Lecture Notes on
A Concurrent Logical Framework

15-816: Linear Logic
Frank Pfenning

Lecture 21
April 9, 2012

Today we will put many of pieces together that we have put in place so far, by moving from a *logic* to a *type theory*. This means being explicit about proof terms. The end goal is to develop the concurrent logical framework CLF [WCPW02, CPWW02].

1 Propositions

We start with the language of propositions, which are now identified entirely with types. It is based upon the negative and positive proposition from the last lecture.

$$\begin{array}{llcl}
\text{Async (neg)} & A & ::= & P^- \mid S \multimap A \mid A_1 \mathbin{\&} A_2 \mid \Pi u{:}A_1.\, A_2 \mid \{S\} \\
 & & \mid & \top \qquad \text{(omitted from CLF)} \\
\text{Sync (pos)} & S & ::= & S_1 \otimes S_2 \mid \mathbf{1} \mid \exists u{:}A.\, S \mid A \mid {!A} \mid @A \\
 & & \mid & P^+ \mid S_1 \oplus S_2 \mid \mathbf{0} \qquad \text{(omitted from CLF)}
\end{array}$$

A few notes on this grammar: we eliminated $\top$ and $\mathbf{0}$, but added affine resources, marked with $@A$ to take their place. We will explain later why these are not present. We also eliminated positive atoms, which is mostly a historical omission. However, it turns out in our examples it is easily possible to get by without positive atoms, at the cost of additional focusing phases.

The biggest change is to replace the type τ in the quantifiers with negative proposition, now acting as a type of proof terms. To build the type theory we now need to develop a notion of proof.

2 Negative Proof Terms

In order to be able to instantiate the quantifiers, and also to understand the structure of proofs as objects of study, we now need to assign proof terms to the types we have introduced above. For example, in our encoding of substructural operational semantics so far we can examine the state during the computation, but we do not have an object describing the computation itself. That will be role of the proof terms.

We start with backward chaining. The pure backward chaining fragment of the logic is obtained by removing the lax monad $\{S\}$ and leaving everything else intact. Sequents during the inversion phase of backward chaining will have the form

$$\Delta \to N : A$$

where N are the normal terms that arise from the search. We call them normal, because they do not contain a cut (or a redex, its natural deduction analog).

We start with linear implication, which may at first seem straightforward, but which we will have to revisit later. In the presentation of the rules, we omit persistent and affine resources, and concentrate on the linear ones.

$$\frac{\Delta, x{:}S \to N : A}{\Delta \to \lambda x.\, N : S \multimap A} \multimap R? \qquad \frac{\Delta \to N_1 : A_1 \quad \Delta \to N_2 : A_2}{\Delta \to \langle N_1, N_2 \rangle : A_1 \,\&\, A_2} \,\&R$$

How do we treat $\Pi u{:}A_1.\, A_2$? It will work very similar to the universal quantifier, except that the new parameter does not come form a separate domain, but is just a typing assumption. This hypothesis is persistent, however.

$$\frac{\Gamma, u{:}A_1 \;;\; \Xi \;;\; \Delta \to N : A_2}{\Gamma \;;\; \Xi \;;\; \Delta \to \lambda u{:}A_1.\, N : \Pi u{:}A_1.\, A_2} \,\Pi R$$

Note that both occurrences of u in the conclusion of this rule are bound variables. We call them u in order to simplify the presentation of the rule.

At a negative atom, we have to focus on some resource in Γ, Ξ, or Δ, assuming for now the sequent is stable.

$$\frac{\Gamma \;;\; \Xi \;;\; \Delta, x{:}[A] \to R : P^-}{\Gamma \;;\; \Xi \;;\; \Delta, x{:}A \to R : P^-} \,\mathsf{foc}L \qquad \frac{\Gamma \;;\; \Xi \;;\; \Delta, a{:}[A] \to R : P^-}{\Gamma \;;\; \Xi, a{:}A \;;\; \Delta \to R : P^-} \,\mathsf{foc}L@$$

$$\frac{u{:}A \in \Gamma \quad \Gamma \;;\; \Xi \;;\; \Delta, u{:}[A] \to R : P^-}{\Gamma \;;\; \Xi \;;\; \Delta \to R : P^-} \,\mathsf{foc}L!$$

So one possibility for a normal term N is a new kind of term R, which represents proof constructed during focusing. We call these atomic terms.

The base case for left focusing is that we reach a negative atom. In this case we just return the proof term constructed for the formula in focus as the proof term R.

$$\frac{}{R{:}[P^-] \to R : P^-}\ \mathsf{id}^-$$

The next situation to consider is that we are focused on a linear implication. We assume we already have constructed an $R{:}[S \multimap A]$.

$$\frac{\Delta, R\,M{:}[A] \to R' : C \quad \Delta' \to M : [S]}{\Delta, \Delta', R{:}[S \multimap A] \to R' : C}\ {\multimap}L$$

We see the evidence for the truth of A is just an application, which is therefore a specific kind of atomic term. Operationally, we have to search for the proof of the first premise first. We cannot yet construct its justification, because we have not yet solved the subgoal S. However, when we succeed in the end we will be able to fill in the whole term.

