

# KNOWLEDGE ACQUISITION MODELING THROUGH DIALOGUE BETWEEN COGNITIVE AGENTS

Mehdi Yousfi-Monod, Violaine Prince

LIRMM, UMR 5506  
161 rue Ada  
34392 Montpellier Cedex 5 - France  
{yousfi,prince}@lirmm.fr



May 26th, 2005

# Synopsis

2

- ① Purpose
- ② Knowledge acquisition between agents
- ③ Our dialogical agents modellisation
- ④ Architecture and implementation
- ⑤ Conclusion, perspectives

# Purpose

## 3

- defining a set of knowledge acquisition algorithms

# Purpose

## 3

- defining a set of knowledge acquisition algorithms
- using only dialogue to acquire knowledge

# Purpose

## 3

- defining a set of knowledge acquisition algorithms
- using only dialogue to acquire knowledge
- being in a teacher-student situation

# Purpose

## 3

- defining a set of knowledge acquisition algorithms
- using only dialogue to acquire knowledge
- being in a teacher-student situation
- using of socratic dialogues

# Purpose

## 3

- defining a set of knowledge acquisition algorithms
- using only dialogue to acquire knowledge
- being in a teacher-student situation
- using of socratic dialogues
- belief revision and reasoning

# Purpose

## 3

- defining a set of knowledge acquisition algorithms
- using only dialogue to acquire knowledge
- being in a teacher-student situation
- using of socratic dialogues
- belief revision and reasoning
- using of the first-order predicate logic (with functions) for their knowledge bases (KB)



# Purpose

## 3

- defining a set of knowledge acquisition algorithms
- using only dialogue to acquire knowledge
- being in a teacher-student situation
- using of socratic dialogues
- belief revision and reasoning
- using of the first-order predicate logic (with functions) for their knowledge bases (KB)
- allowing the student to question the teacher about unknown predicates or functions before revising its KB

# Knowledge acquisition between agents

4

- Modelling of our communicating cognitive agents

# Modelling of our communicating agents

## Cognitive agent

### Knowledge base

*human*(*x*)  $\rightarrow$  *smart*(*x*)

*mortal*(*x*)  $\rightarrow$  *live*(*x*)

lesson to teach

*human*(*x*)  $\rightarrow$  *mortal*(*x*)

*human*(*John*)

*human*(*Mary*)

*human*(*x*)  $\rightarrow$  *not*(*bird*(*x*))

*animal*(*x*)  $\rightarrow$  *living*(*x*)

*animal*(*titi*)

...

# Modelling of our communicating agents

5

## Cognitive agent

### Knowledge base

 $human(x) \rightarrow smart(x)$  $mortal(x) \rightarrow live(x)$ 

lesson to teach

 $human(x) \rightarrow mortal(x)$  $human(John)$  $human(Mary)$  $human(x) \rightarrow not(bird(x))$  $animal(x) \rightarrow living(x)$  $animal(titi)$ 

...

### Model of itself

depends on the task and the competences

Role: Teacher or Student

Goal: Teach a lesson or Learn a lesson

# Modelling of our communicating agents

5

## Cognitive agent

### Knowledge base

$human(x) \rightarrow smart(x)$   
 $mortal(x) \rightarrow live(x)$

lesson to teach

$human(x) \rightarrow mortal(x)$   
 $human(John)$   
 $human(Mary)$

$human(x) \rightarrow not(bird(x))$   
 $animal(x) \rightarrow living(x)$   
 $animal(titi)$   
 ...

### Model of itself

depends on the task and the competences

Role: Teacher or Student

Goal: Teach a lesson or Learn a lesson

### Model of the interlocutor

Identity

Characteristics

Model of its KB

$mortal(x) \rightarrow live(x)$   
 $human(Peter)$   
 ...

# Knowledge acquisition between agents

6

- Modelling of our communicating cognitive agents
- Learning methods

# Knowledge acquisition between agents

- Modelling of our communicating cognitive agents
- Learning methods
  - A. Ravenscroft, R.M. Pilkington, 2000

# Knowledge acquisition between agents

- Modelling of our communicating cognitive agents
- Learning methods
  - A. Ravenscroft, R.M. Pilkington, 2000
- Dialogue models



# Knowledge acquisition between agents

- Modelling of our communicating cognitive agents
- Learning methods
  - A. Ravenscroft, R.M. Pilkington, 2000
- Dialogue models
  - P. Cohen, H. Levesque, 1992

# Knowledge acquisition between agents

- Modelling of our communicating cognitive agents
- Learning methods
  - A. Ravenscroft, R.M. Pilkington, 2000
- Dialogue models
  - P. Cohen, H. Levesque, 1992
  - J.L. Austin, 1975

# Knowledge acquisition between agents

- Modelling of our communicating cognitive agents
- Learning methods
  - A. Ravenscroft, R.M. Pilkington, 2000
- Dialogue models
  - P. Cohen, H. Levesque, 1992
  - J.L. Austin, 1975
  - M. E. Pollack, 1998

