On the Logical Content of Computational Type Theory: A Solution to Curry's Problem*

Matt Fairtlough and Michael Mendler

Department of Computer Science, Sheffield University
211 Portobello Street, Sheffield S1 4DP, UK

Abstract. In this paper we relate the lax modality $\bigcirc$ to Intuitionistic Propositional Logic (IPL) and give a complete characterisation of inhabitation in Computational Type Theory (CTT) as a logic of constraint contexts. This solves a problem open since the 1940's, when Curry was the first to suggest a formal syntactic interpretation of $\bigcirc$ in terms of contexts.

1 Introduction

Recently, modal extensions to type theory have received a lot of attention as a natural enrichment suggested by a variety of typing problems that occur in programming and encodings of logic in type-theoretical frameworks.

Modalities may be used for enriching intuitionistic type theories, which traditionally focus on pure function and data, so as to include also various forms of reactive program features. In type theories for functional programming, for instance, modalities are added to accommodate non-functional ("impure") semantic features such as non-termination, side-effects or different operational modes [21, 20]. In applications to strictness analysis [3] and program optimisation [12, 6] modalities have been used to integrate static and dynamic types, while in other applications they provide a rigorous interface between inductive and non-inductive data-types, for instance in type systems to internalise higher-order abstract syntax [8, 7]. In strong functional programming [19] modalities occur (implicitly) as least or greatest fixpoint operators constructing data and codata types. Modalities have also proved useful in studying the relationship between second order encodings of logic in type theory [1].

In this paper we take a fresh look at one of the oldest modal extensions of type theory which arose with Moggi's influential work on computational monads [18]. This system, which we call *Computational Type Theory* (CTT), is an extension of simple type theory that features a single modality $\bigcirc$ satisfying the axioms

$$\begin{aligned} \bigcirc I &: \varphi \supset \bigcirc\varphi \\ \bigcirc M &: \bigcirc\bigcirc\varphi \supset \bigcirc\varphi \\ \bigcirc S &: (\bigcirc\varphi \wedge \bigcirc\psi) \supset \bigcirc(\varphi \wedge \psi) \end{aligned}$$

* This work is supported by EPSRC grants GR/L86180 and GR/M99637, by the EU Types Working Group IST-EU-29001 and by the British Council.

M. Fairtlough, M. Mendler: On the Logical Content of Computational Type Theory: A Solution to Curry's Problem.
In P. Callaghan, Z. Luo, J. McKinna (eds.), Types for Proofs and Programs, Springer 2002 (LNCS 2277), pp. 63-78.

together with the rule of Modus Ponens as well as the Extensionality Rule *Ext*:

$$\vdash \varphi \supset \psi \;\Rightarrow \vdash \bigcirc\varphi \supset \bigcirc\psi.$$

CTT can be given an alternative formulation as an axiomatic extension of IPL by $\bigcirc I + \bigcirc L : (\varphi \supset \bigcirc\psi) \supset (\bigcirc\varphi \supset \bigcirc\psi)$, which is extensionally equivalent to CTT in the sense that CTT $\vdash \varphi$ iff IPC $+ \bigcirc I + \bigcirc L \vdash \varphi$. We prefer our presentation of CTT because it fits better with other algebraic and categorical treatments of the modality $\bigcirc$. This point is taken up in the concluding remarks to this paper. The logic of inhabitation of CTT is also known as *Propositional Lax Logic* (PLL) [9, 10] or *Computational Logic* [2]. Our analysis illuminates the characteristic rôle and the duality between two particular computational monads: the state readers monad $\bigcirc\varphi \equiv K \supset \varphi$ and the exceptions monad $\bigcirc\varphi \equiv \varphi \vee L$. Specifically, we define a notion of *standard context* as a context $C[\cdot]$ where $C[\varphi]$ has the form $\bigwedge_{i\in I} K_i \supset (\varphi \vee L_i)$ with K_i and L_i arbitrary formulas of IPL. We then show that the set $\mathbb{S}$ of standard contexts forms a Boolean algebra. Each such context provides a sound interpretation of CTT under the correspondence $\varphi \mapsto \varphi^C$, where φ^C is the formula φ with every subformula $\bigcirc\psi$ replaced[1] by $C[\psi]$. The contexts $K \supset \cdot$ and $\cdot \vee L$ have a simple computational interpretation in type theory: $K \supset \varphi$ represents the *state reader lifting* of type φ. Terms of $K \supset \varphi$ may be thought of as elements of type φ which depend on an additional state variable of type K which can be read from but not written to. The weakening $\varphi \vee L$ corresponds to an *exception lifting* of type φ. Terms of type $\varphi \vee L$ either denote proper elements of type φ or raise an exception of type L.

Our main theorem is that standard contexts are sound and complete for CTT, that is, CTT $\vdash \varphi$ iff for any standard context C, IPL $\vdash \varphi^C$. We go on to show that no finite set of standard contexts is complete for CTT. These results answer a question raised by Curry, as we shall shortly show.

Our paper concentrates on the logic of inhabitation of CTT, namely PLL. Of course, type theory is not simply concerned with whether or not types are inhabited. For any concrete interpretation of $\bigcirc$ in CTT, there must be λ-terms corresponding to the axioms and rules. Before proceeding further, we therefore present a set of such terms for every standard context C. For the "state readers plus exceptions" interpretation $C[\varphi] = K \supset (\varphi \vee L)$ we have $(\varphi \supset \bigcirc\varphi)^C = A \supset (K \supset (A \vee L))$ with $A = \varphi^C$, and this type is inhabited by $C_I =_{df} \lambda a.\, \lambda k.\, \iota_1(a)$. Next we have $(\bigcirc\bigcirc\varphi \supset \bigcirc\varphi)^C = (K \supset ((K \supset (A \vee L)) \vee L)) \supset (K \supset (A \vee L))$, and this type is inhabited *inter alia* by

$$C_M =_{df} \lambda c.\, \lambda k.\, \mathsf{case}\ c\ k\ \mathsf{of}\ [\iota_1(e) \to \begin{pmatrix} \mathsf{case}\ e\ k\ \ \mathsf{of} \\ [\iota_1(a) \to \iota_1(a), \\ \ \iota_2(l_1) \to \iota_2(l_1)] \end{pmatrix},\ \iota_2(l_2) \to \iota_2(l_2)]\ .$$

Now let B be ψ^C. For the axiom $\bigcirc S$, $((\bigcirc\varphi \wedge \bigcirc\psi) \supset \bigcirc(\varphi \wedge \psi))^C$ is the type $((K \supset (A \vee L)) \wedge (K \supset (A \vee L))) \supset K \supset ((A \wedge B) \vee L)$ which is inhabited *inter*

[1] Note that *all* occurrences of $\bigcirc$ must be replaced by the *same* constraint context. It would not be sound to permit independent expansion of different $\bigcirc$ occurrences by different contexts. Consider the trivial theorem $\bigcirc\varphi \supset \bigcirc\varphi$. Replacing the first $\bigcirc\varphi$ by $K \supset \varphi$ and the second by $\varphi \vee L$ does not give a theorem of IPL.

