Constructive Logic (15-317), Spring2022
Assignment 4: Quantification and Arithmetic

Instructor: Klaas Pruiksma
TAs: Runming Li, Onyekachi Onyeador, Viraj Puri, Xiao Yu

Due: Thursday, Feb 17, 2022, 11:59 pm

The assignments in this course must be submitted electronically through Gradescope. Written homework PDFs and coding Dcheck files will both go to Gradescope. For this homework, you will be submitting two files:

- `hw4.deriv` (your coding solutions)
- `hw4.pdf` (your written solutions)

1 Quantification

Distribution of Quantifiers

In class, we saw that universal quantification distributes over conjunction, that is,

$$(\forall x : \tau . A(x) \wedge B(x)) \equiv (\forall x : \tau . A(x)) \wedge \forall x : \tau . B(x) \textit{ true}.$$

In this section, we will explore various other distributivity properties of quantifiers.

Task 1 (12 points)**.** Dually, existential quantification distributes over disjunction, that is,

$$(\exists x : \tau . A(x) \vee B(x)) \equiv (\exists x : \tau . A(x)) \vee \exists x : \tau . B(x) \textit{ true}.$$

In this task, you will show this equivalence by giving a natural deduction proof of each of the following directions in Dcheck, using the `AR` system:

a. $(\exists x : \tau . A(x) \vee B(x)) \supset (\exists x : \tau . A(x)) \vee \exists x : \tau . B(x)$ *true*

b. $(\exists x : \tau . A(x)) \vee (\exists x : \tau . B(x)) \supset \exists x : \tau . A(x) \vee B(x)$ *true*

Your solution should go in `hw4.deriv`. For clarification on how to write quantification proof in Dcheck, please look at the course website[1] and the Dcheck documentation[2].

Classy Quantifiers

Task 2 (8 points)**.** For each of the following judgments, give a constructive natural deduction proof in Dcheck if it is constructively valid. If it is not constructively valid, state this by returning a proof of $\top$ using the TI rule. *The following judgments are all classically valid.*[3]

a. $(\neg \forall x : \tau . \neg A(x)) \supset \exists x : \tau . A(x)$ *true*

b. $(\exists x : \tau . A(x)) \supset \neg \forall x : \tau . \neg A(x)$ *true*

Your solution should go in `hw4.deriv`. Note that while this task is autograded, there will not be instant feedback.

For All the Exists

Task 3 (8 points)**.** Consider the following proposition

$$((\forall x : \tau . \forall y : \tau . A(x) \supset A(y)) \wedge T) \supset \exists x : \tau . \forall y : \tau . A(x) \supset A(y)$$

When trying to prove this proposition you will run into issues. In this problem, you will fix the proposition so that it becomes provable.

Replace T with some proposition that makes the whole proposition provable. You may use the connectives $\exists, \forall, \wedge, \vee, \supset, T$, but not F (which could trivialize the problem). You also may not use the predicate '$A(\_)$.

a. Provide a proposition that would make the above statement provable if it replaced T. Your solution should go in `hw4.pdf`.

b. Provide a proof of the now provable statement. Your solution should go in `hw4.deriv`. While this task will be autograded to ensure that you have given a valid proof, it will be manually graded to check that the proposition being proved meets the requirements above.

[1] `https://www.andrew.cmu.edu/user/kpruiksm/15317s22/example.deriv`
[2] `https://www.andrew.cmu.edu/user/kpruiksm/15317s22/dcheck.pdf`
[3] This problem first appeared in 15-317 HW's a decade ago. So like these propositions, it is truly a classic

Heyting Arithmetic

Thus far, the propositions we've discussed have been true for generic atoms. With Heyting Arithmetic, we will finally be able to reason about the properties of math. In our case, the naturals and equality.

Rules of Equality

Task 4 (12 points)**.** Prove the following proposition in Dcheck, using the `AR` system:

$$\forall x : \texttt{nat}.\forall y : \texttt{nat}.\forall z : \texttt{nat}.\ (x = y \wedge y = z) \supset x = z$$

For clarification on how to use Heyting Arithmetic, please look at the course website[4]. Your solution should go in `hw4.deriv`.

[4] `https://www.andrew.cmu.edu/user/kpruiksm/15317s22/example.deriv`