# Knowledge acquisition between agents

- Modelling of our communicating cognitive agents
- Learning methods
  - A. Ravenscroft, R.M. Pilkington, 2000
- Dialogue models
  - P. Cohen, H. Levesque, 1992
  - J.L. Austin, 1975
  - M. E. Pollack, 1998
  - G. Sabah *et al*, 1998

# Knowledge acquisition between agents

- Modelling of our communicating cognitive agents
- Learning methods
  - A. Ravenscroft, R.M. Pilkington, 2000
- Dialogue models
  - P. Cohen, H. Levesque, 1992
  - J.L. Austin, 1975
  - M. E. Pollack, 1998
  - G. Sabah *et al*, 1998
- Reasoning

# Knowledge acquisition between agents

- Modelling of our communicating cognitive agents
- Learning methods
  - A. Ravenscroft, R.M. Pilkington, 2000
- Dialogue models
  - P. Cohen, H. Levesque, 1992
  - J.L. Austin, 1975
  - M. E. Pollack, 1998
  - G. Sabah *et al*, 1998
- Reasoning
  - J.R. Josephson, S.G. Josephson, 1994

# Our approach

- KB specifications

# Our approach

- KB specifications
  - bootstrap



# Our approach

- KB specifications
  - bootstrap
  - **Connexity**

# The connexity notion

8

## A connex KB

$$t(z) \wedge p(x) \rightarrow q(y)$$

$$r(x) \rightarrow q(y)$$

$$s(x) \rightarrow r(y)$$

$$q(a)$$

$$r(b)$$

## The corresponding graph

$$t(z) \wedge p(x) \rightarrow q(y)$$

# The connexity notion

8

## A connex KB

$$t(z) \wedge p(x) \rightarrow q(y)$$

$$r(x) \rightarrow q(y)$$

$$s(x) \rightarrow r(y)$$

$$q(a)$$

$$r(b)$$

## The corresponding graph

$$t(z) \wedge p(x) \rightarrow q(y)$$

$$r(x) \rightarrow q(y)$$

# The connexity notion

## A connex KB

$$t(z) \wedge p(x) \rightarrow q(y)$$

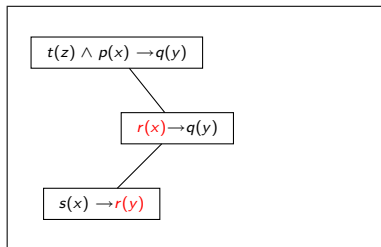
$$r(x) \rightarrow q(y)$$

$$s(x) \rightarrow r(y)$$

$$q(a)$$

$$r(b)$$

## The corresponding graph



# The connexity notion

## A connex KB

$$t(z) \wedge p(x) \rightarrow q(y)$$

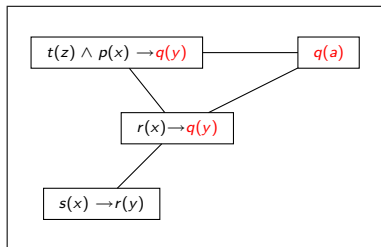
$$r(x) \rightarrow q(y)$$

$$s(x) \rightarrow r(y)$$

$$q(a)$$

$$r(b)$$

## The corresponding graph



# The connexity notion

8

## A connex KB

$$t(z) \wedge p(x) \rightarrow q(y)$$

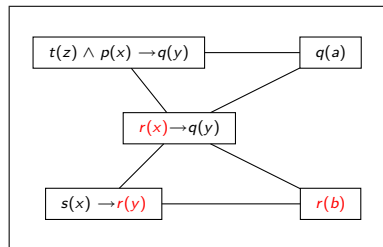
$$r(x) \rightarrow q(y)$$

$$s(x) \rightarrow r(y)$$

$$q(a)$$

$$r(b)$$

## The corresponding graph



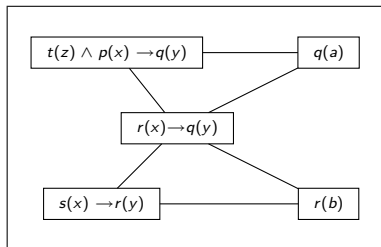
# The connexity notion

8

## A connex KB

$$\begin{aligned} &t(z) \wedge p(x) \rightarrow q(y) \\ &r(x) \rightarrow q(y) \\ &s(x) \rightarrow r(y) \\ &q(a) \\ &r(b) \end{aligned}$$

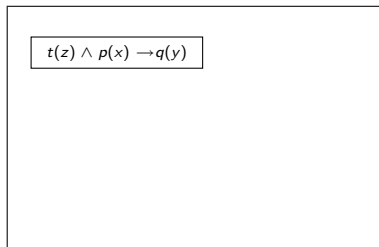
## The corresponding graph



## A **non** connex KB

$$\begin{aligned} &t(z) \wedge p(x) \rightarrow q(y) \\ &r(x) \rightarrow q(y) \\ &s(x) \rightarrow u(y) \\ &q(a) \\ &u(b) \end{aligned}$$

