JFP **19** (1): 17–26, 2009. © 2008 Cambridge University Press
doi:10.1017/S0956796808006734 First published online 6 August 2008 Printed in the United Kingdom

THEORETICAL PEARL

Control effects as a modality

HAYO THIELECKE
University of Birmingham, Birmingham, UK
(*e-mail:* H.Thielecke@cs.bham.ac.uk)

Abstract

We combine ideas from types for continuations, effect systems and monads in a very simple setting by defining a version of classical propositional logic in which double-negation elimination is combined with a modality. The modality corresponds to control effects, and it includes a form of effect masking. Erasing the modality from formulas gives classical logic. On the other hand, the logic is conservative over intuitionistic logic.

1 Introduction

Under the Curry–Howard (or formulas-as-types) correspondence, type systems for functional languages correspond to certain logics. In particular, purely functional languages give rise to an intuitionistic logic, whereas languages with first-class control operators give classical logic, as the type of the control operators is double-negation elimination (Griffin 1990) or some variant of it, like Peirce's law.

Effect systems are extensions of type systems that are more fine-grained and expressive with regard to computational effects (Lucassen & Gifford 1988). Moreover, for the special case of control effects, they extend the classical typing (Jouvelot & Gifford 1988). The explicit tracking of effectful computations in an effect system is reminiscent of the way effects are encapsulated in a monad in the approach of "monads as notions of computations" (Moggi 1989; Wadler 1998). It has also been noted that the monad type constructor T is a sort of modal operator on top of intuitionistic logic (Benton *et al.* 1998).

Thus there is a wealth of connections between these logical accounts. However, type and effect systems for realistic programming languages can become quite complex. The aim of this note is to abstract away as much complication as possible, and to present a simplified view, in the style of the classical double-negation transforms as found in proof theory texts (Troelstra & Schwichtenberg 2000).

We consider a version of classical logic in which non-intuitionistic inferences are tracked by a modality, inspired by effect systems. The modality can be eliminated under certain conditions, just as effects that stay in some sense local can be *masked* in an effect system. The intention is that the logic mediates between intuitionistic and classical logic, rather as effect systems mediate between functional and imperative programming. If we erase the modality, allowing classical inference

https://doi.org/10.1017/S0956796808006734 Published online by Cambridge University Press

$$\frac{\Gamma \vdash_{\diamond \mathbf{c}} M : (A \to B) \qquad \Gamma \vdash_{\diamond \mathbf{c}} N : A}{\Gamma \vdash_{\diamond \mathbf{c}} (MN) : B}\ (\to\mathrm{E}) \qquad \frac{\Gamma, x : A, \Gamma' \vdash_{\diamond \mathbf{c}} M : B}{\Gamma, \Gamma' \vdash_{\diamond \mathbf{c}} (\lambda x.M) : (A \to B)}\ (\to\mathrm{I})$$

$$\frac{}{\Gamma, x : A, \Gamma' \vdash_{\diamond \mathbf{c}} x : A}\ (\mathrm{Var}) \qquad \frac{\Gamma \vdash_{\diamond \mathbf{c}} M : \bot}{\Gamma \vdash_{\diamond \mathbf{c}} (\mathscr{A} M) : A}\ (\bot\mathrm{E})$$

$$\frac{\Gamma, \Gamma' \vdash_{\diamond \mathbf{c}} M : \diamond A \qquad \Gamma, x : A, \Gamma' \vdash_{\diamond \mathbf{c}} N : \diamond B}{\Gamma, \Gamma' \vdash_{\diamond \mathbf{c}} (\mathtt{let}\ x = M\ \mathtt{in}\ N) : \diamond B}\ (\mathrm{Let}) \qquad \frac{\Gamma \vdash_{\diamond \mathbf{c}} M : A}{\Gamma \vdash_{\diamond \mathbf{c}} (\mathtt{unit}\ M) : \diamond A}\ (\diamond\mathrm{Unit})$$

$$\frac{\Gamma \vdash_{\diamond \mathbf{c}} M : (A \to \diamond \bot) \to \diamond \bot}{\Gamma \vdash_{\diamond \mathbf{c}} (\mathscr{C}_\diamond\ M) : \diamond A}\ (\mathscr{C}_\diamond) \qquad \frac{\Pi \vdash_{\diamond \mathbf{c}} M : \diamond P}{\Pi \vdash_{\diamond \mathbf{c}} (\mathtt{mask}\ M) : P}\ (\diamond\mathrm{Mask})$$

Fig. 1. The modally classical logic $\diamond \mathbf{c}$.

to be untrammelled, we get classical logic (Section 3). But an entailment between pure formulas (those not containing the modality) is intuitionistically valid even if double-negation elimination and masking have been used in its derivation (Section 4).

