Logic and Applications

Task 11

Hand in as a `.pdf` before the deadline indicated in Brightspace

And save your Coq file(s) on the Coq server within your **HOMxxx** account before that time as well!

- This is a group assignment, so please define groups of two persons.
- Make sure to include all names in your submission.
- Learning tasks for this course typically consist of two parts:
  - Part A, which is for practice and discussion in the group tutorials.
  - Part B, which will count for the average of the weekly assignments.
- Exercises of Part A will not be graded at all.
- All exercises of part B will be graded with a score between 0 and 10.
- The final grade for this task is the sum of all the graded exercises, divided by the number of exercises that were graded in total.
- At the end of the semester, the average score for these weekly tasks should be at least 5.5.
- This year, there are 12 tasks scheduled in the full course.
- Because Part A does not influence your grade, some of you might be tempted not to do them, however the exercises in Part A can also be asked at the exam!
- Note that your submission should be typed and in English and converted to a `.pdf` file.
- **For security reasons, the Coq server is only reachable via VPN.**

- Just like last week, this week's task consists of doing proofs on paper and doing proofs in Coq. However, you may be tempted to do all proofs in Coq first, and then
  - either simply copy the tree you get from Coq as a picture,
  - or recreate the tree on paper

  to your solution to this task.

  But please, don't do this! The format used for displaying the tree in Coq, does not correspond to the format you are supposed to use, so the first option is definitely wrong. And for the second option it holds that you will not be able to use Coq at your exam, and then you will have to do it on paper directly. As previous years have shown, this is really different from making proofs in Coq. **So please, practice with pen and paper.**
- **In the natural deduction exercises of this task, you may use all the rules that are listed on the schema that you can find on Brightspace.**

Background

Natural deduction for predicate logic with numerical domains

In this task you will show that you know how to create derivation trees for predicate logic using numbers and intervals, both on paper as well as with Coq.

Reading list

Please read this and make sure that you understand the main concepts.

1. The slides.
2. The grammar for propositional logic.
3. The grammar for predicate logic.
4. The rules for natural deduction.

Goals

After completion of this assignment, you are able to:

1. Make derivation trees with natural deduction with the rules for quantifiers for theorems in predicate logic with numbers and intervals.
2. Reuse a problem that has already been solved to find a proof for a similar theorem.
3. Use negation rules for problems with numbers.
4. Make derivation trees with natural deduction with the rules for quantifiers for theorems in predicate logic with numbers and intervals.
5. Prove both simple and difficult theorems in predicate logic with numbers and intervals with natural deduction in Coq.
6. Prove a difficult theorem in predicate logic with numbers and intervals with natural deduction in Coq.

Instructions

Part A

Exercise 1

Make derivation trees for the following theorems. There is no formal definition of the domain model, but you may assume that each predicate gets exactly the right number and type of parameters it needs and $\mathbb{R}$ indicates the domain of real numbers.

a) $\forall x : \mathbb{R}, \neg(\exists u : \mathbb{R}, U\ (x+7)\ u) \quad \vdash \quad \forall x : \mathbb{R}, \forall y : \mathbb{R}, \neg U\ (x-5)\ (y+3)$

b) $\forall t : \mathbb{R}, X\ t \to Y\ (t+1) \vee Y\ (t+2) \quad \vdash \quad \forall t : \mathbb{R}, X\ t \to (\exists u : \mathbb{R}, u \geq t+1 \wedge Y\ u)$

c) $\exists x : \mathbb{R}, x \in (3,7) \wedge S\ x \quad \vdash \quad \exists y : \mathbb{R}, y \in [3,7] \wedge S\ y$

Exercise 2

A few weeks ago we covered the so-called "three minute light". The associated slides contain several specifications for it. In this exercise you will prove that some of these specifications imply each other.

- Log in to the server `https://proofweb[0-9].cs.ru.nl/new`.
- Open the file `Task11_three_minutes_basic.v`.
- This file contains three different specifications in natural language.
- Subsequently a theorem is proven from which it follows that specification `spec1` implies the specification `spec2`.
- Go through that proof in Coq and try to truly understand what is happening.
- There is also a theorem stating that `spec2` implies `spec1`. Finish the proof at the location of the placeholder `(*! benb_proof *)`. Obviously without the use of `tauto`.
- Save your file with “Save”.
- If you got a red cross, Coq did not accept your proof. If you see an orange flag appearing, you changed something where you were not allowed to change anything or tried to sneak in a `tauto` anyway. Make sure you get a green check mark!

a) Give a short reflection as usual on theorem `spec2ImpliesSpec1`.

- Now open the file `Task11_three_minutes_negations.v`.
- This file contains the same three specifications, but this time you will do something with `spec3` which contains negations.
- Finish the proof at the location of the placeholder `(*! benb_proof *)`. Obviously without the use of `tauto`.
- Save your file with “Save”.
- If you got a red cross, Coq did not accept your proof. If you see an orange flag appearing, you changed something where you were not allowed to change anything or tried to sneak in a `tauto` anyway. Make sure you get a green check mark!

b) Give a short reflection as usual on theorem `spec1ImpliesSpec3`.

