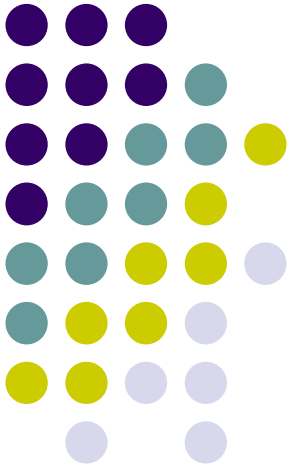
Web Sémantique
Master2

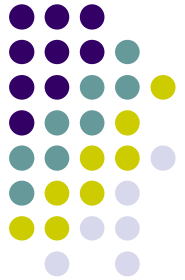
# Chapitre V: Logiques de description

## V.2 familles des logique de description



*Mr Bougherara.S*

# Plan du cours



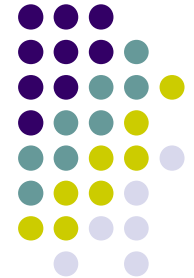
- Introduction
- Le langage  basique AL
- Extensions du  langage AL
- La famille SH
- Remarques (équivalence entre langages)
- Conclusion

# introduction

- Il existe plusieurs langage LD :AL,ALC,ALE ….

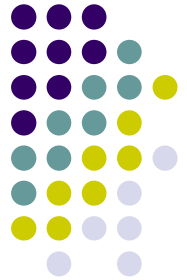
- Les  LD se distinguent par les constructeurs qu'elles proposent :
- plus elles ont de constructeurs, plus elles sont expressives, et ont des chances d'être non décidables ou de complexité très élevée
- les LD trop peu expressives ne permettent pas de représenter des domaines complexes.

# Le langage basique AL



- Le langage AL (pour Attributive Langage) a été introduit comme un langage minimal ayant un intérêt pratique.
- Les autres langages de la famille AL en sont une extension

# Le langage basique AL



- **Conventions**
- A, B :dénotent des concepts atomiques
- C,D : dénotent des concepts définis

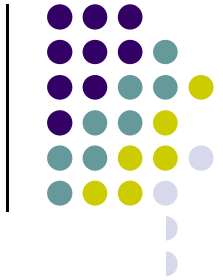
# Le langage basique AL

## Syntaxe

- Les descriptions sont générées selon les règles syntaxiques suivantes

$$
\begin{array}{lll}
C, D \rightarrow & A \mid & \text{(concept atomique)} \\
& \top \mid & \text{(concept universel)} \\
& \bot \mid & \text{(concept bottom)} \\
& \neg A \mid & \text{(negation atomique)} \\
& C \sqcap D \mid & \text{(intersection)} \\
& \forall R.C \mid & \text{(restriction de valeur)} \\
& \exists R.\top & \text{(quantifiquateur existentiel limité)}
\end{array}
$$

# Le langage basique AL



**Constructeurs d'$\mathcal{AL}$:**

| | |
|---|---|
| $\neg A$ | la négation atomique |
| $C \sqcap D$ | l'intersection de concepts |
| $\forall R.C$ | la restriction de valeur (quantification universelle complète) |
| $\exists R.\top$ | la quantification existentielle limitée * |

- Remarques
  - la négation n'est appliquée qu'aux concepts atomiques
  - Seulement le concept universel est permis à la portée du quantificateur existentiel
  - ne permet pas la spécification de rôles composés

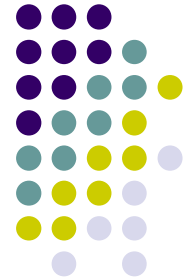
# Le langage basique AL

## Signification des constructeurs avec quantification

- **∃R.T** :les individus qui ont au moins une relation de type R avec un individu de n'importe quel type.
- **∀R.C :**les individus dont toutes les relations de type R se font avec des individus de type C.
- Exemples: sur les cités .
- citéModerne =∃ contientEcole.T
- citéCompacte= ∀ contient.Logement
- ∀ contient.¬Ecole,
- ∃ voisin.T,
- ∃ voisin.(∃ contientEcole.T)

# Le langage basique AL

## •Sémantique ALC



Un interprétation $I$ est une paire $(\Delta^I, \cdot^I)$ où

- $\Delta^I$ est un ensemble (le domaine) et $\cdot$
- $\cdot^I$ est une fonction d'interprétation :

