Abstract. These notes are meant as complements to the paper [5] which served as backbone for most of the material covered in my lectures delivered at the Oregon Programming Languages Summer School 2011 “Types, Semantics and Verification” (videos should be soon available at http://www.cs.uoregon.edu/Activities/summerschool/summer11/). In particular, we give a brief account of Munch-Maccagnoni’s polarised classical realisability [17].

1 Plan of the lectures

1.1 Lectures 1–3

1. Recapitulate Frank’s course, and set up the objectives of this course: emphasis on sequent calculus as a “typing system for abstract machines”, and on classical logic, because sequent calculus is more natural in a classical setting, and of course also because there is a nice computational interpretation of classical logic.

2. Typing Krivine abstract machine: left implication introduction as “putting the argument on the stack”.

3. Introduce classical sequent calculus, discussing choices of reversible versus irreversible rules. Settle on a choice of presentation where both conjunction and disjunction are positive.

4. Introduce a syntax for these classical proofs (no focalisation yet!), using the syntactic kit of “system L” : µ, ˜µ, and three different kinds of judgements: v, e, c.

5. Discuss the slogan “commutative cut rules as explicit substitutions”.

6. Exercise. Implication can be encoded as

\[ \neg A \lor B \neg (A \land \neg B) \]

inducing respectively

\[ \langle \lambda x. M | N \cdot E \rangle \rightarrow \langle M[N/x] | E \rangle \]
\[ \langle \lambda x. M | N \cdot E \rangle \rightarrow \langle N | \tilde{\mu}x.(M | E) \rangle \]

7. Lafont’s critical pair.

8. Krivine abstract machine continued, with Felleisen’s C, and a call-by-value machine (see also [4]).

9. Reflection on the advantages of sequent calculus with respect to natural deduction when it comes to classical logic: no need to introduce new axioms, just relax the existing rules by allowing several formulas on the right (which amounts to allow contraction on the right).

10. Focalised proofs.

11. Counter-example: a non-focalised proof exhibiting, say \textit{inl}(\mu \alpha.c).

12. Syntax for focalised proofs.

13. Dynamics: critical pair resolved!

14. An example: a focused proof of (\vdash P \lor \neg P \vdash (P \lor \neg P) \vdash !)

15. Encoding call-by-value \xi-calculus in LKQ.

16. Meta-theoretical properties: confluence and strong normalisation hold for LKQ.

17. Completeness of the focused system can be proved by translating non-focused proofs into focused proofs with cuts and then eliminating them (see also exercise in lecture 5).

1.2 Lecture 4

1. For intuitionistic minds: what is the meaning of µ and ˜µ? Anticipating CPS (lecture 5):

(a) read \( e : (\Gamma \vdash \ldots, \alpha : P, \ldots) \) as \( \Gamma, \ldots, k : P \supset R, \ldots \vdash e : R \);

(b) read \( \Gamma \vdash e : P \supset \ldots, \alpha : P, \ldots \) as \( \Gamma, \ldots, k : P \supset R, \ldots \vdash e : P \supset R \). In particular, \( \tilde{\mu}x.e \) reads as \( \lambda x.e \);

(c) finally, considering (a), one is naturally led to read \( \mu \alpha.e \) as \( \Gamma, \ldots, k : P \supset R, \ldots \vdash \lambda k.e : (P \supset R) \supset R \).

2. Dual system LKT: encoding of call-by-name \xi-calculus (section 2 of these notes).
3. For intuitionistic minds again: by opening the space to the classical world, we could synthesise call-by-name and call-by-value from a *unique*, more basic system (classical sequent calculus based on conjunction, disjunction and negation).

4. Discussion on positive / negative and left / right (anticipating item 4 of lecture 5). We have respectively

\[ LKQ \quad LKT \quad \text{monilateral} \]

\[ \ldots, P, \ldots \vdash \ldots, Q, \ldots, \ldots, N, \ldots \vdash \ldots, N, \ldots \vdash \ldots, N, \ldots, P, \ldots \]

In the first case we do have terms and contexts, but only for CBV languages / positive connectives. In the second case, we also have terms and contexts, but only for CBN languages / negative connectives. In the third case, we can mix CBN and CBV, but we have lost the division into contexts and terms, i.e., we cannot distinguish a context of negative type from a term of positive type.