M. Fairtlough, M. Mendler: On the Logical Content of Computational Type Theory: A Solution to Curry's Problem.
In P. Callaghan, Z. Luo, J. McKinna (eds.), Types for Proofs and Programs, Springer 2002 (LNCS 2277), pp. 63-78.

alia by

$$C_S =_{df} \lambda(c_1, c_2).\, \lambda k.\, \mathsf{case}\ c_1\ k\ \mathsf{of}\ \left[\iota_1(a) \to \begin{pmatrix} \mathsf{case}\ c_2\ k\ \ \mathsf{of} \\ [\iota_1(b) \to \iota_1(a,b), \\ \iota_2(l_1) \to \iota_2(l_1)] \end{pmatrix},\ \iota_2(l_2) \to \iota_2(l_2)\right].$$

Finally, if $f : A \supset B$ then

$$C_{Ext}(f) =_{df} \lambda c.\, \lambda k.\, \mathsf{case}\ c\ k\ \mathsf{of}\ [\iota_1(a) \to \iota_1(f\ a),\ \iota_2(l) \to \iota_2(l)]$$

inhabits $(\bigcirc A \supset \bigcirc B)^C = (K \supset (A \vee L)) \supset (K \supset (B \vee L))$. In fact, as PLL has the deduction property, the term $\lambda f.\, C_{Ext}(f)$ inhabits $((A \supset B) \supset (\bigcirc A \supset \bigcirc B))^C$. Suppose we have a suitable set of λ-terms for C_1 and C_2. We now wish to find λ-terms for the conjunction of constraints $C_1 \sqcap C_2$, defined by $(C_1 \sqcap C_2)[\varphi] = C_1[\varphi] \wedge C_2[\varphi]$. Suppose therefore that for $j = 1, 2$ and $f : A \supset B$ we have terms

$$\begin{aligned} C_{j\,I} &: A \supset C_j[A]\,, \\ C_{j\,M} &: C_j[C_j[A]] \supset C_j[A]\,, \\ C_{j\,S} &: C_j[A] \wedge C_j[B] \supset C_j[A \wedge B]\,, \\ C_{j\,Ext}(f) &: C_j[A] \supset C_j[B]\ . \end{aligned}$$

Then we may define the following λ-terms

$$\begin{aligned} (C_1 \sqcap C_2)_I &=_{df} \lambda a.\, (C_{1I}(a),\, C_{2I}(a))\,, \\ (C_1 \sqcap C_2)_M &=_{df} \lambda(c_1,\, c_2).(C_{1M}(C_{1F}(\pi_1)\ c_1),\, C_{2M}(C_{2F}(\pi_2)\ c_2))\,, \\ (C_1 \sqcap C_2)_S &=_{df} \lambda((c_{11},\, c_{21}),\, (c_{12},\, c_{22})).(C_{1S}(c_{11},\, c_{12}),\, (C_{2S}(c_{21},\, c_{22})))\,, \\ (C_1 \sqcap C_2)_{Ext}(f) &=_{df} \lambda(c_1,\, c_2).(C_{1Ext}(f)\ c_1,\, C_{2Ext}(f)\ c_2)\,, \end{aligned}$$

and assign types to them as follows:

$$\begin{aligned} (C_1 \sqcap C_2)_I &: A \supset (C_1 \sqcap C_2)[A] = A \supset C_1[A] \wedge C_2[A]\,, \\ (C_1 \sqcap C_2)_M &: (C_1 \sqcap C_2)^2[A] \supset (C_1 \sqcap C_2)[A]\,, \\ (C_1 \sqcap C_2)_S &: (C_1 \sqcap C_2)[A] \sqcap (C_1 \sqcap C_2)[B] \supset (C_1 \sqcap C_2)[A \wedge B]\,, \\ (C_1 \sqcap C_2)_{Ext}(f) &: (C_1 \sqcap C_2)[A] \supset (C_1 \sqcap C_2)[B]\ . \end{aligned}$$

In the above we assume the following syntax for typed λ-terms:

$$\frac{\Gamma \vdash p : A \qquad \Gamma \vdash q : B}{\Gamma \vdash (p, q) : A \wedge B} \qquad \frac{\Gamma \vdash r : A \wedge B}{\Gamma \vdash \pi_1(r) : A} \qquad \frac{\Gamma \vdash r : A \wedge B}{\Gamma \vdash \pi_2(r) : B}$$

$$\frac{\Gamma \vdash r : A \vee B \qquad \Gamma, y : A \vdash p : C \qquad \Gamma, z : B \vdash q : C}{\Gamma \vdash \mathsf{case}\ r\ \mathsf{of}\ [\iota_1(y) \to p,\ \iota_2(z) \to q] : C}$$

$$\frac{\Gamma \vdash p : A}{\Gamma \vdash \iota_1(p) : A \vee B} \qquad \frac{\Gamma \vdash q : B}{\Gamma \vdash \iota_2(q) : A \vee B}$$

$$\frac{\Gamma, z : A \vdash : B}{\Gamma \vdash \lambda z\,.\, p : A \supset B} \qquad \frac{\Gamma \vdash p : A \supset B \qquad \Gamma \vdash q : A}{\Gamma \vdash p\, q : B}$$

The interpretation of the results of this paper with respect to these λ-terms is left as future work.

M. Fairtlough, M. Mendler: On the Logical Content of Computational Type Theory: A Solution to Curry's Problem.
In P. Callaghan, Z. Luo, J. McKinna (eds.), Types for Proofs and Programs, Springer 2002 (LNCS 2277), pp. 63-78.

2 Curry's Problem

In [16, 17, 9] we proposed to read $\bigcirc\varphi$ as a weakened notion of validity, *viz.* "φ up to constraints." This constraint interpretation is related to an idea originally suggested by Curry in his 1948 Notre Dame Lectures on a *Theory of Formal Deducibility* reprinted as [5]. Indeed, Curry was probably the first to study the constructive modality $\bigcirc$ (see also [4]) and to suggest a formal interpretation inside IPL in terms of constraints and hidden assumptions.

Curry's proposal was to take $\bigcirc\varphi$ as the statement "*in some outer (stronger) theory, φ holds.*" As examples of such nested systems of reasoning (with two levels) he suggested Mathematics as the inner and Physics as the outer system, or Physics as the inner system and Biology as the outer. In both examples the outer system is more encompassing than the inner system where reasoning follows a more rigid notion of truth and deduction. The modality $\bigcirc$, which Curry conceived of as a modality of possibility, is a way of reflecting the relaxed, outer, notion of truth within the inner system.

Assuming that the outer theory can be axiomatised by some, possibly very complex, formula K inside the inner system the formal semantics of $\bigcirc$ would come down to

$$\bigcirc_K\varphi = \text{“}\varphi \textit{ under the assumption } K\text{”} = K \supset \varphi.$$

It is evident that this interpretation within IPL provably satisfies the axioms of $\bigcirc$, hence provides a sound semantics of PLL. In fact, $\bigcirc_K\varphi$ is a particular constraint interpretation in the sense of [17], with fixed implicational constraint K. However, Curry's guess [4, § 5,p.261] that the interpretation $\bigcirc_K\varphi$ generates the theory PLL is unjustified: $\bigcirc_K\varphi$ validates $(\bigcirc_K\varphi \supset \bigcirc_K\psi) \supset \bigcirc_K(\varphi \supset \psi)$ which is not a theorem of PLL (see [9]).

So, is Curry's idea ill-conceived? Let us follow the constraint paradigm a bit further. Surely, implicational contexts are not the only way to weaken a proposition by constraints. Another, dual, way for doing this are the *disjunctive constraint contexts* $\bigcirc^L\varphi = \varphi \vee L$. Again, it is not difficult to show that this interpretation provably satisfies the axioms of $\bigcirc$. The contexts $K \supset \cdot$ and $\cdot \vee L$ are dual in the following semantical sense: Let $M(\psi)$, for proposition ψ and Kripke model M, denote the set of worlds of M where ψ is true. Then $M \models K \supset \varphi$ iff $M(K) \subseteq M(\varphi)$, while $M \models \varphi \vee L$ iff $\overline{M(L)} \subseteq M(\varphi)$, where $\overline{M(L)}$ is the complement of $M(L)$. Intuitively speaking, the difference is that the weakening of φ by $K \supset \varphi$ is obtained by "switching φ on" only in those worlds where the constraint K is *true*, while in $\varphi \vee L$ the weakening is obtained by switching φ on where L is *false*.

Let us call any syntactic context $C[\cdot]$ for which the axioms and rule for $\bigcirc$ are provable in IPL a *constraint context* or simply a *constraint*. Let the set of constraint contexts be $\mathbb{C}$. Now if $\bigcirc$ is to be the modality of truth under constraints then there should be a natural collection of constraints that completely characterise PLL. A suitable refinement of the original problem posed by Curry, then, is this:

M. Fairtlough, M. Mendler: On the Logical Content of Computational Type Theory: A Solution to Curry's Problem.
In P. Callaghan, Z. Luo, J. McKinna (eds.), Types for Proofs and Programs, Springer 2002 (LNCS 2277), pp. 63-78.

