An Intuitionistic Logic for Sequential Control

Chuck Liang[1] and Dale Miller[2]

1 **Department of Computer Science, Hofstra University, Hempstead, NY, US**
`chuck.c.liang@hofstra.edu`
2 **INRIA Saclay & LIX/Ecole Polytechnique, Palaiseau, France**
`dale.miller@inria.fr`

Abstract

We introduce a propositional logic *ICL,* which adds to intuitionistic logic elements of classical reasoning without collapsing it into classical logic. This logic includes a new constant for *false*, which augments *false* in intuitionistic logic and in minimal logic. We define Kripke models for ICL and show how they translate to several other forms of semantics. We define a sequent calculus and prove cut-elimination. We then formulate a natural deduction proof system with a term calculus that gives a direct, computational interpretation of contraction. This calculus shows that ICL is fully capable of typing programming language control operators such as *call/cc* while maintaining intuitionistic implication as a genuine connective.

1 Introduction

It is now well known that the Curry-Howard correspondence can be extended beyond intuitionistic logic. Since Griffin ([7]) showed the relationship between certain classical axioms and control operators, several *constructive* classical systems have been formulated, including Girard's *LC* proof system [5] and Parigot's deduction system, from which is derived $\lambda\mu$-calculus [14] and its variants.

However, the isomorphism between λ-abstraction and intuitionistic implication is a very strong one. If one collapses intuitionistic logic into classical logic altogether and considers the whole arena of classical proofs, then one is confronted with the fact that *classical implication* does not have the same strength as its intuitionistic counterpart. For example, intuitionistic implication corresponds to the programming notion of localized *scope.* In classical logic, however, $(A \Rightarrow B) \vee C$ is equivalent to $B \vee (A \Rightarrow C)$, which means that the assumption A is not localized to the left disjunct. The constructive meaning of classical logic is dependent on how we *choose* to interpret classical implication: for example, $\neg A \vee B$ and $\neg(A \wedge \neg B)$ convey different kinds of procedural information.

On the other hand, if we embed classical logic into intuitionistic logic via a double-negation translation, then the constructive meaning of classical proofs is also changed, for we can only expect λ-terms from such a translation, and not $\lambda\mu$-terms.

We propose a new logic that is an amalgamation of intuitionistic and classical logics, one that does not collapse one into the other. We refer to this logic as *Intuitionistic Control Logic* (ICL). In contrast to *intermediate logics,* we do not add new axioms to intuitionistic logic but a new logical constant for *false*. We distinguish between two symbols: 0 and $\bot$. The constant 0 is *false* in intuitionistic logic. The two constants will allow us to define two forms of negation: $\sim A$ and $\neg A$. For example, $A \vee \neg A$ will be provable but not $A \vee \sim A$. On the other hand, neither form of negation is *involutive* because both negations are defined by *intuitionistic* implication ($A \supset 0$ and $A \supset \bot$).

ICL can also be described as intuitionistic logic plus a version of Peirce's law. However, that description alone is unsatisfactory: we desire clear semantics and proof systems with cut-elimination procedures that make computing possible. We define several forms of semantics, including versions of Kripke models and cartesian closed categories. These intuitionistic structures do not become degenerate in ICL. The cut-elimination procedures we define will include a form of *structural reduction* as found in $\lambda\mu$ calculus, thereby showing that control operators can be obtained without a complete collapse into classical logic.

Valid	Invalid
$\neg A \vee A$	$\sim A \vee A$
$(\neg \mathbf{P} \supset \mathbf{P}) \supset \mathbf{P}$	$((P \supset Q) \supset P) \supset P$
$0 \supset A$	$\bot \supset A$
$\neg A \vee B \equiv \neg(A \wedge \neg B)$	$\sim A \vee B \equiv \sim(A \wedge \sim B)$
$\neg(A \wedge B) \equiv (\neg A \vee \neg B)$	$\neg(A \wedge \neg B) \equiv \neg\neg(A \supset B)$
$\neg\neg A \equiv A \vee \bot$	$\sim\sim A \supset A$
$\sim\neg \mathbf{A} \supset \mathbf{A}$	$\neg\neg A \supset A$
$A \supset \neg\sim A$	$\neg \sim A \supset A$
$A \supset \neg\neg A$	$A \supset \sim\neg A$
$A \supset \sim\sim A$	$(\neg B \supset \neg A) \supset (A \supset B)$

Table 1 Sample Truths and Falsehoods; $\neg A = A \supset \bot$, $\sim A = A \supset 0$, $(A \equiv B) = (A \supset B) \wedge (B \supset A)$.

2 Syntax; Sample Truths and Falsehoods

We consider only propositional logic in this presentation. The formulas of ICL are freely composed from atomic formulas, the binary connectives $\wedge$, $\vee$ and $\supset$, and the logical constants $\top$, 0 and $\bot$. Although there are three constants, two for *false* and one for *true*, ICL should not be confused with a "three-valued logic:" this will be obvious from its semantics. Without $\bot$, ICL is *identical* to intuitionistic logic, which is not finitely truth-valued.