2 A logic with a control effect modality

The inference rules for our modally classic logic are presented in Figure 1. The proof terms consist of a functional language with a control operator. This system is built up from an intuitionistic subset by adding rules for a modality $\diamond$, a modal variant of the classical axiom for double-negation elimination, and a form of effect masking. We discuss these in turn.

The modality $\diamond A$ is reminiscent of containing effects in monads in Haskell. These effects are propagated using rules from Benton, Bierman and de Paiva's CL logic (Benton *et al.* 1998), corresponding to the multiplication and unit of a monad (Moggi 1989):

$$\frac{\Gamma, \Gamma' \vdash_{\diamond \mathbf{c}} M : \diamond A \qquad \Gamma, x : A, \Gamma' \vdash_{\diamond \mathbf{c}} N : \diamond B}{\Gamma, \Gamma' \vdash_{\diamond \mathbf{c}} (\mathtt{let}\ x = M\ \mathtt{in}\ N) : \diamond B}\ (\mathrm{Let}) \qquad \frac{\Gamma \vdash_{\diamond \mathbf{c}} M : A}{\Gamma \vdash_{\diamond \mathbf{c}} (\mathtt{unit}\ M) : \diamond A}\ (\diamond\mathrm{Unit})$$

While these rules were inspired by monads rather than effect systems, they work similarly (Wadler 1998). In an effect system, arrow types are annotated with effects. For instance, $A \xrightarrow{e} B$ is the type of functions from A to B with effect e. Rather than having multiple arrows, we can decompose the effectful arrow into a pure arrow $\to$ and the effect modality $\diamond$. One of the most basic inference rules in an effect system (Lucassen & Gifford 1988) joins together the effects that may happen in an application: the operator, the operand, or the body of the function being applied may have effects, all of which are unleashed when the application is evaluated. To see how that is handled in the modal setting, suppose we want to make an inference from the following two judgements, where the '?' indicates that a modality may be present:

$$\Gamma \vdash M : ?(A \to ?B) \quad \text{and} \quad \Gamma \vdash N : ?A$$

https://doi.org/10.1017/S0956796808006734 Published online by Cambridge University Press

If there is a modality in any of the indicated positions, then there must also be a modality in front of B in the conclusion $\Gamma \vdash \ldots M \ldots N \ldots : ?B$. It is not hard to see that the rule (Let) combined with ($\to$E) conforms to this condition, as (Let) may be used to strip off instances of $\diamond$ in the premise to make ($\to$E) applicable, but requiring a $\diamond$ to appear in the conclusion.

Next, we consider the rule that is specific to our effects at hand. The double-negation elimination rule crucially interacts with the $\diamond$-modality:

$$\frac{\Gamma \vdash_{\diamond \mathbf{c}} M : (A \to \diamond \bot) \to \diamond \bot}{\Gamma \vdash_{\diamond \mathbf{c}} (\mathscr{C}_\diamond\ M) : \diamond A}\ (\mathscr{C}_\diamond)$$

This rule stems from a control effect system (Jouvelot & Gifford 1988; Thielecke 2003), simplified to fit the modal setting. To stay as close as possible to standard proof theory presentations (Troelstra & Schwichtenberg 2000), we use double-negation elimination, corresponding to a typed version of Felleisen's $\mathscr{C}$ operator (Griffin 1990; Felleisen & Hieb 1992), rather than Peirce's law $((A \to B) \to A) \to A$, corresponding to typed `call/cc`. These control operators are interdefinable under mild conditions. The modality is inserted in those places where the effect system would specify a control effect, with the combination of $\to$ and $\diamond$ emulating the effectful function type. Intuitively, the first occurrence of $\diamond$ in the premise of the rule is required because the seized continuation jumps when applied to an argument of type A, while the second occurrence is needed because the function using the continuation may apply it and cause a jump. The occurrence of $\diamond$ in the conclusion of the rule is needed because at this point the continuation is seized by the control operator.