Does there exist a class $\mathbb{S}$ *of constraint contexts such that* $\mathsf{PLL} \vdash \varphi$ *iff* $\forall C \in \mathbb{S}.\, \mathsf{IPL} \vdash \varphi^C$ *?*

To see that this question is non-trivial we first observe that neither one of the simple constraints $K \supset \cdot$ or $\cdot \vee L$ is sufficient as a semantics for PLL. We already saw that Curry's implicational contexts $K \supset \cdot$ validate the scheme $(\bigcirc\varphi \supset \bigcirc\psi) \supset \bigcirc(\varphi \supset \psi)$, which is not a theorem of PLL. The disjunctive contexts $\cdot \vee L$ on the other hand give rise to the theorem $\bigcirc(\varphi \vee \psi) \supset \bigcirc\varphi \vee \bigcirc\psi$, which is not part of PLL either (see [9]). Similarly, it is not enough to take for $\mathbb{S}$ the collection of implicational and disjunctive constraints, since then the disjunction

$$\bigcirc(\varphi \vee \psi) \supset \bigcirc\varphi \vee \bigcirc\psi \vee (\bigcirc\varphi \supset \bigcirc\psi) \supset \bigcirc(\varphi \supset \psi)$$

would be validated. For whatever context we choose, $\bigcirc[\cdot] = K \supset \cdot$ or $\bigcirc[\cdot] = \cdot \vee L$, one of the two disjuncts would be provable, and thus the disjunction itself. Yet the disjunction is not a theorem of PLL since neither disjunct is and PLL satisfies the disjunction property (see [9]).

3 An Algebra of Constraint Contexts

The solution lies in considering combinations of implicational and disjunctive contexts. For instance, we can combine both in the contexts

$$[K, L)\, \varphi =_{df} K \supset (\varphi \vee L),$$

called *basic constraints*, which also provide a sound interpretation of $\bigcirc$. The interval notation $[K, L)$ is suggested by the fact that the constraint weakening $K \supset (\varphi \vee L)$ switches φ on in the "interval" of worlds between K (inclusive) and L (exclusive) as illustrated in Fig. 1. The shaded area, indicating this interval, measures the extent to which φ needs to be true in the model to validate $[K, L)\, \varphi$.

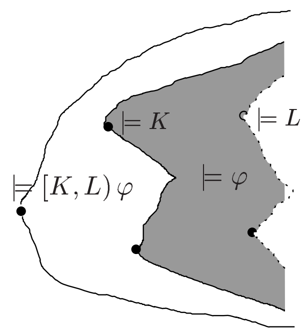


Fig. 1. Interval Constraint $[K, L)$

It is not difficult to see that basic constraints by themselves are still not quite enough to characterise PLL. They are a model of $\bigcirc(\bigcirc(\varphi \vee \psi) \supset (\bigcirc\varphi \vee \bigcirc\psi))$, which is not a theorem of PLL. However, so it turns out, finite conjunctions of basic constraints do the job. Specifically, if C_1 and C_2 are constraints, then

$$(C_1 \sqcap C_2)[x] =_{df} C_1[x] \wedge C_2[x]$$

is a constraint, too. A finite conjunction of basic constraints $\prod_{i=0}^{n-1} [K_i, L_i)$ is called a *standard constraint*. We refer to n as the *depth* of the constraint. It

M. Fairtlough, M. Mendler: On the Logical Content of Computational Type Theory: A Solution to Curry's Problem.
In P. Callaghan, Z. Luo, J. McKinna (eds.), Types for Proofs and Programs, Springer 2002 (LNCS 2277), pp. 63-78.

will be expedient to include the degenerate case $n = 0$, and put $\bigsqcap_{i=0}^{0-1} C_i =_{df}$ $[\mathit{false}, \mathit{false})$. It will turn out that the collection of standard constraints of arbitrary depth, henceforth called $\mathbb{S}$, is a suitable class of contexts to characterise PLL. But before we go into the details of the proof it is worthwhile to study some of the algebraic properties of $\mathbb{S}$.

We take constraints up to equivalence, *i.e.* consider two constraints identical if they have the same action on all propositions. Formally, $C_1 = C_2$ iff $\forall x.\ C_1[x] \equiv C_2[x]$. We shall now show that $\mathbb{S}$ forms an infinite Boolean algebra. To stress the algebraic standpoint let us write $x \leq y$ to indicate that $x \supset y$ is a theorem of IPL, while for constraints we take $C \leq D$ to abbreviate the statement that $C[x] \leq D[x]$ for all propositions x. Since $\mathbb{S}$ is a Boolean algebra one could equally well adopt a dual view swapping $\leq$ with $\geq$ and $\sqcap$ with $\sqcup$.

We begin by mentioning a few obvious facts about the algebra $\mathbb{S}$. The bottom element is the constraint $\bot =_{df} [\mathit{true}, \mathit{false})$, *i.e.* the identity modality acting as $\bot[x] = \mathit{true} \supset (x \vee \mathit{false}) \equiv x$. The top element is $\top =_{df} [\mathit{false}, \mathit{false})$, *i.e.* the modality $\top[x] = \mathit{false} \supset (x \vee \mathit{false}) \equiv \mathit{true}$ that forces everything true. As defined above $\top$ is the (unique) standard constraint of depth 0. Note that $\top = [K, K)$ for arbitrary propositions K. Top and bottom elements satisfy $C \sqcap \top = C = \top \sqcap C$ and $C \sqcap \bot = \bot = \bot \sqcap C$. Since $\sqcap$ is essentially conjunction $\wedge$ it is commutative, associative, idempotent, and satisfies $C \leq D$ iff $C \sqcap D = C$. Thus it is the real meet of the algebra.

Generally, a formula L may be identified with the disjunctive constraint $L =_{df} [\mathit{true}, L)$ so that $L[x] = \mathit{true} \supset (x \vee L) \equiv x \vee L$. Its complement is the dual implicational constraint $\overline{L} =_{df} [L, \mathit{false})$, so that $\overline{L}[x] = L \supset (x \vee \mathit{false}) \equiv L \supset x$. We call these (positive and negative) *atomic* constraints. With this lifting of formulas to constraints being understood, we have $\bot = \mathit{false}$ and $\top = \mathit{true}$. One also easily verifies that $\bot = \overline{\mathit{true}}$, $\top = \overline{\mathit{false}}$, and $L \sqcap \overline{L} = \bot$. Indeed as we will see, constraints L and their complements $\overline{L}$ are the generators of $\mathbb{S}$.

The join $C_1 \sqcup C_2$ in $\mathbb{S}$ is given by the following definition.

Definition 1. *Let* $C_1 = \bigsqcap_{i=0}^{m-1} [K_{1i}, L_{1i})$ *and* $C_2 = \bigsqcap_{j=0}^{n-1} [K_{2j}, L_{2j})$ *be standard constraints of depths* m *and* n*, respectively. Then,* $C_1 \sqcup C_2$ *is the standard constraint* $\bigsqcap_{i<m, j<n} [K_{1i} \wedge K_{2j}, L_{1i} \vee L_{2j})$.

Definition 1 gives an explicit representation of $C_1 \sqcup C_2$. For basic constraints we get $[K_1, L_1) \sqcup [K_2, L_2) = [K_1 \wedge K_2, L_1 \vee L_2)$, in particular $\overline{K} \sqcup L = [K, \mathit{false}) \sqcup [\mathit{true}, L) = [K, L)$. In other words, basic constraints are conjunctions of atomic constraints. If atomic constraints are the literals of our algebra then basic constraints are the minterms. Generally, we have

$$\overline{K_1} \sqcup \cdots \sqcup \overline{K_m} \sqcup L_1 \sqcup \cdots \sqcup L_n = [K_1 \wedge \cdots \wedge K_m, L_1 \vee \cdots \vee L_n).$$

Next, let us consider the algebraic properties of $\sqcup$. From the definition of $\sqcup$ we easily get $C \sqcup \bot = C = \bot \sqcup C$ and $C \sqcup \top = \top = \top \sqcup C$ as well as the associativity and commutativity of $\sqcup$. Idempotency $C \sqcup C = C$ and the inequations $C_1 \leq C_1 \sqcup C_2$ and $C_2 \leq C_1 \sqcup C_2$ require a little more work but are straightforward. We show $C_1 \leq C_1 \sqcup C_2$. Suppose as before that $C_1 = \bigsqcap_{i=0}^{m-1} [K_{1i}, L_{1i})$ and

M. Fairtlough, M. Mendler: On the Logical Content of Computational Type Theory: A Solution to Curry's Problem.
In P. Callaghan, Z. Luo, J. McKinna (eds.), Types for Proofs and Programs, Springer 2002 (LNCS 2277), pp. 63-78.

