# Minimal Classical Logic and Control Operators

Zena M. Ariola[1] and Hugo Herbelin[2]

[1] University of Oregon, Eugene, OR 97403, USA
ariola@cs.uoregon.edu
[2] INRIA-Futurs, Parc Club Orsay Université, 91893 Orsay Cedex, France
Hugo.Herbelin@inria.fr

**Abstract.** We give an analysis of various classical axioms and characterize a notion of minimal classical logic that enforces Peirce's law without enforcing Ex Falso Quodlibet. We show that a "natural" implementation of this logic is Parigot's classical natural deduction. We then move on to the computational side and emphasize that Parigot's $\lambda\mu$ corresponds to minimal classical logic. A continuation constant must be added to $\lambda\mu$ to get full classical logic. We then map the extended $\lambda\mu$ to a new theory of control, $\lambda$-$\mathcal{C}^-$-*top*, which extends Felleisen's reduction theory. $\lambda$-$\mathcal{C}^-$-*top* allows one to distinguish between aborting and throwing to a continuation. It is also in correspondence with the proofs of a refinement of Prawitz's natural deduction.

## 1 Introduction

Traditionally, classical logic is defined by extending intuitionistic logic with either Pierce's law, excluded middle or the double negation law. We show that these laws are not equivalent and define *minimal classical logic*, which validates Peirce's law but not Ex Falso Quodlibet (EFQ), i.e. the law $\bot \to A$. The notion is interesting from a computational point of view since it corresponds to a calculus with a notion of control (such as `callcc`) which however does not allow one to abort a computation. We point out that closed typed terms of Parigot's $\lambda\mu$ [Par92] correspond to tautologies of minimal classical logic and not of (full) classical logic. We define a new calculus called $\lambda\mu$-*top*. Tautologies of classical natural deduction correspond to closed typed $\lambda\mu$-*top* terms. We show the correspondence of $\lambda\mu$-*top* with a new theory of control, $\lambda$-$\mathcal{C}^-$-*top*. The calculus $\lambda$-$\mathcal{C}^-$-*top* is interesting in its own right, since it extends Felleisen's theory of control ($\lambda$-$\mathcal{C}$) [FH92]. The study of $\lambda$-$\mathcal{C}^-$-*top* leads to the development of a refinement of Prawitz natural deduction [Pra65] in which one can distinguish between aborting a computation and throwing to a continuation (aborting corresponds to throwing to the top-level continuation). This logic provides a solution to the mismatch between the operational and proof-theoretical interpretation of Felleisen's $\lambda$-$\mathcal{C}$ reduction theory.

More specifically, we devote Section 2 to the definition of the various logics considered in this paper. Sections 3 through 5 explain their computational counterparts. We discuss related work in Section 6 and conclude in Section 7.

$$\frac{}{\Gamma, A \vdash_M A}\ Ax \qquad \frac{\Gamma, A \vdash_M B}{\Gamma \vdash_M A \to B}\ \to_i \qquad \frac{\Gamma \vdash_M A \to B \quad \Gamma \vdash_M A}{\Gamma \vdash_M B}\ \to_e$$

**Fig. 1.** Minimal Natural Deduction

## 2 Minimal, Intuitionistic and Classical Logic

In this paper, we restrict our attention to propositional logic. We assume a set of *formulas*, denoted by roman uppercase letters $A$, $B$, *etc.* , which are built from an infinite set of *propositional atoms* (ranged over by $X, Y$, *etc.* ), a distinguished formula $\bot$ denoting *false*, and *implication* written $\to$. We define *negation* as $\neg A \equiv A \to \bot$. A *named formula* is a pair of a formula and a name taken from an infinite set of *names*. We write $A^x$, $B^\alpha$, *etc.* for named formulas. A *context* is a set of named formulas[3]. We use Greek uppercase letters $\Gamma$, $\Delta$, *etc.* for contexts. We generally omit the names, unless there is an ambiguity. We will consider *sequents* of the form $\Gamma \vdash A$, $\Gamma \vdash$, $\Gamma \vdash ; \Delta$, and $\Gamma \vdash A ; \Delta$. The formulas in $\Gamma$ are the *hypotheses* and the formulas on the right-hand side of the symbol $\vdash$ are the *conclusions*. In each case, the intuitive meaning is that the conjunction of the hypotheses implies the disjunction of the conclusions. Especially, a sequent with no conclusion means the negation of the conjunction of the hypotheses. As initially shown by Gentzen [Gen69] in his sequent calculus LK, classical logic can be obtained by considering sequents with several conclusions. Parigot extended this approach to natural deduction [Par92]. We will see that using sequents with several conclusions allows for a uniform presentation of different logics.

In the rest of the paper, we successively recall the definition of minimal, intuitionistic and classical logic. We consider only formulas built from $\to$ and $\bot$ and we define $\neg A$ as $A \to \bot$. We state simple facts about various classical axioms from which the definition of minimal classical logic emerges. We use natural deduction to formalize the different logics, but we could have used a sequent calculus instead (then the Curry-Howard correspondence would be with Herbelin's calculus [Her94]). If $S$ is a schematic axiom or rule, we denote by $S, \Gamma \vdash A$ the fact that $\Gamma \vdash A$ is derivable using an arbitrary number of instances of $S$.

**Minimal Logic.** *Minimal natural deduction* implements minimal logic [Joh36]. It is defined by the set of (schematic) inference rules given in Figure 1. In minimal logic, $\bot$ is neutral and has no specific rule associated to it.

