Constructive Logic (15-317), Fall 2024
Assignment 3: Proof Terms and Verifications & Uses

Constructive Logic Staff
(Instructor: Karl Crary)

Due: Wednesday, September 18, 2024, 11:59 pm

This assignment will have a written portion and two coding portions. You will submit all three portions through Gradescope, to the assignments labelled "Homework 3 (written)", "Homework 3 (sml)", and "Homework 3 (dcheck)." Please submit files named "`hw.pdf`", "`hw.sml`", and "`hw.deriv`" to each assignment respectively.

We recommend that you typeset your written solutions. Most students use LaTeX, but other software is acceptable. (Please put each task on its own page to speed up grading.) If you choose not to typeset your solutions, be aware that you are answerable for your handwriting. Any that the grader has difficulty reading (in the sole judgement of the grader), will be marked wrong.

For the coding portions you will use Dcheck and Standard ML. You can find documentation on Dcheck at `cs.cmu.edu/~crary/dcheck/dcheck.pdf` and a sample file at `cs.cmu.edu/~crary/dcheck/example.deriv`. (Be aware that the sample file uses several logics that we have not seen yet in class.)

1 Proof Terms

Using Dcheck, give derivations that provide proof terms for each of the following propositions:

Task 1 (12 points).

a. $\neg A \wedge \neg B \supset \neg(A \vee B)$

b. $(A \supset T) \wedge (F \supset A)$

c. $((A \supset B) \wedge (A \supset C)) \supset (A \supset (B \wedge C))$

d. $((A \vee B) \supset C) \supset (A \supset C) \wedge (B \supset C)$

That is, prove `M :: P` for some `M` that you devise and the indicated `P`. Name your derivations `task1a` through `task1d`.

2 Proofs as programs

In this program you will look at the proof-as-programs paradigm not through the lens of theoretical proof terms, but as actual Standard ML programs. We are interested in the propositions:

a. $(A \wedge B \supset C) \supset A \supset B \supset C$

b. $((A \supset B) \supset B) \supset A$

c. $(A \supset B) \supset (\neg B \supset \neg A)$

d. $((A \vee B) \wedge \neg A) \supset B$

Some of these propositions are true, others are false.

Task 2 (8 points). In your SML solution, write a module named `ProofsAsPrograms` with the following signature:

```
signature PROOFS_AS_PROGRAMS =
   sig
      val curry : (('a * 'b -> 'c) -> 'a -> 'b -> 'c) option
      val abba : ((('a -> 'b) -> 'b) -> 'a) option
      val contrapositive : (('a -> 'b) -> (('b -> void) -> ('a -> void))) option
      val exclusion : (('a, 'b) sum * ('a -> void) -> 'b) option
   end
```

Each field should be `SOME` if the corresponding proposition is true, and `NONE` if it is not. **Do not use exceptions, recursion, or any other "cheat."** Your code will execute in an environment containing the following definitions:

```
datatype ('a, 'b) sum = INL of 'a | INR of 'b
type void
val abort : void -> 'a
```

You may find it helpful to derive the proof term using formal system and then translate it into SML.

Note: the autograder will report any SML type errors in your code; but to avoid revealing the answer, the autograder will not provide additional feedback on any problem for which you answer `NONE`.

Task 3 (3 points). In your written solution, translate the following propositions into SML types in a similar fashion to the above. (Do not add the options.)

a. $((A \supset B) \wedge (B \supset C)) \supset (A \supset C)$

b. $A \supset (T \supset A)$

c. $\neg\neg(A \vee \neg A)$

3 Verifications & Uses

Consider the ♡ connective:[1]

$$\dfrac{\begin{array}{cc}[A\ \mathsf{true}]_v & [A\ \mathsf{true}]_w \\ \vdots & \vdots \\ B\ \mathsf{true} & C\ \mathsf{true}\end{array}}{\heartsuit(A,B,C)\ \mathsf{true}}\ \heartsuit I^{v,w}$$

$$\dfrac{\heartsuit(A,B,C)\ \mathsf{true} \quad A\ \mathsf{true} \quad \begin{array}{c}[B\ \mathsf{true}]_u \\ \vdots \\ D\ \mathsf{true}\end{array}}{D\ \mathsf{true}}\ \heartsuit E1^u \qquad \dfrac{\heartsuit(A,B,C)\ \mathsf{true} \quad A\ \mathsf{true} \quad \begin{array}{c}[C\ \mathsf{true}]_u \\ \vdots \\ D\ \mathsf{true}\end{array}}{D\ \mathsf{true}}\ \heartsuit E2^u$$

Task 4 (5 points). Give appropriate rules for ♡ in verifications & uses.

Task 5 (6 points). Using Dcheck, give a derivation of the judgement:

$$(\neg P \wedge Q) \supset ((P \supset Q) \supset (\neg P \supset \neg Q)) \supset F \uparrow$$

Name your derivation `task5`. (Remember that Dcheck takes the propositions `P` and `Q` to be atomic.)

4 Proof Term Rules

Consider again the ♡ connective from the previous section. Let us use the syntax $\mathsf{hearti}(x.M, y.N)$ for the proof term for $\heartsuit I$, and the syntax $\mathsf{heartel}(M, N, x.P)$ and $\mathsf{hearter}(M, N, x.P)$ for the proof terms for $\heartsuit E1$ and $\heartsuit E2$.

Task 6 (4 points). Give proof term deduction rules corresponding to $\heartsuit I$, $\heartsuit E1$, and $\heartsuit E2$.

5 Natural Deduction Mastery

Task 7 (4 points). Using Dcheck, give a derivation of the judgement:

$$\neg A \vee \neg B \supset \neg(A \wedge B)\ \mathsf{true}$$

Name your derivation `task7`. **Instant feedback is turned off for this task, so be extra careful.**

[1] in Latex: `\heartsuit`