For the masking of the modality, we will need a notion of pure formulas P, analogous to types that do not contain any free region identifiers in an effect system (Lucassen & Gifford 1988). The rule for masking control effects is then a form of $\diamond$-elimination:

$$\frac{x_1 : P_1, \ldots, x_n : P_n \vdash_{\diamond \mathbf{c}} M : \diamond P}{x_1 : P_1, \ldots, x_n : P_n \vdash_{\diamond \mathbf{c}} (\mathtt{mask}\ M) : P}\ (\diamond \text{Mask})$$

Masking of effects can be understood generically as keeping effects local, so that they are not observable from the outside via P or any of the $P_1, \ldots, P_n$. Logically, we could read $\diamond$ as a sort of non-standard modality for intuitionists who are not entirely convinced by classical inference. When double-negation elimination is used in rule ($\mathscr{C}_\diamond$), the conclusion is tagged with the modality to flag this intuitionistic doubt, which is then spread by the rule (Let). With this interpretation, the masking rule says that in some restricted cases the doubt should be dispelled even if classical inference was used locally. The main result will be that the masking rule is in fact acceptable, due to a translation to intuitionistic logic (Proposition 4.4).

More formally, we will use the following definitions:

Definition 2.1

Formulas are defined by the grammar

$$A, B ::= A \to B \mid \bot \mid \alpha \mid \diamond A$$

https://doi.org/10.1017/S0956796808006734 Published online by Cambridge University Press

where we assume infinitely many atomic formulas α. We use Γ to range over contexts consisting of sequences of the form $x_1 : A_1, \ldots, x_n : A_n$. Let a *pure* formula be one that does not contain any occurrences of the modality $\diamond$, that is,

$$P ::= P \to P \mid \bot \mid \alpha$$

Let Π range over contexts containing only pure formulas. We will use three kinds of inference, $\vdash_{\diamond\mathbf{c}}$, $\vdash_{\mathbf{i}}$ and $\vdash_{\mathbf{c}}$:

- We write $\vdash_{\diamond\mathbf{c}}$ for inference with the rules from Figure 1.
- We write $\vdash_{\mathbf{i}}$ for the usual intuitionistic inference with $\to$ introduction and elimination, axiom and falsity, given by the subset of rules ($\to$E), ($\to$I), (Var) and ($\bot$E) from Figure 1. Intuitionistic logic consists of derivable pure judgements of the form $\Pi \vdash_{\mathbf{i}} M : P$.
- We write $\vdash_{\mathbf{c}}$ for classical inference, given by the rules for $\vdash_{\mathbf{i}}$ above, together with the double-negation elimination rule (unconstrained by a modality):

$$\frac{\Gamma \vdash_{\mathbf{c}} M : (A \to \bot) \to \bot}{\Gamma \vdash_{\mathbf{c}} (\mathscr{C}\ M) : A}\ (\mathscr{C})$$

Although weakening of contexts is not present as a rule, it is obviously admissible in all these systems.

Lemma 2.2
If $\Gamma \vdash M : A$, then also $\Gamma, x : B \vdash M : A$, where $\vdash$ can be any of $\vdash_{\mathbf{i}}$, $\vdash_{\mathbf{c}}$ or $\vdash_{\diamond\mathbf{c}}$.

Example 2.3
To see how the modality works in derivations, we can derive $P \vdash_{\diamond\mathbf{c}} P$ for some pure formula P by using double-negation elimination, and then masking the resulting modality:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x : P, k : P \to \diamond\bot \vdash_{\diamond\mathbf{c}} k : P \to \diamond\bot}\ (\text{Var}) \qquad \dfrac{}{x : P, k : P \to \diamond\bot \vdash_{\diamond\mathbf{c}} x : P}\ (\text{Var})}{x : P, k : P \to \diamond\bot \vdash_{\diamond\mathbf{c}} (k\ x) : \diamond\bot}\ (\to\text{E})}{x : P \vdash_{\diamond\mathbf{c}} (\lambda k.k\ x) : (P \to \diamond\bot) \to \diamond\bot}\ (\to\text{I})}{x : P \vdash_{\diamond\mathbf{c}} (\mathscr{C}_\diamond\ (\lambda k.k\ x)) : \diamond P}\ (\mathscr{C}_\diamond)}{x : P \vdash_{\diamond\mathbf{c}} \mathtt{mask}\ (\mathscr{C}_\diamond\ (\lambda k.k\ x)) : P}\ (\diamond\,\text{Mask})$$

