

## **exercice01**

### **a) les sources du problème d'apprentissage**

1. La première correspond au nombre de cycles durant lesquels la performance de l'agent reste sous-optimale pour la tâche de décision donnée.
2. La seconde correspond aux ressources de calcul nécessaires durant chaque cycle à l'agent pour réviser sa stratégie et choisir une action.

### **b) une définition du modèle d'apprentissage**

modèle d'apprentissage est un cadre formel donnant une mesure de ces deux sources de complexité. Les observations, les actions et le feedback peuvent influencer sur la difficulté de l'apprentissage

### **c) Expliquer comment les observations, les actions et le feedback peuvent influencer sur la difficulté de l'apprentissage**

not fished

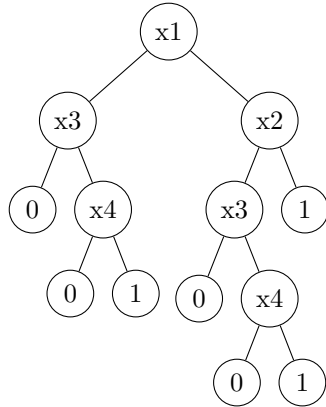
## **exercice2**

### **Donner avec explication les composants du problème l'apprentissage supervisé de porte logique XOR**

Une fonction XOR doit renvoyer une valeur vraie si les deux entrées ne sont pas égales et une valeur fausse si elles sont égales -l'espace des entrées est  $X = \{0, 1\}^n$  et l'espace des sorties est  $Y = \{0, 1\}$ .

### exercice03

x1 x2 x3 x4



### exercice04

a) la différence entre une requête d'appartenance et une requête d'équivalence est

1. Une requête d'appartenance (MQ) associe 'a une instance  $x$  posée par l'apprenant la réponse oui si  $h(x) = 1$ , et non sinon.
2. Une requête d'équivalence (EQ) associe 'a une hypothèse  $h$  posée par l'apprenant la réponse oui si  $h = h$ , et non sinon

b) Si pour toute instance  $(x1, x2, x3)$  quelle est l'hypothèse (le concept) le plus spécifique

est  $h1((x1, x2, x3)) = x1$

c) Si  $h(x1, x2, x3) = x1 \ x2$  et la requête est  $h((0, 1, 1)) = 0$

1. le type de cette requête est requête d'appartenance (MQ)
2. et si  $h^*((0, 1, 1)) = 1$ , requête d'équivalence (EQ)
3. Si le contre exemple est  $(0, 1, 0)$  mettre 'a jour  $h$ .  $h((0, 1, 0)) = 1$

## exercice05

## exercice06

### la différence entre le modèle agnostique et le modèle PAC

Dans le modèle “agnostique”, la distribution  $D$  est arbitraire, ce qui implique qu’il n’existe a priori aucune dépendance fonctionnelle entre une instance  $x$  et une décision  $y$  dans un exemple tiré dans  $D$ . En revanche, dans le modèle (PAC), nous supposons qu’il existe une dépendance fonctionnelle gouvernée par une fonction cible

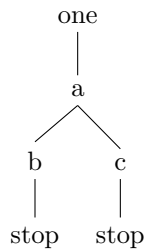
## exercice01

1.  $A \parallel B = B \parallel A \rightarrow$  Commutativité
2.  $A \parallel \text{stop} = \text{stop} \parallel A = A \rightarrow$  Zéro absorption
3.  $A \parallel (B \parallel C) = (A \parallel B) \parallel C \rightarrow$  Associativité

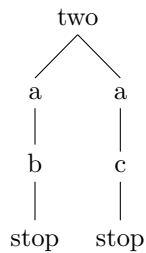
## exercice02

## exercice03

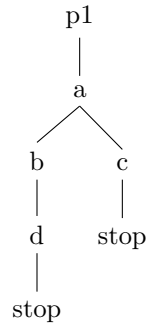
process one [a,b,c] a; (b; stop  $\parallel$  c; stop) endproc



process two [a,b,c] a; b; stop  $\parallel$  a; c; stop Endproc



**P1 := a; (b; d; stop [] c; stop)**



## exercice04

**Ecrire des spécifications lotos pour les circuits logiques:  
and, or et xor**

### AND

```

specification circuitlogiqueAND [a,b,c] :noexit
type BIT is
sorts BIT
opns 0 (*! constructor *),
1 (*! constructor *) : -! BIT
and : BIT ,BIT-! BIT
eqns
ofsort BIT
and (0,0) = 0;
and (0,1) = 0;
and (1,0) = 0;
and (1,1) = 1;
endtype
behaviour
gate_and[a,b,c]
where
processgate_AND[a,b] : noexit :=
a?aa : Bit; b?bb : Bit; c!and(aa,bb); stop
endproc
endspec
  
```

## OR

```
specification circuitlogiqueOR [a,b,c] :noexit
type BIT is
sorts BIT
opns 0 (*! constructor *),
1 (*! constructor *) : -! BIT
or : BIT ,BIT-! BIT
eqns
ofsort BIT
or (0,0) = 0;
or (0,1) = 1;
or (1,0) = 1;
or (1,1) = 1;
endtype
behaviour
gateOR[a, b, c]
where
processgateOR[a, b] : noexit :=
a?aa : Bit; b?bb : Bit; c!or(aa, bb); stop
endproc
endspec
```

## XOR

```
specification circuitlogiqueXOR [a,b,c] :noexit
type BIT is
sorts BIT
opns 0 (*! constructor *),
1 (*! constructor *) : -! BIT
xor : BIT ,BIT-! BIT
eqns
ofsort BIT
and (0,0) = 0;
and (0,1) = 1;
and (1,0) = 1;
and (1,1) = 0;
endtype
behaviour
gateXOR[a, b, c]
where
processgateXOR[a, b] : noexit :=
a?aa : Bit; b?bb : Bit; c!xor(aa, bb); stop
endproc
endspec
```