**Intuitionistic logic.** *Intuitionistic natural deduction* is described in Figure 2. The rule $\bot_e$ introduces a sequent with no conclusion, thus allowing the application of a weakening rule named *Activate*. This logic is strictly stronger

[3] If interested only in provability, one could have defined contexts just as sets of formulas (not as sets of named formulas). But to assign terms to proofs, one needs to be able to distinguish between different occurrences of the same formula. This is the role of names. Otherwise, *e.g.* the two distinct normal proofs of $A, A \vdash A$ (representable by the $\lambda$-terms $\lambda x.\lambda y.x$ and $\lambda x.\lambda y.y$) would have been identified.

$$\frac{}{\Gamma, A \vdash_I A}\,Ax \qquad \frac{\Gamma \vdash_I}{\Gamma \vdash_I A}\,Activate$$

$$\frac{\Gamma \vdash_I \bot}{\Gamma \vdash_I}\,\bot_e \qquad \frac{\Gamma, A \vdash_I B}{\Gamma \vdash_I A \to B}\,\to_i \qquad \frac{\Gamma \vdash_I A \to B \quad \Gamma \vdash_I A}{\Gamma \vdash_I B}\,\to_e$$

**Fig. 2.** Intuitionistic Natural Deduction

than minimal logic[4] but obviously equivalent to minimal logic extended with the schematic axiom $\bot \to A$.

**Proposition 1.** *$\Gamma \vdash_I A$ iff $EFQ, \Gamma \vdash_M A$, but $\nvdash_M EFQ$.*

**Classical axioms.** We now give an analysis in minimal logic of different axiom schemes[5] leading to classical logic.

| | |
|---|---|
| $(\neg A \to A) \to A$ | Weak Peirce's law ($PL_\bot$) |
| $\neg A \vee A$ | Excluded middle (EM) |
| $((A \to B) \to A) \to A$ | Peirce's law (PL) |
| $(A \to B) \vee A$ | Generalized excluded-middle (GEM) |
| $\neg\neg A \to A$ | Double negation law (DN) |

We classify the axioms in three categories: we call PL$_\bot$ and EM *weak classical axioms*, PL and GEM *minimal classical axioms*, and DN a *full classical axiom.* The main results of this section are that none of the classical axioms are indeed derivable in minimal logic and that the weak classical axioms are weaker in minimal logic than the minimal classical axioms, which themselves are weaker than DN. Together with EFQ, weak and minimal classical axioms are however equivalent to DN.

**Proposition 2.** *In minimal logic, we have*

1. *neither PL$_\bot$, PL, EM, GEM nor DN is derivable.*
2. *PL$_\bot$ and EM are equivalent (as schemes).*
3. *GEM and PL are equivalent (as schemes).*
4. *GEM and PL imply EM and PL$_\bot$ but not conversely.*
5. *DN implies GEM and PL but not conversely.*
6. *DN, EM+EFQ, GEM+EFQ, PL$_\bot$+EFQ and PL+EFQ are all equivalent.*

The previous result suggests that there is space for a classical logic which does validate Peirce's law (or GEM) but not EFQ. Let us call this logic *minimal classical logic*. In contrast, EM and PL$_\bot$ without EFQ are weaker than PL, and their addition to minimal logic seems uninteresting. We will investigate a weaker form of EFQ at the end of this section.

[4] This holds for propositional or first-order logic but not for second-order logic, since the second-order formula $\forall X.X$ enjoys $\forall X.X \to A$. However, the rule $\bot_e$ is still not valid for $\forall X.X$.

[5] To reason about excluded-middle, we enrich the set of formulas with disjunction and the usual inference rules.

$$\frac{}{\Gamma, A \vdash_{MC} A; \Delta}\,Ax \qquad \frac{\Gamma \vdash_{MC} ; A, \Delta}{\Gamma \vdash_{MC} A; \Delta}\,Activate \qquad \frac{\Gamma \vdash_{MC} A; A, \Delta}{\Gamma \vdash_{MC} ; A, \Delta}\,Passivate$$

$$\frac{\Gamma, A \vdash_{MC} B; \Delta}{\Gamma \vdash_{MC} A \to B; \Delta}\,\to_i \qquad \frac{\Gamma \vdash_{MC} A \to B; \Delta \qquad \Gamma \vdash_{MC} A; \Delta}{\Gamma \vdash_{MC} B; \Delta}\,\to_e$$

**Fig. 3.** Minimal Classical Natural Deduction

**Minimal Classical Logic.** An axiom-free implementation of minimal classical logic is actually Parigot's classical natural deduction [Par92] (with no special rule for $\bot$). The inference rules are shown in Figure 3. Parigot's convention is to have two kinds of sequents, one with only named formulas on the right, written $\Gamma \vdash ; \Delta$, and one with exactly one unnamed formula on the right, written $\Gamma \vdash A; \Delta$. We now state that minimal Parigot classical natural deduction is equivalent to minimal logic extended with Peirce's law, *i.e.* it implements minimal classical logic[6].

**Proposition 3.** *$\Gamma \vdash_{MC} A$ iff $PL, \Gamma \vdash_M A$*

Note that in minimal classical logic, DN is not provable but $\vdash_{MC} \neg\neg A \to A; \bot$ is.

$$\frac{}{\Gamma, A \vdash_{C} A; \Delta}\,Ax \qquad \frac{\Gamma \vdash_{C} ; A, \Delta}{\Gamma \vdash_{C} A; \Delta}\,Activate \qquad \frac{\Gamma \vdash_{C} A; A, \Delta}{\Gamma \vdash_{C} ; A, \Delta}\,Passivate$$

$$\frac{\Gamma \vdash_{C} \bot; \Delta}{\Gamma \vdash_{C} ; \Delta}\,\bot_e \qquad \frac{\Gamma, A \vdash_{C} B; \Delta}{\Gamma \vdash_{C} A \to B; \Delta}\,\to_i \qquad \frac{\Gamma \vdash_{C} A \to B; \Delta \qquad \Gamma \vdash_{C} A; \Delta}{\Gamma \vdash_{C} B; \Delta}\,\to_e$$

**Fig. 4.** Classical Natural Deduction

**Classical Logic.** To obtain full classical logic from minimal Parigot's classical natural deduction[7] and thus derive DN, we explicitly add the elimination rule for $\bot$. The *(full) Parigot's classical natural deduction* is described in Figure 4. From Propositions 1, 2 and 3, we directly have:

**Proposition 4.** *$\Gamma \vdash_C A$ iff $PL, \Gamma \vdash_I A$ iff $DN, \Gamma \vdash_M A$ iff $EFQ, \Gamma \vdash_{MC} A$.*

[6] The proof proceeds by replacing each instance of *Activate* on $A$ by as many instances of $PL$ as the number of *Passivate* applied on $A$

[7] Parigot's original formulation of classical natural deduction [Par92] does not include the $\bot_e$-rule but gives direct rules for negation which are easily derivable from the elimination rule of $\bot$.

