Correction TD8

Exo 1

1/ Les longueurs en entrées doivent être positives
Il faut aussi que la somme de deux côtés soit suppérieur au troisième

2/ 3 entrées sur 2^{32} possibilités correspond à $(2^{32})^3$ possibilités
Pour convertir le résutat en années on le divise par :
la fréquence pour obtenir le résutat en secondes,
puis par $\underbrace{60}_{\text{secondes par minute}} \times \underbrace{60}_{\text{minutes par heure}} \times \underbrace{24}_{\text{heure par jour}} \times \underbrace{365,25}_{\text{jour par an}}$

$$\frac{(2^{32})^3}{10^9 \; 60^2 \; 24 \times 365,25} \simeq 2,5 \times 10^{12} \text{ ans de calculs}$$ soit 2500 milliards d'années

3/ On propose les classes binaires : $a < 0$, $b < 0$, $c < 0$
$b+c > a$, $a+c > b$, $a+b > c$
$a = b$, $b = c$, $a = c$
$a \neq b$, $b \neq c$, $a \neq c$

C'est le contraire de la ligne précédente, donc c'est inutile

↳ Il reste 9 classes binaires soit $\overbrace{512}^{=2^9}$ possibilités pour le produit cartésien

4/ Pour réduire le nombre de classes, on peut utiliser
une classe " $a<0$ OU $b<0$ OU $c<0$ " c_1
une classe " $b+c>a$ OU $a+c>b$ OU $a+b>c$ " c_2

Notons $V \equiv a>0$ ET $b>0$ ET $c>0$ ET $b+c \leq a$ ET $a+c \leq b$ ET $a+b \leq c$

On introduit les classes
- V ET $a \neq b$ ET $b \neq c$ ET $a \neq c$ $\quad c_3$
- V ET $a = b$ ET $b \neq c$ $\quad c_4$
- V ET $b = c$ ET $a \neq b$ $\quad c_5$
- V ET $a = c$ ET $a \neq b$ $\quad c_6$
- V ET $a = b$ ET $b = c$ $\quad c_7$

Nous n'avons que 7 cas à tester

On génère les tests

$a = -3$	$b = 1$	$c = 2$	(c_1)
$a = 42$	$b = 3$	$c = 21$	(c_2)
$a = 1$	$b = 2$	$c = 3$	(c_3)
$a = 3$	$b = 3$	$c = 2$	(c_4)
$a = 2$	$b = 3$	$c = 3$	(c_5)
$a = 1$	$b = 2$	$c = 1$	(c_6)
$a = 4$	$b = 4$	$c = 4$	(c_7)

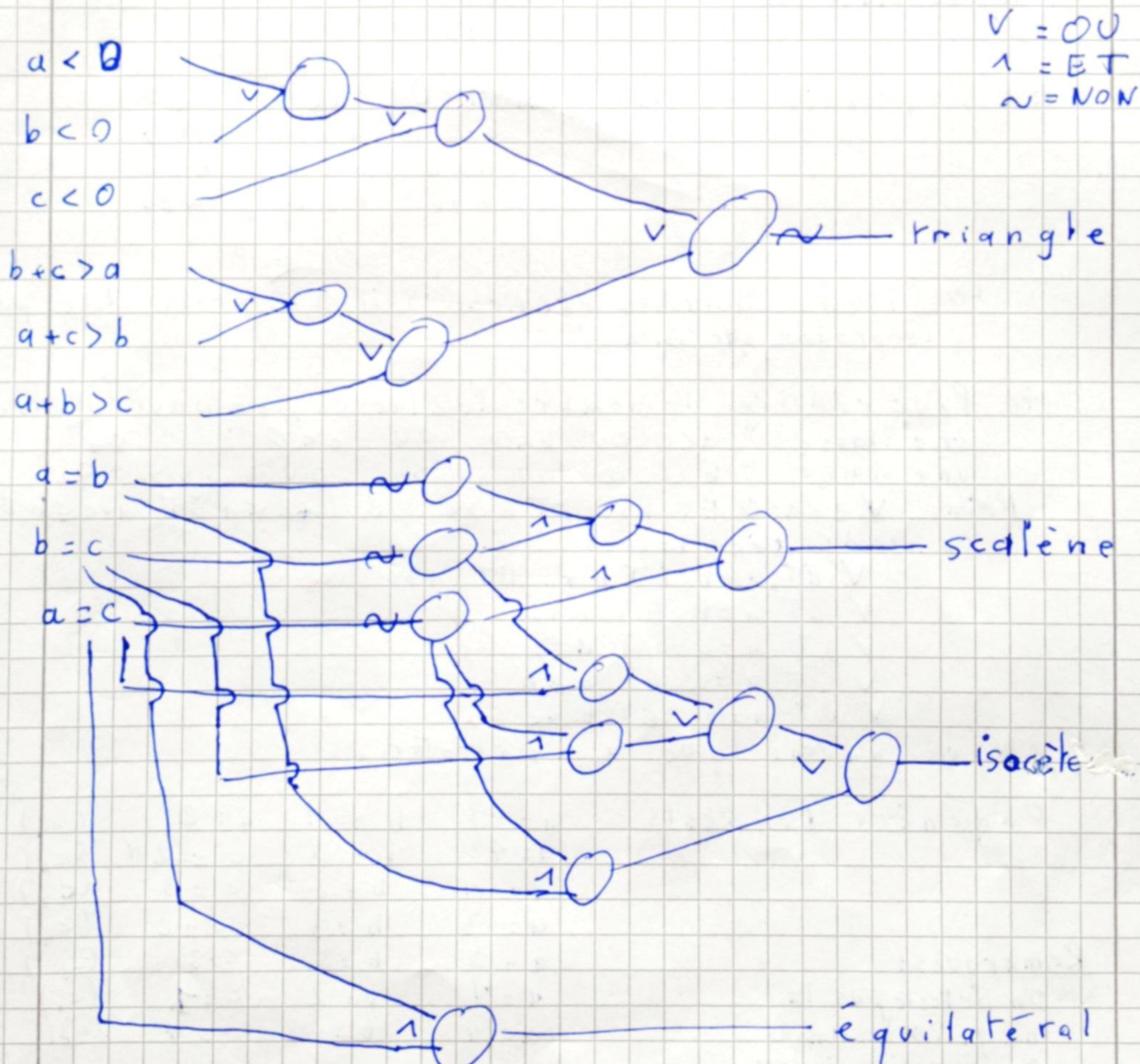
Remarques :
- On ne teste pas la négation des classes
- On pourrait scinder c_1 et c_2 en 3 classes