$C_2 = \prod_{j=0}^{n-1}[K_{2j}, L_{2j})$. Note that if $K' \supset K$ and $L \supset L'$ then $[K, L) \leq [K', L')$. It follows that for a given $i < m$, $[K_{1i}, L_{1i}) \leq [K_{1i} \wedge K_{2j}, L_{1i} \vee L_{2j})$ for every $j < n$ and thus $[K_{1i}, L_{1i}) \leq \prod_{j=0}^{n-1}[K_{1i} \wedge K_{2j}, L_{1i} \vee L_{2j})$. Hence

$$C_1 = \prod_{i=0}^{m-1}[K_{1i}, L_{1i}) \leq \prod_{i=0}^{m-1}\left(\prod_{j=0}^{n-1}[K_{1i} \wedge K_{2j}, L_{1i} \vee L_{2j})\right) = C_1 \sqcup C_2$$

Also Definition 1 immediately implies the dual distributivity law $C \sqcup (D \sqcap E) = (C \sqcup D) \sqcap (C \sqcup E)$. From this the distributivity law $C \sqcap (D \sqcup E) = (C \sqcap D) \sqcup (C \sqcap E)$ and the characterisation of inequality $C \leq D$ iff $C \sqcup D = D$ follow. All this shows that $\mathbb{S}$ is a distributive lattice.

What remains is to define complements. For atoms L we obtained complementation as $\overline{L} = [L, \mathit{false})$. In view of Definition 1 this suffices to define complements for arbitrary standard constraints "by duality." More precisely, given a standard constraint $C = \prod_{i \in I}[K_i, L_i)$ we define its complement as

$$\overline{C} =_{df} \prod_{A \subseteq I}[\bigwedge_{a \in A} L_a, \bigvee_{b \in I \setminus A} K_b).$$

This definition is simply an application of DeMorgan's rule: $\overline{C} = \overline{\prod_{i \in I}[K_i, L_i)} = \bigsqcup_i \overline{[K_i, L_i)} = \bigsqcup_i \overline{\overline{K_i} \sqcup L_i} = \bigsqcup_i (K_i \sqcap \overline{L_i}) = \prod_{A \subseteq I}[\bigwedge_{a \in A} L_a, \bigvee_{b \in I \setminus A} K_b)$, where the last equation is by virtue of the dual distributivity law and Definition 1. We use the convention that empty disjunctions are false and empty conjunctions true. Using this convention one verifies $\overline{[K, \mathit{false})} = [\mathit{true}, K)$ and $\overline{[\mathit{true}, L)} = [L, \mathit{false})$, which confirms that the constraints $K \supset \cdot$ and $K \vee \cdot$ are indeed Boolean complements. It is a routine matter to check that C and $\overline{C}$ are complements for arbitrary C.

We can finally state the main theorem of this section, whose proof is obvious from the discussions above.

Theorem 2. *The collection $\mathbb{S}$ of standard constraints is a Boolean algebra generated by formulas as atomic constraints.*

In the remainder of this paper we show that $\mathbb{S}$ provides an adequate interpretation of PLL, while at the same time no finite subset of $\mathbb{S}$ is sufficient. The proof depends on a model-theoretic characterisation of PLL to be discussed next.

4 Kripke Model-Theory for PLL

Our model-theory is built on Kripke *constraint models* introduced in [9]. The main definitions and the completeness results are as follows:

Definition 3 (Kripke Constraint Models). *A (Kripke)* constraint model *for* PLL *is an intuitionistic modal model $(W, \sqsubseteq_i, \sqsubseteq_m, V, F)$, in which $\sqsubseteq_i$ and $\sqsubseteq_m$ are partial orderings, $\sqsubseteq_m$ is a subrelation of $\sqsubseteq_i$, V is a valuation, i.e. a mapping*

M. Fairtlough, M. Mendler: On the Logical Content of Computational Type Theory: A Solution to Curry's Problem.
In P. Callaghan, Z. Luo, J. McKinna (eds.), Types for Proofs and Programs, Springer 2002 (LNCS 2277), pp. 63-78.

assigning a set of propositional variables to each $w \in W$*, and* $F \subseteq W$*.* V *and* F *are hereditary in the sense that* V *is monotone in* $\sqsubseteq_i$ *and* F *is upper closed under* $\sqsubseteq_i$*, in other words,* $w \in F$ *and* $w \sqsubseteq_i v$ *imply* $v \in F$*.*

The interpretation of $w \sqsubseteq_m v$ put forward in [9] is that v is a *constraining* of w, or v is reachable from w up to a *constraint.* Elements of F are *fallible* worlds and if $w \sqsubseteq_m v$ and $v \in F$, then intuitively the constraint leading to v is inconsistent with world w. Using the "creative mathematician" interpretation of $w \sqsubseteq_i v$ as a construction step, the difference between $\sqsubseteq_i$ and $\sqsubseteq_m$ should be thought of as relating to some intensional feature of the world, such as the resources (time spent, energy dissipated, waste produced, *etc.*) used up in the constructions made. A modal step $w \sqsubseteq_m v$ then amounts to the stronger statement that v may be constructed from w *up to bounded resources*, whereas in a step $w \sqsubseteq_i v$ no such bound can be guaranteed.

Definition 4 (Validity). *Let* $M = (W, \sqsubseteq_m, \sqsubseteq_i, V, F)$ *be a constraint model. Given a proposition* φ *and* $w \in W$*,* φ *is* valid *at* w *in* M*, written* $M, w \models \varphi$ *iff*

- φ *is a propositional constant* α *and* $\alpha \in V(w)$*;*
- φ *is* $\varphi_1 \wedge \varphi_2$ *and both* $M, w \models \varphi_1$ *and* $M, w \models \varphi_2$*;*
- φ *is* $\varphi_1 \vee \varphi_2$ *and* $M, w \models \varphi_1$ *or* $M, w \models \varphi_2$*;*
- φ *is true; or* φ *is false and* $w \in F$*;*
- φ *is* $\varphi_1 \supset \varphi_2$ *and for all* $v \in W$ *such that* $w \sqsubseteq_i v$*,* $M, v \models \varphi_1$ *implies* $M, v \models \varphi_2$*;*
- φ *is of form* $\bigcirc\psi$ *and for all* $v \in W$ *such that* $w \sqsubseteq_i v$*, there exists* $u \in W$ *such that* $v \sqsubseteq_m u$ *and* $M, u \models \psi$*.*

A proposition φ *is* valid *in* M*, written* $M \models \varphi$*, if for all* $w \in W$*,* φ *is valid at* w *in* M*;* φ *is* valid*, written* $\models \varphi$*, if* φ *is valid in any constraint model* M*.*

Constraint models are not the only possible adequate Kripke semantics for PLL. A quite different kind of semantics was given by Goldblatt [11], in which only the intuitionistic part $\supset$ is represented by a frame, while the modality is realized by some extra topological information on the intuitionistic frame. Another kind of topological model has been introduced in [13]. To obtain the results of this paper, however, essential use of the structure of Kripke constraint models will be made, and of the following completeness theorem.

Theorem 5 ([9]). *Let a constraint model* $M = (W, \sqsubseteq_i, \sqsubseteq_m, V, F)$ *be called finite if* W *is finite, and* $V(w)$ *is finite for all* $w \in W$*. Then,* PLL $\vdash \varphi$ *iff for all finite constraint models* M*,* $M \models \varphi$*.*

5 Adequacy of Standard Constraints

We can now state and prove our version of Curry's conjecture. We show that a proposition is provable in PLL iff all its instantiations by standard constraints are provable in IPL. This gives a precise sense in which, proof-theoretically, a lax proposition is properly stronger than its $\bigcirc$-stripped version.

