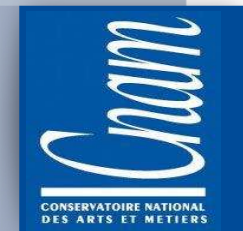


**> Introduction à la Logique Floue  
appliquée aux systèmes décisionnels**



**Présentation**

Franck.Dernoncourt@gmail.com  
26 Janvier 2011

1. Origine

2. Définitions

3. Application : systèmes d'inférence flous

4. Application : requêtes floues

5. Conclusion

6. Références

# 1. Origine

## Le constat :



Les connaissances dont disposent les humains sur le monde ne sont presque jamais parfaites. Ces imperfections peuvent être distinguées en deux classes :

- **Incertitudes** pour désigner les connaissances dont la validité est sujette à question. Par exemple, si nous savons qu'une personne s'est cognée la tête sur un plafond, nous devinons qu'il est probable qu'elle soit très grande.
- **Imprécisions** pour désigner les connaissances qui ne sont pas perçues ou définies nettement. Par exemple, au lieu de dire qu'une personne mesure 2 mètres et 3 centimètres, nous disons usuellement que cette personne est très grande.

# 1. Origine

## Le constat :



Les connaissances dont disposent les humains sur le monde ne sont presque jamais parfaites. Ces imperfections peuvent être distinguées en deux classes :

- ✓ **Incertitudes** pour désigner les connaissances dont la validité est sujette à question. Par exemple, si nous savons qu'une personne s'est cognée la tête sur un plafond, nous devinons qu'il est probable qu'elle soit très grande.
- **Imprécisions** pour désigner les connaissances qui ne sont pas perçues ou définies nettement. Par exemple, au lieu de dire qu'une personne mesure 2 mètres et 3 centimètres, nous disons usuellement que cette personne est très grande.

# 1. Origine

## Le constat :



Les connaissances dont disposent les humains sur le monde ne sont presque jamais parfaites. Ces imperfections peuvent être distinguées en deux classes :

### Probabilités !

- ✓ **Incertitudes** pour désigner les connaissances dont la validité est sujette à question. Par exemple, si nous savons qu'une personne s'est cognée la tête sur un plafond, nous devinons qu'il est probable qu'elle soit très grande.
- **Imprécisions** pour désigner les connaissances qui ne sont pas perçues ou définies nettement. Par exemple, au lieu de dire qu'une personne mesure 2 mètres et 3 centimètres, nous disons usuellement que cette personne est très grande.

# 1. Origine

## Le constat :



Les connaissances dont disposent les humains sur le monde ne sont presque jamais parfaites. Ces imperfections peuvent être distinguées en deux classes :

### Probabilités !

- ✓ **Incertitudes** pour désigner les connaissances dont la validité est sujette à question. Par exemple, si nous savons qu'une personne s'est cognée la tête sur un plafond, nous devinons qu'il est probable qu'elle soit très grande.
- ✗ **Imprécisions** pour désigner les connaissances qui ne sont pas perçues ou définies nettement. Par exemple, au lieu de dire qu'une personne mesure 2 mètres et 3 centimètres, nous disons usuellement que cette personne est très grande.

# 1. Origine

## Le constat :



Les connaissances dont disposent les humains sur le monde ne sont presque jamais parfaites. Ces imperfections peuvent être distinguées en deux classes :

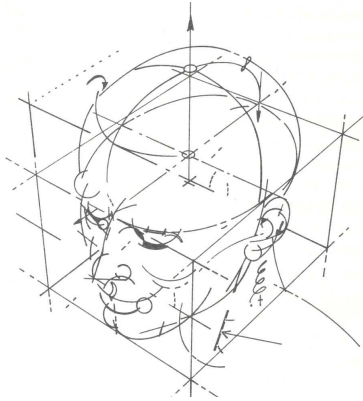
### Probabilités !

- ✓ **Incertitudes** pour des connaissances dont la validité est sujette à question. Par exemple, si nous savons qu'une personne s'est cognée la tête sur un plafond, nous devinons qu'il est probable qu'elle soit très grande.

### Logique floue !

- ✗ **Imprécisions** pour des connaissances qui ne sont pas perçues ou définies nettement. Par exemple, au lieu de dire qu'une personne mesure 2 mètres et 3 centimètres, nous disons usuellement que cette personne est très grande.

# 1. Origine



Comment faire en sorte d'exprimer ces imprécisions en **termes logiques** ?

En **logique classique**, une proposition est **vraie** ou **fausse**.

Exemple : C'est personne est grande. *Vrai* ou *faux* ?

→ Pas flexible...

En **logique multivaluée**, une proposition peut avoir plusieurs valeurs.

Exemple (ternaire) : C'est personne est grande. *Vrai*, à *moitié vrai* ou *faux* ?

→ Un peu plus flexible...

En **logique floue**, une proposition peut avoir autant de valeurs que l'on veut.

Exemple : C'est personne est grande. C'est vrai à 30%.

→ Flexible !

# 1. Origine



**Lotfi Zadeh**, chercheur en théorie des systèmes posa les fondements de la logique floue dans un article en 1965.

La logique floue est une **extension de la logique booléenne** en se basant sur sa théorie mathématique des ensembles flous, qui est une généralisation de la théorie des ensembles classiques.

En introduisant la **notion de degré dans la vérification d'une condition**, permettant ainsi à une condition d'être dans un autre état que vrai ou faux, la logique floue confère une flexibilité très appréciable aux raisonnements qui l'utilisent, ce qui rend possible la prise en compte des imprécisions.

# 1. Origine



## Applications :

- Aide à la décision. (ex. : domaine médical) ;
- Base de données (ex. : requêtes floues) ;
- Commande floue de systèmes (ex. : métro ligne 14) ;
- Data mining (ex. : clustering) ;
- Reconnaissance de forme ;
- ...

1. Origine

2. Définitions

3. Application : systèmes d'inférence flous

4. Application : requêtes floues

5. Conclusion

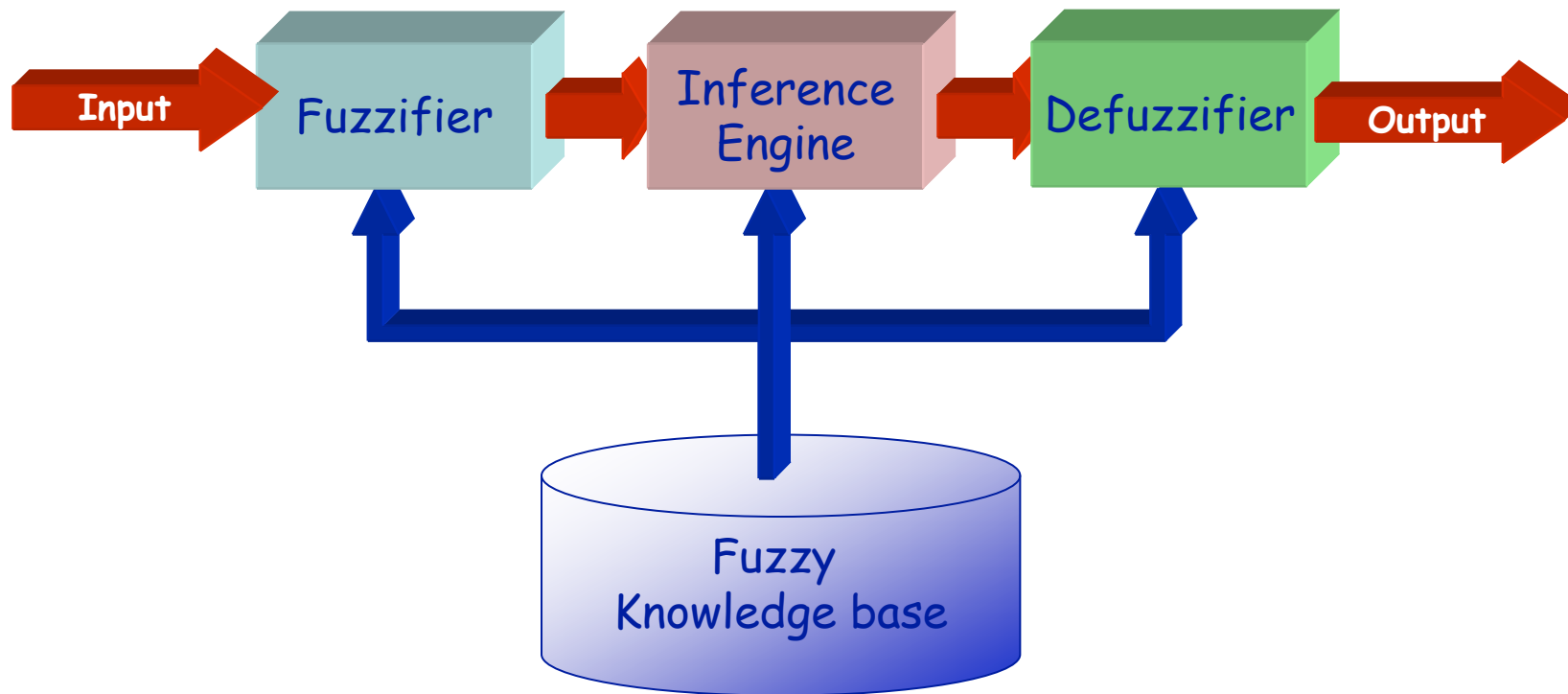
6. Références

## 2. Définitions

Définitions des **concepts principaux de la logique floue** via un exemple de traitement d'image : augmentation du contraste d'une image.



