MASTER IN ARTIFICIAL INTELLIGENCE (UDC - USC - UVigo)
REASONING AND PLANNING exam. January 9th, 2026

Last Name: _____

First Name: _____

INSTRUCTIONS This part of the exam covers units 1-6 and is weighted with a maximum of **45 points (pt) from a total of 100 pt**. Unit 7 (15 pt) is covered in a separated sheet. For the test, use the original statement sheet and avoid corrections or unclear marking (ask for a new blank sheet if needed).

— EXAM —

Exercise 1 (20pt). Each question has at least one correct answer and its total score depends on whether you check: some incorrect answer = **-3pt**; all correct answers and nothing else = **5pt**; only correct answers, but not all = **3pt**; blank = **0pt**. A total negative score in Ex. 1 counts as 0pt in the rest of the exam.

1.1) Mark those interpretations that are *countermodels* of the formula $p \rightarrow (q \rightarrow p)$ in classical propositional logic:

- ☐ $\{p, q\}$
- ☐ $\{p\}$
- ☐ $\emptyset$
- ☐ None of the above

1.2) The logic program P `a :- b, c.  c :- a.  b :- d.` is obviously stratified since it contains no negation. Mark each rule below such that, if added to P, the result would not be stratified.

- ☐ `d :- b`
- ☐ `b :- not a`
- ☐ `c :- not d`
- ☐ `d :- not c`

1.3) Mark the stable models of the following logic program `p :- not q.  q :- p.  r :- not q.  q :- not r.`

- ☐ $\emptyset$
- ☐ $\{p\}$
- ☐ $\{q\}$
- ☐ $\{p, q\}$
- ☐ $\{r\}$
- ☐ $\{p, r\}$
- ☐ $\{q, r\}$
- ☐ $\{p, q, r\}$

1.4) The logic program P_1 `q :- not p.  q :- p.` corresponds to the formula $(\neg p \rightarrow q) \wedge (p \rightarrow q)$. Mark below the Here-and-There (HT) interpretations that are models of P_1 in HT:

- ☐ $H = \{p\}, T = \emptyset$
- ☐ $H = \emptyset, T = \{p\}$
- ☐ $H = \emptyset, T = \{p, q\}$
- ☐ $H = \emptyset, T = \{q\}$

Exercise 2 (5pt). The previous program P_1 seems to produce the same effect as program P_2 with just the fact `q.`. However, you should have found some HT-model of P_1 above that is not HT-model of q. As a consequence, P_1 and P_2 are *not strongly equivalent*. **Find a single rule** R using atoms p, q such that $P_1 \cup R$ and $P_2 \cup R$ have different behaviour (that is, they have some difference in their respective stable models). Explain the result of adding R.

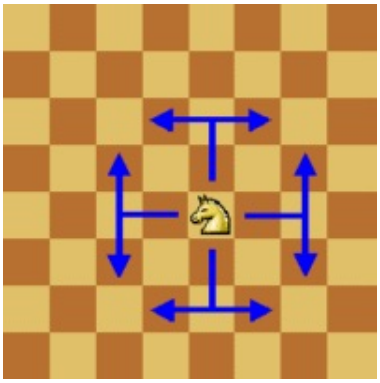
Exercise 3 (10pt). Given a chessboard of `n x n` with `n` ≥ 1 we want to place `m` > 0 knights so they do not attack each other, using predicate `horse(X,Y)` to mean that cell `X`, `Y` contains a horse. NOTE: a chess knight attacks as shown in the figure below: attacked positions correspond to the cells with arrow heads. Complete the following ASP code to solve the problem (try to write as few rules as possible).

```
#const n=3.
#const m=5.
row(1..n). col(1..n).

% Generate all possible placements of m Knights



% Forbid mutual attacks
:- horse(X,Y), horse(X+1,Y+2).            % (*)
```



Exercise 4 (5pt). Explain why constraint `(*)` does not need to check the conditions `X+1<n` and `Y+2<n`. How many ground rules does rule `(*)` generate when $n = 3$? [ ] (explain the answer below too)

Exercise 5 (5pt). In the taxi routing problem of the second lab assignment, write telingo code to forbid that a taxi picks somebody that was already in a station (you may use your own predicates).

```
#program dynamic.
```

In general, this constraint removes valid plans that are not too useful (because of moving some passenger already located at a station). But suppose we allowed a passenger to be initially at a station: could there be scenarios in which all the possible plans are ruled out by the constraint above? If so, draw a possible initial configuration where this may happen.