# Dependent Type Theory in Polarised Sequent Calculus

Étienne Miquey[1], Xavier Montillet[2], and Guillaume Munch-Maccagnoni[2]

[1] CNRS, LSV, ÉNS Paris-Saclay, Inria, Cachan, France
[2] Inria, LS2N CNRS, Nantes, France

Thanks to several works on classical logic in proof theory, it is now well-established that continuation-passing style (CPS) translations in call by name and call by value correspond to different polarisations of formulae (Girard, 1991; Danos, Joinet, and Schellinx, 1997; Laurent, 2002). Extending this observation and building on Curien and Herbelin's abstract-machine-like calculi (2000), the last author proposed a term assignment for a polarised sequent calculus (where the polarities of formulae determine the evaluation order) in which various calculi from the literature can be obtained with *macros* responsible for the choices of polarities (Munch-Maccagnoni, 2013). It aims to explain several CPS translations from the literature by decompositing them through a *single* CPS for sequent calculus. It has later proved to be a fruitful setting to study the addition of effects and resource modalities (Curien, Fiore, and Munch-Maccagnoni, 2016), providing a categorical proof theory of *Call By Push Value* semantics (Levy, 2004).

We propose to bring together a dependently-typed theory (ECC) and polarised sequent calculus, by presenting a calculus $\mathbf{L_{dep}}$ suitable as a vehicle for compilation and representation of effectful computations. As a first step in that direction, we show that $\mathbf{L_{dep}}$ advantageously factorize a dependently typed continuation-passing style translation for **ECC**+`call/cc`. To avoid the inconsistency of type theory with control operators, we restrict their interaction. Nonetheless, in the pure case, we obtain an unrestricted translation from **ECC** to itself, thus opening the door to the definition of dependently typed compilation transformations.

**Overview of $\mathbf{L_{dep}}$** Recall that the key notion of term assignments for sequent calculi is that of a *command*, written $\langle t \parallel e\rangle$, which can be understood as a state of an abstract machine, representing the evaluation of an proof (or *expression*) $t$ against a counter-proof $e$ that we call *context*. Their typing judgements are of the form $\Gamma \vdash t : A \mid \Delta$ and $\Gamma \mid e : A \vdash \Delta$, which correspond respectively to underlying sequents $\Gamma \vdash A, \Delta$ and $\Gamma, A \vdash \Delta$, in which $A$ is in both cases the *principal formula* of the sequent. The command $\langle t \parallel e\rangle$ is the result of applying the cut rule with $t$ and $e$ as premises: $\langle t \parallel e\rangle : (\Gamma \vdash \Delta)$. It represents a cut rule with no principal formula.

But, in comparison to other presentations of sequent calculi, and like in Girard's original formulation of **LC**, our logic features a negation operator $\cdot^{\perp}$ which is involutive strictly: $A = A^{\perp\perp}$. This involution allows us to represent any sequent $c : (\Gamma \vdash \Delta)$ (resp. $\Gamma \vdash t : A \mid \Delta$) as a sequent $c : (\vdash \Gamma^{\perp}, \Delta)$ (resp. $\vdash \Gamma^{\perp}, \Delta \mid t : A$) with all formulae on the right. Thus, we are able to use a single grammar to describe both expressions and contexts.

The sequent calculus we propose is, in term of expressiveness, an extension of Luo's ECC. Namely, ECC contains dependent products $\Pi(x : A).B$ (becoming here a dependent $\parr$) and dependent sums $\Sigma(x : A).B$ (becoming here a dependent $\otimes$), a cumulative hierarchy of universes $\square_i$ and an impredicative propositional universe $\mathbb{P}$, the inductive type of booleans with dependent elimination $\mathbb{B}$, and equalities between terms $t = u$:

| | | | |
|---|---|---|---|
| **Atoms** | $C ::= x \mid \mathbb{B} \mid \mathbb{P} \mid \square_i \mid t = u$ | **Values** | $V ::= x \mid A \mid V \otimes_A V' \mid \texttt{true} \mid \texttt{false} \mid \texttt{refl}$ |
| **Types**$^+$ | $P ::= C \mid A \otimes x.B \mid \Downarrow A$ | | $\mid \mu(x \otimes_A y).c \mid \mu^{\ominus}x.c \mid \mu[c_1 \mid c_2] \mid \mu{=}c \mid \hat{\mu}c$ |
| **Types**$^\ominus$ | $N ::= C^{\perp} \mid A \parr x.B \mid \Uparrow A$ | **Terms** | $t ::= \mu^+x.c \mid \hat{} \mid V^{\diamond}$ |
| **Types** | $A ::= P \mid N$ | **Commands** | $c ::= \langle t \parallel V\rangle^+$ |

where the notations $\mu^+x.c/\mu^{\ominus}x.c$ distinguish the binder according to the polarity of the corresponding type.