As expected, full classical logic is conservative over minimal classical logic for formulas not mentioning the $\bot$ formula, as stated by the following:

**Proposition 5.** *If $\bot$ does not occur in $A$ then $\vdash_C A$ iff $\vdash_{MC} A$ .*

*Remark 1.* Minimal classical natural deduction without the *Passivate* rule yields minimal logic, since the context $\Delta$ is inert and can only remain empty in a derivation for which the end sequent has the form $\Gamma \vdash A$; (even the *Activate* rule cannot be applied). Similarly, classical natural deduction without the *Passivate* rule yields intuitionistic logic. As a consequence, minimal and intuitionistic natural deduction can both be seen as subsystems of classical natural deduction.

$$\frac{}{\Gamma, A \vdash_{RAA} A}\,Ax \qquad \frac{\Gamma \vdash_{RAA} \bot^c}{\Gamma \vdash_{RAA} A}\,Activate \qquad \frac{\Gamma, \neg_c A \vdash_{RAA} \bot^c}{\Gamma \vdash_{RAA} A}\,RAA_c$$

$$\frac{\Gamma \vdash_{RAA} \bot}{\Gamma \vdash_{RAA} \bot^c}\,\bot_e^c \qquad \frac{\Gamma, A \vdash_{RAA} B}{\Gamma \vdash_{RAA} A \to B}\,\to_i \qquad \frac{\Gamma \vdash_{RAA} A \to B \quad \Gamma \vdash_{RAA} A}{\Gamma \vdash_{RAA} B}\,\to_e$$

**Fig. 5.** Natural Deduction with RAA$_c$

**Minimal Prawitz Classical Logic.** Prawitz defines classical logic as minimal logic plus the Reductio Ad Absurdum rule (RAA) [Pra65]: from $\Gamma, \neg A \vdash \bot$ deduce $\Gamma \vdash A$. This rule implies EFQ (as DN implies EFQ) and hence yields full classical logic. In this section, we are interested in exploring the possibility of defining minimal classical logic from minimal logic and RAA but without deriving EFQ. Equivalently, we would like to devise a restricted version of EFQ that would allow one to prove PL from PL$_\bot$. This alternative formulation of (minimal) classical logic is obtained by distinguishing two different notions of $\bot$: $\bot$ for commands (written as $\bot^c$) and $\bot$ for terms, see Figure 5, where $\neg_c A$ stands for $A \to \bot^c$. If the context $\Delta$ is the set of formulas $A_1, \cdots, A_n$, then we write $\neg_c \Delta$ for the set $\neg_c A_1, \cdots, \neg_c A_n$. Sequents are of the form $\Gamma, \neg_c \Delta \vdash A$ or $\Gamma, \neg_c \Delta \vdash \bot_c$ and $\bot^c$ is not allowed to occur in $\Gamma$, $\Delta$ and $A$. The minimal subset does not contain the $\bot_e^c$ rule and is denoted by $\vdash_{MRAA}$.

**Proposition 6.** *Given a formula $A$ and contexts $\Gamma$ and $\Delta$, all with no occurrences of $\bot^c$, we have* (1) *$\Gamma \vdash_{MC} A; \Delta$ iff $\Gamma, \neg_c \Delta \vdash_{MRAA} A$;* (2) *$\Gamma \vdash_C A; \Delta$ iff $\Gamma, \neg_c \Delta \vdash_{RAA} A$.*

## 3 Computational Content of Minimal Logic + DN

To reason about Scheme programs, Felleisen *et al.* [FH92] introduced the $\lambda$-$\mathcal{C}$ calculus. $\mathcal{C}$ provides *abortive continuations*: the invocation of a continuation reinstates the captured context *in place* of the current one. Griffin was the first to observe that $\mathcal{C}$ is typable with $\neg\neg A \to A$. This extended the Curry-Howard isomorphism to classical logic [Gri90].

**Proposition 7 (Griffin).** *A formula $A$ is provable in classical logic iff there exists a closed $\lambda$-$\mathcal{C}$ term $M$ such that $\vdash M : A$ is provable, where $\vdash M : A$ is simple typability in $\lambda$ calculus.*

Felleisen also developed the $\lambda$-$\mathcal{K}$ calculus which axiomatizes the `callcc` (*i.e.* call-with-current-continuation) control operator. In contrast to $\mathcal{C}$, $\mathcal{K}$ leaves the current context intact as explicitly described in its usual encoding: $\mathcal{K}(M) = \mathcal{C}(\lambda k.k(Mk))$. $\mathcal{K}$ is not as powerful as $\mathcal{C}$ [Fel90]. In order to define $\mathcal{C}$ (of type DN) we need the abort primitive $\mathcal{A}$ (of type EFQ): $\mathcal{C}(M) = \mathcal{K}(\lambda k.\mathcal{A}(Mk))$. An alternative encoding $\mathcal{K}(M) = \mathcal{C}(\lambda k.k(M\lambda x.\mathcal{A}(kx)))$ obeys the same reduction rules and shows that $\mathcal{K}$ can be typed with PL. Following Proposition 3, we have:

**Proposition 8.** *A formula $A$ is provable in minimal classical logic iff there exists a closed $\lambda$-$\mathcal{K}$ term $M$ such that $\vdash M : A$ is provable.*