5. Monolateral presentation of focalising system L (section 3 of these notes).

6. Classical polarised realisability (section 4 of these notes).

1.3 Lecture 5

1. Classical polarised realisability (continued).
2. Illustrating classical polarised realisability.
3. **Exercise.** Revisit the proof of completeness of focalisation given by Frank (Theorem 1 of [18]) to unveil its underlying reducibility flavour (there is an orthogonal implicit in case (ii) of this statement).
4. “mixed call by name and call-by-value”, i.e., restoring bilaterality: negative on the left, negative on the right, etc...
   Illustration: Frank’s focused intuitionistic calculus with a positive disjunction, a negative implication, and both a positive and a negative conjunction (section 5 of these notes).
5. Synthetic connectives (cf. “big-step derived rules” in [18]).
6. Translation of focalised system L to intuitionistic logic: almost textual! The double negation is only introduced at the root of terms \( v \).
7. Reflection: focalisation as “CPS in direct style”. All the work on encoding the order of evaluation has been done in the process of translating your favourite functional programming language (possibly with control operators) in focalised sequent calculus. This provides a factoring of CPS, where the second step, from focalised sequent calculus to intuitionistic logic, hurts more than anything else, blurring distinctions such as ordinary variables and continuation variables. Why not taking focalising system L as the target of CPS?
8. A short primer in linear logic (LL).
9. Perspective: LL versus focalised classical logic (section 7 of these notes).

2 System LKT

**Note.** that in the lectures, for the syntax of LKQ, we have used \( e^\downarrow \) and \( \bar{\mu}\alpha^\downarrow.c \) in place of the notation orginally used in [5]. We also gave up on marking explicitly the coercion from values \( V \) to terms \( V^\circ \).

The rationale for the different syntaxes that follow is that structured \( \mu \) and \( \bar{\mu} \) are for “absolute” negative introductions, and that more precisely \( \bar{\mu} \) (resp. \( \mu \)) is for introduction of a positive on the left (resp. negative on the right).

System LKT is defined by duality from LKQ:

\[
N ::= \overline{X} | N \not\exists N | N \& N | \neg N
\]

**Commands**

\[
e ::= \langle v | e \rangle | c[\sigma]
\]

**Expressions**

\[
v ::= x | \mu x.c | \mu x^\uparrow.c | \mu[\alpha_1, \alpha_2].c | \mu(\alpha_1[fst].c_1, \alpha_2[snd].c_2) | v[\sigma]
\]

**Contexts**

\[
e ::= E | \bar{\mu}x.c | e[\sigma]
\]

**Covalues**

\[
E ::= x | [E, E] | E[fst] | E[snd] | v^\dagger | E[\sigma]
\]
The operational semantics is left to you (mirror image of LKQ).

3 Monolateral presentation of focalising system L

The following monolateral presentation is yet another isomorphic copy of LKQ, where the sequents are folded on the right, the positive formulas on the left becoming negative. This change of point of view leads us to recognise that $\neg P$ decomposes as $\bar{P}$, where $\bar{P}$ is the involutive de Morgan duality, and where the $\uparrow$ operator, together with its dual $\downarrow$, is an explicit “modality” governing the changes of phase in the focalised proofs. (There are connections with Paul-André Melliès’ tensorial logic, yet to be explored.)

$$P ::= X \mid P \otimes Q \mid P \oplus Q \mid \downarrow N$$
$$N ::= \overline{N} \mid N \& N \mid \uparrow P$$

Commands $c ::= (t^+ \mid t^-) \mid c[\sigma]$

Positive terms $t^+ ::= T \mid \mu x.c \mid t^+[\sigma]$

Values $T ::= x \mid (T, T) \mid \text{inl}(T) \mid \text{inr}(T) \mid (t^-)^\downarrow \mid T[\sigma]$

Negative terms $t^- ::= \alpha \mid \mu x.c \mid \mu \alpha^+.c \mid \mu[x_1,x_2].c \mid \mu(x_1[\text{fst}],c_1,x_2[\text{snd}],c_2) \mid t^-[\sigma]$