## 2. Définitions

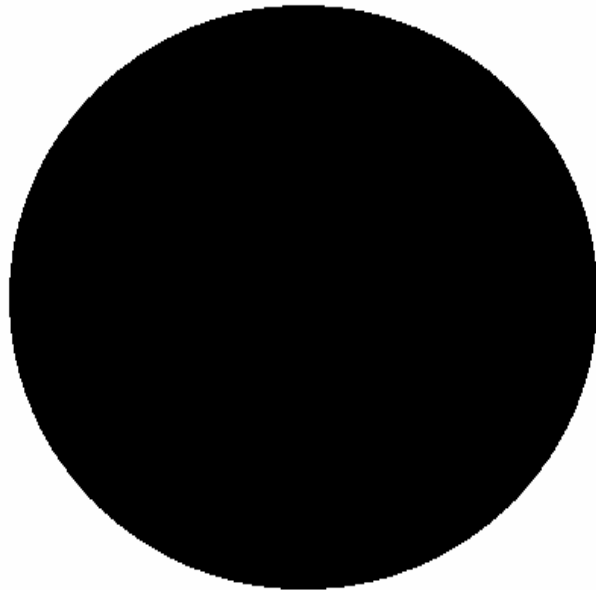


## 2. Définitions

### 1) Ensemble flou et degré d'appartenance

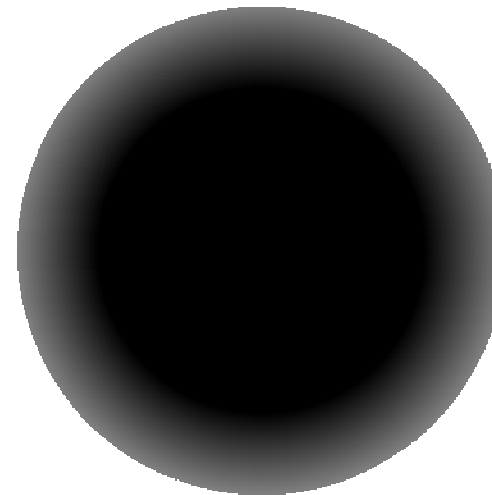
Fuzzifier

Ensemble classique



Appartenance : 0 ou 1  
→ **Binaire** en logique classique

Ensemble flou



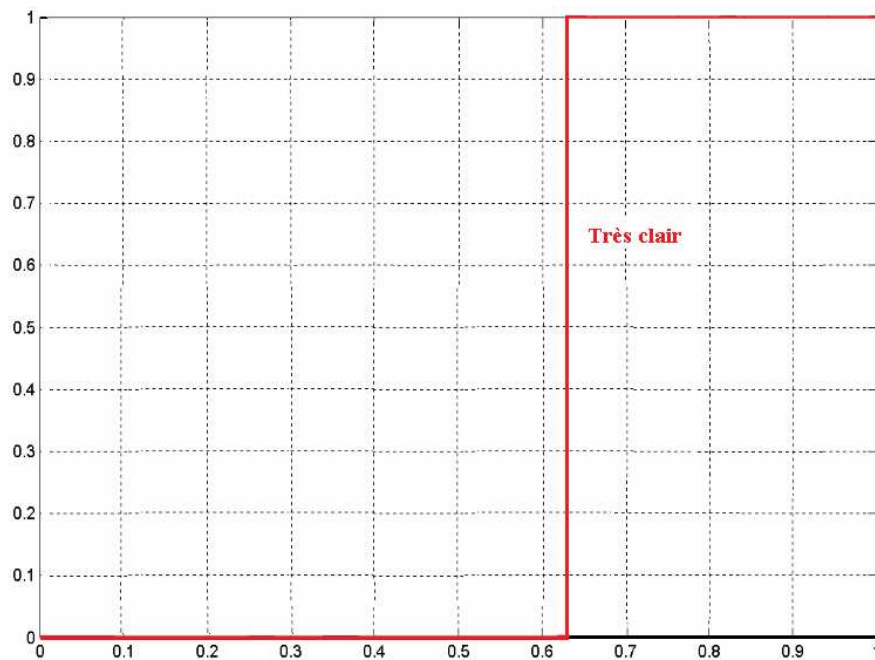
Appartenance : entre 0 et 1 (eg 0.867)  
→ **degré d'appartenance** en logique floue

## 2. Définitions

Fuzzifier

### 1) Ensemble flou et degré d'appartenance

Logique classique



Appartenance : 0 ou 1  
→ Binaire en logique classique

Logique floue

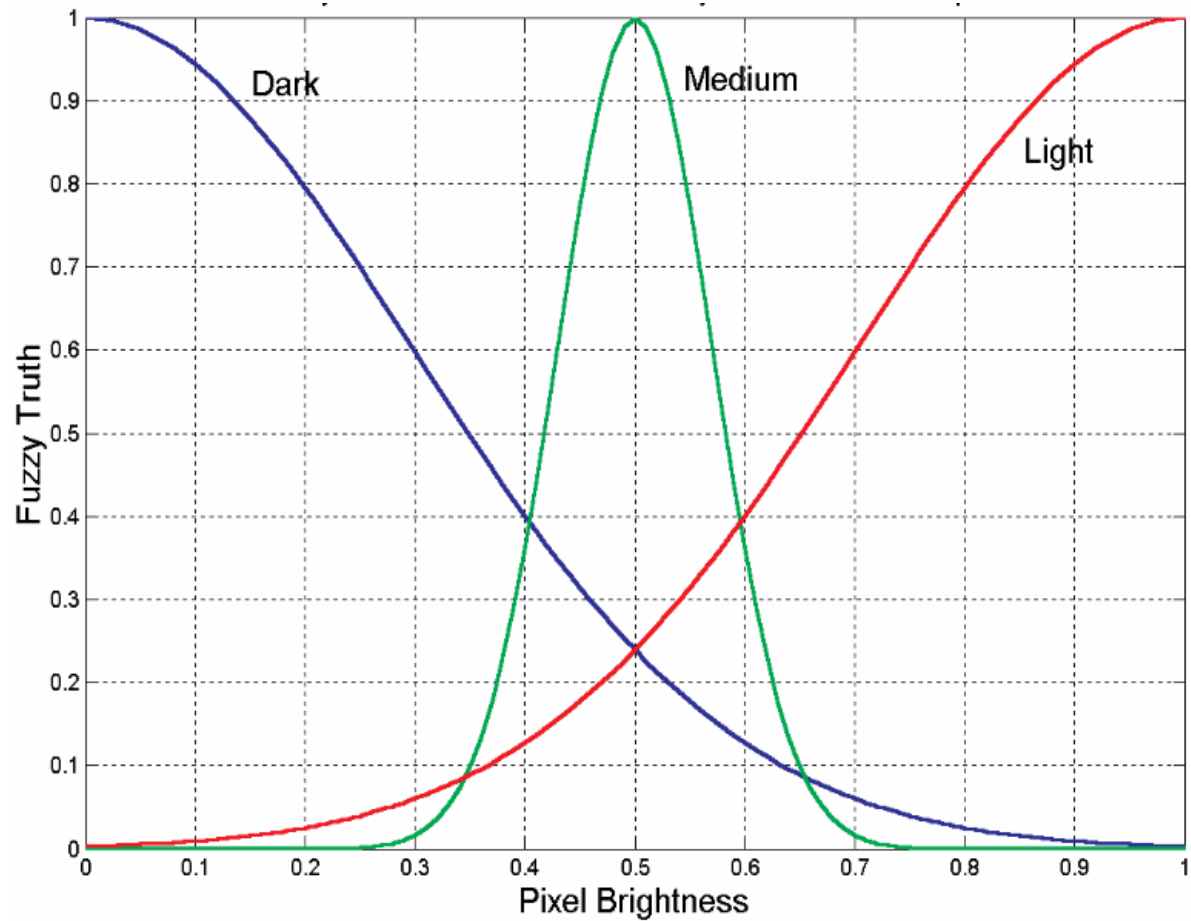


Appartenance : entre 0 et 1 (eg 0.867)  
→ **degré d'appartenance** en logique floue

