Les trois parties sont indépendantes.

Partie 1 : circuits logiques

▶ Parmi les instructions dont dispose un ordinateur, on trouve `INR` (incrémentation) : cette instruction permet d'ajouter 1 à une case mémoire. Si cet ordinateur possède une architecture n bits, l'addition est effectuée modulo 2^n. On se propose d'étudier un circuit logique effectuant cette opération, dans les cas $n = 1$ et $n = 2$.

▶ On note $\mathcal{B} = \{0,1\}$. Cet ensemble est muni des deux lois $+$ et $\cdot$ décrites par les tables ci-dessous :

$+$	0	1
0	0	1
1	1	1

$\cdot$	0	1
0	0	0
1	0	1

Si $x \in \mathcal{B}$, on note $\overline{x} = 1 - x$.

▶ Les portes logiques *élémentaires* sont la porte ET à deux entrées, la porte OU à deux entrées, et la porte NON (à une entrée) ; elles sont décrites dans la figure 1.

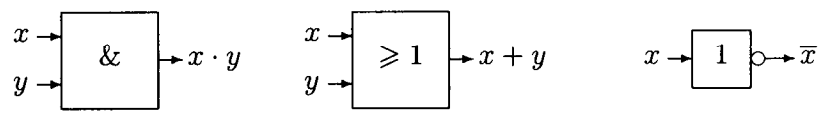


Figure 1: les trois portes logiques élémentaires

Question 1 Lorsque $n = 1$, quel circuit très simple réalise l'incrémentation ?

▶ Dans le cas $n = 2$, le circuit logique qui nous intéresse possède deux entrées e_0 et e_1, et deux sorties s_0 et s_1. La fonction de ce circuit est parfaitement définie par la relation :

$$s_0 + 2s_1 \equiv e_0 + 2e_1 + 1 \pmod 4$$

Question 2 Dressez une table donnant les valeurs de s_0 et s_1 pour chacun des quatre cas de figure possibles.

Question 3 Donnez des formules exprimant s_0 et s_1 en fonction de e_0 et e_1.

Question 4 Dessinez le circuit logique d'incrémentation ; vous placerez les entrées sur le côté gauche de la figure, et les sorties sur le côté droit.

Partie 2 : automates finis

▶ Dans cette partie, Σ désigne l'alphabet $\{a,b\}$. Soit L un langage sur cet alphabet ; on note $\Phi(L)$ le langage déduit de L en «ajoutant» une lettre a en fin de chacun de ses mots :

$$u \in \Phi(L) \iff \exists v \in L \text{ tel que } u = va$$

Question 5 Existe-t-il des langages finis tels que $\Phi(L) = L$?

Question 6 On note $L = \{a, ab, aaa\}$. Dessinez un automate fini reconnaissant L, puis un automate fini reconnaissant $\Phi(L)$.

Question 7 On note M le langage constitué des mots contenant un nombre pair de lettres b. M est-il rationnel ? Dessinez un automate fini reconnaissant $\Phi(M)$

Question 8 Soit L un langage rationnel. Montrez que $\Phi(L)$ est lui aussi un langage rationnel.

Partie 3 : arbres binaires

▶ On considère dans cette partie des arbres binaires de recherche, dont les nœuds sont étiquetés par des entiers, et les feuilles ne portent pas d'information. On rappelle la propriété caractéristique d'un arbre binaire de recherche : l'étiquette de chaque nœud majore strictement toutes les étiquettes des nœuds du sous-arbre gauche et minore strictement toutes les étiquettes des nœuds du sous-arbre droit. La figure 2 présente un tel arbre, dans lequel on n'a pas fait figurer les feuilles par souci de simplification.

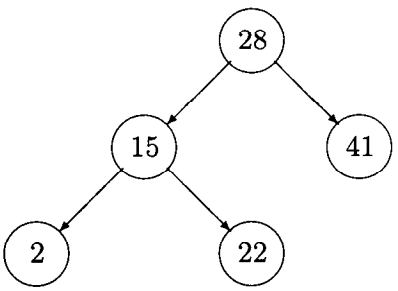


Figure 2: un arbre binaire de recherche.

▶ Soient a un arbre binaire et s un entier. On note $\mathcal{T}_s(a)$ l'arbre binaire déduit de a par l'opération consistant à ajouter 1 à chaque étiquette de a au moins égale au seuil s. La figure 3 représente l'image par $\mathcal{T}_{20}$ de l'arbre binaire présenté à la figure 2.

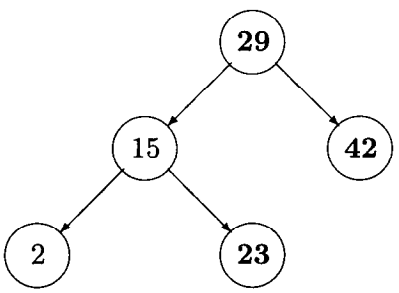


Figure 3: l'image par $\mathcal{T}_{20}$ de l'arbre binaire de la figure 2.

Question 9 Montrez que si a est un arbre binaire de recherche, alors $\mathcal{T}_s(a)$ est lui aussi un arbre binaire de recherche.

Programmation en Caml

Cette partie ne s'adresse qu'aux étudiants qui programment en langage Caml.

▶ Pour décrire les arbres binaires de recherche étiquetés par des entiers, on définit le type Caml suivant :

```
type abr = F | N of abr * int * abr;;
```

L'interprétation de ce type est la suivante : `F` est un arbre binaire réduit à une feuille ; `N(g,x,d)` est un arbre binaire dont la racine est étiquetée x, dont le sous-arbre gauche est g et dont le sous-arbre droit est d. Par exemple, l'arbre binaire de la figure 2 est décrit par :

```
N(N(N(F,2,F),15,N(F,22,F)),28,N(F,41,F))
```

Question 10 Rédigez en langage Caml une fonction de signature :

```
compte : int -> abr -> int
```

spécifiée comme suit : `compte s a` compte le nombre de nœuds de l'arbre a, dont l'étiquette est au moins égale à s.

Question 11 Rédigez en langage Caml une fonction de signature :

```
incr : int -> abr -> abr
```

spécifiée comme suit : `incr s a` construit l'arbre $\mathcal{T}_s(a)$.

Programmation en Pascal

Cette partie ne s'adresse qu'aux étudiants qui programment en langage Pascal.

▶ Pour décrire les arbres binaires de recherche étiquetés par des entiers, on définit le type Pascal suivant :

```
type
 Arbre = ^Noeud;
 Noeud = record
    etiquette : integer;
    gauche, droit : Arbre
   end;
```

Un arbre réduit à une feuille sera représenté par la constante `nil`.

Question 10 Rédigez en langage Pascal une fonction d'en-tête :

```
function compte(s:integer; a:Arbre):integer;
```

spécifiée comme suit : `compte(s,a)` compte le nombre de nœuds de l'arbre a, dont l'étiquette est au moins égale à s.

Question 11 Rédigez en langage Pascal une procédure d'en-tête :

```
procedure incr(s:integer; a:Arbre);
```

spécifiée comme suit : `incr(s,a)` modifie sur place l'arbre a pour le transformer en l'arbre $\mathcal{T}_s(a)$.

FIN