$$\vdash t_1^+ : P \mid \Delta \quad \vdash t_2 : \bar{P} \mid \Delta$$

$$\vdash \alpha : N \mid \Delta \quad \vdash \alpha : N \mid \Delta$$

$$\vdash \mu x.c : N \mid \Delta \quad \vdash \mu \alpha.c : P \mid \Delta$$

$$\vdash \mu \alpha^+.c : P \mid \Delta$$

$$\vdash t : N \mid \Delta$$

$$\vdash T_1 : P_1 \mid \Delta \quad \vdash T_2 : P_2 \mid \Delta$$

$$\vdash \mu x_1.x_2 : N \mid \Delta$$

$$\vdash \mu[x_1,x_2].c : N \& N \mid \Delta$$

$$\vdash \mu(x_1[\text{fst}],c_1,x_2[\text{snd}],c_2) : N \mid \Delta$$

$$\vdash \mu \alpha^+.c : P \mid \Delta$$

$$\vdash \mu \alpha^+.c : P \mid \Delta$$

$\vdash V : P \mid \Delta$ ... $\vdash e : Q \mid \Delta$ ... $\vdash c[\ldots, q : P, \ldots] : \Delta, \ldots, \alpha : Q, \ldots$

$$\vdash c[\ldots, V/q, \ldots, e/\alpha] : (\neg \Delta)$$

$\text{(idem } v[\sigma], V[\sigma], e[\sigma])$
we say that \( t \perp \subseteq \{ \}
One chooses a fixed set and a fortiori no closed terms of closed types.)

(Note that in the absence of base inhabited type constants or of second-order quantification, there are no closed types and a fortiori no closed terms of closed types.)

One chooses a fixed set \( \perp \subseteq \{ c \} \) of commands, closed under backward reduction: if \( c \in \perp \) and \( c' \rightarrow c \), then \( c' \in \perp \).

we say that \( t^+ \) is orthogonal to \( t^- \) (and may use the notation \( t^+ \perp t^- \) for this) when \( \langle t^+ | t^- \rangle \in \perp \). One then defines:

for any \( X \subseteq \{ t^+ \} \), \( X^\perp = \{ t^- | \forall t^+ \in X \langle t^+ | t^- \rangle \in \perp \} \)

for any \( Y \subseteq \{ t^- \} \), \( Y^\perp = \{ t^+ | \forall t^- \in X \langle t^+ | t^- \rangle \in \perp \} \)

We have the usual properties. Let \( Z \subseteq \{ t^+ \} \) or \( Z \subseteq \{ t^- \} \):

\[ Z \subseteq Z^\perp \perp \quad Z^\perp = Z^\perp \perp \]

(the latter following from antimonotonicity of \( Z \mapsto Z^\perp \)). We say that \( Z \subseteq \{ t^+ \} \) (resp. \( Z \subseteq \{ t^- \} \)) is a positive (resp. negative) behaviour if \( Z = Z^\perp \perp \) (this terminology comes from [10]). Note that any \( Z^\perp \perp \) is a behaviour.

We then associate with each positive type \( P \) a set (not a behaviour) \( \mathcal{V}[P] \subseteq \{ T \} \) (parameterised by an environment), as follows, by induction on the syntax of positive formulas:

\[
\mathcal{V}[X] \rho = \rho(X) \\
\mathcal{V}[P \otimes Q] \rho = \{(V_1, V_2) | V_1 \in \mathcal{V}[P] \rho \text{ and } V_2 \in \mathcal{V}[Q] \rho\} \\
\mathcal{V}[P \oplus Q] \rho = \{ \text{inl}(V_1) | V_1 \in \mathcal{V}[P] \rho \} \cup \{ \text{inr}(V_2) | V_2 \in \mathcal{V}[Q] \rho\} \\
\mathcal{V}[\perp] \rho = \{ (t^-)^\perp | t^- \in [N] \rho \}
\]

One then sets:

\[ \mathcal{V}[P] = [P]^{\perp \perp} \quad [N] = [N]^\perp \]

In other words, the \( \mathcal{V}[P] \)'s determine the interpretations \( \mathcal{V}[P] \subseteq \{ t^+ \} \) and \( [N] \subseteq \{ t^- \} \), for all positive and negative formulas. (If you are worried by the occurrence of \([N]\) in the definition of \( \mathcal{V}[\perp] \), replace it with \( \mathcal{V}[\perp] \).) We have:

\[ [P] = [P]^{\perp \perp} \text{ and } [N] = [N]^{\perp \perp} \text{ i.e., } [P] \text{ and } [N] \text{ are behaviours} \]

\[ [P] \rho = \mathcal{V}[N] \rho \]

(We omit \( \rho \) in what follows for keeping light notations. Also, think in this section in terms of implicit substitutions)

