



Promo : 2^{ème} Tronc commun licence Informatique

Module : Bases de données

Durée : 01h30

Date : 29 Mai 2023

Epreuve de moyenne durée

Partie I (05.00 pts)

(Q1). Soient deux relations R et S de clés primaires respectives ar et as. Définir la clé primaire de chaque relation résultat des opérations suivantes :

- Restriction quelconque de R
- Projection quelconque de R
- Jointure de R et S
- Différence de R et S

(Q2). On considère la relation **VEHICULE** (NV, Type, Marque, Puissance, Couleur) munie de l'ensemble des dépendances fonctionnelles $F = \{NV \rightarrow Type ; Type \rightarrow Marque ; Type \rightarrow Puissance ; NV \rightarrow Couleur\}$

Soit la décomposition de la relation **VEHICULE** en R1 (NV, Type), R2 (Type, Puissance, Couleur) et R3 (Type, Marque). Cette décomposition est-elle SPD (Sans Perte de Dépendances fonctionnelles) ? Justifier votre réponse ?

Partie II

Exercice I (06.00 pts)

Soit le schéma d'une partie de la base de données de la gestion du laboratoire de recherche :
Laboratoire (Code-lab, Nom-Lab, Sigle, Nom-directeur, Date-1création, date-renouvellement, C-Directeur)

Projet (C-projet, titre-projet, c-chef-Projet, code-lab)

Chercheur (C-chercheur, Nom, prénom, date-nais, Email, Grade)

Affectation (C-chercheur, C-projet, Code-Lab, date-Rattachement)

Collaboration (Code-Lab, collaborateur, C-Projet)

Collaborateur (Collaborateur)

Hypothèses :

- Un laboratoire est créé une seule fois avec un seul directeur, mais son renouvellement peut être modifié à des dates différentes.
- Chaque projet a un directeur et ne peut être rattaché qu'à un seul laboratoire.
- La date de rattachement correspond à la date où un chercheur est rattaché à un projet.
- Un collaborateur ne collabore qu'avec un laboratoire que pour un et un seul projet (les directeurs et les chefs de projet sont des chercheurs).

Question

Pour chaque relation obtenue, préciser dans le tableau [1] sa forme normale.

Exercice II (09.00 pts)

Soit le schéma suivant :

Voyage (NumV, NmV, TypePens)

Ville (NumVil, Nomv, Superf)

Voyag-Tarif(CT, Dtadeb, prix, #NumV)

Etape(Net, duree, #Numv, #NumVil)

Questions

Exprimez les requêtes suivantes en Sql :

Q1. Donner, pour chaque voyage, sa durée totale ainsi que son nombre d'étapes.

Q2. Donner, pour chaque voyage, sa durée totale ainsi que son nombre d'étapes (seulement pour les voyages de numéro supérieur à 5 et ayant plus de 3 étapes).

Q3. Nombre de départs pour chacun des voyages dont le nom commence par 'C' ou par 'R' ?

Q4. Numéros des voyages coûtant le même prix (quelque soit la date de départ), pour chaque voyage?

Q5. Numéros de voyages coûtant moins de 50000DA.

Q6. Donner, pour chaque date de départ, le nombre de voyages qui partent à cette date.

Q7. Nous supposons que la requête suivante [**la durée moyenne des étapes par voyage**] se répète fréquemment, que ferons-nous pour éviter de l'écrire tant de fois ?

CORRIGE TYPE DE L'EXAMEN SEMESTRIEL BDD L2 INFO

PARTIE I [05.00 Pts]

(Q1). Soient deux relations R et S de clés primaires respectives ar et as. Définir la clé primaire de chaque relation résultat des opérations suivantes :

- a. Restriction quelconque de R **(0.5 Pts)**
Clé : ar
- b. Projection quelconque de R **(0.5 Pts)**
Selon les attributs sur lesquels on fait la projection, si la clé y figure dont ar.
- c. Jointure de R et S **(0.5 Pts)**
ar+as
- d. Différence de R et S **(0.75 Pts)**
1^{er} cas : si les relations ont le même schéma alors ar//as
2^{ème} cas : si les relations n'ont pas le même schéma → on ne peut pas faire l'opération d' U