The call-by-value and call-by-name reduction semantics of $\lambda$-$\mathcal{C}$ are presented in Figure 6. An important point to clarify is the presence of the abort operations in the right-hand sides of the reduction rules. As far as evaluation is concerned, they are not necessary. They are important in order to obtain a satisfying correspondence between the operational and reduction semantics. For example, the term $\mathcal{C}(\lambda k.(k\ \lambda x.x)N)$ evaluates to $\lambda x.x$. However, the absence of the abort from the reduction rules makes impossible to get rid of the control context $\lambda f.f\ N$. The abort steps signal that $k$ is not a normal function but is an abortive continuation. As we explain in Section 5, these abort steps are different from the abort used in defining $\mathcal{C}$ in terms of `callcc`: $\mathcal{C}(M) = \mathtt{callcc}(\lambda k.\mathcal{A}(Mk))$. The aborts in the reduction rules correspond to throwing to a user defined continuation (*i.e.* a *Passivate* step), whereas the abort in the definition of $\mathcal{C}$ corresponds to throwing to the predefined top-level continuation (*i.e.* a $\bot_e$ step).

$$
\lambda_n\text{-}\mathcal{C}\quad
\left\{\begin{array}{lll}
\beta : & (\lambda x.M)N & \rightarrow M[x := N] \\
\mathcal{C}_L : & (\mathcal{C}M)N & \rightarrow \mathcal{C}(\lambda k.M(\lambda f.\mathcal{A}(k(fN)))) \\
\mathcal{C}_{top} : & \mathcal{C}M & \rightarrow \mathcal{C}(\lambda k.M(\lambda f.\mathcal{A}(kf))) \\
\mathcal{C}_{idem} : & \mathcal{C}(\lambda k.\mathcal{C}M) & \rightarrow \mathcal{C}(\lambda k.M(\lambda x.\mathcal{A}(x))) \\
\mathcal{C}_{elim} : & \mathcal{C}(\lambda k.kM) & \rightarrow M \qquad k \notin FV(M)
\end{array}\right.
$$

$$
\begin{array}{c}\lambda_v\text{-}\mathcal{C}\\ V ::= x \mid \lambda x.M\end{array}\quad
\left\{\begin{array}{lll}
\beta : & (\lambda x.M)V & \rightarrow M[x := V] \\
\mathcal{C}_L : & (\mathcal{C}M)N & \rightarrow \mathcal{C}(\lambda k.M(\lambda f.\mathcal{A}(k(fN)))) \\
\mathcal{C}_R : & V(\mathcal{C}M) & \rightarrow \mathcal{C}(\lambda k.M(\lambda x.\mathcal{A}(k(Vx)))) \\
\mathcal{C}_{top} : & \mathcal{C}M & \rightarrow \mathcal{C}(\lambda k.M(\lambda f.\mathcal{A}(kf))) \\
\mathcal{C}_{idem} : & \mathcal{C}(\lambda k.\mathcal{C}M) & \rightarrow \mathcal{C}(\lambda k.M(\lambda x.\mathcal{A}(x)))
\end{array}\right.
$$

**Fig. 6.** $\lambda_n$-$\mathcal{C}$ and $\lambda_v$-$\mathcal{C}$ reduction rules

*Remark 2.* Parigot in [Par92] criticized Griffin's work because the proposed $\mathcal{C}$-typing did not fit the operational semantics. Actually, the only rule that breaks subject reduction is the top-level computation rule (*i.e.* $\mathcal{C}M \mapsto M(\lambda x.\mathcal{A}(x))$ (not

$$t :: x \mid \lambda x.t \mid tt \mid \mu\alpha.c \qquad c ::= [\beta]t \mid [top]t$$

$$\frac{}{\Gamma, A^x \vdash x : A; \Delta}\,Ax \qquad \frac{c : \Gamma \vdash; A^\alpha, \Delta}{\Gamma \vdash \mu\alpha.c : A; \Delta}\,Activate \qquad \frac{\Gamma \vdash t : A; A^\alpha, \Delta}{[\alpha]t : \Gamma \vdash; A^\alpha, \Delta}\,Passivate$$

$$\frac{\Gamma \vdash t : \bot; \Delta}{[top]t : \Gamma \vdash; \Delta}\,\bot_e \qquad \frac{\Gamma \vdash t{:}A \to B; \Delta \quad \Gamma \vdash s : A; \Delta}{\Gamma \vdash ts{:}B; \Delta}\,\to_e \qquad \frac{\Gamma, A^x \vdash t : B; \Delta}{\Gamma \vdash \lambda x.t : A \to B; \Delta}\,\to_i$$

**Fig. 7.** $\lambda\mu$ and $\lambda\mu$-*top* calculi

mentioned on Figure 6) which forces a conversion from $\bot$ to the top-level type. To solve the problem, instead of reducing $M$, Griffin proposed to reduce $C(\lambda\alpha.\alpha M)$, where $\alpha M$ is of type $\bot$. As detailed in the next section, the classical version of Parigot's $\lambda\mu$ requires a similar intervention (a free continuation constant – that we call *top* – is needed).

## 4 Computational Content of Classical Natural Deduction

Figure 7 describes Parigot's $\lambda\mu$ calculus [Par92] which is a term assignment for his classical natural deduction. The *Passivate* rule reads as follows: given a term producing a value of type $A$, if $\alpha$ is a continuation variable waiting for something of type $A$ (*i.e.* $A$ *cont*), then by invoking the continuation variable we leave the current context. Terms of the form $[\alpha]t$ are called *commands*. The *Activate* rule reads as follows: given a command (*i.e.* no formula is focused) we can select which result to get by capturing the associated continuation. If $A^\alpha$ is not present in the precondition then the rule corresponds to weakening. Note that the rule $\bot_e$ differs from Parigot's version. In [Par92], the elimination rule for $\bot$ is interpreted by an unnamed term $[\gamma]t$, where $\gamma$ is any continuation variable (not always the same for every instance of the rule). In contrast, the rule is here systematically associated to the same primitive continuation variable *top* considered as a constant. This was also observed by Streicher *et al.* [SR98]. Parigot would represent DN as the term $\lambda y.\mu\alpha.[\gamma](y\lambda x.\mu\delta.[\alpha]x)$ whereas our representation is $\lambda y.\mu\alpha.[top](y\lambda x.\mu\delta.[\alpha]x)$. We use $\lambda\mu$-*top* to denote the whole calculus with $\bot_e$ and $\lambda\mu$ to denote the calculus without $\bot_e$. The need for an extra continuation constant to interpret the elimination of $\bot$ can be emphasized by the following statement:

