Un seul état d'arrêt ?

- Ajouter q_f
- $\forall T(q,\alpha)$ non définie ajouter $(q,\alpha) \mapsto (\alpha, \triangledown, q_f)$
- Aucune transition sur q_f

Pas de transitions sur place ?

Si $T(q,\alpha) = (\beta, \triangledown, q')$:

- Ajouter r
- $(q,\alpha) \mapsto (\beta, \triangleright, r)$
- $(r,x) \mapsto (x, \triangleleft, q')$ pour tout $x \in V$

Alphabet ? limitation à $\{0,1\}$ idée : codage en binaire

- Pour tout $x \in V$, codage sur k bits (k suffisant pour tout V)
  Codage de $\mathbb{B}$ par 0^k
- Case de $\mathscr{M}$, codage par k cases pour $\mathscr{M}'$

Par transition pour $\mathscr{M}$:

- Lecture des k cases : (mémorisation dans état)

$$\begin{array}{lcl@{\qquad}lcl} (q,0) & \mapsto & (0,\triangleright,q_0) & (q_w,0) & \mapsto & (0,\triangleright,q_{w0}) \\ (q,1) & \mapsto & (1,\triangleright,q_1) & (q_w,1) & \mapsto & (1,\triangleright,q_{w1}) \end{array}$$

  In fine : q_v où v : code sur k bits d'un $\alpha \in V$ (2^k états...)

Alphabet ? limitation à $\{0,1\}$ idée : codage en binaire

- Pour tout $x \in V$, codage sur k bits (k suffisant pour tout V)
  Codage de $\mathbb{B}$ par 0^k
- Case de $\mathscr{M}$, codage par k cases pour $\mathscr{M}'$

Par transition pour $\mathscr{M}$:

- Lecture des k cases : q_v où v : code sur k bits d'un $\alpha \in V$
- $T(q,\alpha)$ non défini : Retour sur k cases et arrêt
- $T(q,\alpha) = (\beta, \blacktriangle, q')$
  1. Retour sur k cases avec écriture du code de β $\longleftarrow$
  2. Déplacement de k cases selon $\blacktriangle$
  3. Passage à q'

Machines de Turing — machines simples

- Pas de transitions sur place
- Alphabet : $\{0, 1\}$

$\rightsquigarrow$ Machine de Turing simple

Proposition.
Si L défini par une MT, alors L défini par MT simple
Si f Turing calculable, alors f calculable par MT simple

Machines de Turing — variantes

Plusieurs bandes (k), une tête :
$T : Q \times V^k \to V^k \times \{\triangleleft, \triangledown, \triangleright\} \times Q$

- Mot à reconnaître sur bande 1
- Entrée/sortie de fonction sur bande 1

Simulation par MT à 1 bande : contenu des k bandes sur k cases...
k lectures : $(q, \alpha) \mapsto (\alpha, \triangleright, q_\alpha)$ $(q_w, \alpha) \mapsto (\alpha, \triangleright, q_{w\alpha})$
In fine : q_v où $v = \alpha_1 \cdots \alpha_k$: mot sur k symboles des k bandes

- $T(q, \alpha_1 \cdots \alpha_k)$ non défini : Retour sur k cases et arrêt
- $T(q, \alpha_1 \cdots \alpha_k) = (\beta_1 \cdots \beta_k, \blacktriangle, q')$
  - Retour sur k cases avec écriture des β_i (de droite à gauche)
  - Déplacement de k cases selon $\blacktriangle$ 3. Passage à q'

Machines de Turing — variantes

Plusieurs bandes (k), plusieurs têtes (k) :
$T : Q \times V^k \to V^k \times \{\triangleleft, \triangledown, \triangleright\}^k \times Q$

- Mot à reconnaître sur bande 1
- Entrée/sortie de fonction sur bande 1

Simulation par MT à k bandes et 1 tête : case présence + case contenu...
Lecture de la totalité des bandes par transition (marqueur à gauche)

- Lecture complète, mémorisation dans état
- Si arrêt : remettre tête à la bonne place (position tête 1)
- Sinon : parcours droite-gauche complet avec écritures idoines
- Passage dans l'état adéquat

Proposition.

Si L_1 et L_2 récursifs alors :

- $L_1 \cup L_2$ récursif
- $L_1 \cap L_2$ récursif
- $\overline{L_1}$ récursif

Proposition.

Si L_1 et L_2 r.é. :

- $L_1 \cup L_2$ r.é.
- $L_1 \cap L_2$ r.é.

Proposition.

