

Modèles de calcul - TD1

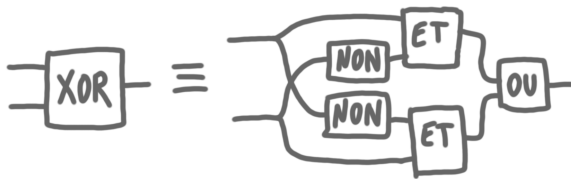
Exercice 1 Circuits : des modèles équivalents

- On fixe un nombre n d'entrées. On considère des circuits avec des portes ET, OU et NON. Montrer que l'ajout des portes suivantes ne rend pas le modèle de calcul plus puissant/expressif. (On le prouve directement, sans utiliser de théorème du cours.)
 - La porte XOR (qui prend deux entrées, et sort 1 si et seulement si les entrées sont différentes).
 - La porte ET avec un nombre arbitraire d'entrées (qui sort 1, si et seulement si toutes les entrées sont 1).
- Quel théorème du cours permet de conclure pour n'importe quelle porte supplémentaire ?

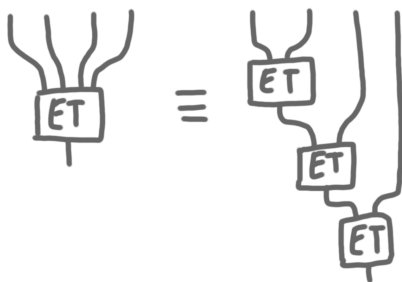
Solution de l'exercice 1

- On montre que chaque nouvelle porte peut être remplacée par un circuit utilisant seulement les portes de base. Si on prend un circuit utilisant une nouvelle porte, et qu'on remplace chaque nouvelle porte par le petit circuit, alors on calcule exactement la même fonction, donc on a la même puissance de calcul.

- Pour le XOR :



- Pour le ET avec un nombre arbitraire d'entrée :



- Le théorème du cours qui dit que toutes les fonctions à n variables peuvent être calculées par un circuit booléens (ET, OU, NON) à n entrées.

Exercice 2 Circuits : un modèle plus faible

On fixe un nombre n d'entrées, montrer que les circuits utilisant seulement le ET et le OU ne peuvent pas calculer n'importe quelle fonction.

Solution de l'exercice 2

Prenons la fonction f qui vaut 1 si et seulement si toutes les entrées sont à 0. Supposons qu'il y ait un circuit qui calcule f . Quand on met le circuit sur l'entrée 0^* , alors en sortie de chaque porte on a toujours des 0, donc on aura jamais le 1 souhaité.

Exercice 3 Circuits : codage en binaire

Décrire informellement un algorithme d'encodage qui prend en entrée un circuit et retourne une chaîne binaire telle que :

- Deux circuits différents ont forcément des encodages différents
- Si le circuit a s portes, alors la chaîne a de l'ordre de $s \log s$ bits (disons inférieur à $10s \log s$).

Solution de l'exercice 3

Rapidement :

1. On numérote les portes et on écrit le circuit comme une série de description de portes, par exemple :
porte $i = ET(\text{porte } j, \text{ porte } k)$.
2. On écrit chaque caractère en code ascii binaire, et on concatène.

Pour coder les numéros de portes, on utilise des nombres entre 1 et $\log s$, et pour les quelques caractères, c'est constant. Pour s portes, on a du $s \log s$, avec une constante inférieure à 10.

Exercice 4 Automates : déterminisme et non déterminisme

Pour chacun des deux automates de Figure 1, dire s'il est déterministe ou pas. S'il ne l'est pas, proposer un automate équivalent.

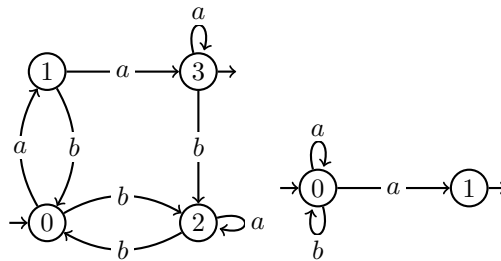
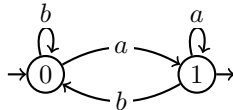


FIGURE 1 – Automates de l'exercice 4 .

Solution de l'exercice 4

Le premier est déterministe. Pour le deuxième un automate déterministe est :



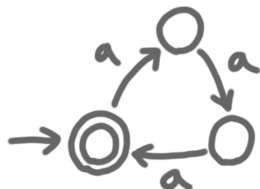
Exercice 5 Automates : modèles plus faible

On considère le langage des mots utilisant seulement la lettre a , et dont la longueur est un multiple de 3 (par exemple le mot vide, aaa et $aaaaaa$).

1. Montrer que ce langage est reconnaissable par un automate déterministe.
2. (*) Montrer que si on se restreint aux automates finis à deux états (avec un seul état de départ), alors on ne peut pas reconnaître ce langage.

Solution de l'exercice 5

1. L'automate :



2. Nommons les deux états 1 et 2, avec 1, l'état de départ. Comme le mot vide est accepté, l'état 1 est aussi un état final. Si il y a une autoboucle sur l'état 1, alors le mot a est accepté, contradiction. Sinon, le mot aaa doit forcément être reconnu par un run dont la liste des états est : 1221. Et donc il y a une auto-boucle sur l'état 2. Mais alors le mot $aaaa$ est aussi accepté car le run 12221 est possible.

Exercice 6 Circuits : borne inférieure du nombre de portes (*)

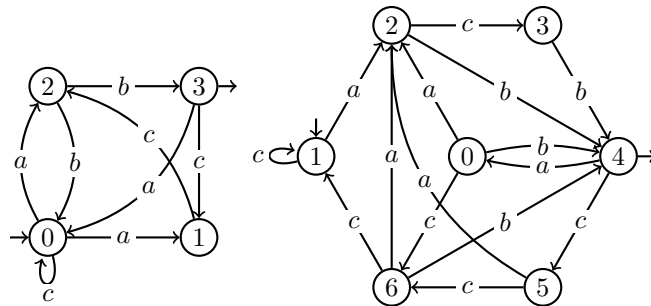
Montrer que pour tout n assez grand, il existe une fonction f à n variables, tel que le plus petit circuit qui calcule f a besoin d'au moins $2^{n/2}$ portes.

Solution de l'exercice 6

Supposons que pour chaque fonction f , il existe un circuit C_f avec strictement moins que $2^{n/2}$ portes. On code chacun de ces circuits sur $10 \cdot 2^{n/2} \cdot (n/2)$ bits grâce à un exercice précédent. Comme les fonctions sont différentes, les circuits doivent être différents, donc les encodages doivent être différents. Avec $10 \cdot 2^{n/2} \cdot (n/2)$ bits, on a au plus $2^{10 \cdot 2^{n/2} \cdot (n/2)}$ encodages différents. Donc il devrait y avoir au plus ce nombre de fonctions. Mais il y a 2^{2^n} fonction booléennes à n variables, ce qui est strictement plus grand, pour n assez grand.

Exercice 7 Automates : déterminisme et non déterminisme II

Les mots $acab$ et $abcabcb$ appartiennent-ils au langage accepté par ces automates ?



Solution de l'exercice 7

Le premier n'est reconnu par aucun, le deuxième par les deux. En fait, celui de droite est le déterminisé de celui de gauche. On voit que c'est beaucoup plus galère dans un non-déterministe.