## 2. Définitions

### 1) Fonctions d'appartenance (phase de fuzzification) en entrée

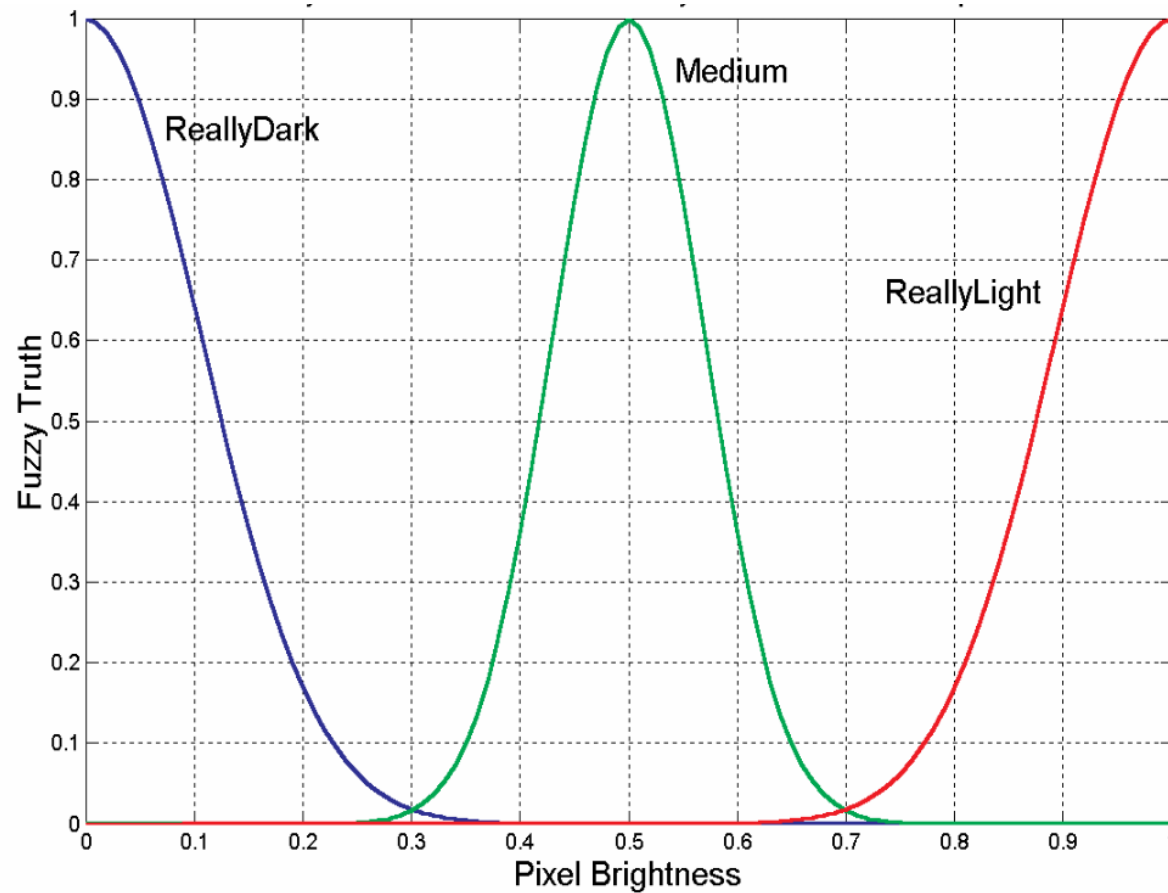
Fuzzifier



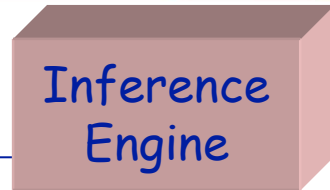
## 2. Définitions

### 1) Fonctions d'appartenance (phase de fuzzification) en sortie

Fuzzifier



## 2. Définitions



### 2) Redéfinitions des règles de bases

Logique classique

- Négation
- Et
- Ou
  
- Implication
- Modus ponens

Logique floue

## 2. Définitions



### 3) Matrice des décisions

Entrée	Sortie
Clair	Très clair
Moyen	Moyen
Foncé	Très foncé

**La puissance de la logique floue !  
Un système logique qui s'exprime  
directement en langage naturel**

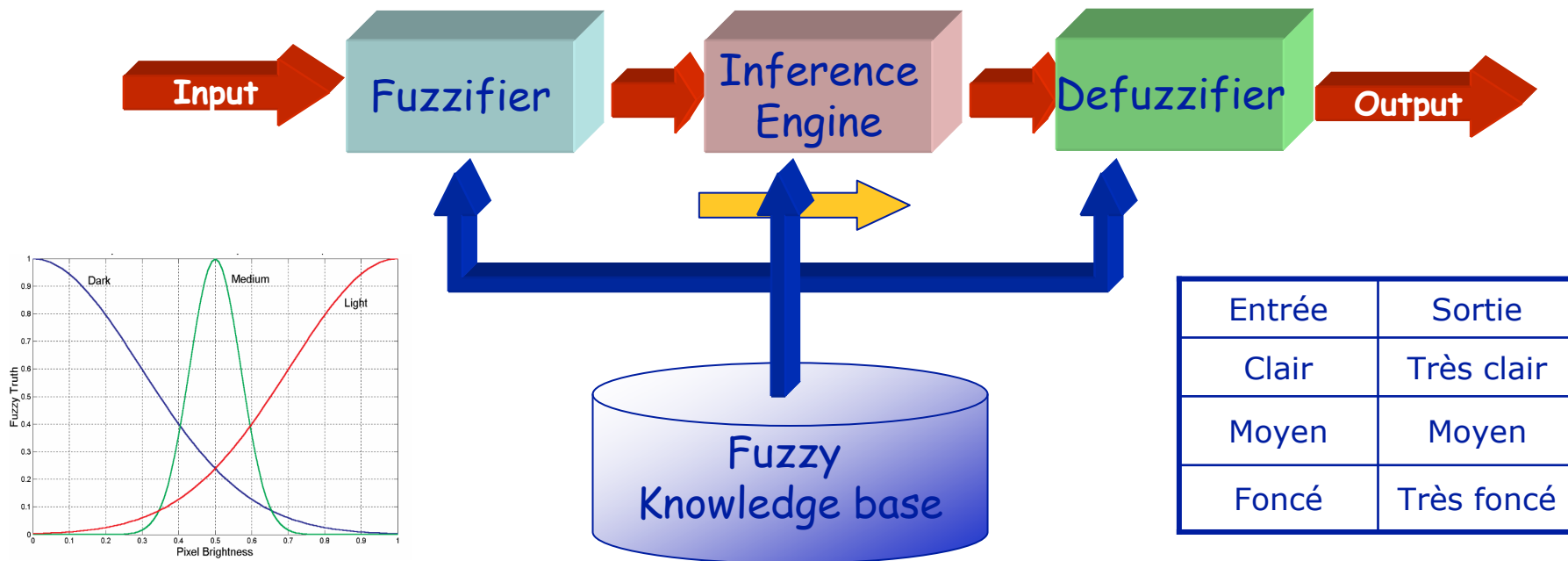
## 2. Définitions

### 4) Défuzzification !

Defuzzifier



## 2. Définitions



1. Origine

2. Définitions

3. Application : systèmes d'inférence flous

4. Application : requêtes floues

5. Conclusion

6. Références

### 3. Application : décisionnel

#### Exemple de décision

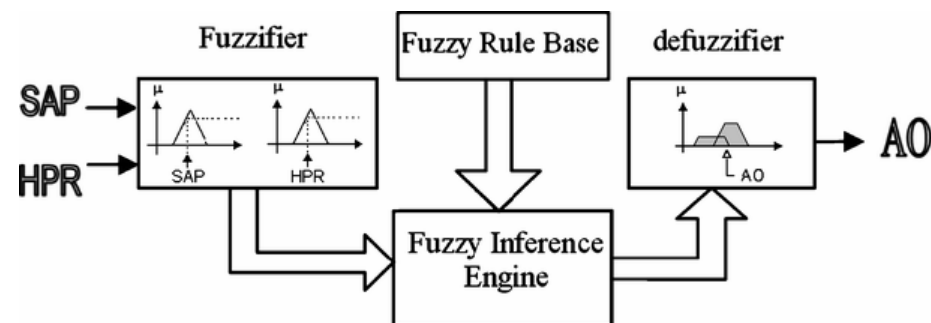
Un anesthésiste doit décider du contenu du gaz pendant une opération chirurgicale : ce gaz doit être plus ou moins anesthésiant selon la condition du patient à instant donné.