5/

a < 0	1	*	*	*	*	*	0	0	0	0	0
b < 0	*	1	*	*	*	*	0	0	0	0	0
c < 0	*	*	1	*	*	*	0	0	0	0	0
b+c ⩾ a	*	*	*	0	*	*	1	1	1	1	1
a+c ⩾ b	*	*	*	*	0	*	1	1	1	1	1
a+b ⩾ c	*	*	*	*	*	0	1	1	1	1	1
a = b	*	*	*	*	*	*	0	1	0	0	1
b = c	*	*	*	*	*	*	0	0	1	0	1
c = a	*	*	*	*	*	*	0	0	0	1	1
triangle	0	0	0	0	0	0	1	1	1	1	1
scalène	*	*	*	*	*	*	1	0	0	0	0
isocèle	*	*	*	*	*	*	0	1	1	1	1
équilatéral	*	*	*	*	*	*	0	0	0	0	1

Ici avec la table de décision on fait plutôt plein de classes, mais les "*" permettent de réduire le nombre de possibilités du produit cartésien. On a aussi supprimé les cas impossible (ex a=b, b=c, a≠c)

Voici le graphe de décision :

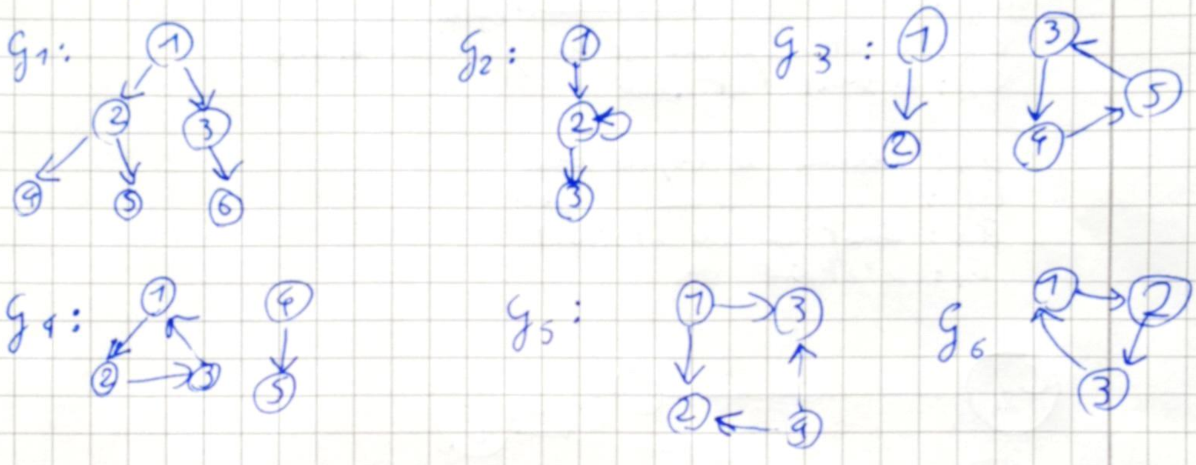


Exo 2

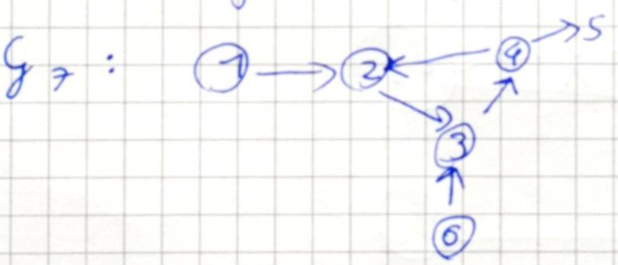
Ici il faut trouvé les cas caractéristiques : (on va supposer que le graphe à tester est orienté

