

Fundamentals of Artificial Intelligence and Knowledge Representation – Module 4

Prof. Federico Chesani – 2nd of September, 2022

Available time: 30 min.

- 1) The candidate is invited to define a Prolog meta-interpreter **solve(Goal)** that solves the given Goal, but using a right-to-left selection rule of the subgoal to be solved instead of the usual left-to-right rule.

For example, given the following knowledge base:

```
p(1) :- write('I am solving P.').  
r(1) :- write('I am solving R.').  
q(X) :- p(X), r(X).
```

and the goal:

```
:- solve( p(X) ).
```

The expected outcome is:

```
I am solving R.  
I am solving P.  
Yes, X/1.
```

Notice that the invocation **solve(write(something))** will fail, because **write/1** is not defined in the clause database (it is a predefined, built-in predicate, with no logical meaning). The candidate should take care of extending the meta-interpreter to deal with such built-in predicate, by directly invoking it (through **call/1** meta-predicates).

- 2) The candidate is invited to describe the predicates/terminology used in the definition of the Event Calculus Framework.

Fundamentals of Artificial Intelligence and Knowledge Representation – Module 4

Prof. Federico Chesani – 2nd of September, 2022

Available time: 30 min.

Solution

1) *The candidate is invited to ...*

```
p(1) :- write('I am solving P.').  
r(1) :- write('I am solving R.').  
q(X) :- p(X), r(X).
```

```
solve(true) :- !.  
solve(write(X)) :- !, call(write(X)).  
solve( (A,B) ) :- !, solve(B), solve(A).  
solve(A) :- clause(A,B), solve(B).
```

2) *The candidate is invited ...*

See the slides