## The corresponding graph



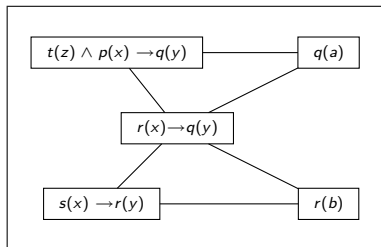
# The connexity notion

8

## A connex KB

$$\begin{aligned} &t(z) \wedge p(x) \rightarrow q(y) \\ &r(x) \rightarrow q(y) \\ &s(x) \rightarrow r(y) \\ &q(a) \\ &r(b) \end{aligned}$$

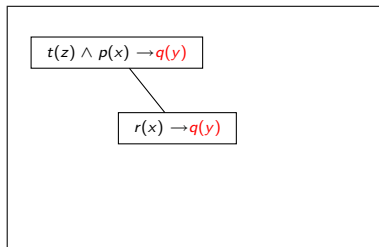
## The corresponding graph



## A **non** connex KB

$$\begin{aligned} &t(z) \wedge p(x) \rightarrow q(y) \\ &r(x) \rightarrow q(y) \\ &s(x) \rightarrow u(y) \\ &q(a) \\ &u(b) \end{aligned}$$

## The corresponding graph





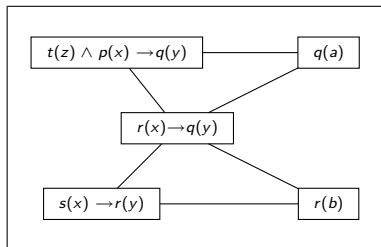
# The connexity notion

8

## A connex KB

$$\begin{aligned}
 &t(z) \wedge p(x) \rightarrow q(y) \\
 &r(x) \rightarrow q(y) \\
 &s(x) \rightarrow r(y) \\
 &q(a) \\
 &r(b)
 \end{aligned}$$

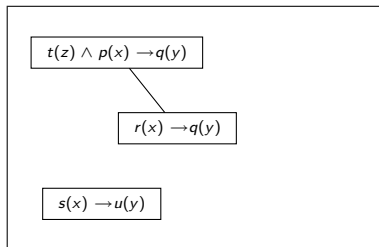
## The corresponding graph



## A **non** connex KB

$$\begin{aligned}
 &t(z) \wedge p(x) \rightarrow q(y) \\
 &r(x) \rightarrow q(y) \\
 &s(x) \rightarrow u(y) \\
 &q(a) \\
 &u(b)
 \end{aligned}$$

## The corresponding graph



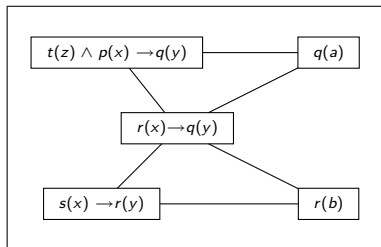
# The connexity notion

8

## A connex KB

$$\begin{aligned} &t(z) \wedge p(x) \rightarrow q(y) \\ &r(x) \rightarrow q(y) \\ &s(x) \rightarrow r(y) \\ &q(a) \\ &r(b) \end{aligned}$$

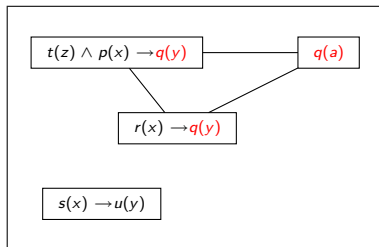
## The corresponding graph



## A **non** connex KB

$$\begin{aligned} &t(z) \wedge p(x) \rightarrow q(y) \\ &r(x) \rightarrow q(y) \\ &s(x) \rightarrow u(y) \\ &\textcolor{red}{q(a)} \\ &u(b) \end{aligned}$$

## The corresponding graph



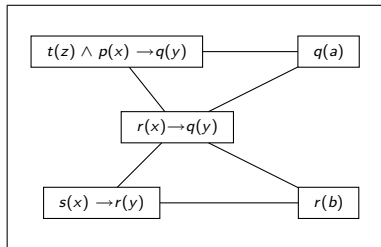
# The connexity notion

8

## A connex KB

$$\begin{aligned} &t(z) \wedge p(x) \rightarrow q(y) \\ &r(x) \rightarrow q(y) \\ &s(x) \rightarrow r(y) \\ &q(a) \\ &r(b) \end{aligned}$$

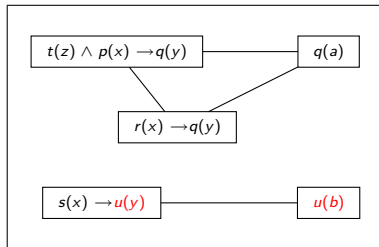
## The corresponding graph



## A **non** connex KB

$$\begin{aligned} &t(z) \wedge p(x) \rightarrow q(y) \\ &r(x) \rightarrow q(y) \\ &s(x) \rightarrow u(y) \\ &q(a) \\ &u(b) \end{aligned}$$

## The corresponding graph



# Our approach

- KB specifications
- the fonctional roles as modeller

# Our approach

- KB specifications
- the functional roles as modeller
  - ① give-knowledge

# Our approach

- KB specifications
- the functional roles as modeller
  - ① give-knowledge  
→ give-knowledge( $\text{cat}(x) \rightarrow \text{mortal}(x)$ ): "Cats are mortal."