- le graphe est un arbre/un cycle
- le graphe contient au moins un arc d'un noeud sur lui même
- le graphe contient deux composantes connexes en particulier il faut penser que la fonction puisse ne traiter qu'une seule composante
- le graphe ne possède pas de cycles, mais si les arcs n'étaient pas orientés ce serait le cas

On propose alors les DTs :



On doit aussi ajouter un/des cas classiques :



Exo 3

1/ Chaque caractéristique possède deux "options", et il y en a n, donc le nombre de possibilités est de : 2^n

On effectue un test par possibilité ce qui nous donne 2^n tests (On dit aussi qu'il y a un nombre exponentiel de tests à faire en fonction du nombre de caractéristique.

2/ Dans l'approche pairwise on a besoin de moins de $2n$ tests : pour illustrer l'explication on suppose que les deux éléments de S_i sont vrais et faux (1 et 0), il suffira d'adapter si ce n'est pas le cas :

On choisit les combinaisons de caractéristique

a_k = " $c_i = 0$ si $i \leq k$, 1 sinon "
b_k = " $c_i = 1$ si $i \leq k$, 0 sinon "   pour $k \in [\![1, n]\!]$
(on suppose $i \in [\![1, n]\!]$)

Vérifions que chaque combinaisons est possible pour deux paires quelconques c_α et c_β (avec $\alpha < \beta$ quitte à les inverser)

($c_\alpha = 0$ et $c_\beta = 0$) dans a_β ($c_\alpha = 0$ et $c_\beta = 1$) dans a_α
($c_\alpha = 1$ et $c_\beta = 0$) dans b_α ($c_\alpha = 1$ et $c_\beta = 1$) dans b_β

(Je recommande de visualiser cela sur un tableau avec $n = 5$ par exemple)

Note : on dit que le nombre de tests requis est linéaire en le nombre de caractéristiques

Note 2 : il y a peut-être moyen de faire moins de tests mais l'ordre de grandeur est satisfaisant

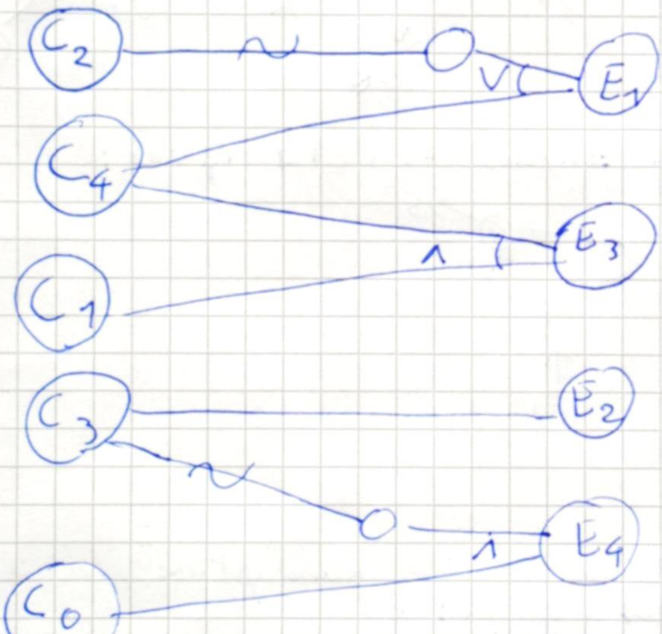
Conclusion : l'approche pairwise réduit drastiquement le nombre de tests requis (s'il y a 6 caractéristiques par exemple, on le réduit de 64 à 12)

Exo 4 :

Choisissons les effets et les causes :

C_0 : la couleur détectée est verte
C_1 : la couleur détectée est rouge
C_2 : — ——————— bleu
C_3 : l'interrupteur est en position haute
C_4 : le bruit est élevé

E_1 : allume le voyant bleu
E_2 : ——————— vert
E_3 : émet un cri strident
E_4 : s'éteint



C_0	1	*	0	0	0
C_1	*	*	*	*	1
C_2	*	*	*	0	*
C_3	0	1	*	*	*
C_4	*	*	1	*	1
E_1	*	*	1	1	*
E_2	*	1	*	*	*
E_3	*	*	*	*	1
E_4	1	0	0	0	0

Dans cette table si le robot s'éteint, il n'est pas logique de demander un autre effet, d'où les * sur E_1, E_2, E_3. Sinon on peut combiner certaines colonnes

Il est possible d'ajouter certaines colonnes à cette table