# Constructive Logic (15-317), Spring 2023
# Assignment 7/8: Logic Programming (90 points)

Instructor: Frank Pfenning

Due: Tuesday, April 25, 2023, 11:59 pm

This assignment has a written portion and a Prolog programming portion. You can find instructions for running the Ciao implementation of Prolog on Andrew on the course software page. You may also install a standard Prolog implementation (for example, Ciao, SWI Prolog, or GNU Prolog) on your own machine.

You will submit all portions through Gradescope.

We recommend that you typeset your written solutions. Most students use LaTeX, but other software is acceptable. If you choose not to typeset your solutions, be aware that your handwriting must be **legible**.

## 1 Backward Chaining and Certification (50 points)

In this problem we explore logic programming in the pure Horn fragment of Prolog, as introduced in Lectures 18–22.

**Task 1.** (20 points) Augment the rules for backward chaining from Lecture 19 (summarized in Appendix A) to maintain proof terms as introduced for natural deduction in Lecture 4. You should assume a distinct $c : D$ for every clause in the program $\Gamma$. From the natural deduction perspective these would be considered hypothesis; from the proof term perspective, these would be considered variables.

We call these rules for *certifying backward chaining*.

**Task 2.** (10 points) State the theorem expressing that if a sequent in the certifying backward chaining calculus is derivable, then the extracted proof term is correct in natural deduction. Since backward chaining uses several forms of sequents, this should be a multipart statement directly suitable for a proof by simultaneous rule induction. You do not need to prove it.

**Task 3.** (20 points) Extend the metainterpreter meta.pl from Lecture 19 to include proof terms as designed in Task 1. Since the metainterpreter does not explicitly deal with quantifiers, your proof terms should not explicitly represent quantifier instantiation but leave this to (sound!) unification. The result should be an interpreter for Horn clauses that searches for derivations like Prolog, but also maintains proof terms.

At the top level there should be a predicate

```
backchain(Gamma, M, atom(P))
```

where `Gamma` is the suitably augmented program, `M` is a proof term, and `P` is the representation of an atomic predicate.

- If `M` is a logic variable, its answer substitution should be the proof term for `P` found by Prolog search.
- If `M` is a complete proof term (not containing an logic variables), then `backchain` should act as a checker that `M` represents a proof of `P`.
- If `M` is a proof term and `P` is a logic variable, then the instantiation of `P` should be the (most general) proposition proved by `M`. This corresponds to *type inference* similar to what we live-coded in (lec18.pl) during Lecture 18.

## 2 Forward Inference (40 points)

**Task 4.** (20 points) Show the cases for $\vee R_1$ and $\vee L$ in the completeness proof of the forward sequent calculus as in Theorem 2 in Lecture 22. This proof is in the language without falsehood, so the $\vee L$ rule has the form

$$\frac{\Gamma_1, A \longrightarrow \gamma \quad \Gamma_2, B \longrightarrow \gamma}{\Gamma_1, \Gamma_2, A \vee B \longrightarrow \gamma} \vee L$$

For the next two tasks, you should use the forward sequent calculus in Figure 1 in Lecture 22 to prove or refute the given proposition for an *atomic proposition* $A$. You should start with inversion, and then specialize the rules to the left/right subformulas of your goal sequent. We leave it to decide whether you introduce names for subformulas or not.

**Task 5.** (10 points) $(\neg\neg A) \supset A$

**Task 6.** (10 points) $\neg\neg(A \vee \neg A)$

## A Backward Chaining

**Horn clauses.**

$$
\begin{array}{llll}
\text{Programs} & \Gamma & ::= & D \mid \cdot \mid \Gamma_1, \Gamma_2 \\
\text{Horn clauses} & D & ::= & \forall x.\, D \mid G \supset P \mid P \\
\text{Goals} & G & ::= & P \mid G_1 \wedge G_2 \mid \exists x.\, G(x)
\end{array}
$$

**Judgments.**

$$
\begin{array}{l}
\Gamma \xrightarrow{\mathsf{C}} P \\
\Gamma, [D] \xrightarrow{\mathsf{FL}} P \\
\Gamma \xrightarrow{\mathsf{FR}} [G]
\end{array}
$$

**Choice.**

$$
\frac{D \in \Gamma \quad \Gamma, [D] \xrightarrow{\mathsf{FL}} P}{\Gamma \xrightarrow{\mathsf{C}} P}\ \mathsf{FLC}
$$

**Left Focus.**

$$
\frac{}{\Gamma, [P] \xrightarrow{\mathsf{FL}} P}\ \mathsf{id}
\qquad
\frac{\Gamma \xrightarrow{\mathsf{FR}} [G]}{\Gamma, [G \supset P] \xrightarrow{\mathsf{FL}} P}\ {\supset}L
\qquad
\frac{\Gamma, [D(t)] \xrightarrow{\mathsf{FL}} P}{\Gamma, [\forall x.\, D(x)] \xrightarrow{\mathsf{FL}} P}\ \forall L
$$

**Right Focus.**

$$
\frac{\Gamma \xrightarrow{\mathsf{FR}} [G_1] \quad \Gamma \xrightarrow{\mathsf{FR}} [G_2]}{\Gamma \xrightarrow{\mathsf{FR}} [G_1 \wedge G_2]}\ \wedge R
\qquad
\frac{\Gamma \xrightarrow{\mathsf{FR}} [G(t)]}{\Gamma \xrightarrow{\mathsf{FR}} [\exists x.\, G(x)]}\ \exists R
\qquad
\frac{\Gamma \xrightarrow{\mathsf{C}} P}{\Gamma \xrightarrow{\mathsf{FR}} [P]}\ \mathsf{CFR}
$$