ \sigma_{q_1(i)};\ -\delta_{r(i)})$. L'ensemble de ceux-ci ne constituera généralement pas une machine de Turing; mais lorsqu'il en sera ainsi, on dira que MT est « réversible », et l'on donnera à la nouvelle machine le nom d'image inverse MT* de MT. Les ε^*_p seront dits images des ε_p. La substitution d'un ε^*_p à un ε_p dans une configuration instantanée U_k sera dite passage à U_k à sa configuration image U^*_k. Les suites de configurations de MT* sont images de celles de MT, mais MT* les parcourt dans l'ordre inverse. Considérons maintenant la machine R (MT), dont l'ensemble de quintuples est

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

où $(\varepsilon_p; \sigma_q)_{\mathrm{stop}}$ désigne tout couple pour lequel MT stoppe. Si l'on fait partir MT et R (MT) d'une même configuration instantanée U_0, elles passent par les mêmes configurations tant que MT ne stoppe pas (donc éventuellement indéfiniment). Lorsque MT stoppe, R (MT) continue, parcourt en ordre inverse les configurations images des configurations parcourues, et passe par l'image de la configuration initiale. R (MT) sera dite couplage de MT avec son image inverse.

3. Représentation de machines de Turing par des épimorphismes (ou des isomorphismes) de codes. — Revenons au cas où MT est quelconque. A chacun de ses quintuples comportant le mouvement $+1$, soit par exemple $(\varepsilon_g, \sigma_h, \varepsilon_j, \sigma_k, 1)$, on associe trois relations, à savoir ici : $\{\, \alpha_g\sigma_h \to \sigma_k\alpha_j; \omega_g\sigma_h \to \sigma_k\alpha_j; \sigma_h\beta_g \to \sigma_k\alpha_j \,\}$. Aux $(\varepsilon_g, \sigma_h, \varepsilon_j, \sigma_k, 0)$, on associe : $\{\, \alpha_g\sigma_h \to \omega_j\sigma_k; \omega_g\sigma_h \to \omega_j\sigma_k; \sigma_h\beta_g \to \omega_j\sigma_k \,\}$. Aux $(\varepsilon_g, \sigma_h, \varepsilon_j, \sigma_k, -1)$, on associe $\{\, \alpha_g\sigma_h \to \beta_j\sigma_k; \omega_g\sigma_h \to \beta_j\sigma_k; \sigma_h\beta_g \to \beta_j\sigma_k \,\}$. Enfin, à tout symbole σ_q de MT, on associe $\sigma_q \to \sigma_q$. On peut vérifier, par le procédé donné au paragraphe 1, *d*, que l'ensemble de toutes ces relations définit un épimorphisme de codes. Soit $\tau_{\max}$ celui-ci, $\tau_{\max}$ est une représentation de MT, car il en définit l'alphabet

et les quintuples. On peut, d'autre part, trouver pour les configurations instantanées de MT des notations telles que pour tout couple de configurations successives u_i, u_{i+1} on ait $u_{i+1} = \tau_{\max} u_i$. Pour cela, une configuration se composera d'une suite de symboles σ (le mot du ruban) dans laquelle on intercalera une des lettres α, ω ou β, avec un indice p égal à celui de l'état ε_p de la machine, et de façon à indiquer, non seulement la position π_1 du prochain symbole à lire, mais encore celle π_2 du symbole précédemment écrit (acev convention particulière pour la configuration initiale). Un α_p signifie que π_1 est le premier symbole à sa droite, π_2 le premier à sa gauche. Un β_p, l'inverse. Un ω_p signifie que π_1 et π_2, confondus, sont tous deux le premier symbole à droite de ce ω_p. On a bien alors $u_{i+1} = \tau_{max} u_i$. Si certains états ε_p ne peuvent apparaître que sous deux seulement, ou une seule, des formes $\alpha_p, \omega_p, \beta_p$, et soit $\tau_{\min}$ obtenu en retranchant de $\tau_{\max}$ toutes les relations contenant des formes qui n'apparaissent jamais, alors, $\tau_{\min}$ est encore tel que $u_{i+1} = \tau_{\min} u_i$. Si $\tau_{\min}$ est un isomorphisme de codes, MT est réversible.