Computationally, this derivation amounts to seizing the current continuation and then immediately invoking it. As the continuation does not escape, the control effect can be masked. In this case, it is easy to see that we could have avoided the classical rule $\mathscr{C}_\diamond$ altogether, deriving the intuitionistic

$$x : P \vdash_{\mathbf{i}} x : P$$

Section 4 shows that such an intuitionistic derivation always exists if the effect can be masked, despite the fact that $\vdash_{\diamond\mathbf{c}}$ contains classical logic $\vdash_{\mathbf{c}}$ in a sense as shown in Section 3.

https://doi.org/10.1017/S0956796808006734 Published online by Cambridge University Press

3 Classical logic and the modality

We will show that classical inferences in $\vdash_{\mathbf{c}}$ can always be represented in $\vdash_{\diamond\mathbf{c}}$, although the formulas in the latter may need to contain instances of the modality. To translate from classical logic to the modally classical one, we systematically add the modality to all arrows.

Definition 3.1
We define a translation $(\cdot)^+$ that introduces $\diamond$ as follows:

$$\begin{aligned}(A \to B)^+ &= A^+ \to \diamond(B^+)\\ \alpha^+ &= \alpha\\ \bot^+ &= \bot\end{aligned}$$

The translation is extended to contexts pointwise. We use the same notation $(\cdot)^+$ for the corresponding translation on terms:

$$\begin{aligned}(MN)^+ &= \texttt{let}\ m = M^+\ \texttt{in}\ \texttt{let}\ n = N^+\ \texttt{in}\ (m\,n)\\ (\lambda x.M)^+ &= \texttt{unit}\ (\lambda x.(M^+))\\ x^+ &= \texttt{unit}\ x\\ (\mathscr{A}\,M)^+ &= \texttt{let}\ m = M^+\ \texttt{in}\ (\mathscr{A}\,m)\\ (\mathscr{C}\,M)^+ &= \texttt{let}\ m = M^+\ \texttt{in}\ (\mathscr{C}_\diamond\,m)\end{aligned}$$

Lemma 3.2
If $\Gamma \vdash_{\mathbf{c}} M : A$, then $\Gamma^+ \vdash_{\diamond\mathbf{c}} M^+ : \diamond A^+$.

Proof
The proof proceeds by induction over the derivation of $\Gamma \vdash_{\mathbf{c}} M : A$, using the rule (Let) to strip off $\diamond$ where necessary. Consider the rule $\mathscr{C}$. By the induction hypothesis, we have

$$\Gamma^+ \vdash_{\diamond\mathbf{c}} M^+ : \diamond((A^+ \to \diamond\bot) \to \diamond\bot)$$

We infer the required $\Gamma^+ \vdash_{\diamond\mathbf{c}} \texttt{let}\ m = M^+\ \texttt{in}\ (\mathscr{C}_\diamond\ m) : \diamond A^+$ using (Let) and this inference:

$$\dfrac{\dfrac{}{\Gamma^+, m : (A^+ \to \diamond\bot) \to \diamond\bot \vdash_{\diamond\mathbf{c}} m : (A^+ \to \diamond\bot) \to \diamond\bot}\,(\text{Var})}{\Gamma^+, m : (A^+ \to \diamond\bot) \to \diamond\bot \vdash_{\diamond\mathbf{c}} \mathscr{C}_\diamond\ m : \diamond A^+}\,(\mathscr{C}_\diamond)$$

Next, we consider the rule

$$\dfrac{\Gamma \vdash_{\mathbf{c}} M : A \to B \qquad \Gamma \vdash_{\mathbf{c}} N : A}{\Gamma \vdash_{\mathbf{c}} M\,N : B}\,(\to \text{E})$$

