Lecture 7: Logic Programming

Mingtong Lin

September 29, 2025

1 Introduction

Lecture outline:

- Quantifiers
- Logic programming

2 Quantifiers

2.1 Grammar

In this lecture, we continue with the polarized sequent calculus from last time, but extended it with quantifiers. The grammar and the blurring rules are the same from last time, but the focusing rules are presented slightly differently.

$$\begin{array}{ll} A, B ::= A^+ \mid A^- & \text{formulas} \\ A^+, B^+ ::= \exists x : \tau.\, A & \text{positive connectives} \\ A^-, B^- ::= A \supset B \mid \forall x : \tau.\, A \mid p & \text{negative connectives} \end{array}$$

$$\frac{u : A^- \in \Gamma \quad \Gamma \mid [A^-] \Rightarrow C}{\Gamma \Rightarrow C}\ \mathsf{focus}_L^u \qquad \frac{\Gamma \Rightarrow [A^+]}{\Gamma \Rightarrow A^+}\ \mathsf{focus}_R$$

$$\frac{\Gamma, A^+ \Rightarrow C}{\Gamma \mid [A^+] \Rightarrow C}\ \mathsf{blur}_L \qquad \frac{\Gamma \Rightarrow A^-}{\Gamma \Rightarrow [A^-]}\ \mathsf{blur}_R$$

$$\frac{}{\Gamma \mid [p^-] \Rightarrow p^-}\ \mathsf{init}^-$$

2.2 Logical Rules

We first present the quantifier rules with some redundancies.

$$\frac{\Gamma, y : \tau \Rightarrow A[y/x]}{\Gamma \Rightarrow \forall x : \tau.\, A}\ \forall R \qquad \frac{\Gamma \Rightarrow t : \tau \quad \Gamma, \forall x : \tau.\, A \mid [A[t/x]] \Rightarrow C}{\Gamma \mid [\forall x : \tau.\, A] \Rightarrow C}\ \forall L$$

$$\frac{\Gamma \Rightarrow t : \tau \quad \Gamma \Rightarrow [A[t/x]]}{\Gamma \Rightarrow [\exists x : \tau.\, A]}\ \exists R \qquad \frac{\Gamma, \exists x : \tau.\, A, t : \tau, u : A[t/x] \Rightarrow C}{\Gamma, \exists x : \tau.\, A \Rightarrow C}\ \exists L$$

$$\frac{\Gamma, u : A \Rightarrow B}{\Gamma \Rightarrow A \supset B}\ \supset R \qquad \frac{\Gamma, A \supset B \Rightarrow [A] \quad \Gamma, A \supset B \,|\, [B] \Rightarrow C}{\Gamma \,|\, [A \supset B] \Rightarrow C}\ \supset L$$

The optimized rules are:

$$\frac{\Gamma, x : \tau \Rightarrow A}{\Gamma \Rightarrow \forall x : \tau.\, A}\ \forall R \qquad \frac{\Gamma \Rightarrow t : \tau \quad \Gamma \,|\, [A[t/x]] \Rightarrow C}{\Gamma \,|\, [\forall x : \tau.\, A] \Rightarrow C}\ \forall L$$

$$\frac{\Gamma \Rightarrow t : \tau \quad \Gamma \Rightarrow [A[t/x]]}{\Gamma \Rightarrow [\exists x : \tau.\, A]}\ \exists R \qquad \frac{\Gamma, \exists x : \tau.\, A, t : \tau, u : A[t/x] \Rightarrow C}{\Gamma, \exists x : \tau.\, A \Rightarrow C}\ \exists L$$

$$\frac{\Gamma, u : A \Rightarrow B}{\Gamma \Rightarrow A \supset B}\ \supset R \qquad \frac{\Gamma \Rightarrow [A] \quad \Gamma \,|\, [B] \Rightarrow C}{\Gamma \,|\, [A \supset B] \Rightarrow C}\ \supset L$$

3 Logic Programming

For the rest of the lecture, we remove $\exists R$ and $\exists L$. Since existential quantifier is the only positive connective in our grammar, we have essentially removed the entire syntax category A^+. Also, since we no longer have any positive rules, we may remove blur_L and focus_R as well. Then, we are left with the negative fragment of the logic.

For the following example:

$$\cdot \Rightarrow p \supset (p \supset q) \supset (q \supset r) \supset (p \supset r) \supset r$$

First, we apply the trivial $\supset R$ rules:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{?}{\Gamma = u_1 : p, u_2 : p \supset q, u_3 : q \supset r, u_4 : p \supset r \Rightarrow r}\ \supset R}{u_1 : p, u_2 : p \supset q, u_3 : q \supset r \Rightarrow (p \supset r) \supset r}\ \supset R}{u_1 : p, u_2 : p \supset q \Rightarrow (q \supset r) \supset (p \supset r) \supset r}\ \supset R}{u_1 : p \Rightarrow (p \supset q) \supset (q \supset r) \supset (p \supset r) \supset r}\ \supset R}{\cdot \Rightarrow p \supset (p \supset q) \supset (q \supset r) \supset (p \supset r) \supset r}\ \supset R$$

Let's pause and rewrite the sequent as $\Gamma \Rightarrow C$. Now, think about what are the possible rules to apply. Clearly, the only rule that gets us foward is focus_L^u, but which u should we choose?