# Our approach

- KB specifications
- the functional roles as modeller
  - ① give-knowledge
  - ② askfor/give-information

# Our approach

- KB specifications
- the functional roles as modeller
  - ① give-knowledge
  - ② askfor/give-information
    - askfor-information(cat(Folley)): "Is Folley a cat?"



# Our approach

- KB specifications
- the functional roles as modeller
  - ① give-knowledge
  - ② askfor/give-information
    - askfor-information(cat(Folley)): "Is Folley a cat?"
    - give-information(true): "Yes."

# Our approach

- KB specifications
- the functional roles as modeller
  - 1 give-knowledge
  - 2 askfor/give-information
  - 3 give-explanation (predicate case)

# Our approach

- KB specifications
- the functional roles as modeller
  - 1 give-knowledge
  - 2 askfor/give-information
  - 3 give-explanation (predicate case)
    - $\text{give-explanation}(\text{cat}(x) \leftrightarrow (\text{animal}(x) \wedge \text{pet}(x)))$ :
    - "A cat is a pet animal."

# Our approach

- KB specifications
- the functional roles as modeller
  - 1 give-knowledge
  - 2 askfor/give-information
  - 3 give-explanation (predicate case)
  - 4 say-(dis)satisfaction

# Our approach

- KB specifications
- the fonctional roles as modeller
- the tutored learning

# Our approach

- KB specifications
- the functional roles as modeller
- the tutored learning
  - the fundamental axioms

# Our approach

- KB specifications
- the functional roles as modeller
- the tutored learning
  - the fundamental axioms  
→ Z. Manna, 1974

# Our approach

- KB specifications
- the functional roles as modeller
- the tutored learning
  - the fundamental axioms
  - **FR interpretation axioms**



# Our approach

- KB specifications
- the functional roles as modeller
- the tutored learning
  - the fundamental axioms
  - FR interpretation axioms
  - the curious students

# Our approach

- KB specifications
- the functional roles as modeller
- the tutored learning
  - the fundamental axioms
  - FR interpretation axioms
  - the curious students
    - ① enlarge the base of one of their predicate

# Our approach

- KB specifications
- the functional roles as modeller
- the tutored learning
  - the fundamental axioms
  - FR interpretation axioms
  - the curious students
    - ① enlarge the base of one of their predicate  
→ learns  $p(a)$ , knows  $q(a)$ , then ask if  $q(x) \rightarrow p(x)$

# Our approach

- KB specifications
- the functional roles as modeller
- the tutored learning
  - the fundamental axioms
  - FR interpretation axioms
  - the curious students
    - ① enlarge the base of one of their predicate
    - ② increase its KB connexity

# Our approach

- KB specifications
- the functional roles as modeller
- the tutored learning
  - the fundamental axioms
  - FR interpretation axioms
  - the curious students
    - 1 enlarge the base of one of their predicate
    - 2 increase its KB connexity
    - 3 learn new terms

# Our approach

- KB specifications
- the functional roles as modeller
- the tutored learning
  - the fundamental axioms
  - FR interpretation axioms
  - the curious students
    - 1 enlarge the base of one of their predicate
    - 2 increase its KB connectivity
    - 3 learn new terms
    - 4 understand implications

# Our approach

- KB specifications
- the functional roles as modeller
- the tutored learning
  - the fundamental axioms
  - FR interpretation axioms
  - the curious students
    - 1 enlarge the base of one of their predicate
    - 2 increase its KB connexity
    - 3 learn new terms
    - 4 understand implications  
 $\rightarrow$  knows  $q(x) \rightarrow p(x)$ , asks if  $q(x) \rightarrow ? \rightarrow p(x)$

# Our approach

- KB specifications
- the functional roles as modeller
- the tutored learning
- the knowledge derivation strategies



# Our approach

- KB specifications
- the functional roles as modeller
- the tutored learning
- the knowledge derivation strategies
  - ① lesson strategies

# Our approach

- KB specifications
- the functional roles as modeller
- the tutored learning
- the knowledge derivation strategies
  - ① lesson strategies
  - ② **dialogue strategies**

# Our approach

- KB specifications
- the functional roles as modeller
- the tutored learning
- the knowledge derivation strategies
  - ① lesson strategies
  - ② dialogue strategies
    - the socratic method for the teacher