Since sequent calculi allow us to manipulate classical logic, we need to restrict dependencies to avoid logical inconsistencies (Herbelin, 2005). Following previous works (Herbelin, 2012; Miquey, 2019), we only allow *negative-elimination-free* (NEF) terms within types, which are *thunkable* (value-like) terms. In fact, we relax this constraint into that of Girard's stoup (Girard, 1991), which similarly implies thunkability/linearity (Munch-Maccagnoni, 2013, IV.6). We take advantage of delimited control operators (in the form of $\hat{\mu}c$ and $\hat{\ }$) to separate regular and dependent typing modes:

$$\frac{\vdash \Gamma \mid t : P \qquad \vdash \Gamma \mid V : P^{\perp}}{\langle t \,\|\, V\rangle^{+} : (\vdash \Gamma)} \qquad\qquad \frac{\vdash \Gamma \mid t : P \qquad \vdash_{B[\bullet]} \Gamma \mid V : P^{\perp}}{\langle t \,\|\, V\rangle^{+} : (\vdash_{B[t]} \Gamma)}\; t \in \mathsf{NEF}$$

$$\frac{c : (\vdash \Gamma, x : N)}{\vdash \Gamma \mid \mu^{\ominus}x.c : N} \qquad \frac{c : (\vdash_{N} \Gamma)}{\vdash \Gamma \mid \hat{\mu}c : N} \qquad\qquad \frac{c : (\vdash_{B[x]} \Gamma, x : N)}{\vdash_{B[\bullet]} \Gamma \mid \mu^{\ominus}x.c : N} \qquad \frac{\bullet \notin B}{\vdash_{B} \Gamma \mid \hat{\ } : B^{\perp}}$$

**Regular mode** **Dependent mode**

Observe that in the latter, the turnstile is annotated with a return type whose dependencies evolve with the typing derivation (see Miquey 2019 for more details). For instance, considering the type:

$$T(b) = \mu^{+}x.\big\langle b \,\big\|\, \mu[\langle \mathbb{P} \,\|\, x\rangle^{+} \mid \langle \mathbb{B} \,\|\, x\rangle^{+}]\big\rangle^{+}$$

which verifies that $T(\mathtt{true}) \equiv \mathbb{P}$ and $T(\mathtt{false}) \equiv \mathbb{B}$, we can inhabit it with the following term:

$$\dfrac{\dfrac{\vdash b : \mathbb{B}^{\perp} \mid b : \mathbb{B} \qquad \dfrac{\dfrac{\vdash\mid \Pi(X : \mathbb{P}).X : \Uparrow T(\mathtt{true})}{\langle \Pi(X : \mathbb{P}).X \,\|\, \hat{\ }\rangle^{\ominus} : (\vdash_{\Uparrow T(\mathtt{true})})} \qquad \dfrac{\vdash\mid \mathtt{true} : \Uparrow T(\mathtt{false})}{\langle \mathtt{true} \,\|\, \hat{\ }\rangle^{\ominus} : (\vdash_{\Uparrow T(\mathtt{false})})}}{\vdash_{\Uparrow T(\bullet)} b : \mathbb{B}^{\perp} \mid \mu[\langle \Pi(X : \mathbb{P}).X \,\|\, \hat{\ }\rangle^{\ominus} \mid \langle \mathtt{true} \,\|\, \hat{\ }\rangle^{\ominus}] : \mathbb{B}^{\perp}}}{\big\langle b \,\big\|\, \mu[\langle \Pi(X : \mathbb{P}).X \,\|\, \hat{\ }\rangle^{\ominus} \mid \langle \mathtt{true} \,\|\, \hat{\ }\rangle^{\ominus}]\big\rangle^{+} : (\vdash_{\Uparrow T(b)} b : \mathbb{B}^{\perp})}}{\vdash b : \mathbb{B}^{\perp} \mid \hat{\mu}\big\langle b \,\big\|\, \mu[\langle \Pi(X : \mathbb{P}).X \,\|\, \hat{\ }\rangle^{\ominus} \mid \langle \mathtt{true} \,\|\, \hat{\ }\rangle^{\ominus}]\big\rangle^{+} : \Uparrow T(b)}$$