(Q2). On considère la relation VEHICULE (NV, Type, Marque, Puissance, Couleur) munie de l'ensemble des dépendances fonctionnelles $F = \{NV \rightarrow \text{Type} ; \text{Type} \rightarrow \text{Marque} ; \text{Type} \rightarrow \text{Puissance} ; NV \rightarrow \text{Couleur}\}$

Soit la décomposition de la relation VEHICULE en R1 (NV, Type), R2 (Type, Puissance, Couleur) et R3 (Type, Marque). Cette décomposition est-elle SPD (Sans Perte de Dépendances fonctionnelles) ? Justifier votre réponse ?

On va tirer les DFs à partir de chaque relation

R1 (NV, Type) alors $F_1 = \{NV \rightarrow \text{Type}\}$ **(0.25 Pts)**

R2 (Type, Puissance, Couleur) $F_2 = \{\text{Type} \rightarrow \text{Puissance} ; \text{Type} \rightarrow \text{Couleur}\}$ **(0.25 Pts)**

R3 (Type, Marque) $F_3 = \{\text{Type} \rightarrow \text{Marque}\}$ **(0.25 Pts)**

Si on fera l'union de l'ensemble des DFs [fermeture] **(1.00 Pts)**

$(F_1 \cup F_2 \cup F_3)^+ = \{NV \rightarrow \text{Type} ; \text{Type} \rightarrow \text{Puissance} ; \text{Type} \rightarrow \text{Couleur} ; \text{Type} \rightarrow \text{Marque} ; \underline{NV \rightarrow \text{Couleur}}\}$ = l'ensemble initial F^+ donné initialement

Par conséquent : La décomposition de R en R1, R2 et R3 **préserve** les dépendances fonctionnelles.

De la, on s'assure que la décomposition est sans perte d'information comme suit : **(1.00 Pts)**

R12 = Jointure (R1, R2, R1.type =R2.type)

R123 = Jointure (R12, R3, R12.type = R3.type)

Si R = R123 (Exactement la même extension ou les mêmes tuples), la décomposition sera sans perte d'information.

PARTIE II

Exo II [09.00 PTS]

Exprimez les requêtes suivantes en sql :

Q1. Donner, pour chaque voyage, sa durée totale ainsi que son nombre d'étapes
(1.25 Pts)

.....
SELECT
numVoyage,SUM(duree) AS "durée totale" ,COUNT(*) AS
"nb étapes"
FROM Etape
GROUP BY numVoyage ;

Q2. Donner, pour chaque voyage, sa durée totale ainsi que son nombre d'étapes
(seulement pour les voyages de numéro supérieur à 5 et ayant plus de 3 étapes)
(1.25 Pts)

.....
SELECT
numVoyage,SUM(duree) AS "durée totale" ,COUNT(*) AS "nb
étapes"
FROM Etape
WHERE numVoyage > 5
GROUP BY numVoyage
HAVING COUNT(*) > 3

Q3. Nombre de départs pour chacun des voyages dont le nom commence par
'C' ou par 'R' ? (1.25 Pts)

.....
SELECT
V.numVoyage, COUNT(*)
FROM Voyage V, Tarif T
WHERE
V.numVoyage=T.numVoyage
AND (nomVoyage LIKE 'C%' OR nomVoyage LIKE 'R%')
GROUP BY V.numVoyage

.....
**Q4. Numéros des voyages coûtant le même prix (quelque soit la date de départ)
? (1.25 Pts)**

.....
**SELECT
numVoyage
FROM Tarif
GROUP BY numVoyage
HAVING MAX(prix) = MIN(prix)
Variante
HAVING COUNT(DISTINCT prix) = 1**

.....
Q5. Numéros de voyages coûtant moins de 50000DA (1.25 Pts)

.....
**SELECT numVoyage
FROM Tarif
WHERE prix<50000;**

.....
**Q6. Donner, pour chaque date de départ, le nombre de voyages qui partent à
cette date (1.25 Pts)**

.....
**SELECT
dateDeb, COUNT(*)
FROM Tarif
GROUP BY dateDeb**

.....
**Q7. Nous supposons que la requête suivante [la durée moyenne des étapes par
voyage] se répète fréquemment, que ferons-nous pour éviter de l'écrire tant de
fois ? (1.5 Pts)**