**Proposition 9.** *A formula $A$ is provable in minimal classical logic (resp. classical logic) iff there exists a* closed *$\lambda\mu$ term (resp. $\lambda\mu$-top term) $t$ such that $\vdash t : A$ is provable.*

We write $\lambda\mu_n$ and $\lambda\mu_v$ (resp. $\lambda\mu_n$-*top* and $\lambda\mu_v$-*top*) for the $\lambda\mu$ calculus (resp. $\lambda\mu$-*top* calculus) equipped with call-by-name and call-by-value reduction rules, respectively. The reduction rules are given in Figure 8 (substitutions $[[\alpha](ws)/[\alpha]w]$ and $[[\alpha](sw)/[\alpha]w]$ are defined as in [Par92]). Note that the rules

| Calculus | Rule | Redex | | Reduct |
|---|---|---|---|---|
| $\lambda\mu_n$ and $\lambda\mu_n$-*top* | Logical rule: | $(\lambda x.t)s$ | $\rightarrow$ | $t[x := s]$ |
| | Structural rule: | $(\mu\alpha.t)s$ | $\rightarrow$ | $(\mu\alpha.t[[\alpha](ws)/[\alpha]w])$ |
| | Renaming rule: | $\mu\alpha.[\beta]\mu\gamma.u$ | $\rightarrow$ | $\mu\alpha.u[\beta/\gamma]$ |
| | Simplification rule: | $\mu\alpha.[\alpha]u$ | $\rightarrow$ | $u \quad \alpha \notin FV(u)$ |
| $\lambda\mu_v$ and $\lambda\mu_v$-*top* $(v ::= x \mid \lambda x.t)$ | Logical rule: | $(\lambda x.t)v$ | $\rightarrow$ | $t[x := v]$ |
| | Left structural rule: | $(\mu\alpha.t)s$ | $\rightarrow$ | $(\mu\alpha.t[[\alpha](ws)/[\alpha]w])$ |
| | Right structural rule: | $v(\mu\alpha.t)$ | $\rightarrow$ | $(\mu\alpha.t[[\alpha](vw)/[\alpha]w])$ |
| | Renaming rule: | $\mu\alpha.[\beta]\mu\gamma.u$ | $\rightarrow$ | $\mu\alpha.u[\beta/\gamma]$ |
| | Simplification rule: | $\mu\alpha.[\alpha]u$ | $\rightarrow$ | $u \quad \alpha \notin FV(u)$ |

**Fig. 8.** Call-by-name and call-by-value $\lambda\mu$ and $\lambda\mu$-*top* reduction rules

are the same for the $\lambda\mu$ and $\lambda\mu$-*top* calculi. $\lambda\mu_n$ is Parigot's original calculus, while our presentation of $\lambda\mu_v$ is similar to Ong and Stewart [OS97]. Both sets of reduction rules are well-typed and enjoy subject reduction.

Instead of showing a correspondence between the $\lambda\mu$-*top* calculi and the $\lambda$-$\mathcal{C}$ calculi, as in [dG94], we have searched for an isomorphic calculus, which turns out to be interesting in its own right since it extends the expressive power of Felleisen $\lambda$-$\mathcal{C}$ and provides a term assignment for Prawitz classical logic.

## 5 Computational Content of Prawitz Classical Deduction

$$M ::= x \mid MM \mid \lambda x.M \mid \mathcal{C}^-(\lambda k.N) \quad N ::= k'M \mid topM$$

$$\frac{}{\Gamma, x : A \vdash x : A}\,Ax \qquad \frac{\Gamma \vdash N : \bot^c}{\Gamma \vdash \mathcal{C}^-(\lambda q.N) : A}\,Activate \qquad \frac{\Gamma, k : \neg_c A \vdash N : \bot^c}{\Gamma \vdash \mathcal{C}^-(\lambda k.N) : A}\,RAA_c$$

$$\frac{\Gamma \vdash M : \bot}{\Gamma \vdash topM : \bot^c}\,\bot^c_e \qquad \frac{\Gamma \vdash M : A \rightarrow B \quad \Gamma \vdash M' : A}{\Gamma \vdash MM' : B}\,\rightarrow_e \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \rightarrow B}\,\rightarrow_i$$