Tel que

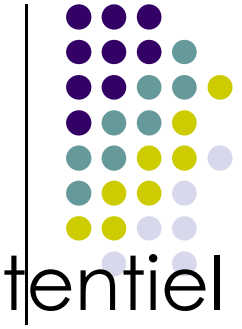
$$
\begin{array}{lcl}
\top^{\mathcal{I}} & = & \triangle^{\mathcal{I}} \\
\bot^{\mathcal{I}} & = & \emptyset \\
(\neg A)^{\mathcal{I}} & = & \triangle^{\mathcal{I}} \backslash A^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} & = & \{a \in \triangle^{\mathcal{I}} | \forall b.(a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\} \\
(\exists R.\top)^{\mathcal{I}} & = & \{a \in \triangle^{\mathcal{I}} | \exists b.(a,b) \in R^{\mathcal{I}}\}
\end{array}
$$

# Extensions du langage AL (la famille AL)

$$\mathcal{AL} = \{\top, \bot, \neg \mathtt{A}, \mathtt{C} \sqcap \mathtt{D}, \forall \mathtt{r}.\mathtt{C}, \exists \mathtt{r}\}$$

- $\mathcal{ALC} = \mathcal{AL} \cup \{\neg \mathtt{C}\}$ (négation de concepts primitifs ou définis).
- $\mathcal{ALU} = \mathcal{AL} \cup \{\mathtt{C} \sqcup \mathtt{D}\}$ (disjonction de concepts).
- $\mathcal{ALE} = \mathcal{AL} \cup \{\exists \mathtt{r}.\mathtt{C}\}$ (quantification existentielle typée).
- $\mathcal{ALN} = \mathcal{AL} \cup \{\geq \mathtt{n}\,\mathtt{r}, \leq \mathtt{n}\,\mathtt{r}\}$ (cardinalité des rôles, en particulier, $(\exists \mathtt{r}) \equiv (\geq 1\,\mathtt{r})$).
- $\mathcal{ALR} = \mathcal{AL} \cup \{\mathtt{r}_1 \sqcap \mathtt{r}_2\}$ (conjonction de rôles).
- Et encore les rôles inverses ($\mathcal{I}$), une hiérarchie de rôles ($\mathcal{H}$), le constructeur $\mathtt{range}$ et la restriction sur les co-domaines ($\mathcal{Q}$) ...

# Extensions du langage AL

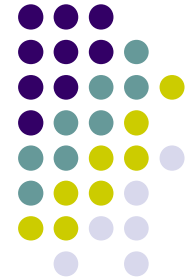


**Quantification existentielle complète** ∃R.C (ALE)

- En AL, on peut utiliser le quantificateur existentiel pour spécifier qu'une entité doit avoir au moins une relation avec un autre objet mais on ne peut pas spécifier la classe de cet autre objet.Ceci limite fortement l'expressivité du langage
- *Exemple : les gens possédant au moins une voiture*
- *Solution :étendre AL par quantification existentielle complète* ∃R.C
- *Sémantique :*

$$\mathcal{I}(\exists R.C) = \{a \in \mid \exists b.(a, b) \in \mathcal{I}(R) \wedge b \in \mathcal{I}(C)\}$$

# Extensions du langage AL



## Négation sans restriction (ALC)

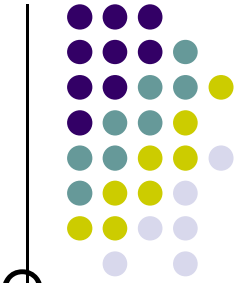
- on peut étendre le langage en ajoutant la possibilité d'utiliser la négation sans aucune restriction
- Cette caractéristique est désignée par le symbole C (pour *complément)*
- *Sémantique :* $I(\neg C) = \{a \in \Delta \, I(C)\}$
- *Exemple:*
- les gens qui n'ont pas d'enfants de sexe masculin:
- $\neg \exists$aEnfant.Homme

# Extensions du langage AL

## L'union (ALU)

- On peut représenter l'ensemble des individus qui appartiennent soit à la classe C, soit à la classe D,
- on écrirait la description C ⊔ D
- Sémantique : $\mathcal{I}$(C ⊔ D) = $\mathcal{I}$(C) ∪ $\mathcal{I}$(D)
- *Exemple*:
- Un magasin qui ne vend que des chats et des chiens : Magasin ⊓ ∀vend.(Chat ⊔ Chien)

# Extensions du langage AL



## Constructeur de fonctions (ALF)

- Pur spécifier que la relation R en en fait une fonction.