Next we consider the universal quantifier, which is part of the backward chaining semantics. When focused on that, we will fill in a variable and let unification during proof search determine its value. At the present level of abstraction in the description of the system, we just write a premise that "guesses" a term of the right type. Because quantification is over persistent terms only, this term must be closed with respect to the linear (and affine) variables.

$$\frac{\cdot \to N : A_1 \quad \Delta, R\,N{:}[A_2\{N/u\}] \to R' : C}{\Delta, R{:}[\Pi u{:}A_1.\, A_2] \to R' : C}\ \Pi L$$

Since the variable u can appear in A_2, we have to substitute the term N for u in the type of $R\,N$. This is the essence of a *dependent function type*.

To summarize, at this point we have judgments

$$\begin{array}{l} \Delta \to N : A \\ \Delta, R{:}[A] \to R' : P^- \\ \Delta \to M : [S] \end{array}$$

with terms

$$\begin{array}{rrcl} \text{Normal terms} & N & ::= & \lambda x.\, N \mid \lambda u.\, N \mid \langle N_1, N_2 \rangle \mid R \\ \text{Atomic terms} & R & ::= & x \mid a \mid u \mid R\,N \mid \pi_1\,R \mid \pi_2 R \\ \text{Positive terms} & M & ::= & \ldots \end{array}$$

where we have elided the obvious left rules $\& L_1$ and $\& L_2$ which requires atomic terms $\pi_1\, R$ and $\pi_2\, R$.

The terms for the positive constructors will be next.

3 Positive Proof Terms

The judgment that arose so far was just S in right focus. We start with tensor and unit.

$$\frac{\Delta_1 \rightarrow M_1 : [S_1] \quad \Delta_2 \rightarrow M_2 : [S_2]}{\Delta_1, \Delta_2 \rightarrow M_1 \otimes M_2 : [S_1 \otimes S_2]}\ \otimes R \qquad\qquad \frac{}{\cdot \vdash \mathbf{1} : [\mathbf{1}]}\ \mathbf{1}R$$

For an existential in right focus, the considerations are similar for a universal in left focus. In the rules we guess the term; in the operational semantics we use a variable together with unification instead to determine it.

$$\frac{\cdot \rightarrow N : A \quad \Delta \rightarrow M : [S\{N/u\}]}{\Delta \rightarrow N \otimes M : [\exists u{:}A.\, S]}\ \exists R$$

Finally, for the inclusion of negative propositions in positive ones, we just have a corresponding inclusion of negative terms in positive ones. We lose focus in either case.

$$\frac{\Gamma\, ;\, \Xi\, ;\, \Delta \vdash N : A}{\Gamma\, ;\, \Xi\, ;\, \Delta \vdash N : [A]}\ \mathsf{blur}R \quad \frac{\Gamma\, ;\, \Xi\, ;\, \cdot \rightarrow N : A}{\Gamma\, ;\, \Xi\, ;\, \cdot \rightarrow @N : [@A]}\ @R \quad \frac{\Gamma\, ;\, \cdot\, ;\, \cdot \vdash N : A}{\Gamma\, ;\, \cdot\, ;\, \cdot \vdash !N : [!A]}\ !R$$

We made the persistent and affine assumptions explicit here, since they play a significant role in these rules.

$$\text{Positive terms} \quad M \quad ::= \quad M_1 \otimes M_2 \mid \mathbf{1} \mid N \otimes M \mid N \mid @N \mid !N$$

But what happened to the (invertible) left rules for the positive propositions S? Going back to our very first rule

$$\frac{\Delta, x{:}S \rightarrow N : A}{\Delta \rightarrow \lambda x.\, N : S \multimap A}\ \multimap R?$$

we see that we have no rules to decompose $x{:}S$. We also see that the *left* rule $\multimap L$ will apply the functional term to a positive term. So instead of simply abstracting over a variable we abstract over a *pattern* that matches the structure of the positive term.

$$\frac{\Delta, p{:}S \to N : A}{\Delta \to \lambda p.\, N : S \multimap A}\ {\multimap}R$$

Again, the mode of this judgment is somewhat strange, because the pattern p is generated by the demposition of S.