By the induction hypothesis, we have $\Gamma^+ \vdash_{\diamond\mathbf{c}} M^+ : \diamond(A^+ \to \diamond B^+)$ and $\Gamma^+ \vdash_{\diamond\mathbf{c}} N^+ : \diamond A^+$. By weakening (Lemma 2.2), the latter implies $\Gamma^+, m : A^+ \to \diamond B^+ \vdash_{\diamond\mathbf{c}} N^+ : \diamond A^+$. Furthermore, from two instances of (Var), we infer using $(\to$ E$)$ that

$$\Gamma^+, m : A^+ \to \diamond B^+, n : A^+ \vdash_{\diamond\mathbf{c}} m\,n : \diamond B^+$$

By applying (Let) to that and $\Gamma^+, m : A^+ \to \diamond B^+ \vdash_{\diamond\mathbf{c}} N^+ : \diamond A^+$, we have

$$\Gamma^+, m : A^+ \to \diamond B^+ \vdash_{\diamond\mathbf{c}} \texttt{let}\ n = N^+\ \texttt{in}\ m\,n : \diamond B^+$$

https://doi.org/10.1017/S0956796808006734 Published online by Cambridge University Press

Finally, by applying (Let) again, we have

$$\Gamma^+ \vdash_{\diamond \mathbf{c}} \mathtt{let}\ m = M^+\ \mathtt{in}\ \mathtt{let}\ n = N^+\ \mathtt{in}\ (m\,n) : \diamond B^+$$

as required. For the rule ($\bot$E), we again use (Let), whereas for ($\to$I) and (Var) we use ($\diamond$Unit) to insert the modality. □

If we erase the modality $\diamond$, then $\vdash_{\diamond \mathbf{c}}$ gives us just the usual presentation of classical logic with the rule for double-negation elimination.

Definition 3.3
We define a translation to formulas not containing $\diamond$ as follows:

$$\begin{aligned}(\diamond A)^\circ &= A^\circ \\ (A \to B)^\circ &= (A^\circ) \to (B^\circ) \\ \alpha^\circ &= \alpha \\ \bot^\circ &= \bot\end{aligned}$$

For contexts, Γ° is defined pointwise. Here is how we translate terms:

$$\begin{aligned}(M\,N)^\circ &= (M^\circ)\,(N^\circ) \\ (\lambda x.M)^\circ &= \lambda x.(M^\circ) \\ x^\circ &= x \\ (\mathscr{A}\,M)^\circ &= \mathscr{A}\,(M^\circ) \\ (\mathtt{let}\ x = M\ \mathtt{in}\ N)^\circ &= (\lambda x.(N^\circ))\,(M^\circ) \\ (\mathtt{unit}\ M)^\circ &= M^\circ \\ (\mathscr{C}_\diamond\ N)^\circ &= \mathscr{C}\,(M^\circ) \\ (\mathtt{mask}\ M)^\circ &= M^\circ\end{aligned}$$

Lemma 3.4
If $\Gamma \vdash_{\diamond \mathbf{c}} M : A$, then $\Gamma^\circ \vdash_{\mathbf{c}} M^\circ : A^\circ$.

Proof
The proof proceeds by induction over the derivation of $\Gamma \vdash_{\diamond \mathbf{c}} M : A$, constructing a derivation $\Gamma^\circ \vdash_{\mathbf{c}} M^\circ : A^\circ$ by cases on each derivation step. The intuitionistic rules are mapped to themselves; $\mathscr{C}_\diamond$ is taken to $\mathscr{C}$, while instances of ($\diamond$Mask) and ($\diamond$Unit) are omitted from the derivation in $\vdash_{\mathbf{c}}$. An instance of the rule (Let) is taken to the following derivation steps:

$$\frac{\dfrac{\Gamma^\circ, x : A^\circ \vdash_{\mathbf{c}} N^\circ : B^\circ}{\Gamma^\circ \vdash_{\mathbf{c}} (\lambda x.N)^\circ : A^\circ \to B^\circ}\,(\to\mathrm{I}) \qquad \Gamma^\circ \vdash_{\mathbf{c}} M^\circ : A^\circ}{\Gamma^\circ \vdash_{\mathbf{c}} (\lambda x.N^\circ)\,M^\circ : B^\circ}\,(\to\mathrm{E})$$

This idiom is familiar as the encoding of let-bindings in terms of lambda abstraction and application. □

Proposition 3.5
We have $\Gamma \vdash_{\mathbf{c}} M : A$ for some term M if and only if $\Gamma^+ \vdash_{\diamond \mathbf{c}} N : \diamond A^+$ for some term N.