2 critères principaux (input) :

- SAP : systolic arterial pressure (tension artérielle)
- HPR : heart pulse rate (rythme cardiaque)

1 décision en sortie (output) :

- AO : Taux de produits anesthésiques dans le gaz



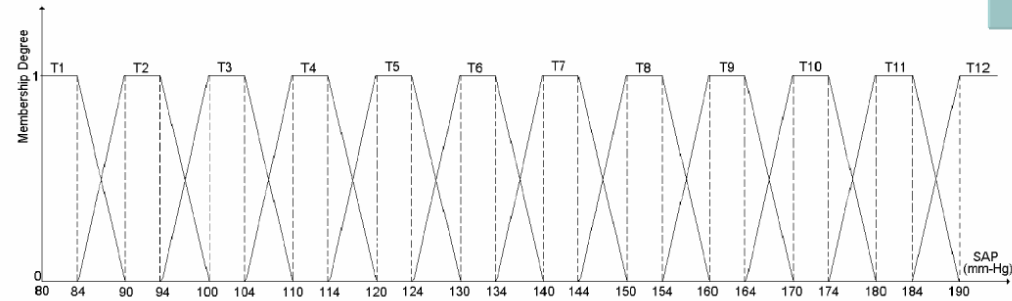
### 3. Application : décisionnel



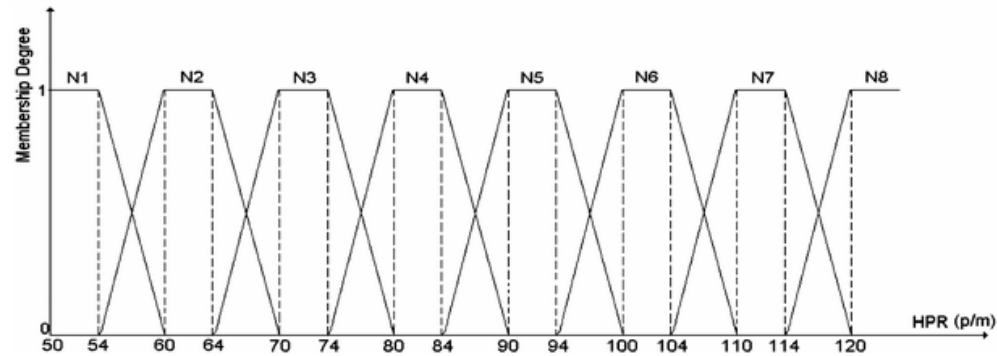
Fuzzifier

#### Fonctions d'appartenance

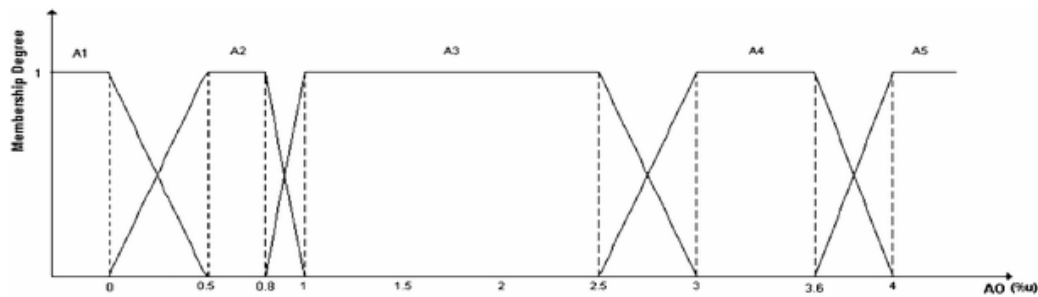
Input 1



Input 2



Output



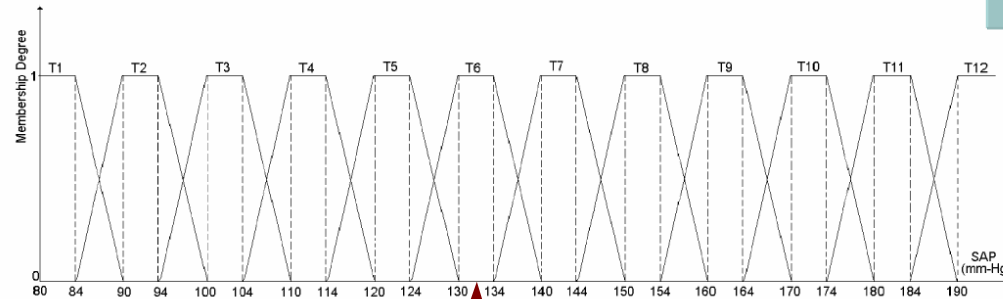
### 3. Application : décisionnel



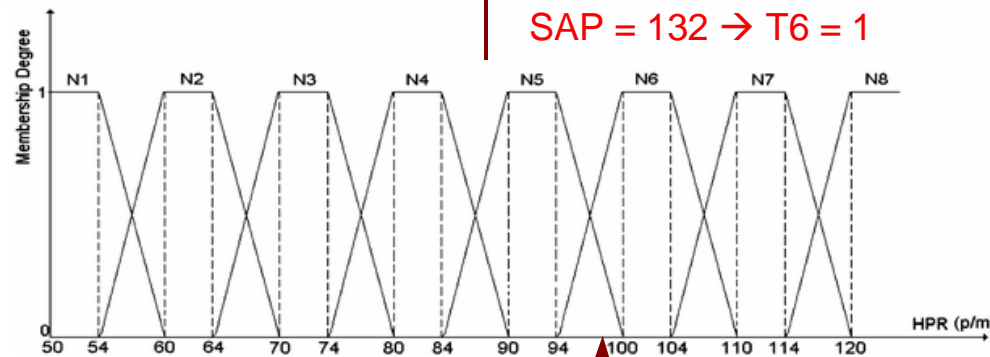
Fuzzifier

#### Fonctions d'appartenance

Input 1  
SAP = 132



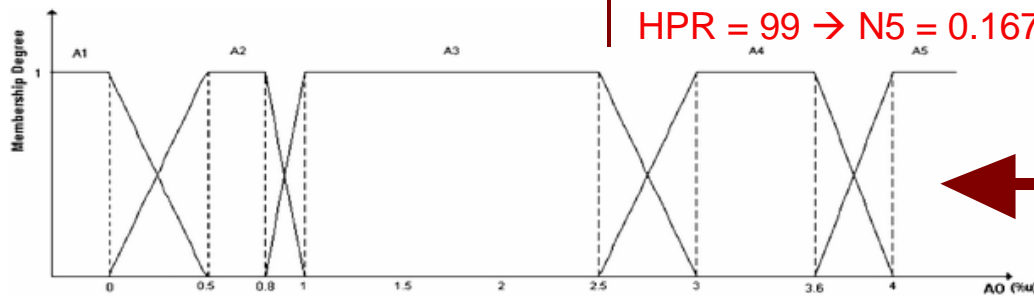
Input 2  
HPR = 99



SAP = 132 → T6 = 1

HPR = 99 → N5 = 0.167, N6 = 0.834

Output



### 3. Application : décisionnel

Inference Engine

Matrice des décisions

	N1	N2	N3	N4	N5	N6	N7	N8
T1	A1	A1	A2	A2	A2	S	S	S
T2	A2	A2	A3	A3	A3	A4	A4	A4
T3	A2	A3	A3	A3	A3	A4	A4	A4
T4	A2	A3	A3	A3	A3	A4	A4	A4
T5	A2	A3	A3	A3	A3	A4	A4	A4
T6	A2	A3	A3	A3	A3	A4	A4	A4
T7	A2	A3	A3	A3	A3	A4	A4	A5
T8	S	A4	A4	A4	A4	A5	A5	A5
T9	S	A4	A4	A4	A4	A5	A5	A5
T10	S	A4	A4	A4	A4	A5	A5	A5
T11	S	A5	A5	A5	A5	A5	A5	A5
T12	S	A5	A5	A5	A5	A5	A5	A5

Lire : Si N5 et T10, alors on décide A4

### 3. Application : décisionnel



#### Matrice des décisions

*HPR = 99 d'où N5 = 0.167 et N6 = 0.834*

*SAP = 132*

*D'où :*

- T6 = 1*

	N1	N2	N3	N4	N5	N6	N7	N8
T1	A1	A1	A2	A2	A2	S	S	S
T2	A2	A2	A3	A3	A3	A4	A4	A4
T3	A2	A3	A3	A3	A3	A4	A4	A4
T4	A2	A3	A3	A3	A3	A4	A4	A4
T5	A2	A3	A3	A3	A3	A4	A4	A4
T6	A2	A3	A3	A3	A3	A4	A4	A4
T7	A2	A3	A3	A3	A3	A4	A4	A5
T8	S	A4	A4	A4	A4	A5	A5	A5
T9	S	A4	A4	A4	A4	A5	A5	A5
T10	S	A4	A4	A4	A4	A5	A5	A5
T11	S	A5	A5	A5	A5	A5	A5	A5
T12	S	A5	A5	A5	A5	A5	A5	A5