M. Fairtlough, M. Mendler: On the Logical Content of Computational Type Theory: A Solution to Curry's Problem.
In P. Callaghan, Z. Luo, J. McKinna (eds.), Types for Proofs and Programs, Springer 2002 (LNCS 2277), pp. 63-78.

Theorem 6. *Let φ be a lax proposition. Then,* $\mathsf{PLL} \vdash \varphi$ *iff for all standard constraints* $C \in \mathbb{S}$*,* $\mathsf{IPL} \vdash \varphi^C$*.*

The soundness direction of the theorem is straightforward. For every standard constraint context $C[x]$ the axioms and rules of PLL are derivable in IPL, as shown by the λ-terms given in the introduction. Hence, $\mathsf{PLL} \vdash \varphi$ implies $\mathsf{IPL} \vdash \varphi^C$. The challenge lies in the completeness direction. In view of Theorem 5 it suffices to find for every finite model M a standard constraint C such that for all φ, $M \models \varphi$ iff $M \models \varphi^C$. In other words, we are done if we can expand the meaning of $\bigcirc$ relative to a fixed finite model in terms of a single standard constraint. This is indeed possible, as stated in Lemma 7 below. The proof makes essential use of the particular structure of constraint models.

To state the lemma we need three constructions for finite constraint models $M = (W, \sqsubseteq_i, \sqsubseteq_m, V, F)$: First, we call a world $w \in W$ *stable* if it has no proper modal successors, *i.e.* for all $v \in W$, $w \sqsubseteq_m v$ implies $v = w$. We denote by $\mathit{Stab} \subseteq W$ the set of stable worlds of M. As one shows without difficulty, the semantics of φ and $\bigcirc\varphi$ coincides on stable worlds $w \in \mathit{Stab}$, *i.e.* $M, w \models \varphi$ iff $M, w \models \bigcirc\varphi$. Second, for any $w \in W$ let $\mathit{iSucc}(w) \subseteq W$ be the (finite) set of immediate successors of w, in other words, $\mathit{iSucc}(w) =_{df} \{v \in W \mid (w \sqsubset_i v) \;\&\; (\forall u.\, w \sqsubset_i u \sqsubseteq_i v \Rightarrow u = v)\}$. Third, for any finite set of propositional variables U, let $M^*_U = (W, \sqsubseteq_i, \sqsubseteq_m, V^*, F)$ be a *semantic completion of M avoiding U*, where V^* is determined by a choice of new propositional variables $\{\, \alpha_w \mid w \in W \,\}$ such that $M^*, v \models \alpha_w$ iff $w \sqsubseteq_i v$. We construct M^*_U to ensure that each new variable α_w does not occur in the range of V or in U and we drop the subscript U from M^*_U when it is clear from the context. The model M^* generates the same theory as M with respect to propositions whose variables are disjoint from $\{\alpha_w \mid w \in W\}$ except that it also has every one of its worlds w explicitly represented by a propositional variable α_w.

Lemma 7. *Let M be a finite constraint model and the sets Stab and iSucc defined as above. Then, if φ is any formula whatsoever, we have $M^* \models \bigcirc\varphi \equiv \bigwedge_{w \in \mathit{Stab}} [K_w, L_w)\, \varphi$, where $K_w =_{df} \alpha_w$ and $L_w =_{df} \bigvee_{w' \in \mathit{iSucc}(w)} \alpha_{w'}$. If $\mathit{iSucc}(w) = \emptyset$ then $\bigvee_{w' \in \mathit{iSucc}(w)} \alpha_{w'} = \mathit{false}$.*

Proof. Let M be a finite constraint model and $w \in W$ an arbitrary world. We show that

$$M^*, w \models \bigcirc\varphi \Leftrightarrow \forall u \in \mathit{Stab}.\; M^*, w \models \alpha_u \supset (\varphi \vee \bigvee_{u' \in \mathit{iSucc}(u)} \alpha_{u'}).$$

We prove direction ($\Rightarrow$) first, and assume $M^*, w \models \bigcirc\varphi$. Let $u \in \mathit{Stab}$ and $v \in W$ such that $w \sqsubseteq_i v$ and $M^*, v \models \alpha_u$ be given. The latter, $M^*, v \models \alpha_u$, means $u \sqsubseteq_i v$. There are two possibilities for this: either (i) $u = v$, or (ii) there exists an immediate successor $u' \in \mathit{iSucc}(u)$ of u such that $u' \sqsubseteq_i v$. In case (ii) we have $M^*, v \models \alpha_{u'}$, which implies $M^*, v \models \varphi \vee \bigvee_{u' \in \mathit{iSucc}(u)} \alpha_{u'}$ as desired. In the first case, (i), we get $v = u \in \mathit{Stab}$ and also, by hereditariness of truth, $M^*, v \models \bigcirc\varphi$. But since $v \in \mathit{Stab}$, this implies $M^*, v \models \varphi$, which in turn entails $M^*, v \models \varphi \vee \bigvee_{u' \in \mathit{iSucc}(u)} \alpha_{u'}$. This completes direction ($\Rightarrow$).

M. Fairtlough, M. Mendler: On the Logical Content of Computational Type Theory: A Solution to Curry's Problem.
In P. Callaghan, Z. Luo, J. McKinna (eds.), Types for Proofs and Programs, Springer 2002 (LNCS 2277), pp. 63-78.

Next, we show the other direction ($\Leftarrow$), where we assume that for all $u \in \mathit{Stab}$,

$$M^*, w \models \alpha_u \supset (\varphi \vee \bigvee_{u' \in iSucc(u)} \alpha_{u'}).$$

We must demonstrate $M^*, w \models \bigcirc\varphi$, *i.e.* that for all u, $w \sqsubseteq_i u$, there exists some v, $u \sqsubseteq_m v$, such that $M^*, v \models \varphi$. To this end, let such u be given. Because of the finiteness of the model there must exists a world v, $u \sqsubseteq_m v$ and $v \in \mathit{Stab}$. Otherwise, by definition of Stab and the properties of $\sqsubseteq_m$ we could construct an infinite sequence $u = u_0, u_1, u_2, \ldots$ of worlds with $u_i \neq u_{i+1}$ and $u_i \sqsubseteq_m u_{i+1}$. Since $w \sqsubseteq_i v$, $M^*, v \models \alpha_v$, and $v \in \mathit{Stab}$, our assumption gives us $M^*, v \models \varphi \vee \bigvee_{v' \in iSucc(v)} \alpha_{v'}$. However, since $v' \not\sqsubseteq_i v$ for every proper (immediate) successor v' of v, $M^*, v \not\models \bigvee_{v' \in iSucc(v)} \alpha_{v'}$ which implies $M^*, v \models \varphi$ as desired. This, finally, proves $M^*, w \models \bigcirc\varphi$. □

We are now ready to prove the completeness direction of Theorem 6.

Proof. (Theorem 6) Let φ be given such that $\mathsf{PLL} \not\vdash \varphi$. Then, by Theorem 5 there exists a finite constraint model M such that $M \not\models \varphi$. But then we must have $M^* \not\models \varphi$ for any semantic completion M^* of M that avoids the variables of φ, since M^* will coincide with M on all propositional variables contained in φ. From Lemma 7 we get $M^* \models \bigcirc\psi \equiv \bigwedge_{w \in \mathit{Stab}} [K_w, L_w)\,\psi$ for arbitrary ψ, where $K_w = \alpha_w$ and $L_w = \bigvee_{v \in iSucc(w)} \alpha_v$. Thus, because of the extensionality of PLL,—namely that whenever $M' \models \psi \equiv \varphi$ then $M' \models C[\psi] \equiv C[\varphi]$ for arbitrary contexts $C[\cdot]$—we have $M^* \not\models \varphi'$ where φ' is obtained from φ by replacing each occurrence of a subproposition $\bigcirc\psi$ of φ by $\bigwedge_{w \in \mathit{Stab}} [K_w, L_w)\,\psi$. Thus, we have found a single standard context $C =_{df} \prod_{w \in \mathit{Stab}} [K_w, L_w)$ such that $\not\models \varphi^C$. □

