
TD02 - Circuits de santé et spas

Exercice 1.*On joue aux bool ?*

1. Montrer que toute fonction booléenne peut être exprimée en n'utilisant que les connecteurs AND, OR et NOT.
2. Montrer qu'on peut même se passer de OR.
3. Peut-on exprimer n'importe quelle fonction booléenne en n'utilisant que le connecteur NOR? Quid de NAND?
4. Qu'en est-il des autres connecteurs binaires?
5. Une fonction booléenne $f : \{0, 1\}^n \mapsto \{0, 1\}$ sera dite monotone croissante si pour tout i , et pour tous $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$ on a :

$$f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n) \geq f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n).$$

Montrer que les fonctions monotones croissantes sont exactement celles qu'on peut exprimer en n'utilisant que les connecteurs AND, OR, et les constantes 0 et 1.

Exercice 2.*Si recuit*

1. Réaliser un circuit effectuant l'addition de deux entiers codés en binaire $x = \overline{x_{n-1} \dots x_0}$ et $y = \overline{y_{n-1} \dots y_0}$ en utilisant l'algorithme naïf appris à l'école. Quelle est la taille (nombre de portes) et la profondeur (plus long chemin entre une entrée et la sortie) de ce circuit?
2. On suppose ici que n est une puissance de 2. Réaliser un circuit effectuant l'addition de x et y en utilisant la méthode récursive suivante : on effectue à la fois le calcul de $\overline{x_{n-1} \dots x_{n/2}} + \overline{y_{n-1} \dots y_{n/2}}$ et le calcul de $\overline{x_{n-1} \dots x_{n/2}} + \overline{y_{n-1} \dots y_{n/2}} + 1$ et on sélectionne le bon en fonction de la retenue sortante du calcul de $\overline{x_{n/2-1} \dots x_0} + \overline{y_{n/2-1} \dots y_0}$. Quelle est la taille et la profondeur de votre circuit?
3. Soit (C_n) une famille de circuits de profondeur $O(\log(n)^k)$ pour une certaine constante k . Que pouvez-vous dire de la taille de C_n ?
On note NC^k la classe des fonctions calculables par une famille de circuits de taille polynomiale et de profondeur $O(\log(n)^k)$. Que pouvez-vous dire de la complexité de l'addition binaire de deux entiers?
4. On note AC^0 la classe des fonctions calculables par une famille de circuits booléens de taille polynomiale et de profondeur constante composés exclusivement de portes NOT, ainsi que de portes AND et OR ayant un nombre d'entrées arbitraire.
Montrer que l'addition binaire est dans AC^0 (conseil : exprimer le fait qu'une retenue sera présente sur le i -ème bit à l'aide d'une formule logique).
Que devient la classe AC^0 si on supprime la condition de taille polynomiale?
5. Montrer que $AC^0 \subseteq NC^1$.

Exercice 3.*Bien habillé*

On rappelle qu'un *système acceptable de programmation* est une énumération (surjective) des fonctions calculables : $\varphi_0, \varphi_1, \dots, \varphi_n, \dots$ vérifiant :

- *universalité* : il existe une fonction récursive totale f telle que $\forall n, x, f(\langle n, x \rangle) = \varphi_n(x)$. On note généralement u un numéro tel que $f = \varphi_u$.
- *s-n-m* : pour tous n et m , il existe une fonction récursive totale s_n^m telle que

$$\forall a, x_1, \dots, x_n, y_1, \dots, y_m, \varphi_{s_n^m(\langle a, x_1, \dots, x_n \rangle)}(\langle y_1, \dots, y_m \rangle) = \varphi_a(\langle x_1, \dots, x_n, y_1, \dots, y_m \rangle).$$

1. Montrer qu'on peut se contenter de s_1^1 .

2. Montrer qu'il existe une fonction totale c telle que pour tous a et b , $\varphi_{c(\langle a, b \rangle)} = \varphi_a \circ \varphi_b$.

3. Soit une énumération vérifiant l'universalité et l'existence d'une telle fonction c . Nous allons montrer l'existence de s_1^1 .

Soient $\varphi_p : y \mapsto \langle 0, y \rangle$, $\varphi_q : \langle x, y \rangle \mapsto \langle x + 1, y \rangle$. En utilisant ces fonctions, montrer qu'il existe une fonction récursive totale R telle que $\forall n, y, \varphi_{R(n)}(y) = \langle n, y \rangle$.

Conclure.