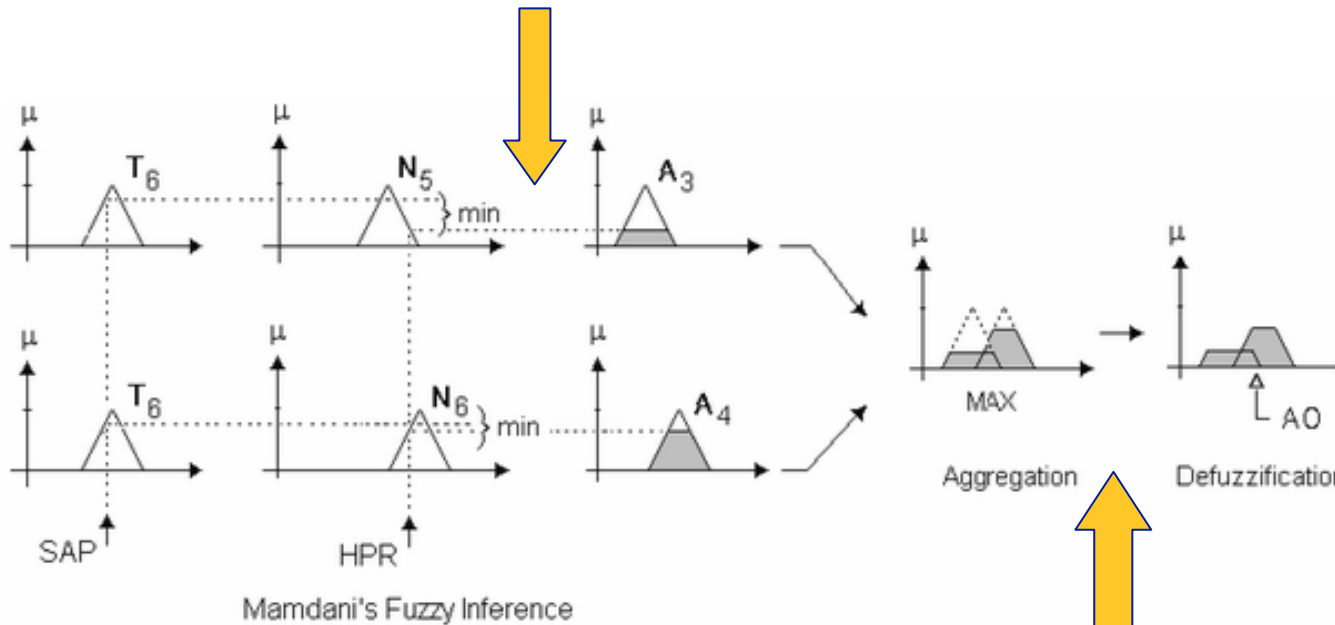
*On décide donc A3 et A4 !*

### 3. Application : décisionnel

Input : HPR = 99 → N5 = 0.167, N6 = 0.834  
 SAP = 132 → T6 = 1

Defuzzifier

T6 = 1	N5 = 0.167	=>	A3 = 0.167
T6 = 1	N6 = 0.834	=>	A4 = 0.5



A3 = 0.167    A4 = 0.5    =>    **Output = 3.368**

### 3. Application : décisionnel



### Anesthésiste vs logique floue ?

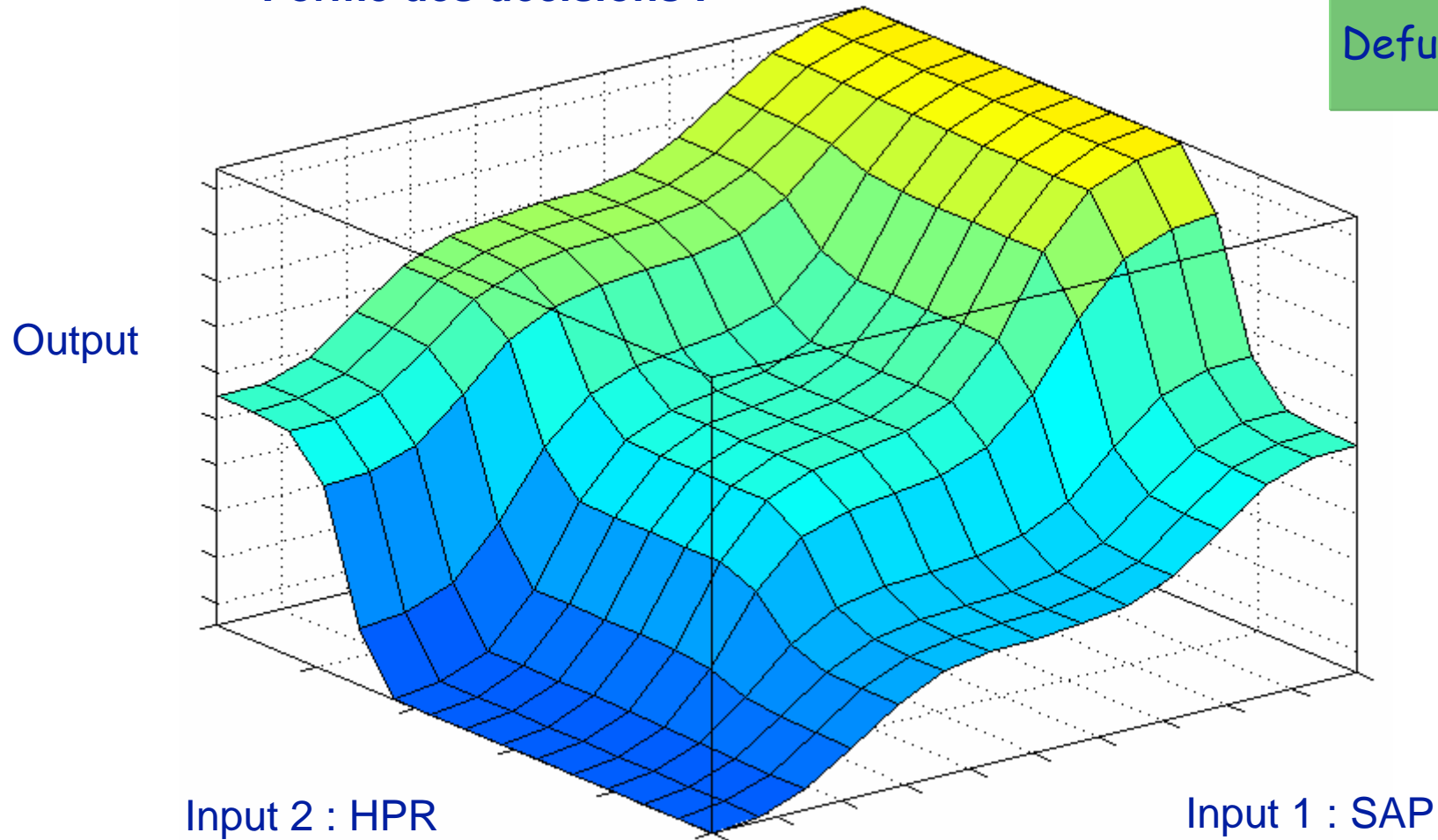
Table 5 Comparison of anesthetist prediction and Fuzzy Logic prediction

											Total	
Patient 1	SAP (mmHg)		164	161	192	156	172	161	152	157	154	
	HPR (p/m)		65	96	99	80	72	81	73	78	73	
	AO	Anesthetist	2	2.5	4	3	3.5	2.5	2.5	2.5	2.5	25
		Fuzzy Logic	3.25	3.29	3.88	2.18	3.27	3.27	1.83	2.38	1.83	25.2
											+1	
Patient 2	SAP (mmHg)		149	184	139	182	163	158	158			
	HPR (p/m)		88	94	98	74	78	90	87			
	AO	Anesthetist	2	4	2	5	4	2	2			21
		Fuzzy Logic	1.85	3.27	2.61	3.27	3.27	2.61	2.52			19.4
											-7.6	
Patient 3	SAP (mmHg)		110	130	130	110	130	120	120			
	HPR (p/m)		90	70	70	68	72	75	76			
	AO	Anesthetist	2.5	2	2	2	2	1.5	1			12
		Fuzzy Logic	1.83	1.83	1.83	1.85	1.83	1.84	1.85			11.1
											-7.5	
Patient 4	SAP (mmHg)		150	110	115	110	110					
	HPR (p/m)		96	74	107	95	90					
	AO	Anesthetist	3	2	2	2	2					11
		Fuzzy Logic	2.18	1.83	3.26	2.00	1.83					11.1
											+1	

### 3. Application : décisionnel

Forme des décisions :

Defuzzifier



1. Origine

2. Définitions

3. Application : systèmes d'inférence flous

4. Application : requêtes floues

5. Conclusion

6. Références

## 4. Application : requêtes floues

**XQuery** : langage de requête informatique pour les bases de données XML

*(XQuery joue par rapport aux données XML un rôle similaire à celui du langage SQL vis-à-vis des données relationnelles)*



**Exemple** : un manager veut donner un bonus à tous ses vendeurs performants. Il veut faire une requête dans la base de données pour les trouver.