To illustrate the constructions involved in the proof let us look at an example. Consider the propositional scheme

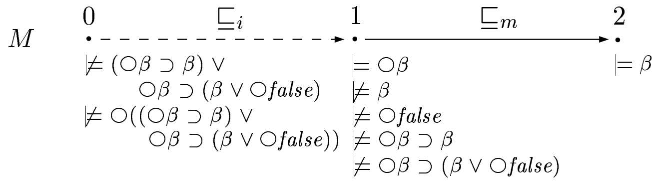
$$\theta =_{df} \bigcirc((\bigcirc\beta \supset \beta) \vee (\bigcirc\beta \supset (\beta \vee \bigcirc \mathit{false}))).$$

If θ is stripped of all $\bigcirc$ it turns into a trivial theorem of IPL. However, because of the way the modalities are placed it is not a theorem of PLL. To explain this in the light of Theorem 6 we expect to find a context $C[\cdot]$ such that the constraint expansion of θ,

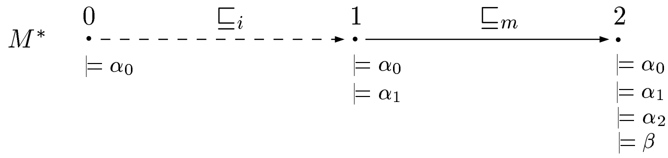
$$\theta^C = C[((C[\beta] \supset \beta) \vee (C[\beta] \supset (\beta \vee C[\mathit{false}])))]$$

is not provable in IPL. Observe that none of the simple contexts $C[x] = K \supset x$, $C[x] = x \vee L$, or $C[x] = K \supset x \vee L$ will work, since for all of them θ^C in fact *is* a theorem of IPL. This means we need a proper meet of contexts to falsify θ^C. As the proof of Theorem 6 shows such a context can be obtained systematically from a counter model for θ. The simplest constraint model that refutes θ is the three-world model $M = (\{0,1,2\}, \sqsubseteq_i, \sqsubseteq_m, V, \emptyset)$ in which the accessibilities are such that $n \sqsubseteq_i m$ iff $n \leq m$ and $\sqsubseteq_m = \{(0,0),(1,1),(2,2),(1,2)\}$, and the valuation is $V(0) = V(1) = \emptyset$, $V(2) = \{\beta\}$. The following picture illustrates the situation and indicates the validity of subpropositions of θ showing that indeed $0 \not\models \theta$ in this model:

M. Fairtlough, M. Mendler: On the Logical Content of Computational Type Theory: A Solution to Curry's Problem.
In P. Callaghan, Z. Luo, J. McKinna (eds.), Types for Proofs and Programs, Springer 2002 (LNCS 2277), pp. 63-78.



Now we consider the semantical completion M^* of M, which is just M but with additional propositional variables $\alpha_0, \alpha_1, \alpha_2$ that are validated at worlds $0, 1, 2$ respectively. So, $V^*(0) = \{\alpha_0\}$, $V^*(1) = \{\alpha_1\}$, $V^*(2) = \{\alpha_2, \beta\}$. In pictures,



We will use the new propositional variables $\alpha_0, \alpha_1, \alpha_2$, which represent the worlds $0, 1, 2$, respectively, to expand the meaning of $\bigcirc$ entirely in terms of propositions, following Lemma 7. Note that in M^*, $\alpha_0 \equiv \mathit{true}$ since α_0 is valid in every world of the model. The stable worlds in M^* are $\mathit{Stab} = \{0, 2\}$ since these have no proper modal successors. Their immediate successor sets are $\mathit{iSucc}(0) = \{1\}$ and $\mathit{iSucc}(2) = \emptyset$. Therefore, by Lemma 7 the following equivalence must hold in M^*, for arbitrary φ:

$$\begin{aligned}
\bigcirc\varphi &\equiv \bigwedge_{w \in \mathit{Stab}} [\alpha_w, \bigvee_{w' \in \mathit{iSucc}(w)} \alpha_{w'})\, \varphi \\
&= [\alpha_0, \bigvee_{w' \in \mathit{iSucc}(0)} \alpha_{w'})\, \varphi \;\wedge\; [\alpha_2, \bigvee_{w' \in \mathit{iSucc}(2)} \alpha_{w'})\, \varphi \\
&\equiv [\alpha_0, \alpha_1)\, \varphi \;\wedge\; [\alpha_2, \bigvee_{w' \in \emptyset} \alpha_{w'})\, \varphi \\
&\equiv [\mathit{true}, \alpha_1)\, \varphi \;\wedge\; [\alpha_2, \mathit{false})\, \varphi \\
&\equiv ([\mathit{true}, \alpha_1) \sqcap [\alpha_2, \mathit{false}))\, \varphi.
\end{aligned}$$

Because of this equivalence and the fact that $M^* \not\models \theta$ it must be the case that $M^* \not\models \theta^C$ where C is the constraint context $C =_{df} [\mathit{true}, \alpha_1) \sqcap [\alpha_2, \mathit{false})$. The proposition θ is an example of a propositional scheme that requires a proper composition of two constraints, *viz.* an implicational $\alpha_2 \supset \cdot$ and a disjunctive one $\cdot \vee \alpha_1$, in order to be outed as a non-theorem of PLL. In general, so it turns out, the discriminative power of *all* standard contexts is needed in order to characterise PLL fully. We show this in the following section.

6 Finite Constraint Collections are Inadequate

Neither the implicational nor the disjunctive contexts alone are sufficient to characterise PLL. The implicational constraints $\bigcirc = [\alpha, \mathit{false})$ validate the scheme

M. Fairtlough, M. Mendler: On the Logical Content of Computational Type Theory: A Solution to Curry's Problem.
In P. Callaghan, Z. Luo, J. McKinna (eds.), Types for Proofs and Programs, Springer 2002 (LNCS 2277), pp. 63-78.

$(\bigcirc\varphi \supset \bigcirc\psi) \supset \bigcirc(\varphi \supset \psi)$ and $\bigcirc(\bigcirc\varphi \supset \varphi)$, while the disjunctive ones $\bigcirc = [\mathit{true}, \beta)$ validate $\bigcirc(\varphi \vee \psi) \supset (\bigcirc\varphi \vee \bigcirc\psi)$ and $\bigcirc\varphi \supset (\varphi \vee \bigcirc\mathit{false})$. It is easy to see that basic constraints $\bigcirc = [\alpha, \beta)$ satisfy the scheme $\bigcirc(\bigcirc\varphi \supset \varphi \vee \bigcirc\mathit{false})$. The general pattern that emerges is as follows. Let $p_1, p_2, \ldots$ be a list of distinct propositional variables. We define a sequence of propositional schemes χ_m such that χ_m has exactly the p_i, $i \leq m$ as its (free) variables:

$$\chi_0 =_{df} \bigcirc\mathit{false}$$
$$\chi_{m+1} =_{df} \bigcirc(\bigcirc p_{m+1} \supset (p_{m+1} \vee \chi_m)).$$

Then χ_m is a valid scheme for all standard constraints of depth at most m.

