

# Concurrence et Répartition

## **Inspector Prolog**

Joao Costa Da Quinta, Lea Heiniger

28.03.2023

# Inspector prolog

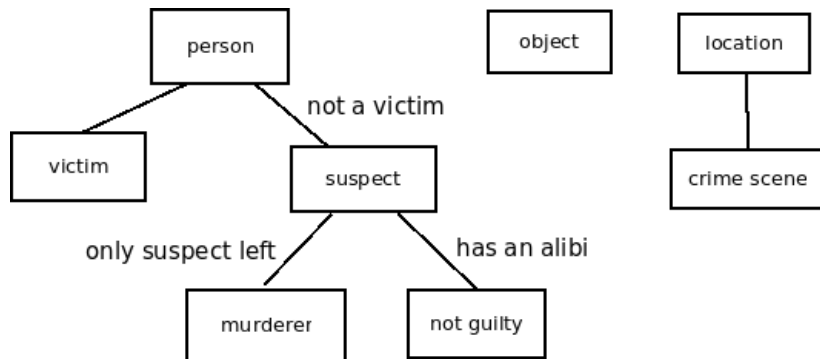
- Educational games to learn formal logic
- Implemented in prolog
- Cases to investigate

# Design of fact database

We designed a global structure that will be used to describe and implement scenarios.

It has to be independent of scenarios content.

# Definitions



# Properties

```
% object  
has_color(object,color).  
found_in(object, location).  
  
% person  
wears(person, object).  
has_distinction(person, distinction).  
  
lost(person, object).  
saw(person, person).
```

# First scenario

We started by writing the scenario.

- **who is the victim**
- **who are the suspects**
- **what are the locations**
- **what is the crime scene**
- **what are the clues**
- **where are the clues**

*Monsieur X*

*Jean Thy, Amy Khale, Sam  
Patick, Chris Minel*

*Chambre, Cuisine, Salon,  
Bibliotheque*

*Chambre*

*mégot, boucle d'oreille,  
couteau, ...*

# Implementation

We used the structure previously created in the fact database to implement the facts described in the scenario.

We had to add some rules for the reasoning :

```
was_in(P, L) :-person(P), forall(location(X), X=L).  
was_in(S, L) :-lost(S, O), found_in(O, L).  
was_in(S, L) :-not_guilty(P), was_in(P, L), saw(P, S).  
  
has_alibi(P) :-was_in(P, L), \+crime_scene(L).
```

# Definition and reasoning problems

- **How to verify an alibi ?**

→ Someone not guilty saw the person in a location (different from the crime scene).

→ The person lost an object at a location different from the crime scene.

- **How to describe a pretended location ?**

*Still not fixed*

- **How to link a distinction to clues for the reasoning ?**

*Still not fixed*



# Prolog problems

- **Negation of a predicate**  
→ `\+predicate()`
- **infinite recursion for was\_in predicate**  
*Still not fixed*

# Our next Goals

- **fix current problems**
- **Add possibilities to the fact data base** (global mechanism for the distinctions, add evidences gathering, ...)
- **Write other scenarios and implement them**
- **Start working on the logical interaction with the player**

# Semester plan

	1	2	3	4	5	6	7	8	9	10	11	12	13
refresh prolog knowledge													
choose game mechanics													
design facts database													
create scenarios													
implement facts database													
implement logic													
implement game --> interface													
generative model													

# Concurrence et Répartition

## **Inspector Prolog**

Joao Costa Da Quinta, Lea Heiniger

28.03.2023