Fundamentals of Artificial Intelligence and Knowledge Representation
Mod. 2

academic years 2022/2023 – 2025/2026: Module 2 (Chesani)
previous academic years: Module 2 (ex-Gaspari) + Module 4 (Chesani)

29th of January 2026 – Available time: 1h.

Exam B

1) Define a predicate `split/3` that receives as input a list of terms `id/2`. The predicate returns in output (by means of the second and third parameters) two lists `First` and `Second`. In particular, `First` will contain the first argument of the `id/2` terms; `Second` will contain the second argument of the `id/2` terms. Unfortunately, the list given in input contains repetitions (the whole term `id/2` is repeated): the candidate is expected to not insert repetitions in the output; it is not necessary to respect the order of the input list. To avoid repetition, the candidate must <u>define and use</u> a `member/2` predicate that returns `True` if a term is present in a list.

   For example, for the query:
   ```
   % split/3: split(InputList, First, Second)
   ?-  split(  [id(12,alice),   id(18,bob),   id(23,charlie),   id(12,alice)],
   First, Second).
   ```

   The expected answer is:
   ```
   Yes, First = [ 18, 23, 12 ]
        Second = [ bob, charlie, alice ]
   ```

2) The candidate is invited to illustrate the "Vanilla" Prolog meta-interpreter, having care of explaining each clause. The candidate should also briefly mention hot to change the meta-interpreter in order to achieve a right-to-left evaluation of conjunction of goals.

3) The candidate is invited to represent, using the EC approach, the following situation:
   A car has a single button that controls front and rear heating fans.
   - *Initially both the fans are off;*
   - *when the button is pressed the first time, only rear fan goes on;*
   - *when the button is pressed a second time, front fan goes on as well;*
   - *when the button is pressed a third time, rear fan goes off;*
   - *when the button is pressed a fourth time, front fan goes off as well; the system then is again in the initial situation.*

   For simplicity, the candidate is asked to model only the behaviour of fluents `front_on` and `rear_on`.

4) The candidate is invited to briefly illustrate the difference between Open World Assumption and Close World Assumption, providing a very short example of formalisms adopting the two different approaches.