Lemma 8. *Let C be a standard constraint of depth n. Then,* $\mathsf{IPL} \vdash \chi_m^C$ *for all $m \geq n$.*

Proof. We prove this by induction on the depth n of the constraint. For $n = 0$ the constraint is $C = [\mathit{false}, \mathit{false})$. Since for all $m \geq 0$, χ_m is of the form $\bigcirc\theta$ we find $\chi_m^C = \mathit{false} \supset \theta \vee \mathit{false}$ which is equivalent to *true*. Now let $C = \prod_{i=0}^{n}[\varphi_i, \psi_i)$ be a constraint of depth $n+1$. Then, $m \geq n+1 \geq 1$ and

$$\chi_m^C = \bigwedge_{i=0}^{n} \varphi_i \supset ((\bigcirc p_m \supset (p_m \vee \chi_{m-1}))^C \vee \psi_i).$$

So, we have to show that for all $i = 0, \ldots, n$, IPL derives $\varphi_i \vdash (\bigcirc p_m \supset (p_m \vee \chi_{m-1}))^C \vee \psi_i$. We will actually show that we can derive $\varphi_i \vdash (\bigcirc p_m \supset (p_m \vee \chi_{m-1}))^C$, or, which amounts to the same thing, that the sequents

$$\varphi_i, \bigwedge_{j=0}^{n} \varphi_j \supset (p_m \vee \psi_j) \vdash p_m \vee \chi_{m-1}^C$$

are derivable in IPL. Since the assumption includes φ_i and the implication $\varphi_i \supset (p_m \vee \psi_i)$ it is enough to show that $\varphi_i, p_m \vee \psi_i \vdash p_m \vee \chi_{m-1}^C$, *i.e.* the two sequents

$$\varphi_i, p_m \vdash p_m \vee \chi_{m-1}^C$$
$$\varphi_i, \psi_i \vdash p_m \vee \chi_{m-1}^C.$$

The first obviously is derivable immediately. For the second we proceed as follows: Let $C_i = \prod_{j \neq i}[\varphi_j, \psi_j)$ be the reduced constraint of depth n where we have dropped the interval $[\varphi_i, \psi_i)$. Then it is not difficult to see that we have $\varphi_i, \psi_i \vdash \chi_{m-1}^C \equiv \chi_{m-1}^{C_i}$. This follows essentially from the equivalence $\varphi_i, \psi_i \vdash \varphi_i \supset (\theta \vee \psi_i) \equiv \mathit{true}$. Thus, to obtain the sequent $\varphi_i, \psi_i \vdash p_m \vee \chi_{m-1}^C$ it suffices to prove $\vdash \chi_{m-1}^{C_i}$. But this follows from the induction hypothesis since C_i is a strictly smaller constraint of depth n and $m - 1 \geq n$. □

Lemma 8 says that standard constraints of depth up to and including m satisfy the axiom χ_m. However, these "characteristic" schemes are not theorems of PLL as we shall now see.

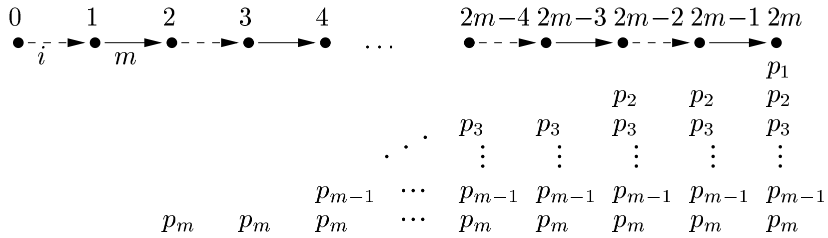
M. Fairtlough, M. Mendler: On the Logical Content of Computational Type Theory: A Solution to Curry's Problem.
In P. Callaghan, Z. Luo, J. McKinna (eds.), Types for Proofs and Programs, Springer 2002 (LNCS 2277), pp. 63-78.

Lemma 9. *For every $m \geq 0$ there is a constraint model M such that $M \not\models \chi_m$.*

Proof. For $m \geq 0$ consider the linear constraint model $M_m =_{df} (W, \sqsubseteq_i, \sqsubseteq_m, V, \emptyset)$, where $W = \{0, 1, \ldots, 2m\}$, $\sqsubseteq_i = \leq$,

$$\sqsubseteq_m = \{ (2k+1, 2k+2) \mid k = 0, \ldots m-1 \} \cup \{ (i,i) \mid i = 0, \ldots, 2m \},$$

and for all $k \leq m$, $V(2k) = V(2k+1) = \{p_{m-k+1}, p_{m-k+2}, \ldots, p_m\}$. So, *e.g.* $V(0) = V(1) = \{\}$ and $V(2) = V(3) = \{p_m\}$, $V(4) = V(5) = \{p_{m-1}, p_m\}$, *etc.* In pictures the model looks like this:



We claim that $M_m \not\models \chi_m$. We show this by induction on n. For $m = 0$ the model M_0 is the trivial non-fallible one-world model which obviously refutes $\chi_0 = \bigcirc \mathit{false}$. Now consider M_{m+1}. Recall that $\chi_{m+1} = \bigcirc(\bigcirc p_{m+1} \supset (p_{m+1} \vee \chi_m))$. We first observe that the suffix model $M_m(2)$ of M_m that starts with world 2 is precisely the same as M_{m-1}, if propositional variable p_{m+1} is ignored, whence by induction hypothesis $2 \not\models \chi_m$, so in particular $1 \not\models \chi_m$. Also, $1 \not\models p_{m+1}$, whence $1 \not\models p_{m+1} \vee \chi_m$. On the other hand, $2 \models p_{m+1}$ and since $1 \sqsubseteq_m 2$ we find $1 \models \bigcirc p_{m+1}$. This shows $0 \not\models \bigcirc p_{m+1} \supset (p_{m+1} \vee \chi_m)$. Finally, since $0 \sqsubseteq_m k$ implies $0 = k$, this implies $0 \not\models \bigcirc(\bigcirc p_{m+1} \supset (p_{m+1} \vee \chi_m))$ as desired. □

The desired theorem is a direct consequence of Lemmas 8 and 9:

Corollary 10. *No finite subset of* $\mathbb{S}$ *is complete for* PLL.

Proof. Let $\mathbb{D} \subset \mathbb{S}$ be a finite subset of standard constraints. Then, there exists a number $m \geq 0$ such that all $D \in \mathbb{D}$ are of depth at most m. By Lemma 8 χ_m^D is a theorem of IPL for each $D \in \mathbb{D}$. On the other hand by Lemma 9 the proposition χ_m is not a theorem of PLL. □

This brings to a satisfactory conclusion the programme of this paper in that we not only showed that the infinite Boolean algebra $\mathbb{S}$ of standard constraints provides a sound and complete interpretation of PLL, but also that no finite subset of $\mathbb{S}$ would suffice.

7 Final remarks

This paper offers one solution to Curry's programme of internally characterising the modality $\bigcirc$ by provability in IPL. A parallel can be found in [15, 14] which

M. Fairtlough, M. Mendler: On the Logical Content of Computational Type Theory: A Solution to Curry's Problem.
In P. Callaghan, Z. Luo, J. McKinna (eds.), Types for Proofs and Programs, Springer 2002 (LNCS 2277), pp. 63-78.

give an abstract representation of nuclei—the algebraic counterparts of $\bigcirc$—on a complete Heyting algebra in terms of implicational and disjunctive nuclei. Let us expand on this a bit. Let $M^* = (W, \sqsubseteq_i, \sqsubseteq_m, V^*, F)$ be a semantically complete constraint model. We form the Alexandroff topology $\Upsilon M^* = (W^+, \subseteq, \cap, \bigcup)$ of subsets of W upward closed under $\sqsubseteq_i$ and containing F. That is, $U \in W^+$ iff $F \subseteq U$ and whenever $u \in U$ and $u \sqsubseteq_i v$ then $v \in U$. ΥM^* is a complete Heyting algebra (cHA). Each formula φ of IPL corresponds to an element $M^*(\varphi) =_{df} \{w \mid M^*, w \models \varphi\}$ of W^+ so that *true* corresponds to W, *false* corresponds to F, $\wedge$ corresponds to $\cap$ and $\vee$ to $\cup$. Also $M^*(\varphi \supset \psi) = M^*(\varphi) \Rightarrow M^*(\psi)$ where $U \Rightarrow V$ is the interior of $\overline{U} \cup V$, *i.e.* the largest upper closed subset of $(W \setminus U) \cup V$. The algebraic counterpart to $\bigcirc$ is a *nucleus* on ΥM^*, that is, a monotone operation j on W^+ satisfying $U \subseteq j(U)$, $j(j(U)) \subseteq j(U)$ and $j(U \cap V) = j(U) \cap j(V)$. Then $\sqsubseteq_m$ determines a specific nucleus j_{M^*} given by $j_{M^*}(U) =_{df} \{u \in W \mid \forall v.\, u \sqsubseteq_i v \Rightarrow \exists r.\, v \sqsubseteq_m r \;\&\; r \in U \cup F\}$. Remarkably, it turns out that the nuclei on an arbitrary cHA H themselves form a cHA $\mathcal{N}(H) = (N(H), \leq, \wedge, \rightarrow, \bigvee)$, where $N(H)$ is the set of nuclei on H and $\leq$, $\wedge$ are given pointwise and $\rightarrow$, $\bigvee$ hardly ever pointwise. In [15] and [14] it is shown that every nucleus on a cHA H can be expressed in the form $\bigvee_{i \in H} o(K_i) \wedge c(L_i)$ where $o(K)$ is the open nucleus sending x to $K \rightarrow x$ and $c(L)$ is the closed nucleus sending x to $x \vee L$. If M^* is finite, then two facts emerge. Firstly we may represent every element $U \in W^+$ syntactically by $\psi_U =_{df} \bigvee\{\alpha_u \mid u \text{ minimal in } U \text{ w.r.t. } \sqsubseteq_i\}$. Secondly, since ΥM^* is also finite, $\mathcal{N}(\Upsilon M^*)$ is a finite Boolean algebra [15]. In this case, we may also represent its join $\bigvee$ syntactically as follows. By DeMorgan's rules, we have