**Fig. 9.** $\lambda$-$\mathcal{C}^-$ and $\lambda$-$\mathcal{C}^-$-*top* calculi

We consider a restricted form of $\lambda$-$\mathcal{C}$, called $\lambda$-$\mathcal{C}^-$-*top*. Its typing system is given in Figure 9. In $\lambda$-$\mathcal{C}^-$-*top*, we distinguish between capturing a continuation and expressing where to go next. We assume the existence of a top-level continuation called *top*. The control operator $\mathcal{C}^-$ can only be applied to a lambda abstraction. Moreover, the body of a $\mathcal{C}^-$-lambda abstraction is always of the form $kM$ for a continuation variable $k$. In $\lambda$-$\mathcal{C}^-$-*top*, K and $\mathcal{C}$ are expressed as $\mathcal{C}^-(\lambda k.k\ M)$ and $\mathcal{C}^-(\lambda k.top\ M)$, respectively. In $\lambda$-$\mathcal{C}^-$-*top*, it is possible to distinguish between aborting a computation and throwing to a continuation. For example, one would write $\mathcal{C}^-(\lambda d.top\ M)$ to abort the computation $M$ and $\mathcal{C}^-(\lambda d.k\ M)$ to invoke continuation $k$ with $M$ ($d$ not free in $M$). Variables and continuation variables are kept distinct. The translation from $\lambda$-$\mathcal{C}$ to $\lambda$-$\mathcal{C}^-$-*top* is expressed as follows: $\overline{\overline{x}} = x$, $\overline{\overline{\lambda x.M}} = \lambda x.\overline{\overline{M}}$, $\overline{\overline{MN}} = \overline{\overline{M}}\;\overline{\overline{N}}$, and $\overline{\overline{\mathcal{C}M}} = \mathcal{C}^-(\lambda k.top(\overline{\overline{M}}(\lambda x.\mathcal{C}^-(\lambda\delta.kx))))$. The call-by-name and call-by-value $\lambda$-$\mathcal{C}^-$-*top* reduction rules are given in Figure 10. Note that one does not need the $\mathcal{C}_{top}$-rule, whose action is to wrap up an application of a continuation with a throw operation. $\mathcal{C}^-_{idem}$ is a generalization of $\mathcal{C}_{idem}$, which is obtained by instantiating the continuation variable $k'$ to *top* (*i.e.* the continuation $\lambda x.\mathcal{A}(x)$): $\mathcal{C}^-(\lambda k.top\ \mathcal{C}(\lambda q.M)) \to \mathcal{C}^-(\lambda k.M[top/q])$. $\mathcal{C}^-_{idem}$ is similar to the rule proposed by Barbanera *et al.* [BB93]: $M(\mathcal{C}N) \to N(\lambda a.(Ma))$, where $M$ has type $\neg A$. Felleisen proposed in [FH92] the following additional rules for $\lambda_v$-$\mathcal{C}$: $\mathcal{C}_E : E[\mathcal{C}M] \to \mathcal{C}(\lambda k.M(\lambda x.\mathcal{A}(k\ E[x])))$ (where $E$ stands for a call-by-value evaluation context) and $\mathcal{C}_{elim} : \mathcal{C}(\lambda k.k\ M) \to M$, where $k$ is not free in $M$. The first rule is a generalization of $\mathcal{C}_L$, $\mathcal{C}_R$, and $\mathcal{C}_{top}$ which adds expressive power to the calculus. The second rule, which also appears in [Hof95], leads to better simulation of evaluation. However, both rules destroy confluence of $\lambda_v$-$\mathcal{C}$. Felleisen left unresolved the problem of finding an extended theory that would include $\mathcal{C}_E$ or $\mathcal{C}_{elim}$ and still satisfy the classical properties of reduction theories. $\mathcal{C}_{elim}$ is already present in our calculi and $\mathcal{C}_E$ is derivable. Thus one may consider our calculi as a solution.

**Proposition 10.** *1. $\lambda_v$-$\mathcal{C}^-$-top and $\lambda_n$-$\mathcal{C}^-$-top are confluent and strongly normalizing.*
*2. Subject reduction: Given $\lambda_v$-$\mathcal{C}^-$-top ($\lambda_n$-$\mathcal{C}^-$-top) terms $M, N$, if $\Gamma \vdash M : A$ and $M \twoheadrightarrow N$ then $\Gamma \vdash N : A$.*

Soundness and completeness of $\lambda_v$-$\mathcal{C}^-$-*top* with respect to $\lambda_v$-$\mathcal{C}$ are stated below, where $\simeq_c$ denotes operational equivalence as defined in [FH92]. A $\lambda_v$-$\mathcal{C}^-$-*top* term $M$ is translated into a $\lambda_v$-$\mathcal{C}$ term $\overline{M}$ by simply replacing $\mathcal{C}^-$ with $\mathcal{C}$ and by erasing the references to the *top* continuation.

**Proposition 11.** *1. Given $\lambda_v$-$\mathcal{C}$ terms $M$ and $N$, if $M \twoheadrightarrow N$ then $\overline{\overline{M}} \twoheadrightarrow \overline{\overline{N}}$.*
*2. Given $\lambda_v$-$\mathcal{C}^-$-top terms $M$ and $N$, if $M \twoheadrightarrow N$ then $\overline{M} \simeq_c \overline{N}$.*

**Relation between the $\lambda\mu$-*top* and the $\lambda$-$\mathcal{C}^-$-*top* calculi.** The $\lambda$-$\mathcal{C}^-$-*top* calculus has been designed in such a way that it is in one-to-one correspondence with the $\lambda\mu$-*top* calculus: $\overline{\lambda x.t} = \lambda x.\overline{t}$, $\overline{ts} = \overline{t}\overline{s}$, $\overline{\mu\alpha.[\gamma]t} = \mathcal{C}^-(\lambda\alpha.\gamma\overline{t})$. The correspondence extends to the reduction rules: Figure 10 matches Figure 8. This is expressed by the following statement:

**Proposition 12.** *Let $t, s$ be $\lambda\mu$-top-terms, then $t \to_{\lambda\mu_n\text{-}top} s$ iff $\overline{t} \to_{\lambda_n\text{-}\mathcal{C}^-\text{-}top} \overline{s}$ and $t \to_{\lambda\mu_v\text{-}top} s$ iff $\overline{t} \to_{\lambda_v\text{-}\mathcal{C}^-\text{-}top} \overline{s}$ .*

**Proposition 13.** *A formula $A$ is provable in Prawitz classical logic iff there exists a* closed *$\lambda\mathcal{C}^-$-top term $M$ such that $\vdash M : A$ is provable.*

We define a subset of $\lambda$-$\mathcal{C}^-$-*top*, which does not allow one to abort a computation, *i.e.* terms of the form $\mathcal{C}^-(\lambda k.topM)$ are not allowed. We call this subset, which is isomorphic to $\lambda\mu$, $\lambda$-$\mathcal{C}^-$ .