https://doi.org/10.1017/S0956796808006734 Published online by Cambridge University Press

Proof
The 'only if' direction is Lemma 3.2 with $N = M^+$. For the converse, suppose $\Gamma^+ \vdash_{\diamond\mathbf{c}} N : \diamond A^+$. By Lemma 3.4, this implies $(\Gamma^+)^\circ \vdash_{\mathbf{c}} N^\circ : (\diamond A^+)^\circ$. As $(B^+)^\circ = B$ for any classical formula B, we have $\Gamma \vdash_{\mathbf{c}} N^\circ : A$. □

Proposition 3.5 means that $\vdash_{\diamond\mathbf{c}}$ accommodates classical logic as the image of $(\cdot)^+$, the subset in which $\to$ and $\diamond$ are always combined, as in $A \to \diamond B$. However, $\vdash_{\diamond\mathbf{c}}$ is more finegrained in that it can distinguish between $A \to \diamond B$ and $A \to B$, just as an effect system differentiates between functions that may or may not have effects. The ability to make this distinction enables $\vdash_{\diamond\mathbf{c}}$ to contain intuitionistic logic as well, without classical inference leaking into intuitionistic ones uncontrollably.

4 CPS transform and conservativity

To relate $\vdash_{\diamond\mathbf{c}}$ to intuitionistic logic, we define a double-negation transform. Unlike the standard transforms, it uses the modality $\diamond$ to decide where to insert double negations.

Definition 4.1
Let α_0 be some designated atomic formula not used elsewhere in the translation of judgements. We define a transformation $\overline{(\cdot)}$ from $\vdash_{\diamond\mathbf{c}}$ to intuitionistic logic as follows:

$$\begin{aligned}
\overline{A \to B} &= \overline{A} \to \overline{B} \\
\overline{\diamond A} &= (\overline{A} \to \alpha_0) \to \alpha_0 \\
\overline{\bot} &= \bot \\
\overline{\alpha} &= \alpha
\end{aligned}$$

The transformation is extended to contexts pointwise. The corresponding translation on terms is as follows:

$$\begin{aligned}
\overline{M\,N} &= \overline{M}\ \overline{N} \\
\overline{\lambda x.M} &= \lambda x.\overline{M} \\
\overline{x} &= x \\
\overline{\mathscr{A}\,M} &= \mathscr{A}\,\overline{M} \\
\overline{\mathtt{let}\ x = M\ \mathtt{in}\ N} &= \lambda k.(\overline{M}\,(\lambda x.\overline{N}k)) \\
\overline{\mathtt{unit}\ M} &= \lambda k.(k\,\overline{M}) \\
\overline{\mathscr{C}_\diamond\ M} &= \lambda k.\overline{M}\,(\lambda x.\lambda h.(k\,x))\,(\lambda y.\mathscr{A}\,y) \\
\overline{\mathtt{mask}\ M} &= \overline{M}\,(\lambda x.x)
\end{aligned}$$

In continuation terms, α_0 is the answer type of our transform. The significance of choosing α_0 to be fresh is that we can therefore substitute other formulas for it without affecting the surrounding formula. To do so, the following lemma will be needed, where the substitution of α by B in A is written as $A[B/\alpha]$.

Lemma 4.2
If $\Gamma \vdash_{\mathbf{i}} M : A$, then also $\Gamma[B/\alpha] \vdash_{\mathbf{i}} M : A[B/\alpha]$.

https://doi.org/10.1017/S0956796808006734 Published online by Cambridge University Press

We can now show that $\overline{(\cdot)}$ is a translation from our modally classical logic to intuitionistic logic, in the sense that judgements are preserved.

Lemma 4.3
If $\Gamma \vdash_{\diamond \mathbf{c}} M : A$, then $\overline{\Gamma} \vdash_{\mathbf{i}} \overline{M} : \overline{A}$.

