

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 – 12th of September, 2023

Available time: 1h.

- 1) A knowledge base of predicates `parent/2` is given, describing the relation between two persons. E.g., `parent(francesco, federico)` means that `francesco` is the son of `federico`.

The candidate is invited to define a Prolog predicate `people/1`, that returns in output a list `L` of the people mentioned in the knowledge base, without repetitions.

For example, if the KB is:

```
parent(francesco, federico).  
parent(chiara, federico).  
parent(francesco, elena).
```

When queried with:

```
:- people(L).
```

The expected outcome is:

Yes, `L = [chiara, federico, francesco, elena]`

- 2) The candidate is invited to explain the meta-interpreter “vanilla”, and to briefly explain the meaning of each clause.
- 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 introduce the terminology used in the Event Calculus Framework having care, in particular, of indicating the framework axioms.

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 – 12th of September, 2023

Available time: 1h.

Solution

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

```
parent(francesco, federico).  
parent(chiara, federico).  
parent(francesco, elena).
```

```
people(L) :-
```

```
    findall((X,Y), parent(X,Y), L1),  
    flat(L1,L2),  
    clean_rep(L2, L).
```

```
flat([], []).
```

```
flat([(X,Y)|T], [X,Y|Rest]) :-  
    flat(T, Rest).
```

```
clean_rep([], []).
```

```
clean_rep([X|T], [X|Rest]) :-  
    \+member(X,T),  
    !,  
    clean_rep(T,Rest).
```

```
clean_rep([_|T], Rest) :-  
    clean_rep(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