# Our approach

- KB specifications
- the functional roles as modeller
- the tutored learning
- the knowledge derivation strategies
  - ① lesson strategies
  - ② dialogue strategies
    - the socratic method for the teacher
  - ③ local strategies

# Our approach

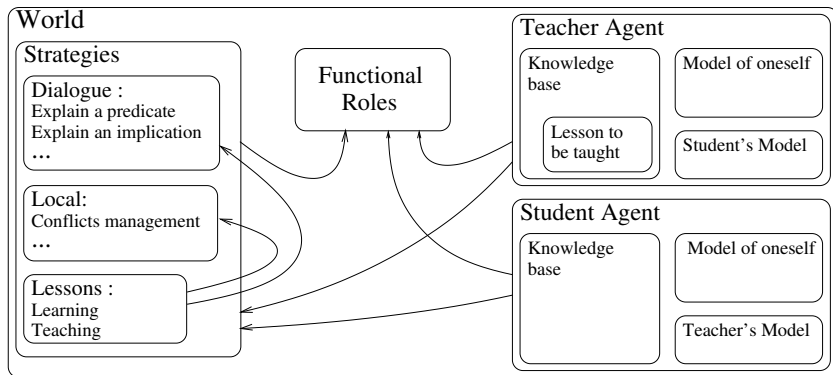
- KB specifications
- the functional roles as modeller
- the tutored learning
- the knowledge derivation strategies
  - ① lesson strategies
  - ② dialogue strategies
    - the socratic method for the teacher
  - ③ local strategies
    - the conflict management for the student

# Architecture, Implementation

10

- General architecture

# Architecture



# Architecture, Implementation

12

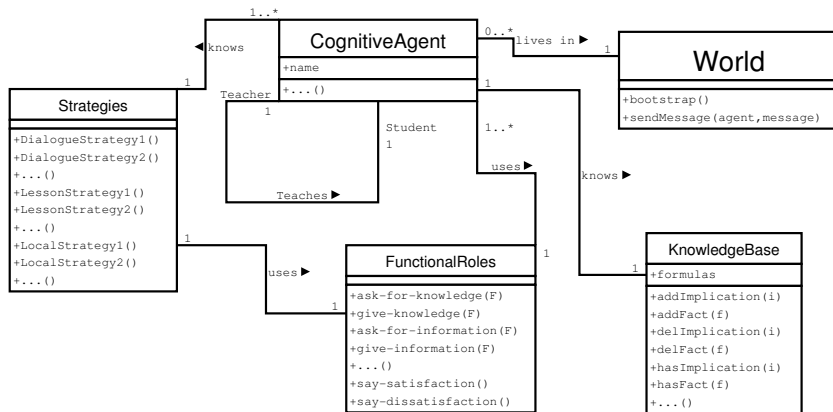
- General architecture
- **Implementation**



## Diagram class

## Diagram class

13



# Architecture, Implementation

14

- General architecture
- Implementation
- An example: a conflict management

# Example: a conflict management

## Teacher's KB:

*living\_being*( $x$ )  $\rightarrow$  *reproduce*( $x$ )

*animal*( $x$ )  $\rightarrow$  *living\_being*( $x$ )

*human*( $x$ )  $\rightarrow$  *mortal*( $x$ )

*human*( $x$ )  $\rightarrow$  *living\_being*( $x$ )

*human*( $x$ )  $\rightarrow$  *reproduce*( $x$ )

## Student's KB:

*human*( $x$ )  $\rightarrow$  *animal*( $x$ )

*animal*( $x$ )  $\rightarrow$  *living\_being*( $x$ )

*living\_being*( $x$ )  $\rightarrow$  *not*(*reproduce*( $x$ ))

# Example: a conflict management

## Teacher's KB:

*living\_being*( $x$ )  $\rightarrow$  *reproduce*( $x$ )

*animal*( $x$ )  $\rightarrow$  *living\_being*( $x$ )

**human**( $x$ )  $\rightarrow$  **mortal**( $x$ )

*human*( $x$ )  $\rightarrow$  *living\_being*( $x$ )

*human*( $x$ )  $\rightarrow$  *reproduce*( $x$ )

## Student's KB:

*human*( $x$ )  $\rightarrow$  *animal*( $x$ )

*animal*( $x$ )  $\rightarrow$  *living\_being*( $x$ )

*living\_being*( $x$ )  $\rightarrow$  *not*(*reproduce*( $x$ ))

Teacher - give-knowledge(*human*( $x$ )  $\rightarrow$  *mortal*( $x$ ))

# Example: a conflict management

## Teacher's KB:

*living\_being*(*x*)  $\rightarrow$  *reproduce*(*x*)  
*animal*(*x*)  $\rightarrow$  *living\_being*(*x*)  
*human*(*x*)  $\rightarrow$  *mortal*(*x*)  
*human*(*x*)  $\rightarrow$  *living\_being*(*x*)  
*human*(*x*)  $\rightarrow$  *reproduce*(*x*)