Proof
The proof proceeds by induction over the derivation of $\Gamma \vdash_{\diamond \mathbf{c}} M : A$. The most important case is the rule

$$\frac{\Pi \vdash_{\diamond \mathbf{c}} M : \diamond P}{\Pi \vdash_{\diamond \mathbf{c}} \mathtt{mask}\ M : P}\ (\diamond \mathrm{Mask})$$

for masking. The idea is the same as the insertion of a control delimiter in a CPS transform whenever a control effect is masked (Thielecke 2003). By the induction hypothesis, we have

$$\overline{\Pi} \vdash_{\mathbf{i}} \overline{M} : (\overline{P} \to \alpha_0) \to \alpha_0$$

Hence by Lemma 4.2, we also have

$$\overline{\Pi}[\overline{P}/\alpha_0] \vdash_{\mathbf{i}} \overline{M} : ((\overline{P} \to \alpha_0) \to \alpha_0)[\overline{P}/\alpha_0]$$

Now, pure formulas are left invariant by the translation and in particular do not contain α_0. Thus $\overline{\Pi} \vdash_{\mathbf{i}} \overline{M} : (\overline{P} \to \overline{P}) \to \overline{P}$. From this and $\overline{\Pi} \vdash_{\mathbf{i}} (\lambda x.x) : \overline{P} \to \overline{P}$, we infer $\overline{\Pi} \vdash_{\mathbf{i}} \overline{M}\,(\lambda x.x) : \overline{P}$, as required.

Of the remaining cases, the most interesting is the modal double-negation elimination ($\mathscr{C}_\diamond$), as it is analogous to typechecking the call-by-value CPS transform of a control operator, using Lemma 2.2 to propagate the current continuation. So to check this case, let B abbreviate the following formula from the translation of the premise of rule ($\mathscr{C}_\diamond$):

$$\begin{aligned} B &= \overline{(A \to \diamond \bot) \to \diamond \bot} \\ &= (\overline{A} \to ((\bot \to \alpha_0) \to \alpha_0)) \to ((\bot \to \alpha_0) \to \alpha_0) \end{aligned}$$

We need to show that $\overline{\Gamma} \vdash_{\mathbf{i}} \overline{M} : B$ implies $\overline{\Gamma} \vdash_{\mathbf{i}} \overline{\mathscr{C}_\diamond\ M} : (\overline{A} \to \alpha_0) \to \alpha_0$. To do so, we first note that

$$\overline{\Gamma}, k : \overline{A} \to \alpha_0 \vdash_{\mathbf{i}} \lambda x.\lambda h.(k\,x) : \overline{A} \to (\bot \to \alpha_0) \to \alpha_0$$

As we have $\overline{\Gamma}, \overline{A} \to \alpha_0 \vdash_{\mathbf{i}} B$ by Lemma 2.2, we infer from that by ($\to$E),

$$\overline{\Gamma}, k : \overline{A} \to \alpha_0 \vdash_{\mathbf{i}} \overline{M}\,(\lambda x.\lambda h.(k\,x)) : (\bot \to \alpha_0) \to \alpha_0$$

Now, we also have by ($\bot$E) and ($\to$I),

$$\overline{\Gamma}, k : \overline{A} \to \alpha_0 \vdash_{\mathbf{i}} (\lambda y.\mathscr{A}\,y) : \bot \to \alpha_0$$

and so by ($\to$E),

$$\overline{\Gamma}, k : \overline{A} \to \alpha_0 \vdash_{\mathbf{i}} \overline{M}\,(\lambda x.\lambda h.(k\,x))\,(\lambda y.\mathscr{A}\,y) : \alpha_0$$

and finally by ($\to$I),

$$\overline{\Gamma} \vdash_{\mathbf{i}} \lambda k.\overline{M}\,(\lambda x.\lambda h.(k\,x))\,(\lambda y.\mathscr{A}\,y) : (\overline{A} \to \alpha_0) \to \alpha_0$$

https://doi.org/10.1017/S0956796808006734 Published online by Cambridge University Press

as required. In continuation-passing terms, $k : \overline{A} \to \alpha_0$ is the current continuation that is seized by the control operator, while $h : \bot \to \alpha_0$ is a continuation that is ignored. □

As is typical for double-negation translations, Lemma 4.3 tells us that we can avoid classical rules at the expense of additional arrows in formulas, just as we can avoid control operators in functional programming by rewriting the code to use explicit continuation passing instead.

Our main result is that the logic $\vdash_{\diamond \mathbf{c}}$ is conservative (Troelstra & Schwichtenberg 2000) over intuitionistic logic.

Proposition 4.4
If $\Pi \vdash_{\diamond \mathbf{c}} M : P$, then $\Pi \vdash_{\mathbf{i}} \overline{M} : P$.