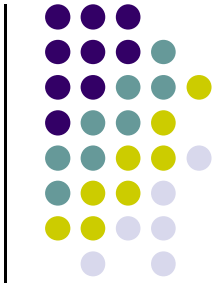
telle que aucune entité ne peut être reliée à plus d'une autre entité

- par cette relation On écrira donc un axiome de la forme ***Fun(R)*** pour dire qu'un rôle R est une fonction.
- *Exemple :*

Fun(aPere)

fils ≡ Homme ⊓ ∃aPere.⊤

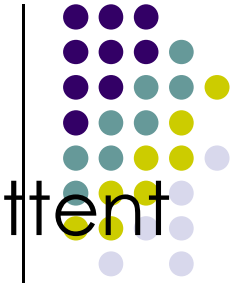
# Extensions du langage AL



## Restriction de cardinalité (ALN)

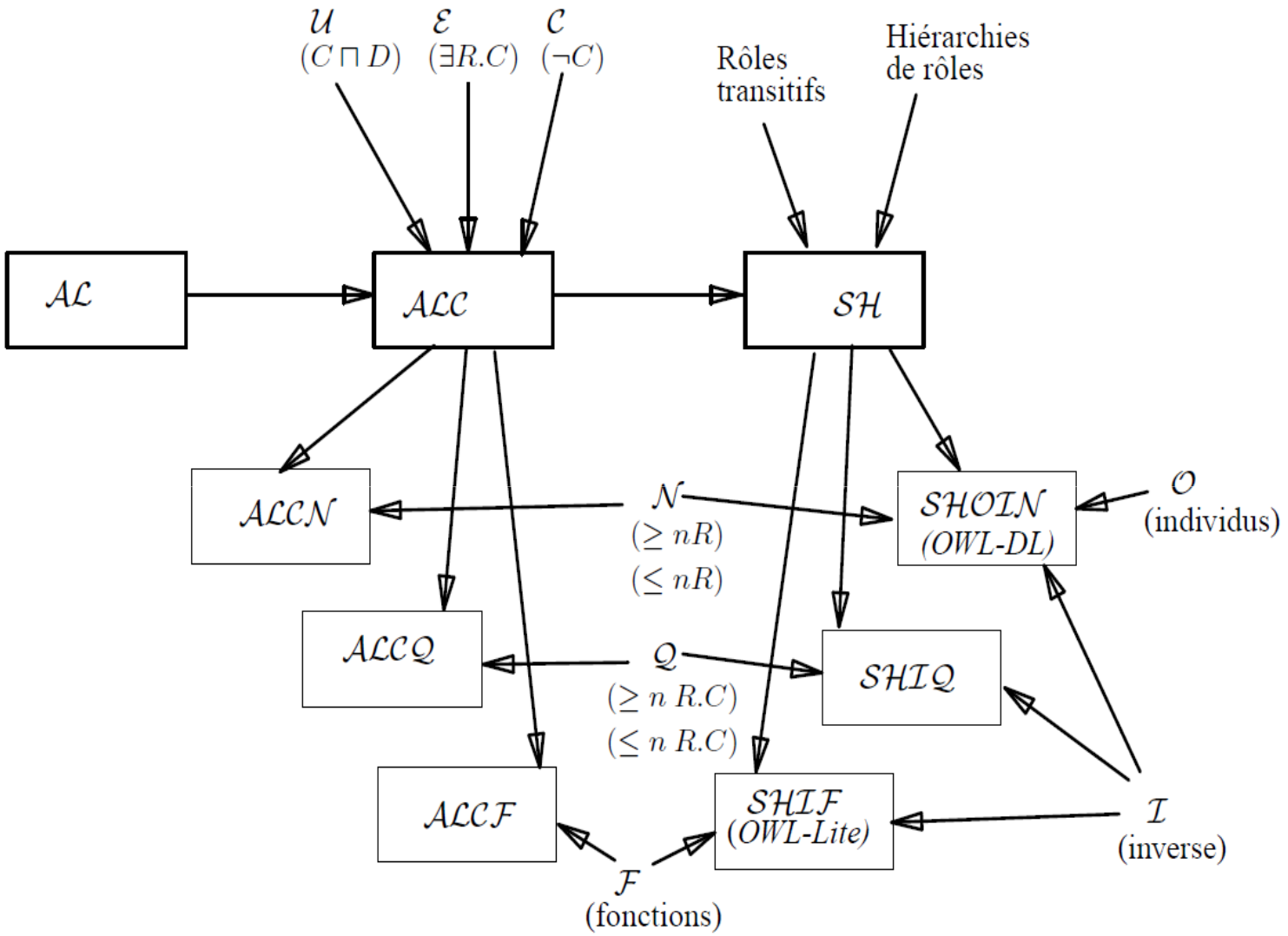
- Deux autres constructeurs, exprimant la *restriction de cardinalité (dénommés N)*,≤ n R et ≥ n R
- Sémantique :
- $\mathcal{I}(\geq n\ R) = \{a \in \Delta^{\mathcal{I}} \mid \{b \mid (a, b) \in \mathcal{I}(R)\} \mid \geq n\}$
- $\mathcal{I}(\leq n\ R) = \{a \in \Delta^{\mathcal{I}} \mid \{b \mid (a, b) \in \mathcal{I}(R)\} \mid \leq n\}$
- *Exemple :*
- le concept de père qui a exactement deux enfants :
  - HommeA2Enf≡Homme ⊓ ≥ 2 aEnfant ⊓ ≤ 2 aEnfant
  - FemmeMariée ≡ Femme ⊓ 1marié Avec