1^{ère} solution : on suppose que l'on cherche la moyenne des durées d'étapes
d'un voyage pour un seul voyage entré en paramètre de la procédure

.....
**CREATE procedure durEtapMoye (IN NV int, OUT moyET float)
Begin
 SELECT AVG (duree) into moyET
 FROM Etape WHERE NumV= NV
 GROUP BY NumV;
End;**

2^{ème} solution : on peut supposer aussi que l'on cherche la moyenne des durées d'étapes d'un voyage pour tout voyage insérer dans la table Etape. Dans ce cas là, on n'a pas besoin du paramètre d'entrée **numéro du voyage**.

```
.....  
CREATE procedure durEtapMoye (OUT moyET float )  
Begin  
    SELECT AVG (duree) into moyET  
    FROM Etape  
    GROUP BY NumV;  
End;  
.....
```

Exo I [06.00 Pts]

TABLEAU [1] (1.2 Pts pour chaque relation)

Nom relation	1FN Vrai Faux	2FN Vrai Faux	3FN Vrai Faux	Justification
Laboratoire	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input checked="" type="checkbox"/>	<p>1FN : * identifiant \exists (✓) - * Atomicité (✓) 2FN : * 1FN (✓) - DFs élémentaires (✓) 3FN : * 2FN (✓) - * DFs directe (x) Cod-Lab \rightarrow C-directeur</p>
Projet	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>	<p>1FN : * identifiant \exists (✓) - * Atomicité (✓) 2FN : * 1FN (✓) - DFs élémentaires (✓) 3FN : * 2FN (✓) - * DFs directe (✓)</p>
Chercheur	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>	<p>1FN : * identifiant \exists (✓) - * Atomicité (✓) 2FN : * 1FN (✓) - DFs élémentaires (✓) 3FN : * 2FN (✓) - * DFs directe (✓)</p>
Affectation	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input checked="" type="checkbox"/>	<input type="checkbox"/> <input checked="" type="checkbox"/>	<p>1FN : * identifiant \exists (✓) - * Atomicité (✓) 2FN : * 1FN (✓) - DFs élémentaires (x) C-chercheur, C-Projet \rightarrow Cod-lab [1] Or, c'est le C-projet \rightarrow Cod-Lab</p>
Collaboration	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>	<p>1FN : * identifiant \exists (✓) - * Atomicité (✓) 2FN : * 1FN (✓) - DFs élémentaires (✓) Car, selon l'hypothèse (4) de l'énoncé, la DF Cod-Lab, collaborateur \rightarrow C-Projet correcte</p>

Corrigé type examen DAW

Questions de cours (4pts)

1. \$_GET,
\$_POST.
2. Il suffit de déposer la description WSDL de votre service Web dans l'annuaire UDDI
3.

```
a:hover{  
    text-decoration: none;  
    color: white;  
    background: #8870FF;  
}
```

Exercice 1

1 – XML (2pts)

```
<?xml version="1.1" encoding="utf-8" ?>  
<Formation>  
    <Code></Code>  
    <Titre></Titre>  
    <Modules>  
        <Module></Module>  
        <Module></Module>  
        <Module></Module>  
    </Modules>  
    <Responsable>  
        <Nom></Nom>  
    </Responsable>  
</Formation>
```

2 – DTD (3pts)

```
<!ELEMENT Formation (Code,Titre,Modules,Responsable?) />  
<!ELEMENT Modules (Module+) />  
<!ELEMENT Responsable (Nom) />  
<!ATTLIST Responsable Date (#CDATA) />  
<!ELEMENT Code (#PCDATA) />  
<!ELEMENT Titre (#PCDATA) />  
<!ELEMENT Module (#PCDATA) />  
<!ELEMENT Nom (#PCDATA) />
```