## Student's KB:

*human*(*x*)  $\rightarrow$  *animal*(*x*)  
*animal*(*x*)  $\rightarrow$  *living\_being*(*x*)  
*living\_being*(*x*)  $\rightarrow$  *not*(*reproduce*(*x*))  
*human*(*x*)  $\rightarrow$  *mortal*(*x*)

Teacher - give-knowledge(*human*(*x*)  $\rightarrow$  *mortal*(*x*))

Student - say-satisfaction()

# Example: a conflict management

## Teacher's KB:

*living\_being*( $x$ )  $\rightarrow$  *reproduce*( $x$ )  
*animal*( $x$ )  $\rightarrow$  *living\_being*( $x$ )  
*human*( $x$ )  $\rightarrow$  *mortal*( $x$ )  
**human**( $x$ )  $\rightarrow$  **living\_being**( $x$ )  
*human*( $x$ )  $\rightarrow$  *reproduce*( $x$ )

## Student's KB:

*human*( $x$ )  $\rightarrow$  *animal*( $x$ )  
*animal*( $x$ )  $\rightarrow$  *living\_being*( $x$ )  
*living\_being*( $x$ )  $\rightarrow$  *not*(*reproduce*( $x$ ))  
*human*( $x$ )  $\rightarrow$  *mortal*( $x$ )

Teacher - give-knowledge(*human*( $x$ )  $\rightarrow$  *mortal*( $x$ ))

Student - say-satisfaction()

Teacher - give-knowledge(*human*( $x$ )  $\rightarrow$  *living\_being*( $x$ ))

# Example: a conflict management

## Teacher's KB:

*living\_being*(*x*)  $\rightarrow$  *reproduce*(*x*)  
*animal*(*x*)  $\rightarrow$  *living\_being*(*x*)  
*human*(*x*)  $\rightarrow$  *mortal*(*x*)  
*human*(*x*)  $\rightarrow$  *living\_being*(*x*)  
*human*(*x*)  $\rightarrow$  *reproduce*(*x*)

## Student's KB:

**human**(*x*)  $\rightarrow$  **animal**(*x*)  
**animal**(*x*)  $\rightarrow$  **living\_being**(*x*)  
*living\_being*(*x*)  $\rightarrow$  *not*(*reproduce*(*x*))  
*human*(*x*)  $\rightarrow$  *mortal*(*x*)

Teacher - give-knowledge(*human*(*x*)  $\rightarrow$  *mortal*(*x*))

Student - say-satisfaction()

Teacher - give-knowledge(*human*(*x*)  $\rightarrow$  *living\_being*(*x*))

Student - say-satisfaction()

# Example: a conflict management

## Teacher's KB:

*living\_being*(*x*)  $\rightarrow$  *reproduce*(*x*)  
*animal*(*x*)  $\rightarrow$  *living\_being*(*x*)  
*human*(*x*)  $\rightarrow$  *mortal*(*x*)  
*human*(*x*)  $\rightarrow$  *living\_being*(*x*)  
*human*(*x*)  $\rightarrow$  *reproduce*(*x*)

## Student's KB:

**human**(*x*)  $\rightarrow$  **animal**(*x*)  
**animal**(*x*)  $\rightarrow$  **living\_being**(*x*)  
**living\_being**(*x*)  $\rightarrow$  **not**(*reproduce*(*x*))  
*human*(*x*)  $\rightarrow$  *mortal*(*x*)

Teacher - give-knowledge(*human*(*x*)  $\rightarrow$  *mortal*(*x*))

Student - say-satisfaction()

Teacher - give-knowledge(*human*(*x*)  $\rightarrow$  *living\_being*(*x*))

Student - say-satisfaction()

Teacher - give-knowledge(*human*(*x*)  $\rightarrow$  *reproduce*(*x*))



# Example: a conflict management

## Teacher's KB:

*living\_being(x) → reproduce(x)*  
*animal(x) → living\_being(x)*  
*human(x) → mortal(x)*  
*human(x) → living\_being(x)*  
*human(x) → reproduce(x)*

## Student's KB:

**human(x) → animal(x)**  
*animal(x) → living\_being(x)*  
*living\_being(x) → not(reproduce(x))*  
*human(x) → mortal(x)*

Teacher - give-knowledge(*human(x) → mortal(x)*)

Student - say-satisfaction()

Teacher - give-knowledge(*human(x) → living\_being(x)*)

Student - say-satisfaction()

Teacher - give-knowledge(*human(x) → reproduce(x)*)

Student - ask-for-information(*human(x) → animal(x)*)

# Example: a conflict management

## Teacher's KB:

$living\_being(x) \rightarrow reproduce(x)$   
 $animal(x) \rightarrow living\_being(x)$   
 $human(x) \rightarrow mortal(x)$   
 $human(x) \rightarrow living\_being(x)$   
 $human(x) \rightarrow reproduce(x)$