# La famille SH



- Les langages de la familles AL nous permettent de faire des descriptions complexes de concepts,
- mais sont plutôt limités en ce qui concernes les rôles
- SH est un une logique expressive :
- (S)la possibilité d'établir qu'une relation est une sous-propriété d'une autre relation(Exemple aDescendant ,parentDe ,mereDe ,pereDe…)
- la possibilité de définir une relation transitive (exemple frèreDe, aDescendant ..)
- S=ALC+TRANSITIVITE ,H :HIERARCHIE DE ROLE

# La famille SH



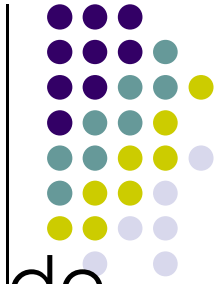
# La famille SH
# Restriction de cardinalité qualifiée (SHQ)

- Les deux constructeurs pour la restriction de cardinalité vus précédemment ne permettent pas de spécifier la classe concernée
- Pour cela, il faut plutôt avoir accès à un constructeur de restriction de cardinalité *qualifiée (dénommée Q)* :
- $\mathcal{I}(\geq n\ R.C) = \{a \in \ |\{b\,|\,(a, b) \in \mathcal{I}(R) \wedge b \in \mathcal{I}(C)\}| \geq n\}$
- $\mathcal{I}(\leq n\ R.C) = \{a \in \ |\{b\,|\,(a, b) \in \mathcal{I}(R) \wedge b \in \mathcal{I}(C)\}| \leq n\}$

## La famille SH
## Constructeur d'inversion (SHI)

- Pour spécifier qu'un rôle R1 est l'inverse d'un deuxième R2
- R1(x,y)-→ R2(y,x)
- *Exemple :*
- *ParentDe ,enfantDE*
- *Regarder , regardéPar*
- sémantique du constructeur d'inversion est définie de la manière suivante :
- $\mathcal{I}(R^-) = \{(y, x) \mid (x, y) \in \mathcal{I}(R)\}$
- *La déclaration se fait* :estRegardéPar ≡ regarde−

## La famille SH
## l'énumération (SHO)



- Le constructeur d'enumération(O) permet de désigner des individus sans classe dans la terminologie en énumérant explicitement la liste des individu de  cette classe.
- $\{a_1, \ldots, a_n\}$ où $a_1, \ldots, a_n$ sont des noms d'individus
- Selon la sémantique suivante:
- $\mathcal{I}(\{a_1, \ldots, a_n\}) = \{\mathcal{I}(a_1), \ldots, \mathcal{I}(a_n)\}$
- *Exemple :*
- MembrePermanentConseilSécurité ≡ {USA,FRANCE,RUSSIE,UK,CHINE}

# Remarques

- **équivalences logiques dans les langages :**
  - $\neg(C \sqcap D) \equiv \neg C \sqcup \neg D$
  - $\neg(C \sqcup D) \equiv \neg C \sqcap \neg D$
  - $\neg\exists R.A \equiv \forall R.\neg A$
  - $\neg\forall R.A \equiv \exists R.\neg A$
  - $\neg(\geq n\ R) \equiv \leq (n-1)\ R$
  - $\neg(\leq n\ R) \equiv \geq (n+1)\ R$
  - $\neg\neg C \equiv C$

# Conclusion

- Différents types :AL ,ALC ,ALCN ….SH ,SHIF,SHOIN ….
- *Permet de représenter des ontologies de différentes complexités selon le langage utilisé*
- *OWL :SHIF*
- *OWL DL:SHOIN (D) ( D pour les propriétés datatype)*