3- XML Schema (3pts)

```
<?xml version="1.0" Encoding="ISO-8859-1" ? >  
<xsd:element name="formation">  
    <xsd:complexType>  
        <xsd:sequence>  
            <xsd:element name="code" type="xsd:string" />  
            <xsd:element name="titre" type="xsd:string" />  
            <xsd:element name="modules" type="xsd:string" />  
            <xsd:complexType>  
                <xsd:sequence>
```

```

        <xsd:element name="module" type="xsd:string" />
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="responsable" type="xsd:string" >
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="Nom" type="xsd:string" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

```

```
</xsd:element>
```

Exercise 2

1. (2pts)

```
function plus_petit_age(){
```

```
var table = document.getElementById('table_1');
```

```
var min = 10000;
```

```
var lignes = table.rows;
```

```
var age_actuel;
```

```
for(var i=1; i<lignes.length; i++){
```

```
age_actuel =
```

```
parseInt(lignes[i].firstElementChild.nextElementSibling.nextElementSibling.textContent);
```

```
if(age_actuel < min) {min = age_actuel;}
```

```
}
```

```
return min;
```

```
}
```

2. (3pts)

```
function supprimer_ligne(i){
```

```
var table = document.getElementById('table_1');
```

```
var lignes = table.rows;
```

```
if(i == 0){
```

```
alert("Impossible de supprimer l'entête du tableau");
```

```
} else if(i >= lignes.length){
```

```
alert("L'indice dépasse la taille du tableau");
```

```
} else {
```

```
lignes[i].parentNode.removeChild(lignes[i]);
```

```
}
```

```
}
```

3. (3pts)

```
document.getElementById("cellule_age_2").parentNode.nextElementSibling.firstElementChild.nextElementSibling.textContent='Akram';
```

University of Ain Temouchent Belhadj Bouchaïb
Faculty of Science and Technology
Department of Mathematics & Computer Science

Academic Year 2022-2023 L2/CS Networks

EMD (Durée : 1H30 Mn)

Exercice 1 (6pts)

1. Représentez sous forme d'un tableau à deux colonnes (Classe, Plage d'adresses) les adresses IP publiques.
2. Soit l'adresse IPv4 $172.16.0.0/22$:
 - Donner le masque de sous réseau correspondant.
 - Combien de sous-réseaux peut-on définir ?
 - Combien peut-on définir de machine par sous réseau ?
 - Soit l'adresse IPv4 $172.16.12.54$ avec le masque $255.255.255.240$:
 - Parmi les adresses suivantes, lesquelles sont des adresses hôtes valides sur le même réseau ?
 - (a) $172.16.12.64$
 - (b) $172.16.12.57$
 - Quelles sont les routes agrégées par l'adresse CIDR $197.50.0.0/19$.

Exercice 2 (6Pts)

La table de routage d'un routeur avec une interface $100.3.4.3$ contient les entrées énumérées dans le Tableau 1.

1. Pour chacune des destinations suivantes, spécifiez s'il est possible de router vers la destination en question. **Justifiez avec une phrase.**
2. Dessinez la topologie du réseau en question, sous la forme d'un réseau PacketTracer. Mentionnez toutes les adresses des interfaces des routeurs et les destinations jointes.
 - a) $221.3.4.1$
 - b) $100.66.85.66$
 - c) $199.22.1.9$
 - d) $222.10.10.7$
 - e) $222.0.44.44$
 - f) $134.6.12.15$
 - g) $22.55.4.56$

Exercice3 (8Pts)

— Décodez la trame ethernet ci-dessous :

AA AA AA AA AA AA AA AB 08 00 20 0A 70 66 08 00 20 0A AC 96
08 00 45 00 00 28 A6 F5 00 00 1A 06 75 94 C0 5D 02 01 84 E3 3D 05 00
15 0F 87 9C CB 7E 01 27 E3 EA 01 50 12 10 00 DF 3D 00 00 20 20 20
20 20 20 9B 52 46 43

Donner les valeurs des champs importants en décimal.