Nous allons comparer les résultats obtenus selon la logique sous-jacente à la requête : logique classique ou logique floue.

## 4. Application : requêtes floues

Logique classique : critère booléen « Plus de 4000 € en 10 jours ! »

*Requête :*

```
<output>
{
  for $emp in doc("sale.xml")/salesman/emp
  let $eid := $emp/empid/text()
  let $en := $emp/ename/text()
  let $sa := $emp/sales/text()
  let $d:=$emp/days/text()
  return if ($sa >=4000 and $d <=10) then
    <emp>
    <empid> {$eid} </empid>
    <ename> {$en} </ename>
    <sales>{$sa}</sales>
    <days>{$d}</days>
    </emp>
  else ()
}</output>
```



## 4. Application : requêtes floues

Logique classique : critère booléen « Plus de 4000 € en 10 jours ! »

```
<output>
<emp>
<empid>1003</empid>
<ename>Joshua</ename>
<sales>5000</sales>
<days>10</days>
</emp>
<emp>
<empid>1004</empid>
<ename>Kevin</ename>
<sales>4998</sales>
<days>3</days>
</emp>
```



[...]

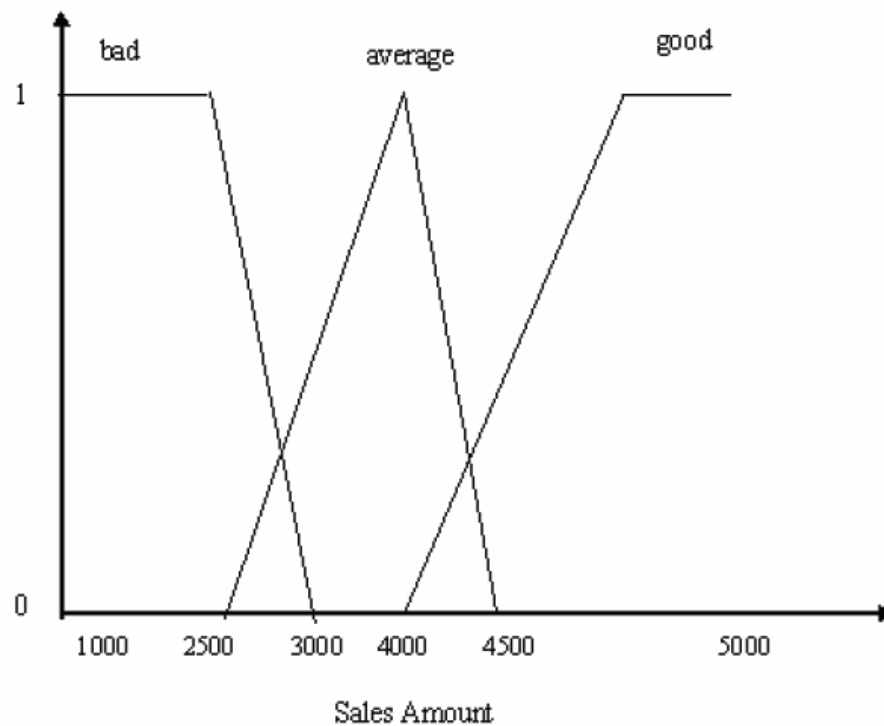
```
<emp>
<empid>1009</empid>
<ename>George</ename>
<sales>4100</sales>
<days>11</days>
</emp>
<emp>
<empid>1008</empid>
<ename>Henry</ename>
<sales>3999</sales>
<days>3</days>
</emp>
</output>
```



## 4. Application : requêtes floues

Logique floue : critère flou « Être bon ! »

Requête :



## 4. Application : requêtes floues

Logique floue : critère flou « Être bon ! »

*Requête :*

```
<output>
{
  for $emp in doc("salesman.xml")/salesman/emp
  let $eid := $emp/empid/text()
  let $en := $emp/ename/text()
  let $sa := $emp/sales/text()
  let $d:=$emp/days/text()
  return if {salesman=good } then
    <emp>
    <empid> {$eid} </empid>
    <ename> {$en} </ename>
    <sales>{$sa}</sales>
    <days>{$d}</days>
    </emp>
  else ()
}</output>
```



## 4. Application : requêtes floues

Logique floue : critère flou « Être bon ! »

```
<output>
<emp>
<empid>1003</empid>
<ename>Joshua</ename>
<sales>5000</sales>
<days>10</days>
</emp>
<emp>
<empid>1004</empid>
<ename>Kevin</ename>
<sales>4998</sales>
<days>3</days>
</emp>
```



[...]

```
<emp>
<empid>1009</empid>
<ename>George</ename>
<sales>4100</sales>
<days>11</days>
</emp>
<emp>
<empid>1008</empid>
<ename>Henry</ename>
<sales>3999</sales>
<days>3</days>
</emp>
</output>
```



## 4. Application : requêtes floues



**SQL** : Langage de requête pour les bases de données

Extensions du SQL intégrant la logique floue :

- SQLf [Bosc 1995] ;
- Summary SQL [Rasmussen 1996].

Requête classique :

```
SELECT LName, Age, Expertise FROM Workers  
WHERE Age < 40 AND Age >= 30 AND Expertise = 'AI';
```

Requête floue (SQLf\_j) :

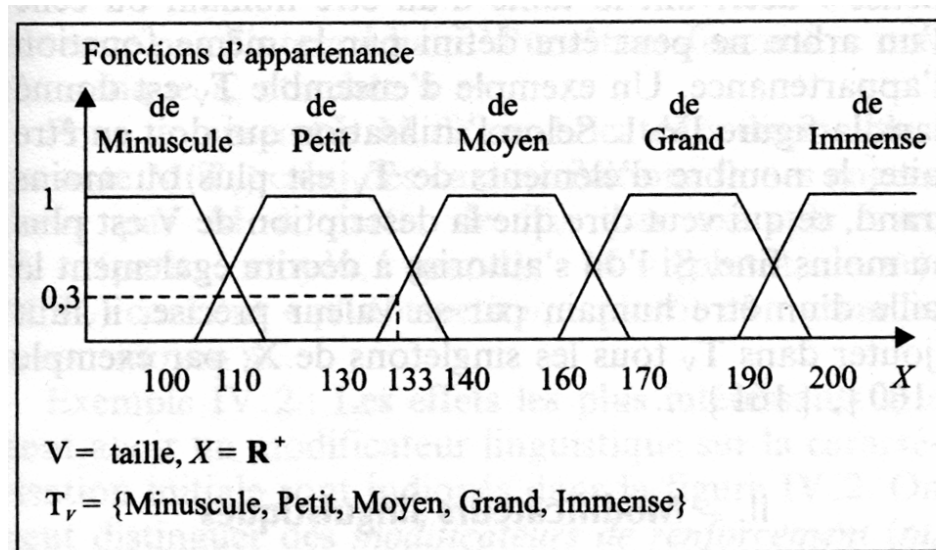
```
SELECT THRESHOLD 0.2 LName, Age, Expertise FROM Workers  
WHERE ~Age IS 'Middle-aged' AND #Expertise IS 'AI';
```