$$\begin{array}{cl}
\begin{array}{c}\lambda_n\text{-}\mathcal{C}^- \\ \text{and} \\ \lambda_n\text{-}\mathcal{C}^-\text{-}top\end{array} &
\left\{\begin{array}{lll}
\beta : & (\lambda x.M)N & \to M[x := N] \\
\mathcal{C}^-_L : & \mathcal{C}^-(\lambda k.M)N & \to \mathcal{C}^-(\lambda k.M[k\ (PN)/k\ P]) \\
\mathcal{C}^-_{idem} : & \mathcal{C}^-(\lambda k.k'\mathcal{C}^-(\lambda q.N)) & \to \mathcal{C}^-(\lambda k.N[k'/q]) \\
\mathcal{C}^-_{elim} : & \mathcal{C}^-(\lambda k.kM) & \to M \quad k \notin FV(M)
\end{array}\right. \\
& \\
\begin{array}{c}\lambda_v\text{-}\mathcal{C}^- \\ \text{and} \\ \lambda_v\text{-}\mathcal{C}^-\text{-}top \\ (V ::= x \mid \lambda x.M)\end{array} &
\left\{\begin{array}{lll}
\beta : & (\lambda x.M)V & \to M[x := V] \\
\mathcal{C}^-_{elim} : & \mathcal{C}^-(\lambda k.kM) & \to M \quad k \notin FV(M) \\
\mathcal{C}^-_L : & \mathcal{C}^-(\lambda k.M)N & \to \mathcal{C}^-(\lambda k.M[k\ (PN)/k\ P]) \\
\mathcal{C}^-_R : & V\mathcal{C}^-(\lambda k.M) & \to \mathcal{C}^-(\lambda k.M[k\ (VP)/k\ P]) \\
\mathcal{C}^-_{idem} : & \mathcal{C}^-(\lambda k.k'\mathcal{C}^-(\lambda q.N)) & \to \mathcal{C}^-(\lambda k.N[k'/q])
\end{array}\right.
\end{array}$$

**Fig. 10.** $\lambda$-$\mathcal{C}^-$ and $\lambda$-$\mathcal{C}^-$-*top* reduction rules

**Proposition 14.** *A formula $A$ is provable in minimal Prawitz classical logic iff there exists a closed $\lambda$-$\mathcal{C}^-$ term $M$ such that $\vdash M : A$ is provable.*

*Remark 3.* The $\lambda$-$\mathcal{C}^-$ term representing PL is $\lambda y.\mathcal{C}^-(\lambda k.k(y(\lambda x.\mathcal{C}^-(\lambda q.kx))))$, which can be written in ML as :

```
- fun PL y = callcc (fn k => (y  (fn x => throw k x)));
val PL = fn : (('a -> 'b) -> 'a) -> 'a
```

Notice how the throw construct corresponds to a weakening step. By Propositions 6, 8 and 14, $\lambda$-$\mathcal{C}^-$ is equivalent to $\lambda$-$\mathcal{K}$, assuming `callcc` is typed with PL, say `callcc`$_{\texttt{pl}}$. However, it might not be at all obvious how to use a continuation in different contexts, since we do not have weakening available. Consider for example the following ML term (with `callcc` and `throw` typable as in [DHM91]):

```
- callcc (fn k => if (throw k 1) then 7 else (throw k 99));
```

We use the continuation in both boolean and integer contexts. How can we write the above expression without making use of weakening or throw? The proof of Proposition 3 gives the answer:

```
- callcc_pl (fn k => callcc_pl (fn q => if q 1 then 7 else k 99));
```

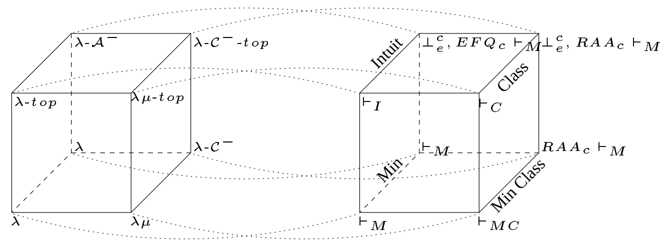
We define a subset of $\lambda$-$\mathcal{C}^-$, called $\lambda$-$\mathcal{A}^-$, in which expressions of the form $\mathcal{C}^-(\lambda d.qM)$ are only allowed when $d$ is not free in $M$ and $q$ is *top*, that is, we only allow throwing to the top-level continuation.

**Proposition 15.** *A formula $A$ is provable in intuitionistic logic iff there exists a closed $\lambda$-$\mathcal{A}^-$ term $M$ such that $\vdash_I M : A$ is provable.*