Proof
The result follows from Lemma 4.3, since for pure P, we have $\overline{P} = P$. □

The conservativity tells us that in judgements only involving pure formulas, we can always avoid the extra rules of $\vdash_{\diamond \mathbf{c}}$ by rewriting the derivation into an intuitionistic one for the same judgement.

5 Conclusions

The logic $\vdash_{\diamond \mathbf{c}}$ is intended as an extremely distilled form of control effect system, including a variant of effect masking. Real effect systems are much more sophisticated, including effect and region polymorphism; on the other hand, they may also seem ad hoc and complicated, and effect masking is often explained in terms of implementation-specific techniques, such as stack allocation. By contrast, Proposition 4.4 aims to look at effect annotations in a more traditional logical setting, as a conservative extension.

This conservativity of effects at the level of types complements Felleisen's notion of expressiveness Felleisen (1991), where it is characteristic of effects that they are *not* conservative extensions in terms of observational equivalence, which is taken as evidence of an increase in expressive power. As a direction for further research, one could aim to reconcile the conservativity and the non-conservativity: it appears that the breaking of contextual equivalences is due to those effects that cannot be masked. More specifically, recall that in Example 2.3, we masked a term whose control effects were not externally visible:

$$x : P \vdash_{\diamond \mathbf{c}} \mathtt{mask}\ (\mathscr{C}_{\diamond}\ (\lambda k.k\, x)) : P$$

It is interesting to note that using the equational axioms for the $\mathscr{C}$ operator Felleisen & Hieb (1992), one can infer $\mathscr{C}\,(\lambda k.k\, x) = x$, and clearly the pure term x witnesses the entailment $x : P \vdash_{\mathbf{i}} x : P$. This situation suggests that terms with masked effects could be re-written into equivalent ones without effects, so that they could not in fact break any equivalences that hold only in the absence of effects. However, proving such a result on terms appears much more technically involved than the simplified

https://doi.org/10.1017/S0956796808006734 Published online by Cambridge University Press

account at the level of types given here. Equational reasoning based on answer type parametricity, as used for the equivalence of two CPS transforms (Thielecke 2004), may be a promising technique. It is also possible that analogous conservativity over intuitionistic logic holds for other effects, such as state with assignment, although in that situation we would no longer have the Curry–Howard correspondence to classical logic that is specific to control effects.

References

Benton, P. N., Bierman, G. M., & de Paiva, V. (1998) Computational types from a logical perspective. *J. Functional Programming* **8**(2), 177–193.

Felleisen, M. (1991) On the expressive power of programming languages. *Sci. Comput. Program*, **17**(1–3), 35–75.

Felleisen, M., & Hieb, R. (1992) The revised report on the syntactic theories of sequential control and state. *Theor. Comp. Sci.* **103**(2), 235–271.

Griffin, T. G. (1990) A formulae-as-types notion of control. In *Principles of Programming Languages (POPL)*. ACM, New York, NY, USA, pp. 47–58.

Jouvelot, P., & Gifford, D. K. (1988) Reasoning about continuations with control effects. In *Programming Language Design and Implementation (PLDI)*. ACM, New York, NY, USA, pp. 218–226.

Lucassen, J. M., & Gifford, D. K. (1988) Polymorphic effect systems. In *Principles of Programming Languages (POPL)*. ACM, New York, NY, USA, pp. 47–57.

Moggi, E. (1989) Computational lambda calculus and monads. In *Logic in Computer Science (LICS)*. IEEE, Pacific Grove, CA, USA, pp. 14–23.

Thielecke, H. (2003) From control effects to typed continuation passing. In *Principles of Programming Languages (POPL)*. ACM, New York, NY, USA, pp. 139–149.

Thielecke, H. (2004) Answer type polymorphism in call-by-name continuation passing. In *European Symposium on Programming (ESOP)*. LNCS, vol. 2986. Springer, Berlin/Heidelberg, pp. 279–293.

Troelstra, A. S., & Schwichtenberg, H. (2000) *Basic Proof Theory*. Cambridge University Press, New York, NY, USA.

Wadler, P. (1998) The marriage of effects and monads. In *International Conference on Functional Programming (ICFP)*. ACM, New York, NY, USA, pp. 63–74.

https://doi.org/10.1017/S0956796808006734 Published online by Cambridge University Press