

$\exists x \text{violinplayer}(x)$

$\exists x (\text{violinplayer}$

# LANGUAGES FOR AI, MODULE 2

FEBRUARY 14, 2024

Time: 2 hours.

Please write the answers to exercises 1-2, 3-4 and 5-6 on three different sheets of paper.

1. (5 points) Consider the language of propositional logic. Use natural deduction to prove that the following holds, or find a counter-example to show that it does not hold (remember that  $\neg F$  is only a shorthand for  $F \rightarrow \perp$ ).

- $\vdash ((A \wedge B) \rightarrow (A \wedge B \wedge C)) \rightarrow C$
- $\vdash ((A \rightarrow C) \wedge (A \rightarrow B) \wedge A) \rightarrow (B \wedge C)$

$\forall y (\text{nurse}(y) \wedge \exists x (\text{violinplayer}(x))$

2. (5 points) Transform the following propositional logic formula into an equivalent formula in Conjunctive Normal Form

$$(A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow (C \rightarrow B))$$

3. (5 points) Bianchi, Rossi, and Neri are suspected of a crime. They testify as follows:

- Bianchi: "Rossi is guilty and Neri is innocent".
- Rossi: "If Bianchi is guilty then so is Neri".
- Neri: "I'm innocent, but at least one of the others is guilty".

Express the testimony of each suspect as a formula in propositional logic. Assuming that all testimonies are true, who is innocent and who is guilty?

4. (5 points) Formalise the following sentence into FOL:

- There is a violin player who is liked by every nurse.
- Some swans are not white.
- If someone is noisy everybody is annoyed.

$\exists x (\text{violinplayer}(x) \rightarrow \dots$

5. (6 points) Using MiniZinc or CLP, create a program that colors the following map keeping in mind that neighboring regions should not have the same color. For simplicity, you can use numbers instead of colors, e.g., 1 = red, 2 = blue, etc. Use the minimum number of different colors.

|     |     |     |
|-----|-----|-----|
| A 1 | 2 D | 3 F |
| 2   | 3 B |     |
| C   | 1   | E   |

6. (6 points) Create a Prolog predicate `multiply(L, N, Res)` that multiplies the all elements of the given list L, N times. For example:

?- `multiply([a, b, 2], 3, Res).`

`Res = [a, a, a, b, b, b, 2, 2, 2].`

Note that you can create and call another predicate that uses an accumulator variable to keep count of how many times you have repeated the current element of the list.

`multiply([], 0, Res).`  
`multiply([_:_], N[_:_], Res) :- multiply([_:_], N[_:_], Res).`

$A \rightarrow B = \neg A \vee B$