We define two forms of negation as abbreviations for the following formulas

$$\sim A \;=\; A \supset 0 \qquad\qquad \neg A \;=\; A \supset \bot$$

We wish to present ICL using a careful balance of syntax and semantics. However, to give the reader a quick overview of the properties of this logic, we list in Table 1 some of the most important valid and invalid formulas. Several of the formulas in the table do not contain $\bot$ as a subformula. It holds that *a formula that does not contain $\bot$ is valid in ICL if and only if it is valid in intuitionistic logic.* Some formulas were selected to emphasize this fact. It will be an over-generalization to say that "$\bot$ is weaker than 0." In particular, $\neg(A \wedge \neg B) \not\equiv \neg\neg(A \supset B)$ whereas the (intuitionistic) equivalence holds if $\neg$ is replaced by $\sim$.

We have highlighted Peirce's formula with Q replaced by $\bot$, $(\neg P \supset P) \supset P$, which we refer to as *"our version of Peirce's formula."* This formula is of central importance to ICL. A similar formula is $\sim\neg A \supset A$, which is $((A \supset \bot) \supset 0) \supset A$. It will allow us to emulate the control operator $\mathcal{C}$. ICL does not have an involutive negation as both $\sim\sim A \supset A$ and $\neg\neg A \supset A$ are invalid. However, with *two* forms of negation there are more combinations to consider.

3 Kripke Semantics

We formally define ICL using a Kripke-style semantics. We consider only Kripke frames that are (finitely) *rooted:* it is known that intuitionistic propositional models can also be assumed to have this restriction. Such frames are the basis of models of the form $\langle \mathbf{W}, \mathbf{r}, \preceq, \models \rangle$, where $\preceq$ is a partial ordering relation on the set of possible worlds $\mathbf{W}$ and $\mathbf{r} \in \mathbf{W}$ is the unique root such that $\mathbf{r} \preceq u$ for all $u \in \mathbf{W}$. The binary relation $\models$ relates elements of $\mathbf{W}$ to sets of atomic formulas; $\models$ is monotonic in that if $u \preceq v$ then $u \models a$ implies $v \models a$. The $\models$ relation is also extended to all formulas in a way that observes the following rules. Here we use the symbols u and v to represent arbitrary possible worlds in $\mathbf{W}$ and the symbol q to represent worlds that are properly above $\mathbf{r}$ ($q \succ \mathbf{r}$).

- $u \models \top$; $u \not\models 0$
- $\mathbf{r} \not\models \bot$
- $q \models \bot$ for all $q \succ \mathbf{r}$
- $u \models A \wedge B$ iff $u \models A$ and $u \models B$
- $u \models A \vee B$ iff $u \models A$ or $u \models B$
- $u \models A \supset B$ iff for all $v \succeq u$, $v \not\models A$ or $v \models B$.

We shall refer to this version of Kripke models as *r-models.* The only differences between forcing rules in r-models and those of regular Kripke models for intuitionistic logic are in regard to $\bot$. *All worlds properly above* $\mathbf{r}$ *force* $\bot$*, but not* $\mathbf{r}$ *itself.* The usual property of monotonicity is established inductively on formulas:

- if $u \preceq v$ then $u \models A$ implies $v \models A$ for all formulas A.

A formula is considered valid in a r-model if it is valid in all worlds: by monotonicity this means that it is valid in $\mathbf{r}$. If a formula A is valid in model M we write $M \models A$. A formula is *valid in ICL* if it is valid in all models. Both 0 and $\bot$ are *inconsistent* as they have no models.

To illustrate reasoning in this semantics, given the root $\mathbf{r}$ of a model, if $\mathbf{r} \not\models A$, then $\mathbf{r} \models A \supset \bot$ because if $q \succ \mathbf{r}$ then $q \models \bot$. Thus all models validate $A \vee \neg A$. In fact, it holds that $\mathbf{r} \models A$ *if and only if* $\mathbf{r} \not\models \neg A$. However, $\neg\neg A \supset A$ remains invalid. For a countermodel, let $\mathbf{r} \not\models a$ for some atomic a and also let $q \not\models a$ for some $q \succ r$. We always have that $q \models \neg\neg a$ since $q \models \bot$, thus $q \models \neg\neg a$ and $q \not\models a$, so $\mathbf{r} \not\models \neg\neg a \supset a$.

Since intuitionistic models can be assumed to have rooted frames, the following is immediate:

▶ **Proposition 1.** *A formula that does not contain* $\bot$ *as a subformula is valid in ICL if and only if it is valid in intuitionistic logic.*

Given that $A \vee \neg A$ is valid, ICL looses the disjunction property for formulas that contain $\bot$, but will gain much in return.