## Student's KB:

$human(x) \rightarrow animal(x)$   
 $animal(x) \rightarrow living\_being(x)$   
 $living\_being(x) \rightarrow not(reproduce(x))$   
 $human(x) \rightarrow mortal(x)$

Teacher - give-knowledge( $human(x) \rightarrow mortal(x)$ )

Student - say-satisfaction()

Teacher - give-knowledge( $human(x) \rightarrow living\_being(x)$ )

Student - say-satisfaction()

Teacher - give-knowledge( $human(x) \rightarrow reproduce(x)$ )

Student - ask-for-information( $human(x) \rightarrow animal(x)$ )

Teacher - give-information(*Unknown*)

# Example: a conflict management

## Teacher's KB:

$living\_being(x) \rightarrow reproduce(x)$   
 $animal(x) \rightarrow living\_being(x)$   
 $human(x) \rightarrow mortal(x)$   
 $human(x) \rightarrow living\_being(x)$   
 $human(x) \rightarrow reproduce(x)$

## Student's KB:

$human(x) \rightarrow animal(x)$   
 $animal(x) \rightarrow living\_being(x)$   
 $living\_being(x) \rightarrow not(reproduce(x))$   
 $human(x) \rightarrow mortal(x)$

Teacher - give-knowledge( $human(x) \rightarrow mortal(x)$ )

Student - say-satisfaction()

Teacher - give-knowledge( $human(x) \rightarrow living\_being(x)$ )

Student - say-satisfaction()

Teacher - give-knowledge( $human(x) \rightarrow reproduce(x)$ )

Student - ask-for-information( $human(x) \rightarrow animal(x)$ )

Teacher - give-information(*Unknown*)

**Student - ask-for-information( $animal(x) \rightarrow living\_being(x)$ )**

# Example: a conflict management

## Teacher's KB:

$living\_being(x) \rightarrow reproduce(x)$   
 $animal(x) \rightarrow living\_being(x)$   
 $human(x) \rightarrow mortal(x)$   
 $human(x) \rightarrow living\_being(x)$   
 $human(x) \rightarrow reproduce(x)$

## Student's KB:

$human(x) \rightarrow animal(x)$   
 $animal(x) \rightarrow living\_being(x)$   
 $living\_being(x) \rightarrow not(reproduce(x))$   
 $human(x) \rightarrow mortal(x)$

Teacher - give-knowledge( $human(x) \rightarrow mortal(x)$ )

Student - say-satisfaction()

Teacher - give-knowledge( $human(x) \rightarrow living\_being(x)$ )

Student - say-satisfaction()

Teacher - give-knowledge( $human(x) \rightarrow reproduce(x)$ )

Student - ask-for-information( $human(x) \rightarrow animal(x)$ )

Teacher - give-information(*Unknown*)

Student - ask-for-information( $animal(x) \rightarrow living\_being(x)$ )

Teacher - give-information(*True*)

# Example: a conflict management

## Teacher's KB:

$living\_being(x) \rightarrow reproduce(x)$   
 $animal(x) \rightarrow living\_being(x)$   
 $human(x) \rightarrow mortal(x)$   
 $human(x) \rightarrow living\_being(x)$   
 $human(x) \rightarrow reproduce(x)$

## Student's KB:

$human(x) \rightarrow animal(x)$   
 $animal(x) \rightarrow living\_being(x)$   
 $living\_being(x) \rightarrow not(reproduce(x))$   
 $human(x) \rightarrow mortal(x)$

Teacher - give-knowledge( $human(x) \rightarrow mortal(x)$ )

Student - say-satisfaction()

Teacher - give-knowledge( $human(x) \rightarrow living\_being(x)$ )

Student - say-satisfaction()

Teacher - give-knowledge( $human(x) \rightarrow reproduce(x)$ )

Student - ask-for-information( $human(x) \rightarrow animal(x)$ )

Teacher - give-information(*Unknown*)

Student - ask-for-information( $animal(x) \rightarrow living\_being(x)$ )

Teacher - give-information(*True*)

Student - ask-for-information( $living\_being(x) \rightarrow not(reproduce(x))$ )

# Example: a conflict management

## Teacher's KB:

$\text{living\_being}(x) \rightarrow \text{reproduce}(x)$   
 $\text{animal}(x) \rightarrow \text{living\_being}(x)$   
 $\text{human}(x) \rightarrow \text{mortal}(x)$   
 $\text{human}(x) \rightarrow \text{living\_being}(x)$   
 $\text{human}(x) \rightarrow \text{reproduce}(x)$

## Student's KB:

$\text{human}(x) \rightarrow \text{animal}(x)$   
 $\text{animal}(x) \rightarrow \text{living\_being}(x)$   
 $\text{living\_being}(x) \rightarrow \text{not}(\text{reproduce}(x))$   
 $\text{human}(x) \rightarrow \text{mortal}(x)$

Teacher - give-knowledge( $\text{human}(x) \rightarrow \text{mortal}(x)$ )

Student - say-satisfaction()