The various forms for this pattern are now straightforward, since they match the terms. When we hit a negative proposition A or an affine or persistent modality, we obtain a variable (in the linear, affine, or persistent context). The first one would have identical premise and conclusion, in our formulation. It could also be seen as going from ${\downarrow}x{:}{\downarrow}A$ to $x{:}A$.

$$\frac{\Gamma \,;\, \Xi, a{:}A \,;\, \Delta \to N : A'}{\Gamma \,;\, \Xi \,;\, \Delta, @a{:}@A \to N : A'} \qquad \frac{\Gamma, u{:}A \,;\, \Xi \,;\, \Delta \to N : A'}{\Gamma \,;\, \Xi \,;\, \Delta, !u{:}!A \to N : A'}$$

The others are even simpler:

$$\frac{\Delta, p_1{:}S_1, p_2{:}S_2 \to N : A'}{\Delta, p_1 \otimes p_2{:}S_1 \otimes S_2 \to N : A'}\ {\otimes}L \qquad \frac{\Delta \to N : A'}{\Delta, \mathbf{1}{:}\mathbf{1} \to N : A'}\ \mathbf{1}L$$

$$\frac{\Gamma, y{:}A \,;\, \Xi \,;\, \Delta, p{:}S \to N : A'}{\Gamma \,;\, \Xi \,;\, \Delta, x \otimes p{:}\exists u{:}A.\, S \to N : A'}\ \exists L$$

At this point our language is complete for the backward-chaining fragment, that is, the fragment that excludes the monad.

$$\begin{array}{rrcl}
\text{Async (neg)} & A & ::= & P^- \mid S \multimap A \mid A_1 \,\&\, A_2 \mid \Pi u{:}A_1.\, A_2 \mid \ldots \\
\text{Sync (pos)} & S & ::= & S_1 \otimes S_2 \mid \mathbf{1} \mid \exists u{:}A.\, S \mid A \mid !A \mid @A \\
\\
\text{Normal terms} & N & ::= & \lambda p.\, N \mid \lambda x.\, N \mid \langle N_1, N_2 \rangle \mid R \\
\text{Atomic terms} & R & ::= & x \mid a \mid u \mid R\,M \mid R\,N \mid \pi_1\,R \mid \pi_2 R \\
\text{Positive terms} & M & ::= & M_1 \otimes M_2 \mid \mathbf{1} \mid N \otimes M \mid N \mid @N \mid !N \\
\text{Positive patterns} & p & ::= & p_1 \otimes p_2 \mid \mathbf{1} \mid x \otimes p \mid x \mid @a \mid !u
\end{array}$$

We have listed to form of abstraction to match the two forms of application. One is over a pattern and application to a positive term (for $S \multimap A$), the other over a persistent variable and appliation to a negative term (for $\Pi u{:}A_1.\, A_2$).

4 Monadic Expressions

If we want to capture all of CLF, we now have to add the monad. It arises if we stumble upon $\{S\}$ during inversion. Write E for terms for $S\ lax$ and write $E \div S\ lax$.

$$\frac{\Delta \vdash E \div S\ lax}{\Delta \vdash \{E\} : \{S\}}\ \{\,\}R$$

First the judgmental rule which is not cut (which is excluded from this system since it does not arise in logic programming). It tells use that a positive term is a possible monad epxression.

$$\frac{\Delta \vdash M : [S]}{\Delta \vdash M \div S\ lax}\ \mathsf{lax}$$

Now we have only one more rule: what happens when we are focused on $\{S\}$ on the left. Recall that we lose focus in this transition.

$$\frac{\Delta, p{:}S \vdash E \div S'\ lax}{\Delta, R{:}[\{S\}] \vdash \mathsf{let}\ p = R\ \mathsf{in}\ E \div S'\ lax}\ \{\,\}L$$

To summarize once more the now complete list:

$$\begin{array}{rrcl}
\text{Async (neg)} & A & ::= & P^- \mid S \multimap A \mid A_1 \mathbin{\&} A_2 \mid \Pi u{:}A_1.\, A_2 \mid \{S\} \\
\text{Sync (pos)} & S & ::= & S_1 \otimes S_2 \mid \mathbf{1} \mid \exists u{:}A.\, S \mid A \mid {!}A \mid @A \\
\\
\text{Normal terms} & N & ::= & \lambda p.\, N \mid \lambda x.\, N \mid \langle N_1, N_2 \rangle \mid R \mid \{E\} \\
\text{Atomic terms} & R & ::= & x \mid a \mid u \mid R\, M \mid R\, N \mid \pi_1\, R \mid \pi_2 R \\
\text{Positive terms} & M & ::= & M_1 \otimes M_2 \mid \mathbf{1} \mid N \otimes M \mid N \mid @N \mid {!}N \\
\text{Positive patterns} & p & ::= & p_1 \otimes p_2 \mid \mathbf{1} \mid x \otimes p \mid x \mid @a \mid {!}u \\
\text{Monadic expressions} & E & ::= & M \mid \mathsf{let}\ \ p = R\ \mathsf{in}\ E
\end{array}$$

References

[CPWW02] Iliano Cervesato, Frank Pfenning, David Walker, and Kevin Watkins. A concurrent logical framework II: Examples and applications. Technical Report CMU-CS-02-102, Department of Computer Science, Carnegie Mellon University, 2002. Revised May 2003.

[WCPW02] Kevin Watkins, Iliano Cervesato, Frank Pfenning, and David Walker. A concurrent logical framework I: Judgments and properties. Technical Report CMU-CS-02-101, Department of Computer Science, Carnegie Mellon University, 2002. Revised May 2003.