**CPS translations for ECC** Following the approach advocated in Boulier, Pédrot, and Tabareau (2017), the soundness of our system is proved by means of a syntactic model. In other words, we define a typed translation from our system to (an extension of) Luo's **ECC** (1990). In broad lines, this translation follows the structure of the call-by-value continuation-passing style translation highlighted in Miquey (2019): we use dependent and parametric return types for continuations, and we translate NEF terms $t$ at two different levels $[t]_0$ and $[t]_1$ in a way that is reminiscent of parametricity translations. For instance, the translations of a (closed and NEF) $b$ boolean verify:

$$[b]_1 : \Pi(R : \mathbb{B} \to \mathbb{P}).((\Pi(x : \mathbb{B}).R\,x) \to R\,[b]_0))$$

Observe that by parametricity, this implies in particular that for any continuation $k$ of parametric return type $R$, we have $[b]_1\,R\,k \;\equiv\; k\,[b]_0$, emphasizing that such a translation is only compatible with NEF terms that observationally behave like values.

Insofar as we can easily embed **ECC**+`call/cc` (evaluated in call by value) in our system, this translation allows us to factorize a CPS translation from this calculus to the (pure) **ECC**:

$$\mathbf{ECC} + \mathtt{call/cc} \quad \xrightarrow{\text{macros}} \quad \mathbf{L_{dep}} \quad \xrightarrow{\text{CPS}} \quad \mathbf{ECC}$$

Interestingly, by considering only the pure (by-value) **ECC**, we can define a dependently typed translation to itself without any kind of restriction on dependent types[1]. Our translation improves over Bowman, Cong, Rioux, and Ahmed (2017) in that no extra assumption (in particular, we do not require an extensional type theory) are necessary to prove its soundness.

[1] A Coq development formalizing some aspects of these ideas is available at: https://www.irif.fr/~emiquey/content/CPS_ECC.v

# References

Simon Boulier, Pierre-Marie Pédrot, and Nicolas Tabareau. 2017. The Next 700 Syntactical Models of Type Theory. In *CPP*. https://doi.org/10.1145/3018610.3018620

William J. Bowman, Youyou Cong, Nick Rioux, and Amal Ahmed. 2017. Type-Preserving CPS Translation of $\Sigma$ and $\Pi$ Types is Not Not Possible. *Proc. ACM Program. Lang.* 2, POPL, Article Article 22, 33 pages. https://doi.org/10.1145/3158110

Pierre-Louis Curien, Marcelo Fiore, and Guillaume Munch-Maccagnoni. 2016. A Theory of Effects and Resources: Adjunction Models and Polarised Calculi. In *Proceedings of POPL '16*. https://doi.org/10.1145/2837614.2837652

Pierre-Louis Curien and Hugo Herbelin. 2000. The duality of computation. *ACM SIGPLAN Notices* 35 (2000), 233–243.

Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. 1997. A new deconstructive logic: linear logic. *Journal of Symbolic Logic* (1997). https://doi.org/10.2307/2275572

Jean-Yves Girard. 1991. A new constructive logic: classic logic. *Mathematical Structures in Computer Science* (1991). https://doi.org/10.1017/S0960129500001328

Hugo Herbelin. 2005. On the Degeneracy of Sigma-Types in Presence of Computational Classical Logic. In *Proceedings of TLCA 2005 (LNCS)*, Pawel Urzyczyn (Ed.), Vol. 3461. Springer, 209–220. https://doi.org/10.1007/11417170_16

Hugo Herbelin. 2012. A Constructive Proof of Dependent Choice, Compatible with Classical Logic. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25-28, 2012*. IEEE Computer Society, 365–374. https://doi.org/10.1109/LICS.2012.47

Olivier Laurent. 2002. *Étude de la polarisation en logique*. Thèse de Doctorat. Université Aix-Marseille II. https://tel.archives-ouvertes.fr/tel-00007884

Paul Blain Levy. 2004. *Call-By-Push-Value: A Functional/Imperative Synthesis (Semantics Structures in Computation, V. 2)*. Kluwer Academic Publishers. https://doi.org/10.1007/978-94-007-0954-6

Zhaohui Luo. 1990. *An Extended Calculus of Constructions*. PhD Thesis. University of Edinburgh. https://era.ed.ac.uk/bitstream/handle/1842/12487/Luo1990.Pdf

Étienne Miquey. 2019. A Classical Sequent Calculus with Dependent Types. *ACM Transactions on Programming Languages and Systems* 41 (2019). https://doi.org/10.1145/3230625

Guillaume Munch-Maccagnoni. 2013. *Syntax and Models of a non-Associative Composition of Programs and Proofs*. Theses. Université Paris-Diderot - Paris VII. https://tel.archives-ouvertes.fr/tel-00918642