Informatique Théorique 3
2004/05

Semaine 11

Voir [HMU03] §8 (en particulier §8.2).

Voir [HMU03] §8.2.6 et [Sch95] §2.2.

```
main()
{
    printf("hello, world");
}
```

```
int exp(int i n)
/* computes i to the power n */
...

main()
{
  int n, total, x, y, z;
  scanf("%d", &n);
  total = 3;
  while (1) {
    for   (x=1; x<=total-2;   x++)
      for (y=1; y<=total-x-1; y++) {
        z = total - x - y;
        if ( exp(x,n) + exp(y,n) == exp(z,n) )
          printf("hello, world")
      }
    total++
  }
}
```

4.1.1 Définition (TM) Une *machine de Turing* (TM) est un septuplet

$$M = (Q, \Sigma, \Gamma, \delta, q_0, \mathrm{B}, F)$$

où

- Q est un ensemble fini d'*états*,
- Σ est un ensemble fini de symboles, *l'alphabet d'entrée*,
- $\Gamma \supset \Sigma$ est un ensemble fini de symboles, *l'alphabet du ruban*,
- $\delta : (Q \times \Gamma) \to (Q \times \Gamma \times \{-1, 0, 1\})$ est la *fonction de transition*,
- $q_0 \in Q$ est l'*état initial* (ou de *départ*),
- $\mathrm{B} \in \Gamma \setminus \Sigma$ est le *symbole blanc*,
- $F \subseteq Q$ est l'ensemble des *états accepteurs* (ou *finaux*).

En général, δ est une fonction partielle.

Il existe aussi des versions des TM où δ est de type

multi/k-rubans : $(Q \times \Gamma^k) \to (Q \times \Gamma^k \times \{-1, 0, 1\}^k)$

non-déterministe : $(Q \times \Gamma) \to \mathcal{P}(Q \times \Gamma \times \{-1, 0, 1\})$

4.1.4 Définition (Configurations et calculs)

Soit $M = (Q, \Sigma, \Gamma, \delta, q_0, \mathrm{B}, F)$ une TM.

1. Une *configuration* de M est un triplet $\boxed{(q, i, \mathrm{rub})}$ où
   - $q \in Q$ est l'état dans lequel se trouve la machine M,
   - $i \in \mathbb{Z}$ est l'emplacement courant de la tête de lecture/écriture, et
   - $\mathrm{rub} : \mathbb{Z} \to \Gamma$ est une application représentant l'état du ruban telle que $\mathrm{rub}(k) \neq \mathrm{B}$ pour au plus un nombre fini d'entiers k.

   Si $\boxed{q \in F}$, on dit que (q, i, rub) est une *configuration acceptante* de M.

2. Soit $\boxed{c = (q, i, \mathrm{rub})}$ une configuration de M et $X = \mathrm{rub}(i)$.
   Si $\boxed{\delta(q, X) = (q', Y, d)}$,
   alors la configuration c *se réduit en une étape de calcul* en la configuration $\boxed{c' = (q', i + d, \mathrm{rub}\{i \mapsto Y\})}$, noté $\boxed{c \vdash_M c'}$, avec

$$\boxed{\begin{array}{lll} \mathrm{rub}\{i \mapsto Y\} : \mathbb{Z} \to \Gamma & & \\ & k \mapsto Y & \text{si } k = i \\ & k \mapsto \mathrm{rub}(k) & \text{sinon} \end{array}}$$

   Comme d'habitude, on note $\boxed{\vdash_M^*}$ la fermeture réflexive et transitive de la relation $\vdash_M$ ainsi définie.

3. On identifie tout mot $w \in \Gamma^*$ avec le ruban $\boxed{w : \mathbb{Z} \to \Gamma}$ défini par :

$$\boxed{\begin{array}{lll} w : \mathbb{Z} \to \Gamma & & \\ & k \mapsto (w)_k & \text{si } 1 \leq k \leq |w| \\ & k \mapsto \mathrm{B} & \text{sinon} \end{array}}$$

4. Soit $w \in \Sigma^*$. Un $\boxed{\textit{calcul de } M \textit{ pour } w}$ est une séquence

$$\boxed{c_0 \vdash_M c_1 \vdash_M \ldots \vdash_M c_i \vdash_M \ldots}$$

   où pour tout k, c_k est une configuration de M et $\boxed{c_0 = (q_0, 1, w)}$ est la configuration initiale du calcul.

5. Soit c un configuration de M. S'il n'existe aucune configuration c' telle que $c \vdash_M c'$, on note $\boxed{c \not\vdash_M}$.

Lorsque la machine M dont on parle est claire dans le contexte, on écrit simplement $\vdash$ à la place de $\vdash_M$.