Fundamental lemma: Let

\[ \Delta = \ldots, x : N, \ldots, \alpha : P, \ldots \quad \ldots, T \in \mathcal{V}[N], \ldots, t^- \in [P], \ldots \]

Then:

\[ c : \{ \vdash \Delta \} \quad \vdash T' : P ; \Delta \quad \vdash t^+ : P ; \Delta \quad \vdash t^- : N | \Delta \]

\[ \begin{align*}
\{ c[\ldots, T/x, \ldots, t^-/\alpha, \ldots] \in \perp \} & \quad \{ T'[\ldots, T/x, \ldots, t^-/\alpha, \ldots] \in \mathcal{V}[P] \} \\
\{ t^+[\ldots, T/x, \ldots, t^-/\alpha, \ldots] \in [P] \} & \quad \{ t^-[\ldots, T/x, \ldots, t^-/\alpha, \ldots] \in [N] \}
\end{align*} \]
The proof is elementary, by induction on terms. Note that, say, for the case \( \mu[x, y]c \), we need to use that \( \bot \) is closed under backward reduction.

Illustration: If \( \Gamma \vdash t^+ : P \oplus Q \) and \( \alpha \) is fresh, then either \( \langle t^+ \mid \alpha \rangle \rightarrow^{*} \langle \text{inl}(T_1) \mid \alpha \rangle \) for some \( T_1 \) or \( \langle t^+ \mid \alpha \rangle \rightarrow^{*} \langle \text{inr}(T_2) \mid \alpha \rangle \) for some \( T_2 \).

Proof: Take \( \bot = \{ t^+ \mid \exists T_1 \langle t^+ \mid \alpha \rangle \rightarrow^{*} \langle \text{inl}(T_1) \mid \alpha \rangle \lor \exists T_2 \langle t^+ \mid \alpha \rangle \rightarrow^{*} \langle \text{inr}(T_2) \mid \alpha \rangle \} \). We have to check that \( \langle t^+ \mid \alpha \rangle \in \bot \). But by the fundamental lemma we have \( t^+ \in \mathcal{V}[P \oplus Q]^{\bot} \). We then conclude noticing that \( \alpha \in \mathcal{V}[P \oplus Q]^{\bot} \).

5 A language for a polarised intuitionistic system

Here is a term assignment for the polarised focusing logic presented in Frank’s last lecture at this school [18]. Because intuitionistic logic has a positive disjunction and a negative conjunction, we are lead to bilateral sequents that have both positive and negative formulas on the left and on the right. More precisely, Frank has three kinds of judgements (which become five in system L style).

<table>
<thead>
<tr>
<th>Frank’s notation</th>
<th>Focalising system L notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Gamma^+ \rightarrow B^- )</td>
<td>( c : (\Gamma^+ \vdash B^-) ) ( \Gamma^+ \vdash v : B^- \mid \Gamma^+ \vdash e : A^+ \vdash B^- )</td>
</tr>
<tr>
<td>( \Gamma^+ \rightarrow [B^+] )</td>
<td>( \Gamma^+ \vdash V : B^+ \mid \Gamma^+ \vdash E : A^- \vdash B^- )</td>
</tr>
<tr>
<td>( \Gamma^+ ; [A^-] \rightarrow B^- )</td>
<td>( A^- := X^- \mid A^+ \lor A^- \mid A^- \land A^- \mid \uparrow A^+ )</td>
</tr>
<tr>
<td>( A^+ := A^+ \lor A^+ \mid A^+ \land A^+ \mid \downarrow A^- )</td>
<td></td>
</tr>
<tr>
<td>( c ::= \langle \mu \cdot e \rangle \mid { v, { } } )</td>
<td>( v ::= \lambda \cdot c \mid \mu([\text{fst}], c_1, [\text{snd}], c_2) \mid V^+ )</td>
</tr>
<tr>
<td>( V ::= x \mid (V, V) \mid \text{inl}(V) \mid \text{inr}(V) \mid e )</td>
<td>( e ::= \mu[\text{inl}(x_1).c_1, \text{inr}(x_2).c_2] \mid \mu(x, y).c \mid E^+ )</td>
</tr>
<tr>
<td>( E ::= [{ } \mid (V, E) \mid E[\text{fst}] \mid E[\text{snd}] \mid e^+ )</td>
<td>( E ::= [{ } \mid (V, E) \mid E[\text{fst}] \mid E[\text{snd}] \mid e^+ )</td>
</tr>
</tbody>
</table>

