Languages for AI, Module 2
January 12, 2023

Time: 2 hours.

In the following $A, B, \ldots$ are propositional variables, $a, b, \ldots$ constant symbols, $f, g, \ldots$ function symbols, $X, Y \ldots$ variables, $p, q, \ldots$ predicate symbols and $F, G, \phi, \psi, \ldots$ formulas (unless differently specified).

1. 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
   - $(A \to (B \to C)) \to ((A \wedge B) \to C)$
   - $(F \to G) \to (\neg G \to F)$

2. (4 points) Prove that that $\phi \models \psi$ (ψ is a logical consequence of ϕ) or that $\phi \not\models \psi$ for the following formulas:
   - $\phi : (A \to B) \wedge (A \to C)$ and $\psi : \neg B \vee \neg C \vee A$
   - $\phi : A \vee B \vee \neg C$ and $\psi : (A \wedge B) \to C$

3. A robbery took place in Wherevercity. Investigators found that if the robbery occurred at midnight then surely the robber(s) fled the city. If the robbery didn't happen at midnight then it involved two robbers. The sheriff of Wherevercity claims that the robber is out of the city or there were two of them. Formalize the situation with propositional logic and establish whether the three statements are contradictory. Provide a short motivation for your answer.

   Solution

   We can define the following propositional variables:

   M ::= The robbery occurred at midnight

   C ::= The Robber(s) fled the city

   T ::= Two robbers are involved

   Then we can formalize the tree statements as follow:

   1. M $\to$ C
   2. not(M) $\to$ T
   3. C v T

   They are not contradictory because the formula consisting of their conjunctios in satisfiable, meaning that the three claims can be simultaneously satisfied.

4. Let us consider the language $\mathcal{L} = \{Y, C, U, W, K\}$ where Y, C, U, W are unary relational symbols ($Y(x)$ stands for "x is under 30 years old", $G(x)$ stands for "x is a girl", $C(x)$ stands for "x is enrolled in college", $U(x)$ stands for "x can use a computer", $W(x)$ stands for "x will find work"), while K is a binary relational symbol ($K(x, y)$ stands for "x knows y"). Formalize the following sentences:

   (a) All people under 25 years old who is enrolled in college know how to use a computer or know someone who can.

   (b) Among those who don't know how to use a computer there are some who will find work.

   (c) Everyone knows someone under 25 years old who will find work.

   Solution

   (a) $\forall x.((Y(x) \wedge C(x)) \to (U(x) \vee \exists y.(K(x, y) \wedge U(y)))$

   (b) $\exists x.(\neg C(x) \wedge W(x))$

   (c) $\forall x \exists y.(Y(y) \wedge K(x, y) \wedge W(y))$

5. Professor X has published all the exams' dates for the whole year at once. Even though this looks like a student-friendly measure, his ultimate goal is to write the lowest number of exercises. In fact, Professor X's exam contains two exercises, A and B, which he wants to recycle between two dates. In order for him to do so, no student should attend both dates but that is unfortunately what Chris, Diana and Ethan have done. By carefully examining the student lists of the 5 available exams, X notices that:

- Chris is in the list for date 1, 3, and 5.
- Diana is in the list for date 1 and 2.
- Ethan is in the list for date 2 and 4.

Additionally, to avoid rumors of his exam always looking the same, Professor X does not want any exercise to appear in the same position twice (e.g. exercise 1 cannot appear twice in position A). Will X manage to prepare all the exams using only 6 exercises?

If that is the case, how can the three students be sure to make sure that, if Professor X only uses 6 exercises, one of them will for sure find the same exercise twice? To this end, note that a student can attend a maximum of 3 dates. Model the problem using CLP and show a possible solution.

Solution:

One variable for each position for each exam: Ai, Bi with domain 1..5

No exercise can appear twice in the same position:

alldifferent([A1, A2, A3, A4, A5])

alldifferent([B1, B2, B3, B4, B5])

The exercises in the same exam should all be different:

alldifferent([Ai, Bi]) with i in [1..5]

Exercises of exams with the same student have to be all different:

alldifferent([A1, B1, A3, B3, A5, B5])

alldifferent([A1, B1, A2, B2])

alldifferent([A2, B2, A4, B4])

Possible solution:

Exam 1: (1, 2), Exam 2: (3, 4), Exam 3: (4, 3), Exam 4: (2,1), Exam 5: (5, 6)

The three students can all attend the first two dates and then if two of them also attend a third date they are sure one of them will find the same exercise as long as they go to different ones. For example, Chris could attend dates 1, 2 and 3, Diana dates 1, 2 and 4 and Ethan 1, 2.

6. Define a Prolog predicate that creates the list of integer numbers with first and last element given.

   For example:

   ?- create_list(-1, 7, X).

   X = [-1, 0, 1, 2, 3, 4, 5, 6, 7].

   Note that you can use the predicates seen in the course and the predicate last/2, that returns the last element of a given list:

   ?- last([1,2,3], X).

   X = 3.

   Solution

   create_list(X, X, [X]).

   create_list(X, Y, [X—T]) :- X1 is X+1, create_list(X1, Y, T).

   or (using last/2)

   create_list(X, Y, Res) :- create_list(X, Y, [], Res).

   create_list(X, Y, [], Res) :- create_list(X, Y, [X], Res), !.

   create_list(_,Y,Res,Res) :- last(Res, Y).

   create_list(X,Y,Acc,Res) :- last(Acc, Z), Z1 is Z+1, append(Acc, [Z1], Acc1), create_list(X, Y, Acc1, Res).