Let's first try $u_1 : p$, then we end up with the derivation:

$$\dfrac{\dfrac{}{\Gamma \,|\, [p] \Rightarrow C}\ ?}{\Gamma \Rightarrow C}\ \mathsf{focus}_L^{u_1}$$

The only choice for ? seems to be init^-:

$$\dfrac{\dfrac{}{\Gamma \,|\, [p] \Rightarrow p^-}\ \mathsf{init}^-}{\Gamma \Rightarrow C}\ \mathsf{focus}_L^{u_1}$$

But then, this means that our goal C must be p, which is not the case. We may also distill the "knowledge" we just learned from this trial into a *synthetic inference rule*:

$$\frac{C = p^-}{\Gamma \Rightarrow C}$$

Similarly, if we choose to focus on u_3, we get:

$$\dfrac{\dfrac{\dfrac{\Gamma \Rightarrow q}{\Gamma \Rightarrow [q]}\,\mathsf{blur}_R \qquad \dfrac{}{\Gamma \mid [r] \Rightarrow r}\,\mathsf{init}^-}{\Gamma \mid [q \supset r] \Rightarrow C}\,{\supset}L}{\Gamma \Rightarrow C}\,\mathsf{focus}_L^{u_3}$$

and the synthetic inference rule:

$$\frac{\Gamma \Rightarrow q}{\Gamma \Rightarrow r}$$

This looks plausible, as our goal is indeed r! Continue with the proof, the full derivation turns out to be:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \mid [p] \Rightarrow p}\,\mathsf{init}^-}{\Gamma \Rightarrow p}\,\mathsf{focus}_L^{u_1}}{\Gamma \Rightarrow [p]}\,\mathsf{blur}_R \qquad \dfrac{}{\Gamma \mid [q] \Rightarrow q}\,\mathsf{init}^-}{\Gamma \mid [p \supset q] \Rightarrow q}\,{\supset}L}{\Gamma \Rightarrow q}\,\mathsf{focus}_L^{u_2}}{\Gamma \Rightarrow [q]}\,\mathsf{blur}_R \qquad \dfrac{}{\Gamma \mid [r] \Rightarrow r}\,\mathsf{init}^-}{\Gamma \mid [q \supset r] \Rightarrow C}\,{\supset}L}{\Gamma \Rightarrow C}\,\mathsf{focus}_L^{u_3}$$

Intuitively, if we focus on a formula of form $p \supset \cdots \supset s$ for proving $\Gamma \Rightarrow C$, the goal C must be exactly s.

4 Logic Programming in Twelf

Twelf allows us to define custom inductive types. Here's a good example of how logic programming languages like Twelf generalize the above recipe to work with them.

```
nat : type.
z : nat.
s : nat -> nat.

lt : nat -> nat -> type.
lt/z : lt z (s N).
lt/s : lt (s M) (s N)
```

```
        <- lt M N.

%solve two-lt-four : lt (s (s z)) (s (s (s (s z)))).
```

The `two-lt-four` in the above Twelf program can be translated to

$$\Gamma \Rightarrow a < b$$

where $a = \mathsf{suc}\,\mathsf{suc}\,\mathsf{zero}$ and $b = \mathsf{suc}\,\mathsf{suc}\,\mathsf{suc}\,\mathsf{suc}\,\mathsf{zero}$, and our context Γ contains the equivalent definitions of `lt`:

$$\begin{aligned} &\texttt{lt/z} : \forall n : \mathsf{nat}\,.\, zero < \mathsf{suc}\, n \\ &\texttt{lt/s} : \forall n : \mathsf{nat}\,.\, \forall m : \mathsf{nat}\,.\, (n < m) \supset (\mathsf{suc}\, n < \mathsf{suc}\, m) \end{aligned}$$

As usual, let's first think about the possible derivation of $\Gamma \Rightarrow C$. If we choose to focus on the `lt/s` rule, we get the derivation:

$$\dfrac{\Gamma \Rightarrow a : \mathsf{nat} \qquad \dfrac{\Gamma \Rightarrow b : \mathsf{nat} \qquad \dfrac{\dfrac{\Gamma \Rightarrow a < b}{\Gamma \Rightarrow [a < b]}\,\mathsf{blur}_R \qquad \Gamma \,|\, [\mathsf{suc}\, a < \mathsf{suc}\, b] \Rightarrow C}{\Gamma \,|\, [(a < b) \supset (\mathsf{suc}\, a < \mathsf{suc}\, b)] \Rightarrow C}\,\supset L}{\Gamma \,|\, [\forall m : \mathsf{nat}\,.\,(a < m) \supset (\mathsf{suc}\, a < \mathsf{suc}\, m)] \Rightarrow C}\,\forall L}{\dfrac{\Gamma \,|\, [\forall n : \mathsf{nat}\,.\,\forall m : \mathsf{nat}\,.\,(n < m) \supset (\mathsf{suc}\, n < \mathsf{suc}\, m)] \Rightarrow C}{\Gamma \Rightarrow C}\,\mathsf{focus}_L}\,\forall L$$