$$\bigvee_{i \in I} o(K_i) \wedge c(L_i) = \bigwedge_{A \subseteq I} \left(\bigvee\nolimits_{i \in A} o(K_i)\right) \vee \left(\bigvee\nolimits_{i \in I \setminus A} c(L_i)\right) \qquad (1)$$

$$= \bigwedge_{A \subseteq I} o\left(\bigwedge\nolimits_{i \in A} K_i\right) \vee c\left(\bigvee\nolimits_{i \in I \setminus A} L_i\right) \qquad (2)$$

where step (2) follows from the equivalences $(K_1 \supset \varphi) \vee (K_2 \supset \varphi) \equiv (K_1 \wedge K_2) \supset \varphi$ and $(\varphi \vee L_1) \vee (\varphi \vee L_2) \equiv \varphi \vee (L_1 \vee L_2)$. It remains to find a syntactic representation for joins of the form $o(K) \vee c(L)$. But [15, Lem. 2.1] tells us that $o(K) \vee j = o(K) \circ j$ for any nucleus j and so we may define $(o(K) \vee c(L))\, \varphi$ to be $K \supset (\varphi \vee L) = [K, L)\, \varphi$. Since we can define $o(K)$ as $[K, \mathit{false})$ and $c(L)$ as $[\mathit{true}, L)$ it makes sense to take as basic constraints those of the form $[K, L)$ as we have done in this paper.

The constraint algebra $\mathbb{S}$ is not the only collection of constraints that might be considered for PLL. In type theory other computational interpretations of $\bigcirc$ have been proposed. For instance a generalisation of double negation $\bigcirc p = (p \supset \alpha) \supset \alpha$, used for typing continuations, yields a sound semantics, too. It can be shown that the context $a(\alpha)(\varphi) = (\varphi \supset \alpha) \supset \alpha$ in general cannot be represented by a fixed standard constraint that does not depend on x. However, given a finite constraint model M^*, results of [15] show that any nucleus j in $\mathcal{N}(\Upsilon M^*)$ may be represented as a meet of nuclei of the form $\bigwedge\{a(x) \mid x \text{ stable}\}$ where x is stable if $j(x) = x$. Letting Stab^+ denote the set of elements U of W^+ satisfying

M. Fairtlough, M. Mendler: On the Logical Content of Computational Type Theory: A Solution to Curry's Problem.
In P. Callaghan, Z. Luo, J. McKinna (eds.), Types for Proofs and Programs, Springer 2002 (LNCS 2277), pp. 63-78.

$j_{M^*}(U) = U$, this means that for finite M^*, $M^* \models \bigcirc\varphi \equiv \bigwedge_{U \in Stab^+}(\varphi \supset \psi_U) \supset \psi_U$ and so the class of constraints of the form $\bigwedge_{i \in I} a(K_i)$ provides another sound and complete interpretation of PLL. More work needs to be done to study the inherent structure of the class of all constraint contexts $C[x]$ expressible in IPL.

Acknowledgements We would like to thank Harold Simmons for a series of email discussions on finite modal algebras, which gave us fruitful insights that are reflected in this work. We are also grateful to Andy Pitts who prompted this investigation by challenging us with the question if Curry contexts are complete for PLL. Finally thanks are due to the anonymous referees who made many good suggestions for improving this paper.

References

[1] P. Aczel. The Russell-Prawitz modality. *Mathematical Structures in Computer Science*, 11(4), 2001. Special issue: *Modalities in Type Theory* (Proceedings of the 1999 Workshop on Intuitionistic Modal Logic and Applications, Trento, Italy).
[2] N. Benton and V. de Paiva. Computational types from a logical perspective. *Journal of Functional Programming*, 8(2), March 1998.
[3] P. N. Benton. A unified approach to strictness analysis and optimising transformations. Technical Report 388, University of Cambridge Computer Laboratory, February 1996.
[4] H. B. Curry. The elimination theorem when modality is present. *Journal of Symbolic Logic*, 17:249–265, 1952.
[5] H. B. Curry. *A Theory of Formal Deducibility*, volume 6 of *Notre Dame Mathematical Lectures*. Notre Dame, Indiana, second edition, 1957.
[6] R. Davies and F. Pfenning. A modal analysis of staged computation. In Jr. Guy Steele, editor, *Proc. 23rd POPL*. ACM Press, January 1996.
[7] J. Despeyroux and P. Leleu. Recursion over objects of functional type. *Mathematical Structures in Computer Science*, 2001.
[8] J. Despeyroux, F. Pfenning, and C. Schürmann. Primitive recursion for higher-order abstract syntax. In *Typed Lambda Calculi and Applications*, pages 147–163. Springer, 1997. LNCS 1210.
[9] M. Fairtlough and M. Mendler. Propositional Lax Logic. *Information and Computation*, 137(1):1–33, August 1997.
[10] M.V.H. Fairtlough and M. Walton. Quantified Lax Logic. Technical Report CS-97-11, Department of Computer Science, The University of Sheffield, July 1997.
[11] R. I. Goldblatt. Grothendieck topology as geometric modality. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 27:495–529, 1981.
[12] J. Hatcliff and O. Danvy. A computational formalisation for partial evaluation. *Math. Struct. in Comp. Science*, 7:507–541, 1997.
[13] B. P. Hilken. Duality for intuitionistic modal algebras. UMCS-96-12-2, Manchester University, Department of Computer Science, 1996. To appear in Journal of Pure and Applied Algebra.
[14] P. T. Johnstone. *Stone Spaces*. Cambridge University Press, 1982.
[15] D. S. Macnab. Modal operators on Heyting algebras. *Algebra Universalis*, 12:5–29, 1981.

M. Fairtlough, M. Mendler: On the Logical Content of Computational Type Theory: A Solution to Curry's Problem.
In P. Callaghan, Z. Luo, J. McKinna (eds.), Types for Proofs and Programs, Springer 2002 (LNCS 2277), pp. 63-78.

[16] M. Mendler. Constrained proofs: a logic for dealing with behavioural constraints in formal hardware verification. In G. Jones and M. Sheeran, editors, *Workshop on Designing Correct Circuits*. Springer, 1991.

[17] M. Mendler. *A Modal Logic for Handling Behavioural Constraints in Formal Hardware Verification*. PhD thesis, Department of Computer Science, University of Edinburgh, ECS-LFCS-93-255, March 1993.

[18] E. Moggi. Notions of computation and monads. *Information and Computation*, 93:55–92, 1991.

[19] D. A. Turner. Elementary strong functional programming. In P. H. Hartel and M. J. Plasmeijer, editors, *Functional Programming Languages in Education FPLE'95*, pages 1–13. Springer, 1995. LNCS 1022.

[20] P. Wadler. Comprehending monads. In *Conference on Lisp and Functional Programming*. ACM Press, June 1990.

[21] P. Wadler. Monads for functional programming. In *Lecture Notes for the Marktoberdorf Summer School on Program Design Calculi*. Springer Verlag, August 1992.

M. Fairtlough, M. Mendler: On the Logical Content of Computational Type Theory: A Solution to Curry's Problem.
In P. Callaghan, Z. Luo, J. McKinna (eds.), Types for Proofs and Programs, Springer 2002 (LNCS 2277), pp. 63-78.