4. Simulation de MT quelconque sur MT$'$ réversible. Application aux isomorphismes de codes. — *a. Propriété.* — On peut simuler une machine de Turing MT quelconque (de configurations v_i) sur une machine de Turing réversible MT_ρ (de configurations $u_{i,j}$) de façon que : 1^0 quand MT passe de v_i à v_{i+1}, MT_ρ passe de $u_{i,0}$ à $u_{i+1,0}$ par l'intermédiaire d'un nombre fini de configurations $u_{i,1}; u_{i,2}; \ldots$; 2^0 on passe d'un v_i au suivant par un épimorphisme de codes τ, d'un $u_{i,j}$ au suivant par un isomorphisme de codes θ; 3^0 si les configurations initiales sont v_0 pour MT et $u_{0,0}$ pour MT_ρ, avec $u_{0,0} = \lambda v_0 \mu \nu$, alors pour tout i, on a $u_{i,0} = \lambda v_i \mu w_i \nu$, où w_i est un mot, et où λ, μ, ν sont trois symboles qui n'apparaissent ni dans v_i ni dans w_i, si bien que $u_{i,0}$ connu donne v_i et w_i; 4^0 il y a des symboles r_k dont chacun représente une relation de τ autre que d'identité; un symbole blanc b; et pour tout i on a $w_i = b^2 r_{k_1} r_{k_2} \ldots r_{k_i} b$, où r_{k_p} est la relation intervenue dans $v_p = \tau v_{p-1}$. Ainsi, w_i représente l'histoire du calcul de MT jusqu'au temps i; 5^0 MT_ρ stoppe sur les $u_{i,0}$ correspondant à des stop de MT, et eux seulement; 6^0 la machine $\mathrm{R}\,(\mathrm{MT}_\rho)$, couplage de MT_ρ avec son image inverse, partie de $u_{0\,0} = \lambda v_0 \mu \nu$ passe par la configuration image $\lambda v_0^* \mu \nu$ si et seulement si MT, partie de v_0, stoppe; 7^0 il existe pour $\mathrm{R}\,(\mathrm{MT}_\rho)$ certaines configurations instantanées u_{st} telles que, partant de $\lambda v_0 \mu \nu$, $\mathrm{R}\,(\mathrm{MT}_\rho)$ ne peut atteindre ces configurations autrement qu'en passant par $\lambda v_0^* \mu \nu$ (c'est-à-dire si MT, partie de v_0, stoppe). On peut ainsi faire en sorte que le retour de $\mathrm{R}\,(\mathrm{MT}_\rho)$ à $\lambda v_0 \mu \nu$ (ou bien, le passage par des u_{st} encadrés par $\lambda' \nu'$ au lieu de $\lambda \nu$) soit conditionné par un stop de MT.

Preuve. — On montre comment passer de τ, supposé donné par un ensemble de relations $\{\, \mathrm{I}_{k,\tau} \,\}_{k \in \mathrm{K}_\tau}$ à l'ensemble des relations $\{\, \mathrm{I}_{j,\theta} \,\}_{j \in \mathrm{J}_\theta}$ définissant θ et MT_ρ. Limitons-nous à donner le principe de cette construction, en montrant comment simuler une instruction $\mathrm{I}_{k_i,\tau}$ de la forme $\alpha_p \sigma_q \to \sigma_f \alpha_g$. A celle-ci, on associera : une instruction $\alpha_p \sigma_q \to \sigma_{f,g,\alpha}\, \varepsilon_{\alpha,\alpha,p,q,f,g,\sigma}$, où le symbole $\sigma_{fg\alpha}$ marque la place où l'on doit modifier v_i, et la nature de la modification; des instructions permettant de conduire le contrôle à gauche de ν par un état $\varepsilon_{\alpha\alpha pqfg\nu}$; une instruction $b \varepsilon_{\alpha\alpha pqfg\nu} \to \varepsilon_s r_{k_i}$, où r_{k_i} représente $I_{k_i,\tau}$, complètant w_i; des instructions de service déplaçant ν et, éventuellement aussi λ, μ et w_i tout entier, pour rétablir les blancs nécessaires dans $u_{i,0}$ puis reportant le contrôle en $\sigma_{f,g,\alpha}$ avec un état ε_σ; une instruction $\sigma_{f,g,\alpha}\varepsilon_\sigma \to \sigma_f\alpha_g$ qui complète v_i dans $u_{i,0}$.

b. THÉORÈME 1. — *Le problème du stop pour une machine de Turing réversible générale est indécidable. Il en est de même du problème du retour à la configuration initiale, et de celui du passage par une configuration déterminée autre que la configuration initiale.*

c. THÉORÈME 2. — *L'équation $w = \theta^n w$, où θ est un isomorphisme de codes, avec $n \in \mathbf{N}$ est récursivement insoluble en n pour w, θ donnés quelconques. L'equation $w_1 = \theta^n w_2$, avec $w_1 \neq w_2$, est aussi récursivement insoluble en n.*

(*) Séance du 21 octobre 1963.

(1) N. CHOMSKY et M. P. SCHÜTZENBERGER, *Computer Programming and Formal Systems*, Hirschberg et 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.*)