Negative rules

\[
\begin{array}{c}
c : (\Gamma^+, x : A^+ \vdash B^-) \\
\Gamma^+ \vdash \lambda x.c : A^+ \land B^- \\
c_1 : (\Gamma^+, x_1 : A^+ \vdash C^-) \\
c_2 : (\Gamma^+, x_2 : B^+ \vdash C^-) \\
\Gamma^+ \vdash \mu[\text{inl}(x_1).c_1, \text{inr}(x_2).c_2] : A^+ \lor B^+ \vdash C^- \\
\end{array}
\]

\[
\begin{array}{c}
c_1 : (\Gamma^+, C^-) \\
c_2 : (\Gamma^+, B^-) \\
c : (\Gamma^+, A^+ \land B^-) \\
\Gamma^+ \vdash \mu[\{ \} \mid \{ v, \{ \} \}] : A^- \land B^- \\
\end{array}
\]

Ending a negative proof-search phase:

\[
\begin{array}{c}
\Gamma^+ ; E : A^- \vdash C^- \\
\Gamma^+ \vdash V : C^+ \mid \Gamma^+ \vdash E^+ : \downarrow A^- \vdash C^- \\
\end{array}
\]

Positive rules

\[
\begin{array}{c}
\Gamma^+ \vdash V : A^+ \\
\Gamma^+ \vdash E : B^- \vdash C^- \\
\Gamma^+ ; [V, E] : A^+ \land B^- \vdash C^- \\
\Gamma^+ ; E : A^- \vdash C^- \\
\Gamma^+ ; E[\text{fst}] : A^- \land B^- \vdash C^- \\
\Gamma^+ ; E[\text{snd}] : A^+ \land B^- \vdash C^- \\
\end{array}
\]

\[
\begin{array}{c}
\text{idem } E[\text{snd}] \\
\Gamma^+ \vdash V_1 : A^+ \\
\Gamma^+ \vdash V_2 : B^- \\
\Gamma^+ \vdash (V_1, V_2) : A^+ \land B^- \\
\Gamma^+ \vdash V : A^+ \\
\Gamma^+ \vdash \text{inl}(V) : A^+ \lor B^+ \\
\end{array}
\]
Getting out of a positive phase:

\[
\begin{align*}
\Gamma^+ &\vdash v: A^- & \quad \Gamma^+ &\vdash e: A^+ \vdash C^- \\
\Gamma^+ &\vdash v^+: \downarrow A^- & \quad \Gamma^+ &\vdash e^+: \uparrow A^+ \vdash C^- \\
\end{align*}
\]

Deactivation (needed to chain the negative rules)

\[
\begin{align*}
\Gamma^+ &\vdash e: A^+ \vdash C^- & \quad \Gamma^+ &\vdash v: C^- \\
\langle x | e \rangle &\vdash (\Gamma^+, x^+: A^+ \vdash C^-) & \quad \langle v [\cdot] \rangle &\vdash (\Gamma^+ \vdash C^-) \\
\end{align*}
\]

**Exercise.** Unroll the definitions of \( \forall \lceil P \to Q \rceil \) and \( [P \to Q] \) (for the CBV encoding of implication). Revisit Amal Ahmed’s first lectures in this context, and exhibit the double orthogonal implicitly lying there.

## 6 Focussing system L in full bilateral form

\[
P ::= X | P \otimes Q \mid P \oplus Q \mid \neg N \mid \downarrow N \\
N ::= \overline{X} | N \& N \mid N \& \neg N \mid \neg P \mid \uparrow P \\
A ::= P \mid N
\]

In sequents, \( \Gamma \) stands for \( \ldots, x^+ : P, \ldots, x^- : N, \ldots \), and \( \Delta \) stands for \( \ldots, \alpha^+ : P, \ldots, \alpha^- : N, \ldots \).