## 4. Application : requêtes floues

### Concept de **variable linguistique**

**Var\_linguistique = (V, X, T<sub>x</sub>) :**

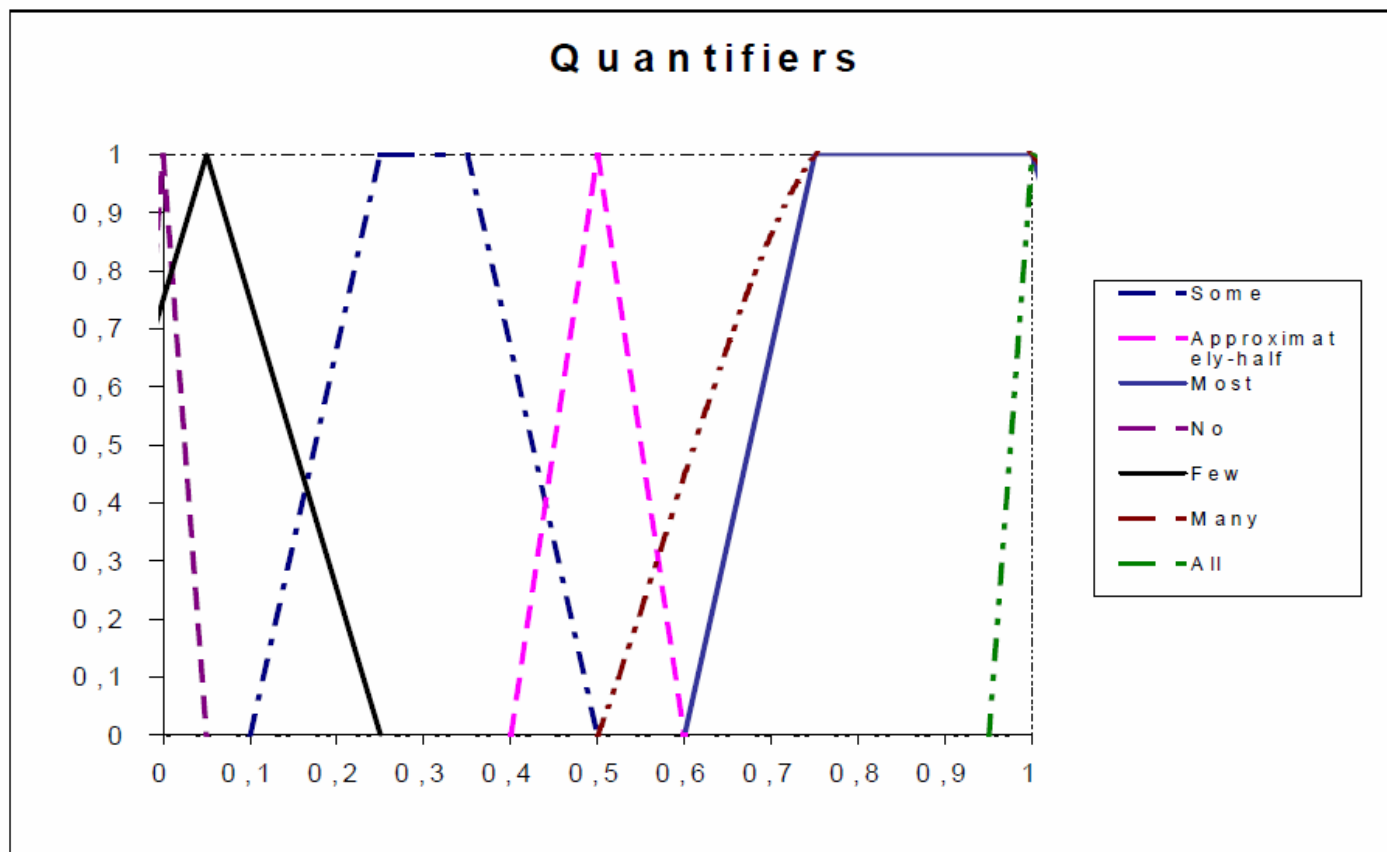
- V : variable (âge, température, etc.) ;
- X : plage de valeurs de la variable ;
- T<sub>x</sub> : sous-ensemble flou.



## 4. Application : requêtes floues

### Concept de quantificateurs

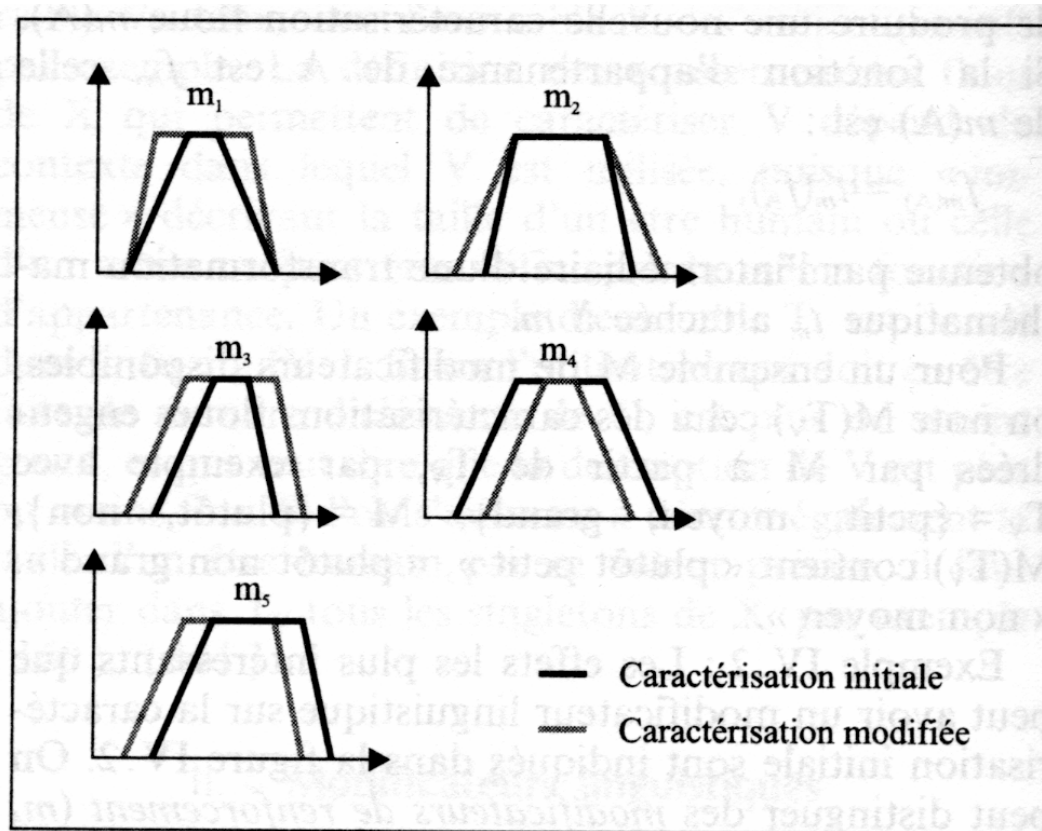
Exemple : « très, un peu, beaucoup, relativement, ... »



## 4. Application : requêtes floues

### Concept de **modificateurs**

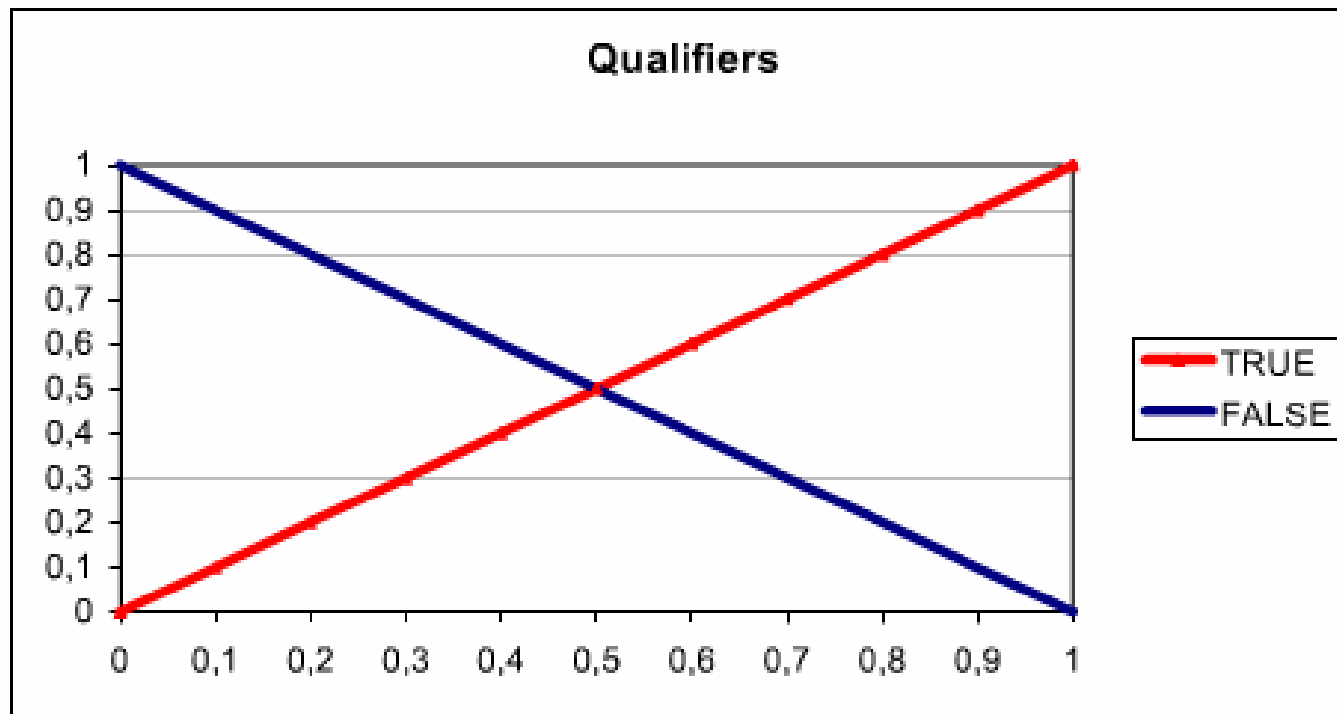
Exemple : « très, un peu, beaucoup, relativement, ... »