Teacher - give-knowledge( $\text{human}(x) \rightarrow \text{living\_being}(x)$ )

Student - say-satisfaction()

Teacher - give-knowledge( $\text{human}(x) \rightarrow \text{reproduce}(x)$ )

Student - ask-for-information( $\text{human}(x) \rightarrow \text{animal}(x)$ )

Teacher - give-information(*Unknown*)

Student - ask-for-information( $\text{animal}(x) \rightarrow \text{living\_being}(x)$ )

Teacher - give-information(*True*)

Student - ask-for-information( $\text{living\_being}(x) \rightarrow \text{not}(\text{reproduce}(x))$ )

Teacher - give-information(*False*)

# Example: a conflict management

## Teacher's KB:

$living\_being(x) \rightarrow reproduce(x)$   
 $animal(x) \rightarrow living\_being(x)$   
 $human(x) \rightarrow mortal(x)$   
 $human(x) \rightarrow living\_being(x)$   
 $human(x) \rightarrow reproduce(x)$

## Student's KB:

$human(x) \rightarrow animal(x)$   
 $animal(x) \rightarrow living\_being(x)$   
 ~~$living\_being(x) \rightarrow not(reproduce(x))$~~   
 $human(x) \rightarrow mortal(x)$

Teacher - give-knowledge( $human(x) \rightarrow mortal(x)$ )

Student - say-satisfaction()

Teacher - give-knowledge( $human(x) \rightarrow living\_being(x)$ )

Student - say-satisfaction()

Teacher - give-knowledge( $human(x) \rightarrow reproduce(x)$ )

Student - ask-for-information( $human(x) \rightarrow animal(x)$ )

Teacher - give-information(*Unknown*)

Student - ask-for-information( $animal(x) \rightarrow living\_being(x)$ )

Teacher - give-information(*True*)

Student - ask-for-information( $living\_being(x) \rightarrow not(reproduce(x))$ )

Teacher - give-information(*False*)

# Example: a conflict management

## Teacher's KB:

$living\_being(x) \rightarrow reproduce(x)$   
 $animal(x) \rightarrow living\_being(x)$   
 $human(x) \rightarrow mortal(x)$   
 $human(x) \rightarrow living\_being(x)$   
 $human(x) \rightarrow reproduce(x)$

## Student's KB:

$human(x) \rightarrow animal(x)$   
 $animal(x) \rightarrow living\_being(x)$   
 ~~$living\_being(x) \rightarrow not(reproduce(x))$~~   
 $human(x) \rightarrow mortal(x)$   
 $human(x) \rightarrow reproduce(x)$

Teacher - give-knowledge( $human(x) \rightarrow mortal(x)$ )

Student - say-satisfaction()

Teacher - give-knowledge( $human(x) \rightarrow living\_being(x)$ )

Student - say-satisfaction()

Teacher - give-knowledge( $human(x) \rightarrow reproduce(x)$ )

Student - ask-for-information( $human(x) \rightarrow animal(x)$ )

Teacher - give-information(*Unknown*)

Student - ask-for-information( $animal(x) \rightarrow living\_being(x)$ )

Teacher - give-information(*True*)

Student - ask-for-information( $living\_being(x) \rightarrow not(reproduce(x))$ )

Teacher - give-information(*False*)



# Example: a conflict management

## Teacher's KB:

$living\_being(x) \rightarrow reproduce(x)$   
 $animal(x) \rightarrow living\_being(x)$   
 $human(x) \rightarrow mortal(x)$   
 $human(x) \rightarrow living\_being(x)$   
 $human(x) \rightarrow reproduce(x)$

## Student's KB:

$human(x) \rightarrow animal(x)$   
 $animal(x) \rightarrow living\_being(x)$   
 ~~$living\_being(x) \rightarrow not(reproduce(x))$~~   
 $human(x) \rightarrow mortal(x)$   
 $human(x) \rightarrow reproduce(x)$

Teacher - give-knowledge( $human(x) \rightarrow mortal(x)$ )

Student - say-satisfaction()

Teacher - give-knowledge( $human(x) \rightarrow living\_being(x)$ )

Student - say-satisfaction()

Teacher - give-knowledge( $human(x) \rightarrow reproduce(x)$ )

Student - ask-for-information( $human(x) \rightarrow animal(x)$ )

Teacher - give-information(*Unknown*)

Student - ask-for-information( $animal(x) \rightarrow living\_being(x)$ )

Teacher - give-information(*True*)

Student - ask-for-information( $living\_being(x) \rightarrow not(reproduce(x))$ )

Teacher - give-information(*False*)

Student - say-satisfaction()

# Conclusion, perspectives

16

## 1 Conclusions

- Dialogue: efficient learning method
- Introduced notion: the Connexity of a KB
- The Functional Roles theory as modeller
- The potential of curious students
- Use of strategies to derive knowledge

## 2 Perspectives

- enrich the formulas type
- define new strategies
- use our learning method in conjunction with other methods