- **Commands**: \( c ::= \langle v^+ | e^+ \rangle \mid \langle v^- | e^- \rangle \)
- **Expressions**: \( v^+ ::= V \mid \mu \alpha^+.c \mid e^- \)
- **Values**: \( v^- ::= \mu \alpha^- .c \mid \mu \alpha^+.c \mid \mu [\alpha_1^-, \alpha_2^-.]c \mid \mu \alpha_1^-[fst].c_1, \alpha_2^- [snd].c_2 \mid e^+ \to \)
- **Contexts**: \( e^- ::= E \mid \mu x^- .c \mid v^+ \)
- **Covvalues**: \( E ::= \alpha^- \mid [E, E] \mid E[fst] \mid E[snd] \mid e^+ \mid V^- \)

We can factorise a few rules using the following mergings:

\[
\begin{align*}
v &::= v^+ \mid v^- \\
\alpha &::= \alpha^+ \mid \alpha^- \\
e &::= e^+ \mid e^- \\
x &::= x^+ \mid x^- \\
\end{align*}
\]

\[
\begin{align*}
\Gamma, x^+: P \vdash x^+: P ; \Delta &\quad \Gamma \mid \alpha^+: P \vdash \alpha^+: P ; \Delta &\quad \Gamma; \alpha^- : N \vdash \alpha^- : N ; \Delta &\quad \Gamma; x^- : N \vdash x^- : N ; \Delta \\
\Gamma \vdash v : A ; \Delta &\quad \Gamma \vdash e : A \vdash \Delta \\
\langle v | e \rangle &\vdash (\Gamma \vdash \Delta) \\
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash \mu \alpha .c : A ; \Delta &\quad \Gamma \vdash \mu x .c : A \vdash \Delta \\
\Gamma \vdash V : P ; \Delta &\quad \Gamma \vdash E : N \vdash \Delta \\
\Gamma \vdash V : P \mid \Delta &\quad \Gamma \vdash E : N \mid \Delta \\
\Gamma \vdash V_1 : P_1 ; \Delta &\quad \Gamma \vdash V_2 : P_2 ; \Delta &\quad \Gamma \vdash V_1 : P_1 ; \Delta \\
\Gamma \vdash v^+: \downarrow N ; \Delta &\quad \Gamma \vdash (V_1, V_2) : P_1 \otimes P_2 ; \Delta &\quad \Gamma \vdash \mu \alpha_1^- .\alpha_2^-.c : N_1 \& N_2 \mid \Delta \\
\Gamma \vdash \mu [\alpha_1^-, \alpha_2^-].c : N_1 \& N_2 ; \Delta &\quad \Gamma \vdash \mu (\alpha_1^- [fst].c_1, \alpha_2^-[snd].c_2) : N_1 \& N_2 ; \Delta
\end{align*}
\]
\[
\begin{array}{c|c|c|c|c}
\Gamma & \vdash A \quad & \vdash v : A \quad & \vdash V : P \quad & \vdash E : N \\
\hline
\Gamma \vdash e^{-} : \neg A & \Gamma \vdash v^{-} : \neg A & \Gamma ; V^{-} : \neg P & \Gamma ; E^{-} : \neg N \\
\hline
\Gamma ; E_{1} : N_{1} & \Gamma ; E_{2} : N_{2} & \Gamma ; E_{1} : N_{1} & \\
\hline
\Gamma ; e : P \vdash & \\
\hline
c : (\Gamma, x^{-} : N \vdash) & c : (\Gamma, x_{1}^{+} : P_{1}, x_{2}^{+} : P_{2} \vdash) & c_{1} : (\Gamma, x_{1}^{+} : P_{1} \vdash) & c_{2} : (\Gamma, x_{2}^{+} : P_{2} \vdash) \\
\hline
\Gamma \vdash \mu x^{-} : \downarrow N & \Gamma \vdash \mu(x_{1}^{+}, x_{2}^{+}) : P_{1} \otimes P_{2} \vdash & \Gamma \vdash \mu([inl(x_{1}^{+}), c_{1}, inr(x_{2}^{+}), c_{2}] : P_{1} \oplus P_{2} \vdash \\
\end{array}
\]