TABLE 1 – Table de routage

Réseau	Routeur prochain saut
100.0.0.0	Directement connecté
22.0.0.0	100.3.5.9
222.0.44.0	100.45.22.224
134.6.0.0	100.56.45.66
199.22.1.0	100.99.23.43

Solution

Exercice 1 (6Pts)

1. (2Pts)

Classe d'adresses	Plage d'adresses public
A	1.0.0.0 - 9.255.255.255 (0.25Pts) 11.0.0.0 - 126-255.255.255 (0.25Pts)
B	128.0.0.0 - 169.253.255.255 (0.25Pts) 169.254.0.1 à 169.254.255.255 (0.25Pts) 170.0.0.0 à 172.15.255.255 (0.25Pts) 172.32.0.0 - 191-255.255.255 (0.25Pts)
C	192.0.0.0- 192.167.255.255 (0.25Pts) à 192.169.0.0 - 223-255.255.255 (0.25Pts)

TABLE 2 – Plages d'adresses public.

2. — Avec l'adresse IPv4 172.16.0.0/22, on peut définir $2^6 = 64$ sous réseaux (**0.5Pts**)
— Masque : 255.255.252.0 (**0.25Pts**)
— Nombre de machine par sous réseau : $2^{10}-2 = 1022$ machines (**0.25Pts**)
3. 172.16.12.54/28 → 172.16.12.**0011**0110 → Le réseau est donc : 172.16.12.48/28 (**0.5Pts**)
4. 172.16.12.61 → 172.16.12.01000000 → est l'adresse du réseau situé après le réseau 172.16.12.48/28 → Elle n'est pas une adresse hôte. (**0.75Pts**)
5. 172.16.12.57 → 172.16.12.**0011**1001 → est une adresse de hôte valide sur le même réseau 17.16.12.48/28 (**0.75Pts**)
6. Routes agrégées : 197.50.0.0/24 → 197.50.31.0/24 (**1Pts**)

Exercice 2 (6Pts)

- a) 221.3.4.1 : **Non**, pas de route vers le réseau de classe C 221.3.4.0 (**0.5Pts**)
- b) 100.66.85.66 : **Oui**, route vers le réseau de classe A 100.0.0.0 (**0.5Pts**)
- c) 199.22.1.9 : **Oui**, route vers le réseau de classe C 199.22.1.0 (**0.5Pts**)
- d) 222.10.10.7 : **Non**, pas de route vers le réseau de classe C 222.10.10.0 (**0.5Pts**)
- e) 222.0.44.44 : **Oui**, route vers le réseau de classe C 222.0.44.0 (**0.5Pts**)

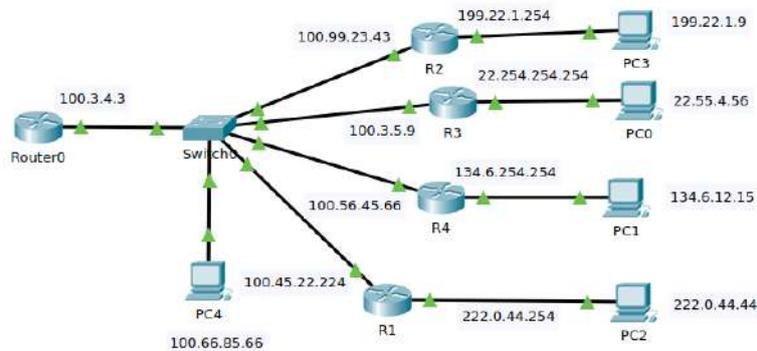


FIGURE 1 – Topologie du réseau. (2.5Pts)=0.23*11

- f) 134.6.12.15 : **Oui**, route vers le réseau de classe B 134.6.0.0 (0.5Pts)
- g) 22.55.4.56 : **Oui**, route vers le réseau de classe A 22.0.0.0 (0.5Pts)

Exercice 3 (8Pts)

