Constructive Logic (15-317), Fall 2024
Assignment 11: Focused and Linear Logic

Constructive Logic Staff
(Instructor: Karl Crary)

Due: Wednesday, November 20, 2024, 11:59 pm

This assignment will have a written portion and a coding portion. You will submit both portions through Gradescope, to the assignments labelled "Homework 11 (written)" and "Homework 11 (code)." Please submit a file named "`hw.pdf`" to the former, and a file named "`hw.deriv`" to the latter.

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 portion you will use Dcheck. 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 Polarization

Task 1 (5 points). Consider the following partially depolarized formula:

$$((A^+ \vee B^-) \supset F) \supset ((A^+ \supset F) \wedge (B^- \supset F))$$

Come up with two *distinct* polarizations of the formula: adding shifts in appropriate places and replacing the conjunction with a polarized conjunction. You do not need to prove them. (This is a written problem. Include your answer in `hw.pdf`.)

2 Focusing

Provide derivations of the following Focused Logic judgements using Dcheck:

Task 2 (5 points). Define a derivation named `task2` that derives:

$$\cdot;\cdot \xrightarrow{R} \downarrow((P^+ \supset Q^-) \wedge^- (P^+ \supset R^-)) \supset (P^+ \supset (Q^- \wedge^- R^-))$$

Task 3 (5 points). Define a derivation named `task3` that derives:

$$\cdot;\cdot \xrightarrow{R} \downarrow P^- \supset \downarrow(\downarrow P^- \supset Q^-) \supset \downarrow((\downarrow P^- \wedge^+ \downarrow Q^-) \supset R^-) \supset R^-$$

Task 4 (5 points). Define a derivation named `task4` that derives:

$$\cdot;\cdot \xrightarrow{R} \downarrow(\uparrow P^+ \wedge^- (P^+ \supset \uparrow Q^+)) \supset \downarrow((Q^+ \vee R^+) \supset \uparrow R^+) \supset \uparrow R^+$$

3 Linear Logic Mastery

Task 5 (5 points). Define a derivation named `task5` that derives:

$$((A \multimap B) \,\&\, 1) \multimap ((A \multimap C) \,\&\, 1) \multimap A \multimap (B \,\&\, C)\ \mathsf{true}$$

Instant feedback is turned off for this task, so be extra careful.

4 Blocks World

In class we looked at *Blocks World*, an example of encoding *state* in linear logic. Blocks World is a class of scenarios in which there is a table, some number of blocks which can be stacked on top of each other, and a robotic arm which can pick up and move blocks. The following atomic predicates are used:

a. $\texttt{ontable}(x)$ means that the block x is sitting directly on the table.

b. $\texttt{on}(x, y)$ means that the block x is directly on top of the block y.

c. $\texttt{clear}(x)$ means that the block x does not have anything on top of it.

d. $\texttt{empty}$ means that the robotic arm's hand is empty.

e. $\texttt{holds}(x)$ means that the robotic arm's hand is holding x.

The possible state transitions for Blocks World are given by the following axioms:

a. $\forall x.\forall y.\, \texttt{on}(x, y) \otimes \texttt{clear}(x) \otimes \texttt{empty} \multimap \texttt{clear}(y) \otimes \texttt{holds}(x)$

b. $\forall x.\, \texttt{ontable}(x) \otimes \texttt{clear}(x) \otimes \texttt{empty} \multimap \texttt{holds}(x)$

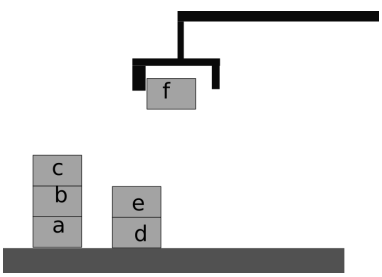
c. $\forall x.\forall y.\, \texttt{clear}(y) \otimes \texttt{holds}(x) \multimap \texttt{on}(x, y) \otimes \texttt{clear}(x) \otimes \texttt{empty}$

d. $\forall x.\, \texttt{holds}(x) \multimap \texttt{ontable}(x) \otimes \texttt{clear}(x) \otimes \texttt{empty}$

Task 6 (10 points). So far we have assumed that the table is infinitely broad and can therefore accomodate any number of blocks. Now consider the case that the table has only a finite number of spaces for blocks on it.

Modify the axioms of Blocks Worlds to support this scenario. You should use an atomic predicate `space`, which represents an open space on the table. A correct solution to this task will not depend on the size of the table.

Task 7 (5 points). Consider the following Blocks World scenario:



Write a formula in linear logic that expresses this configuration, assuming that the table is large enough to accommodate **four** blocks.