## 6 Related Work

The relation between Parigot $\lambda\mu$ and $\lambda$-$\mathcal{C}$ has been investigated by de Groote [dG94], who only considers the $\lambda\mu$ structural rule but not renaming and simplification. As for $\lambda$-$\mathcal{C}$, he only considers $\mathcal{C}_L$ and $\mathcal{C}_{top}$. However, these rules are not the original rules of Felleisen, since they do not contain abort. For example, $\mathcal{C}_{top}$ is $\mathcal{C}M \to \mathcal{C}(\lambda k.M(\lambda f.kf))$ which is in fact a reduction rule for $\lambda$-$\mathcal{F}$ [Fel88]. This work fails in relating $\lambda\mu$ to $\lambda$-$\mathcal{C}$ in an untyped framework, since it does not express continuations as abortive functions. It says in fact that $\mathcal{F}$ behaves as $\mathcal{C}$ in the simply-typed case. Ong and Stewart [OS97] also do not consider the abort step in Felleisen's rules. This could be justified because in a simply-typed setting these steps are of type $\perp \to \perp$. Therefore, it seems we have a mismatch. While the aborts are essential in the reduction semantics, they are irrelevant in the corresponding proof. We are the first to provide a proof theoretic justification for those abort steps, they correspond to the step $\perp \to \perp_c$. In addition to Ong and Stewart, Py [Py98] and Bierman [Bie98] have pointed out the peculiarity of having an open $\lambda\mu$ term corresponding to a tautology. Their solution is to abolish the distinction between commands and terms. A command is a term returning $\perp$. The body of a $\mu$-abstraction is not restricted to a command, but can be of the form $\mu\alpha.t$, where $t$ is of type $\perp$. Thus, one has $\lambda y.\mu\alpha.(y\ \lambda x.[\alpha]x) : \neg\neg A \to A$. We would then represent the term $\mathcal{C}(\lambda k.(kI)x)$ (where $I$ is $\lambda x.x$) as $\mu\alpha.(\alpha I)x$. Whereas $\mathcal{C}(\lambda k.kIx)$ would reduce to $\mathcal{C}(\lambda k.kI)$ according to $\lambda_n$-$\mathcal{C}$ and to $I$ in $\lambda\mu_n$-*top*, it would be in normal form in their calculus. Thus, their work in relating $\lambda\mu$ to $\lambda$-$\mathcal{C}$ only applies to typed $\lambda$-$\mathcal{C}$, whereas our work also applies to the untyped case. Crolard [Cro99] studied the relation between Parigot's $\lambda\mu$ and a calculus with a `catch` and `throw` mechanism. He showed that contraction corresponds to the `catch` operator (*i.e.* $\mu\alpha.[\alpha]t = \texttt{catch}\ \alpha\ t$) and weakening corresponds to the `throw` operator (*i.e.* $\mu\delta.[\alpha]t = \texttt{throw}\ \alpha\ t$ for $\delta$ not free in $t$). He only considers terms of the form $\mu\alpha.[\alpha]t$ and $\mu\beta.[\alpha]t$, where $\beta$ does not occur free in $t$. This property is not preserved by the renaming rule, therefore reduction is restricted. We do not require such restrictions on reduction. We can simulate Ong and Stewart's $\lambda\mu$ and the Crolard calculus via this simple translation: $\mu\alpha.t$ becomes $\mu\alpha.[top]t$ and $[\beta]t$ becomes $\mu\delta.[\beta]t$, where $\delta$ is not free in $t$.

## 7 Conclusions



Our analysis of the logical strength of EFQ, PL (or EM) and DN has led naturally to a restricted form of classical logic, called *minimal classical logic*. Depending on whether EFQ, PL or both are assumed in minimal logic, we get intuitionistic, minimal classical or classical logic. Depending on whether we admit the *Passivate* ($RAA_c$[8]) and $\perp_e$ ($\perp_e^c$) in full classical natural deduction (on

[8] with restrictions on the use of $\perp_c$

top of minimal natural deduction), we get the correspondences with the $\lambda$-calculi considered in the paper, as shown above[9]. Among these systems, $\lambda$-$\mathcal{C}^-$-*top* is a confluent extension of Felleisen's theory of control.

**Acknowledgements.** We thank Matthias Felleisen for the numerous discussions about his theory of control. Miley Semmelroth helped us to improve the presentation of the paper. We also thank the anonymous referees for their comments. The first author has been supported by NSF grant 0204389.

## References

[BB93] F. Barbanera and S. Berardi. Extracting constructive content from classical logic via control-like reductions. In *TLCA'93, LNCS 664*, pages 45–59. 1993.

[Bie98] G.M. Bierman. A computational interpretation of the lambda-mu calculus. In *MFCS'98, LNCS 1450*, pages 336–345, 1998.

[Cro99] T. Crolard. A confluent lambda-calculus with a catch/throw mechanism. *Journal of Functional Programming*, 9(6):625–647, 1999.

[dG94] P. de Groote. On the relation between the lambda-mu calculus and the syntactic theory of sequential control. In *LPAR'94*, pages 31–43. 1994.

[DHM91] B. F. Duba, R. Harper, and D. MacQueen. Typing first-class continuations in ML. In *POPL'91*, pages 163–173, 1991.

[Fel88] M. Felleisen. The theory of practice of first-class prompt. In *POPL '88*, pages 180–190, 1988.

[Fel90] M. Felleisen. On the expressive power of programming languages. In *ESOP '90, LNCS 432*, pages 134–151. 1990.

[FH92] M. Felleisen and R. Hieb. A revised report on the syntactic theories of sequential control and state. *Theoretical Computer Science*, 103(2):235–271, 1992.

[Gen69] G. Gentzen. Investigations into logical deduction. In M.E. Szabo, editor, *Collected papers of Gerhard Gentzen*, pages 68–131. North-Holland, 1969.

[Gri90] T. G. Griffin. The formulae-as-types notion of control. In *POPL'90*, pages 47–57, 1990.

[Her94] H. Herbelin. A lambda-calculus structure isomorphic to Gentzen-style sequent calculus structure. In *CSL'94, LNCS 933*, 1994.

[Hof95] M. Hofmann. Sound and complete axiomatization of call-by-value control operators. *Mathematical Structures in Computer Science*, 5:461–482, 1995.

[Joh36] I. Johansson. Der Minimalkalkül, ein Reduzierter Intuitionistischer Formalismus. *Compositio Mathematica*, 4:119–136, 1936.

[OS97] C.-H. Luke Ong and C. A. Stewart. A Curry-Howard foundation for functional computation with control. In *POPL'97*, pages 215–227. 1997.

[Par92] M. Parigot. Lambda-mu-calculus: An algorithmic interpretation of classical natural deduction. In *LPAR '92*, pages 190–201, 1992.

[Pra65] D. Prawitz. *Natural Deduction, a Proof-Theoretical Study*. Almquist and Wiksell, Stockholm, 1965.

[Py98] W. Py. *Confluence en $\lambda\mu$-calcul*. PhD thesis, Université de Savoie, 1998.

[SR98] T. Streicher and B. Reus. Classical logic: Continuation semantics and abstract machines. *Journal of Functional Programming*, 8(6):543–572, 1998.

[9] $\lambda$-*top* is the subset of $\lambda\mu$-*top* in which expressions of the form $\mu\delta.[\alpha]t$ are only allowed when $\delta$ is not free in $t$ and $\alpha$ is *top*. $EFQ_c$ is the rule $\Gamma \vdash \bot_c$ implies $\Gamma \vdash A$ (the restriction of $RAA_c$ when $\neg_c A$ is not used in the proof).