Si L et $\overline{L}$ r.é. alors L récursif

Question : $\exists$? MT répondant à $w, L \mapsto w \in L$?

Description de MT : avec langage sur $\{q_i \ldots, \triangleright, \triangleleft, 0, 1 \ldots\}$ $\rightsquigarrow$ mot $\langle \mathscr{M} \rangle$

Considérer $\langle \mathscr{M} \rangle$ comme une entrée pour une MT

$\rightsquigarrow$ calculer et décider propriétés sur MT simples...

$\mathscr{M}_u$: 3 bandes

- b_1 : $\langle \mathscr{M} \rangle$ puis w
- b_2 : bande de travail de $\mathscr{M}$
- b_3 : état courant de $\mathscr{M}$

Exécution de $\mathscr{M}_u$:

- Copie de q_0 sur b_3
- Copie de w sur b_2
- Boucle :
  - Recherche dans $\langle \mathscr{M} \rangle$ de transitions pour état en b_3
  - Recherche de la transition pour mot en b_2 (évt sortir)
  - Écriture sur b_2, écriture état sur b_3
- Copie de b_2 sur b_1 (pour fonction)
- Test état b_3 acceptant (pour décision)

Proposition.

L'appartenance à un langage défini par MT est semi-décidable

Machines de Turing — énumérateur

Machine à 2 bandes

- Bande de sortie — tête vers droite uniquement
- Bande de travail

Exécution $\rightsquigarrow \cdots \mathbb{B}\mathbb{B}w_1\mathbb{B}w_2\mathbb{B}w_3\cdots$ — pas canonique

Proposition.
L semi-décidable si et seulement s'il existe énumérateur

Machines de Turing — langage diagonal

Codage en binaire $\rightsquigarrow$ ordre « canonique » (codes vus comme entiers)

$\rightsquigarrow$ Ordonner les w — w_i

$\rightsquigarrow$ Ordonner les MT (en fonction de $\langle \mathscr{M} \rangle$) — $\mathscr{M}_i$ si $\langle \mathscr{M}_i \rangle_{10} = i$

$L_d = \{w_i \mid w_i \notin \mathcal{L}(\mathscr{M}_i)\}$

Proposition.
L_d non récursivement énumérable.

$\rightsquigarrow$ argument diagonal.

Machines de Turing — arrêt

Données : MT simple $\mathscr{M}$, mot w, question : arrêt de $\mathscr{M}$ sur w ?

Proposition.
Arrêt : semi-décidable.

Proposition.
Arrêt : non décidable. (Turing 1936)

Problèmes de décision — réduction

P_1 réduit à P_2 si méthode de décision de P_1 à partir de méthode de décision de P_2.

L_1, L_2 langages, réduction de L_1 à L_2 : fonction Turing-calc. totale f t. q.

$$w \in L_1 \quad \text{ssi} \quad f(w) \in L_2$$

Proposition.
$L_1 \rightsquigarrow L_2$ et L_2 réc. alors L_1 réc.

Proposition.
$P_1 \rightsquigarrow P_2$ et P_1 indécidable alors P_2 indécidable.

Problèmes de décision — vacuité

Données : MT simple $\mathscr{M}$, question : $\mathcal{L}(\mathscr{M}) = \varnothing$?

Proposition.
Problème de vacuité : indécidable.

Preuve : par réduction du problème de l'arrêt.

Et même : non semi-décidable.

Problèmes de décision — égalité de r.é.

Données : MT simples $\mathscr{M}_1$ et $\mathscr{M}_2$, question : $\mathcal{L}(\mathscr{M}_1) = \mathcal{L}(\mathscr{M}_2)$?

Proposition.
Problème de l'égalité de langages r.é. : indécidable.

Preuve : par réduction du problème de la vacuité.

Et même : non semi-décidable.

Problèmes de décision — égalité de fonctions T.-c.

Données : fonctions T-calculables f et g, question : $\forall x, f(x) = g(x)$?

Proposition.
Problème de l'égalité de fonctions Turing-calculables : indécidable.

Preuve : par réduction du problème de l'égalité de langages r.é.

Et même : non semi-décidable.

Problèmes de décision — Rice

Propriété P triviale (pour E) si

- $\forall x \in E, P(x)$ ou bien
- $\forall x \in E, \neg P(x)$.

Théorème. (Rice, 1951) (H. G. Rice, 1920 – 2003)
P non triviale sur langages r.é., alors P indécidable.

Données : machine $\mathscr{M}$, question : $P(\mathcal{L}(\mathscr{M}))$?

Preuve : par réduction du problème de l'arrêt.