- AA AA AA AA AA AA AA AB → Synchronisation+SFD (0.125Pts)
- 08 00 20 0A 70 66 → @MAC destinataire (0.25Pts)
- 08 00 20 0A AC 96 → @MAC émetteur (0.25Pts)
- 08 00 → Type (ici IPv4) (0.125Pts)
- **Entête IPv4 :**
- 4 → Version du protocole IP (IPv4). (0.125Pts)
- 5 → Longueur de l'en-tête (5*32 bits = 160 bits ou 5*4 octets = 20 octets) (0.125Pts)
- 00 → Pas de qualité de service (ToS) (0.125Pts)
- 00 28 → Longueur totale (ici 00 28 en hexadécimal, soit : 2*16 + 8 en décimal donc 40 octets). (0.125Pts)
- A6 F5 → ID du datagramme (numéro quelconque, ne sert que si le datagramme est amené à être fragmenté). (0.125Pts)
- 0000 → Drapeau + Déplacement (0=inutl, 0=DF [fragmentation autorisée]) 0=MF (pas de fragments à suivre, donc dernier fragment). 000000000000 = déplacement, c'est-à-dire position du 1er octet du fragment par rapport au 1er octet du datagramme initial. Ce fragment est le premier et le dernier du datagramme : il s'agit donc d'un datagramme non fragmenté. (0.5Pts)
- 1A → TTL (ici 1A = 1*16 + 10 = 26 routeurs ou secondes). (0.25Pts)
- 06 → Protocole (ici TCP). (0.5Pts)
- 75 94 → Bloc de contrôle d'erreur (sur l'en tête du datagramme seulement). (0.125Pts)
- C0 5D 02 01 → @IP émetteur 192.92.2.1 (0.25Pts)
- 84 E3 3D 05 → @ IP destinataire 132.227.61.5 (0.25Pts)
- **Entête TCP :**
- 00 15 0F 87 9C CB 7E 01 27 E3 EA 01 50 12 10 00 DF 3D 00 00

- 00 15 → port source, ici 21 donc serveur FTP **(0.25Pts)**.
- 0F 87 → port destination 3975, port quelconque du client **(0.25Pts)**.
- 9C CB 7E 01 → Numéro de séquence (numéro du 1^{er} octet émis **(0.25Pts)**).
- 27 E3 EA 01 → Numéro de séquence (numéro du 1^{er} octet attendu en réception. **(0.25Pts)**)
- 5 → Longueur de l'en-tête du segment (20 octets) : (on peut en déduire que ce segment ne contient pas de données. **(0.25Pts)**)
- 0 12 = 000000 01 0010 → Drapeaux :
 - 000000 → 6 bits réservés. **(0.25Pts)**
 - 01 → URG=0 **(0.25Pts)**, ACK=1 **(0.25Pts)**
 - 0010 → PSH=0 **(0.25Pts)**, RST=0 **(0.25Pts)**, SYN=1 **(0.25Pts)**, FIN=0 **(0.25Pts)**
- 10 00 → Taille de la fenêtre, ici a priori 4 096 octets. C'est la quantité de données que l'émetteur est autorisé à envoyer sans accusé de réception **(0.25Pts)**.
- DF 3D → Bloc de contrôle d'erreur sur le segment entier **(0.25Pts)**.
- 00 00 → Pointeur vers les données urgentes (nul ici puisqu'il n'y a pas de données urgentes, bit URG = 0). **(0.25Pts)**
- **Conclusion** : Ce segment contient deux drapeaux mis à 1, les drapeaux SYN et ACK. Il s'agit donc de la réponse positive à une demande d'ouverture de connexion. **(0.25Pts)**

- **Fin de la trame ethernet :**
- 20 20 20 20 20 20 → 6 octets de bourrage pour amener la trame Ethernet à la longueur minimale (64 octets en tout). **(0.5Pts)**
- 9B 52 46 43 → Bloc de contrôle d'erreur de la trame Ethernet. **(0.25Pts)**

Bon courage

L2 Systèmes Informatiques Examen Théorie des langages Durée : 90 min

Nom : Correction Examen Prénom :

Questions de cours : (5 points)

1. Donner la version droite du théorème d'Arden ?

$X = AX + B$ 0,5
 $\Rightarrow X = X^*B$ 0,5

2. Quel est l'origine du mot alphabet ?

α, β
 0,5 0,5

3. Quel est le rôle du mot vide ϵ dans les opérations sur les langages ?

élément neutre 1

4. Quel est le résultat de la détermination d'un AEF non déterministe ?

AEFD équivalent
 0,5 0,5

5. Que signifie $\beta_k \leftrightarrow \beta_{k-1}$?

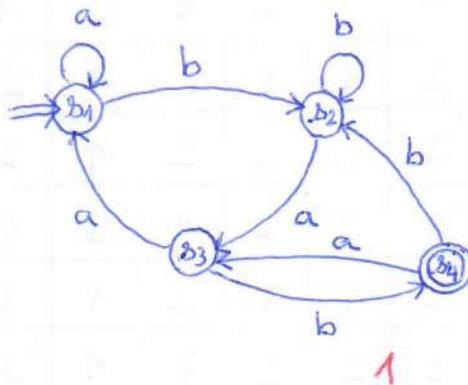
Non stationnarité de β_k 1

Exercice 1 : (5 points)

