

Concurrence et Répartition

Inspector Prolog

Joao Costa Da Quinta, Lea Heiniger

07.03.2023



SWI Prolog

Inspector prolog

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

Prolog

- Refresh our knowledge
- CLP(FD) library
- Function call()

Educational aspect

- Explicit logic formulas
- Select facts to use
- Infer new facts

Levelling the difficulty

- Introduce more complex logic formulas
- More facts available, some of them useless
- Introduce a maximum number of attempts

Implementation methods

The game mechanism will be independent of the content of the cases which allows us to :

- easily modify the content of a case
- be able to easily add or modify cases
- be able to randomly generate cases

Exemple : length of a list

Without CLP(FD) library prolog does not evaluate arithmetic expressions

```
len([], 0).  
len([H|L], Y+1) :-len(L,Y).
```

```
?- len([a,b,c],X).  
X = 0+1+1+1.
```


Exemple : length of a list

We have to modify the definition using CLP(FD) to be able to evaluate the addition.

```
:-use_module(library(clpfd)).  
len([], 0).  
len([H|L], Z) :-len(L,Y), Z #= Y+1.  
  
?- len([a,b,c],X).  
X = 3.
```

CLP(FD)

- Evaluate arithmetic expressions
- Logical reasoning on integers
- Constraint programming
- ...

Predicate as a parameter

To have a general game mechanism we need a way to apply a logical formula that works regardless of the formula selected.

Since logical formulas will be represented by predicates, we need to be able to **pass a predicate as a parameter of another predicate**.

call()

Calls a given predicate on the chosen parameters:

$$\text{call}(f(), p1, p2, p3). \Leftrightarrow f(p1, p2, p3).$$

The number of parameters is flexible since

`call\1`, `call\2`, `call\3`, ... and `call\8`

are defined

design fact database

We have to decide how each elements of a case (clues, witnesses, usable logical formulas, ...) will be represented in order to be able to interact with it.

once this is done we can start to create scenarios and implement them in the fact database

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

07.03.2023