 Operational semantics:
\[
\begin{align*}
\langle V, \mu x^{+}, c \rangle & \to c[V/x^{+}] \\
\langle \mu \alpha^{-}, c \mid E \rangle & \to c[E/\alpha^{-}] \\
\langle v^{-}, \mu x^{-}, c \rangle & \to c[v^{-}/x^{-}] \\
\langle \mu \alpha^{+}, c \mid e^{+} \rangle & \to c[e^{+}/\alpha^{+}] \\
\langle V_{1}, V_{2}, \mu(x_{1}^{+}, x_{2}^{+}), c \rangle & \to c[V_{1}/x_{1}^{+}, V_{2}/x_{2}^{+}] \\
\langle \mu[\alpha_{1}, \alpha_{2}, c \mid [E_{1}, E_{2}] \rangle & \to c[E_{1}/\alpha_{1}, E_{2}/\alpha_{2}] \\
\langle [inl(V_{1}), \mu[inl(x_{1}^{+}), c_{1}, inr(x_{2}^{+}), c_{2}] \rangle & \to c_{1}[V_{1}/x_{1}^{+}] \\
\langle \mu(\alpha_{1}, \mu[fst,c_{1}, \alpha_{2}, c_{2}], E_{1}, [fst]) \rangle & \to c_{1}[E_{1}/\alpha_{1}] \\
\langle v^{-}, \mu x^{-}, c \rangle & \to c[v^{-}/x^{-}] \\
\langle \mu \alpha^{+}, c \mid e^{+} \rangle & \to c[e^{+}/\alpha^{+}] \\
\langle e^{-}, \mu x^{-}, c \rangle & \to \langle e^{-}, c \rangle \\
\langle e^{+}, \mu x^{-}, c \rangle & \to \langle v^{+}, c \rangle
\end{align*}
\]

7 Linear logic

I briefly place focalised classical logic and linear logic in perspective, recalling the rules of LL here for your convenience.

\[
A ::= X \mid \Xi \mid A \otimes A \mid A \oplus A \mid A \otimes A \mid A \otimes A \mid A \otimes A
\]

**AXIOM**

\[
\vdash A, \Gamma_{1} \quad \vdash A^{\perp}, \Gamma_{2}
\]

**CUT**

\[
\vdash A, \Gamma_{1} \quad \vdash A^{\perp}, \Gamma_{2}
\]

**MULTIPLICATIVES**

\[
\vdash A, B, \Gamma \quad \vdash A, \Gamma_{1} \quad \vdash B, \Gamma_{2}
\]

\[
\vdash A \otimes B, \Gamma \quad \vdash A \otimes B, \Gamma_{1}, \Gamma_{2}
\]

**ADDITIVES**

\[
\vdash A, \Gamma \quad \vdash B, \Gamma \quad \vdash A, \Gamma \quad \vdash B, \Gamma
\]

\[
\vdash A \oplus B, \Gamma \quad \vdash A \oplus B, \Gamma \quad \vdash A \otimes B, \Gamma
\]

**EXPONENTIALS**

\[
\vdash \Gamma, A \quad \vdash ?\Gamma, A \quad \vdash \Gamma, A \quad \vdash ?A, \Gamma
\]

**Dereliction**

\[
\vdash \Gamma, ?A
\]

**Promotion**

\[
\vdash ?\Gamma, !A \quad \vdash ?\Gamma, !A \quad \vdash ?A, \Gamma
\]

Notice in particular that the axioms, the cut, and the tensor rules are significantly different from the rules we have given for LKQ. But we could of course have given them in this form, adding (unrestricted) weakening and contraction as explicit rules. The LL formulation of the tensor rule puts forward its irreversible nature.

Both systems share having two conjunctions and disjunctions, and having an involutive defined negation.
Linear logic is primarily (or at least originally) organised around the opposition additive / multiplicative (see table below), and around resource-sensitivity (and the analysis of the complexity of cut-elimination / reduction). The exponentials $\!, \, ?$ control the use of resources.

Focalising logic is primarily organised around the opposition reversible / irreversible (which is related to input / output, lazy / eager), and around “flow-sensitivity”: the shift operators $\uparrow$ and $\downarrow$ control the proof-search process and the direction of cut-elimination (or the CBV vs CBN strategies of reduction).

<table>
<thead>
<tr>
<th>irreversible</th>
<th>reversible</th>
</tr>
</thead>
<tbody>
<tr>
<td>multiplicative</td>
<td>$\otimes$</td>
</tr>
<tr>
<td>additive</td>
<td>$\oplus$</td>
</tr>
</tbody>
</table>

8 Further readings and background references on polarisation, Curry-Howard for classical logic, and abstract machines

For your convenience, these references are extracted from [5, 18] and supplemented by a few more. Feel free to contact me (curien@pps.jussieu.fr) for any questions, corrections, remarks!

References

8. M. Felleisen and D. Friedman, Control operators, the SECD machine, and the $\lambda$-calculus, in Formal Description of Programming Concepts III, 193-217, North Holland (1986).