Part B

Exercise 3

Make derivation trees for the following theorems. There is no formal definition of the domain model, but you may assume that each predicate gets exactly the right number and type of parameters it needs and $\mathbb{R}$ indicates the domain of real numbers.

a) $\neg(\exists x : \mathbb{R}, S\ (x - 4)) \quad \vdash \quad \forall z : \mathbb{R}, \neg S\ (z + 5)$

b) $\forall x : \mathbb{R}, S\ x \quad \vdash \quad \exists y : \mathbb{R}, y \in (3, 7) \wedge S\ (y + 10)$

c) $\vdash \quad (\forall x : \mathbb{R}, \forall y : \mathbb{R}, P\ (x - y) \wedge Q\ (x - y)) \to (\forall x : \mathbb{R}, P\ (x + 7) \vee (\exists y : \mathbb{R}, U\ x\ (y - 3)))$

Exercise 4

Natural deduction is simply a matter of practice. In this exercise you will get the opportunity to show how good you have become at proving theorems in predicate logic with numbers and intervals in Coq. [*Note:* Normally you can only score 10 points for the whole exercise, however, in this exercise, you can score 10 points for each of the subquestions.]

- Log in to the server `https://proofweb[0-9].cs.ru.nl/new`.
- There is a whole series of files with the names `Task11_real*.v`.
- Open the file `Task11_real001.v`.

- Finish the proof at the location of the placeholder `(*! benb_proof *)`. Obviously without the use of `tauto`.
- Save your file with “Save”.
- If you got a red cross, Coq did not accept your proof. If you see an orange flag appearing, you changed something where you were not allowed to change anything or tried to sneak in a `tauto` anyway. Make sure you get a green check mark!

a) Write in your solution to this assignment whether you succeeded in proving this theorem or not. Did you succeed the first time? Or did you have to start over? And if you did not succeed, which proof obligation $\Gamma,\ \Sigma \vdash \alpha$ did you fail to prove?

b) Do the same for `Task11_real005.v`.

c) Do the same for `Task11_real008.v`.

Exercise 5
A few weeks ago we covered the so-called “three minute light” artifact. `SpecOn` and `SpecOff` were two different attempts to formalize the specification. In the slides, it is explained why `SpecOn` is wrong, as it leads to a contradiction.

- Log in to the server `https://proofweb[0-9].cs.ru.nl/new`.
- Open the file `Task11_three_minutes_On_Off.v`.
- This file contains the domain model, the specification `SpecOn`, and the theorem `SpecOnIsWrong` which formalizes that `SpecOn` is wrong with a specific counter-example.
- Finish the proof at the location of the placeholder `(*! benb_proof *)`. Obviously without the use of `tauto`.
- Save your file with “Save”.
- If you got a red cross, Coq did not accept your proof. If you see an orange flag appearing, you changed something where you were not allowed to change anything or tried to sneak in a `tauto` anyway. Make sure you get a green check mark!

a) Give a short reflection as usual on theorem `SpecOnIsWrong`.

Part C

Or how to increase your grade...

For the exercises in Part B you will get **five** grades. Normally, the grade for this task would be the average of these grades.

However, this week it is also possible to increase your grade to a maximum of a 12[1] by doing extra exercises. For every additional theorem that you try to prove in the series `Task11_real*.v` on the Coq server, you will earn the normal score, but the sum of your scores will still be divided by **five**. So even if the five mandatory exercises did not go well, you can still get a high grade for this task by practicing more and **saving** all your attempts on the server! Note that you don’t need to reflect on these extra exercises in this task as you had to do in Part B: simply saving them on the server suffices. The extra exercises contain both easy ones and difficult ones.

Note that you can only increase your grade for Part B if you actually handed in a PDF with a serious attempt to the Part B exercises!

[1]Yes, 12 is strictly more than the usual maximum score of 10, but this is done to encourage you to do your best and make up for the tasks you didn’t do that well earlier this year.

Products

You can use this list to check whether your solution is complete or not. A complete solution consists of:

- For Exercise 1:
  - Three derivation trees for the given theorems.
- For Exercise 2:
  - A reflection on either succeeding or not in proving the theorem `spec2` implies `spec1`.
  - A reflection on either succeeding or not in proving the theorem `spec1` implies `spec3`.
- For Exercise 3:
  - Three derivation trees for the given theorems.
- For Exercise 4:
  - A short reflection on either succeeding or not in proving the three given theorems.
- For Exercise 5:
  - A short reflection on either succeeding or not in proving the given theorem.

Introspection

You can use this list of questions to check if you understood the material. You do not have to hand in the answers, but you are expected to answer them for your self none the less.

1. Do you understand how the rules for repl, lins and int work?
2. Do you know the rules by heart or do you (still) need the template?
3. Do you understand the comments provided in the Coq files for exercise 2?
4. Can you easily create proofs like the ones in exercise 2? in Coq? If so, then the proof in the project should not be that difficult, assuming the theorem is correct.
5. Do you understand how the rules for repl, lins and int work?
6. Do you know the rules by heart or do you (still) need the template?
7. Did Coq use any abbreviations that do not conform to our grammar?
8. Did the explanation in natural language of why `SpecOn` is wrong help you in finding the proof for `SpecOnIsWrong`?