Déterminer l'automate d'état fini suivant :

$A1 = G(\{a,b\}, \{1, 2, 3, 4\}, 1, \{\delta(1,a)=1, \delta(1,b)=\{1,2\}, \delta(2,a)=3, \delta(3,b)=4\}, \{4\})$

$Q \backslash X$	a	b	
E_i $\delta_1 = \{1\}$	$\{1\}$	$\{1, 2\}$	1
$\delta_2 = \{1, 2\}$	$\{1, 3\}$	$\{1, 2\}$	1
$\delta_3 = \{1, 3\}$	$\{1\}$	$\{1, 2, 4\}$	1
E_F $\delta_4 = \{1, 2, 4\}$	$\{1, 3\}$	$\{1, 2\}$	1



Exercice 2 : (5 points)

Trouver le langage $L(A)$ reconnu par l'automate d'état fini suivant :

$$A_2 = G(\{a, b\}, \{1, 2, 3, 4\}, 1, \delta, \{3, 4\})$$

$$\delta = \{ \delta(1,a)=2, \delta(1,b)=4, \delta(2,a)=4, \delta(2,b)=3, \delta(3,a)=\delta(3,b)=3, \delta(4,a)=4 \}$$

$$X_1 = \emptyset \quad 0,5$$

$$X_2 = X_1 a \quad 0,5$$

$$X_3 = X_3(a+b) + X_2 b \quad 0,5$$

$$X_4 = X_4 a + X_1 b + X_2 a \quad 0,5$$

$$Z = X_3 + X_4 \quad 0,5$$

$$X_1 = \emptyset \quad 0,5$$

$$X_2 = a \quad 0,5$$

$$X_3 = ab(a+b)^* \quad 0,5$$

$$X_4 = (b+aa)a^* \quad 0,5$$

$$Z = ab(a+b)^* + (b+aa)a^* \quad 0,5$$

Exercice 3 : (5 points)

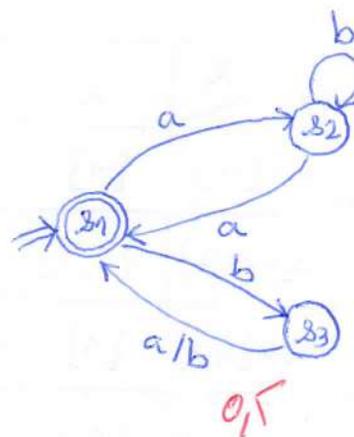
Appliquer l'algorithme de minimisation sur l'automate d'état fini déterministe suivant :

$$A_3 = G(\{a, b\}, \{0, 1, 2, 3, 4, 5, 6\}, 0, \delta, \{0, 3, 5, 6\})$$

$$\delta = \{ \delta(0,a)=1, \delta(0,b)=4, \delta(1,a)=3, \delta(1,b)=2, \delta(2,a)=3, \delta(2,b)=2, \delta(3,a)=1, \delta(3,b)=4, \delta(4,a)=5, \delta(4,b)=6, \delta(5,a)=1, \delta(5,b)=4, \delta(6,a)=1, \delta(6,b)=4 \}$$

- pas d'état inaccessible. 0,5

$B_k \setminus Q$	0	1	2	3	4	5	6	
B_0	1	2	2	1	2	1	1	0,5
a	2	1	1	2	1	2	2	0,5
b	2	2	2	2	1	2	2	0,5
B_1	1	2	2	1	3	1	1	0,5
a	2	1	1	2	1	2	2	0,5
b	3	2	2	3	1	3	3	0,5
B_2	1	2	2	1	3	1	1	0,5



- $B_2 = B_1 \Rightarrow$ Arrêt 0,5