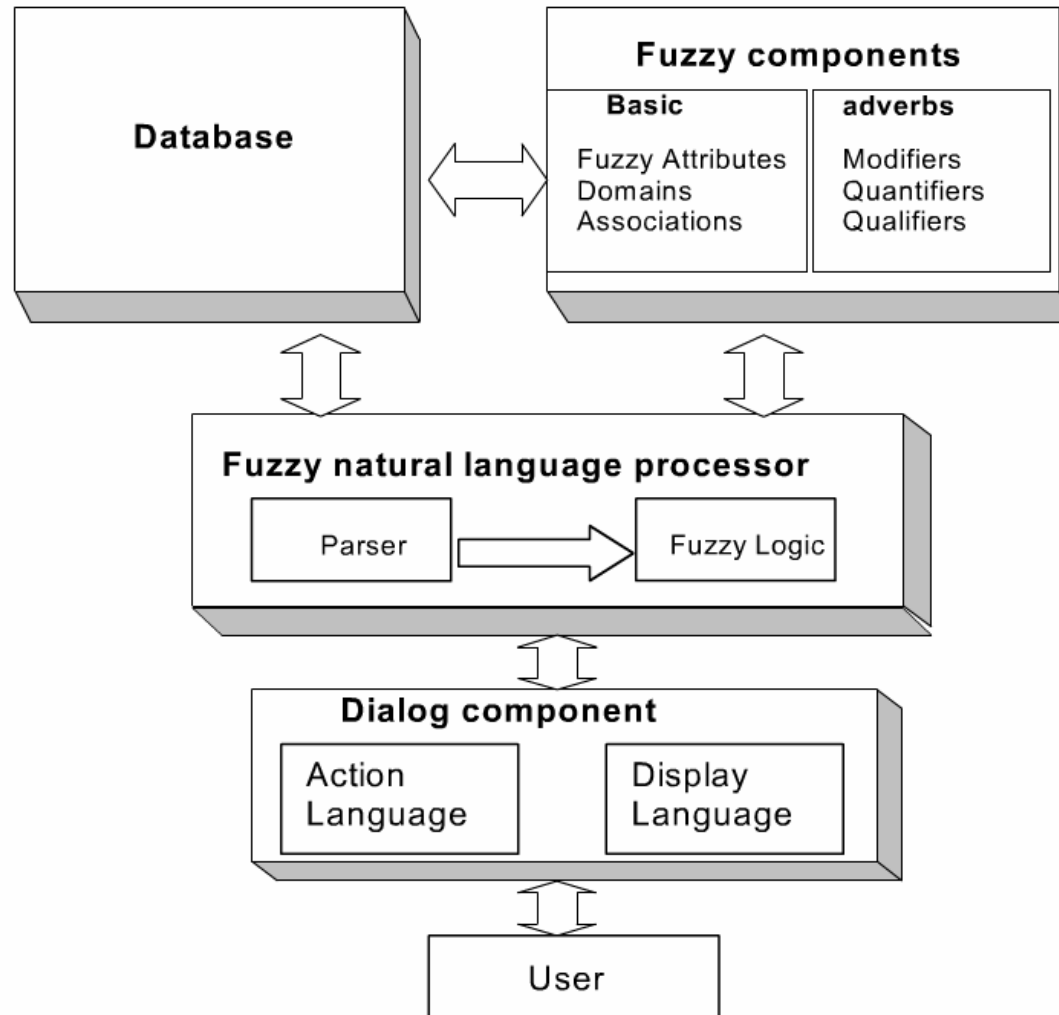
## 4. Application : requêtes floues

Concept de **qualificatifs**

Exemple : « vrai, faux, probable, ... »



## 4. Application : requêtes floues



1. Origine

2. Définitions

3. Application : systèmes d'inférence flous

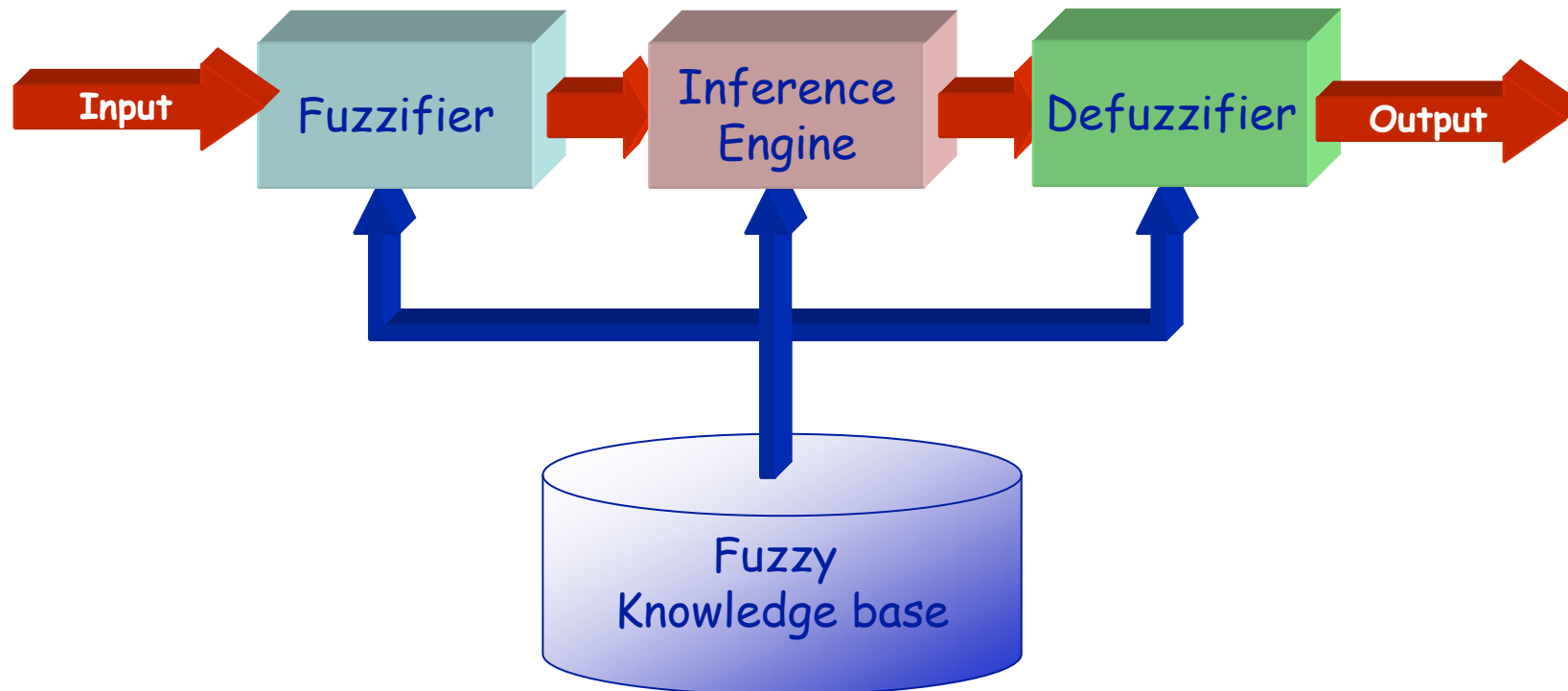
4. Application : requêtes floues

5. Conclusion

6. Références

## 5. Conclusion

En résumé :



# Avenir de la logique floue

- Lien entre raisonnement humain et intelligence artificielle
- Compromis efficacité vs exactitude
- Possibilité de couplage avec d'autres théories (réseaux de neurones, algorithmes évolutionnistes, etc.)

1. Origine

2. Définitions

3. Application : systèmes d'inférence flous

4. Application : requêtes floues

5. Conclusion

6. Références

### Références :

- Bosc, P., Pivert, O.: SQLf: A Relational Database Language for Fuzzy Querying. IEEE Trans. on Fuzzy Systems 3 (1995) 1–17.
- Rasmussen, D. and Yager, R. R.: SummarySQL - A flexible fuzzy query language. Proceedings of the Second Workshop on Flexible Query-Answering Systems, Roskilde, Denmark (1996) 1–18
- Sanli S., Saraoglu H. M., (2007), A fuzzy logic-based decision support system on anesthetic depth control for helping anesthetists in surgeries. Journal of Medical Systems, 31 :511519.
- Zadeh L.A.: Fuzzy Sets. Inform. and Control 8 (1965) 338–353.

[franck.dernoncourt@gmail.com](mailto:franck.dernoncourt@gmail.com)

?

F

D