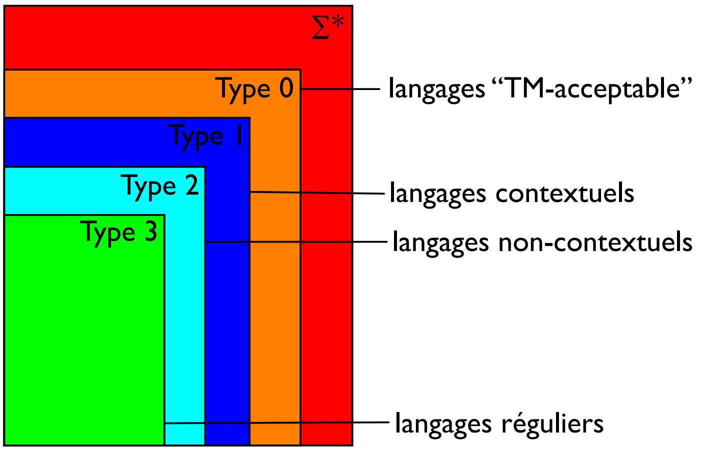
4.1.5 Définition (Langage accepté par une TM)
Soit $M = (Q, \Sigma, \Gamma, \delta, q_0, \mathrm{B}, F)$ une TM.
Le $\boxed{\textit{langage accepté par } M}$ est défini par :

$$\boxed{\mathrm{L}(M)} \boxed{\triangleq \{\, w \in \Sigma^* \mid \text{il existe } c \text{ acceptante telle que } (q_0, 1, w) \vdash_M^* c \,\}}$$

4.1.6 Théorème Soit Σ un alphabet et $L \subseteq \Sigma^*$.
Alors les deux propositions suivantes sont équivalentes :

1. Il existe une TM M telle que $L = \mathrm{L}(M)$.
2. Il existe une grammaire G de type 0 telle que $L = \mathrm{L}(G)$.

Hiérarchie de Chomsky ...



4.1.3 Notation (Graphe d'une TM)

Si $\delta(q, X) = (q', Y, d)$,
on dessine une flèche étiquetée par ~~$X, Y/d$~~ de q à q'.

X / Y, d

4.2.1 Définition (Arrêt)

Soit $M = (Q, \Sigma, \Gamma, \delta, q_0, \mathrm{B}, F)$ une TM.
On dit que $\boxed{M \text{ s'arrête pour } w \in \Sigma^*}$
s'il existe une configuration c telle que $(q_0, 1, w) \vdash_M^* c \not\vdash_M$.
Si M s'arrête pour tout $w \in \Sigma^*$, alors M est dite *totale*.

4.2.2 Notation Soit $M = (Q, \Sigma, \Gamma, \delta, q_0, \mathrm{B}, F)$ une TM.

1. Si l'on ne s'intéresse pas au langage accepté par une TM M, mais seulement à la question de l'arrêt, on peut poser $\boxed{F = \emptyset}$ ou bien omettre F.
2. Si l'on veut imposer qu'une TM s'arrête au moment où elle accepte (c-à-d, où elle passe dans un état accepteur), on peut restreindre le type de δ à

$$(\boxed{(Q \setminus F)} \times \Gamma) \to (Q \times \Gamma \times \{-1, 0, 1\})$$

1. Une fonction $\boxed{f : \Sigma^* \to \Sigma^*}$ est dite $\boxed{\textit{Turing-calculable},}$ s'il existe une TM $M = (Q, \Sigma, \Gamma, \delta, q_0, \mathrm{B}, \emptyset)$ telle que pour tout $x, y \in \Sigma^*$

$$\boxed{\begin{array}{c} f(x) = y \\ \text{ssi} \\ (q_0, 1, x) \vdash^* (q, 1, y) \not\vdash \end{array}}$$

2. Soit $\mathrm{rep}(n)$ la suite $1 \cdots 1$ de longueur $n \in \mathbb{N}$.
Une fonction $\boxed{f : \mathbb{N}^k \to \mathbb{N}}$ est dite $\boxed{\textit{Turing-calculable},}$ s'il existe une TM $M = (Q, \{1\}, \{1, \#\} \cup \Gamma, \delta, q_0, \mathrm{B}, \emptyset)$ telle que pour tout $n_1 \ldots, n_k, m \in \mathbb{N}$

$$\boxed{\begin{array}{c} f(n_1 \ldots, n_k) = m \\ \text{ssi} \\ (q_0, 1, \mathrm{rep}(n_1)\# \cdots \# \mathrm{rep}(n_k)) \vdash^* (q, 1, \mathrm{rep}(m)) \not\vdash \end{array}}$$

4.2.4 Thèse (Church-Turing)

Tout *algorithme*, c-à-d toute *procédure effective*, peut être représenté par une TM.

4.3.1 Définition Soit L un langage sur l'alphabet Σ.

1. L est dit *décidable* (ou : *récursif*)
   s'il existe une TM M telle que $\mathrm{L}(M) = L$ et M est totale.
2. L est dit *semi-décidable* (ou : *récursivement énumerable*, ou : *r.e.*)
   s'il existe une TM M telle que $\mathrm{L}(M) = L$.
3. L est dit *co-semi-décidable* (ou : *co-r.e.*)
   si son complément $\overline{L}$ est semi-décidable.
4. L est dit *indécidable*
   si L n'est pas décidable.
5. L est dit *non-semi-décidable* (ou : *non-r.e.*)
   si L n'est pas semi-décidable.

4.3.2 Proposition Soit L un langage sur l'alphabet Σ.

1. Si L est décidable, alors son complément $\overline{L}$ est aussi décidable.
2. Si L et $\overline{L}$ sont semi-décidables, alors L est décidable.

Hiérarchie de Chomsky ...