

# Fundamentals of Artificial Intelligence and Knowledge Representation –

academic year 2022—2023: Module 2 (Chesani)

previous academic years: Module 2 (ex-Gaspari), Module 4 (Chesani)

Prof. Federico Chesani – 9th of February, 2023

Available time: 1h.

- 1) The candidate is invited to define a Prolog predicate **filter/2**, that receives in input a list **L1** of terms, and returns in output a list **L2**. The list **L2** will contain all the terms appearing in **L1**, with a difference: whenever the term is of the type **p(X)** with **X** a ground number, then in the resulting list **L2** it should be inserted a term **p(Y)** where **Y** is **X** incremented of a unit.

To determine if a term **X** is a ground number, the candidate can use the pre-defined predicate **number/1**, that return **true** if and only if its argument is a ground number.

For example, if invoked with:

```
:- filter([p(1), 3, p(4), p(X)], L2).
```

The expected outcome is:

```
Yes, L2 = [p(2), 3, p(5), p(X)]
```

- 2) The candidate is invited to define a meta-interpreter for the Prolog language, where the selection of the subgoal in the current resolvent is right-most rather than left-most (as it is usually).
- 3) The candidate is invited to introduce the notions of close world assumption and open world assumption, and to briefly discuss how Prolog and Description Logics deal with these aspects.
- 4) The candidate is invited to briefly introduce the ALC Description Logics, mentioning the operators that are supported (negation, AND, ALL, EXISTS), and their meaning (possibly with a short example for each operator).

# Fundamentals of Artificial Intelligence and Knowledge Representation –

academic year 2022—2023: Module 2 (Chesani)

previous academic years: Module 2 (ex-Gaspari), Module 4 (Chesani)

Prof. Federico Chesani – 9th of February, 2023

Available time: 1h.

## Solution

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

```
filter([], []).
```

```
filter([p(X)|T], [p(Y)|Rest]) :-
```

```
    number(X),
```

```
    !,
```

```
    Y is X + 1,
```

```
    filter(T, Rest).
```

```
filter([H|T], [H|Rest]) :-
```

```
    filter(T, Rest).
```

2) *The candidate is invited ...*

See the slides

3) *The candidate is invited ...*

See the slides

4) *The candidate is invited ...*

See the slides