Classical Implication and a General Law of Admissible Rules

One should not expect $\neg\neg A \supset A$ to be valid because $\supset$ represents *intuitionistic implication.* We can define "classical implication" within ICL as a derived connective:

- **Classical Implication:** $A \Rightarrow B \;=\; \neg A \vee B$

Unlike in intuitionistic logic, $\neg A \vee B$ is equivalent to $\neg(A \wedge \neg B)$, so it is not necessary to use a negative translation. If we replaced $\supset$ with $\Rightarrow$, then $\neg\neg A \Rightarrow A$ becomes abbreviation for $\neg\neg\neg A \vee A$, which is valid. It is easy to verify that the contrapositive axiom $(\neg B \Rightarrow \neg A) \Rightarrow (A \Rightarrow B)$ is valid, and that $\Rightarrow$ correctly represents classical implication. Both $A \Rightarrow \bot$ and $A \Rightarrow 0$ are in fact equivalent to $\neg A$, and in this sense $\neg$ can be called classical negation.

Equivalent forms of classical implication include $A \supset (B \vee \bot)$ and $A \supset \neg\neg B$. These alternatives each carry different procedural information, as reflected by the structure of proofs. However, $\neg\neg(A \supset B)$ is *not* an equivalent definition of $A \Rightarrow B$. Intuitionistic implication does not collapse into $\Rightarrow$ even in the scope of $\neg\neg$. This means that it is possible to *mix* classical and intuitionistic reasoning in ICL, e.g., the axiom schema $\neg\neg A \Rightarrow A$ may be instantiated with a formula containing $\supset$ without loosing the meaning of $\supset$. In contrast, $\sim\!\sim(A \supset B)$ collapses to a classical implication by Glivenko's theorem.

Although $\neg\neg A \supset A$ is not valid in all models, if in any model $\mathbf{r} \models \neg\neg A$ then it must be that $\mathbf{r} \not\models \neg A$, which in turn means that $\mathbf{r} \models A$ because all q above $\mathbf{r}$ forces $\neg A$. Thus it is an *admissible rule* of ICL that *if* $\neg\neg A$ *is valid then* A *is also valid.* In fact, we can prove a general property of admissible rules as follows

▶ **Proposition 2.** *If the classical implication* $A \Rightarrow B$ *is valid, then* A *is valid implies* B *is valid.*

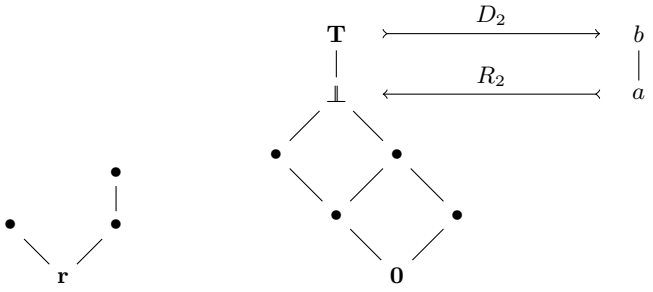

Figure 1 A Kripke frame with root (on the left) and a Heyting algebra with $\mathbb{\perp}$ (on the right) along with an adjunction with a two element boolean algebra.

As proof, if in any model $\mathbf{r} \models \neg A \vee B$ and $\mathbf{r} \models A$, then it cannot be that $\mathbf{r} \models \neg A$ since $\mathbf{r} \not\models \perp$. Therefore it must be that $\mathbf{r} \models B$.

Thus, *every classically valid implication corresponds to at least an admissible rule of ICL.*

In intuitionistic logic, an admissible rule can also be obtained from a proof of $\sim A \vee B$ (by the disjunction property). However, $\sim A \vee B$ does *not* represent classical implication. That would require a *negative* translation such as $\sim(A \wedge \sim B)$ or $\sim\sim(A \supset B)$: these formulas are not equivalent to $\sim A \vee B$, and do not give us a general admissibility property for intuitionistic logic.

The simplicity of the proof of Proposition 2 illustrates the utility of having a Kripke semantics for ICL.

4 Further Semantic Characterizations

In this section we describe several alternative semantic interpretations of ICL, culminating in a category-theoretic description that will lead to the proof theory of later sections.

4.1 $\perp$ in a Finite Heyting Algebra

To those familiar with the translation of Kripke models into Heyting algebras by forming a lattice of upwardly closed subsets of the Kripke frame (see Fitting [4, Chapter 1]), it should be clear that for any frame with a *root,* there is a unique, *second largest* element of the lattice. This element is the upwardly closed set of all elements of the poset *except the root.* This element denotes $\perp$; i.e., it is the set of all possible worlds that force $\perp$. Figure 1 includes an example of such a Heyting algebra. We write $\mathbb{\perp}$ for the semantic denotation of $\perp$. $\mathbb{\perp}$ and the top element $\mathbf{T}$ form a natural, two-element boolean algebra, which is all that is required for propositional classical logic. ICL formulas such as $A \supset \perp$ and $A \vee \perp$ are all interpreted *inside* this subalgebra. The connectives are interpreted as in intuitionistic logic: $\wedge$ is meet, $\vee$ is join and $\supset$ is the relative pseudo-complement.

4.2 $\perp$ in the Metric Space of Reals

Since Gödel showed that intuitionistic logic is not finitely truth-valued, no finite Heyting algebra suffices to interpret all of propositional intuitionistic logic. For this purpose we require the Heyting algebra derived from the topology of a *dense-in-itself* metric space, such as that of the real line $\mathbb{R}$. Here, $\vee$ and $\wedge$ are interpreted by the union and intersection of open sets respectively, and $A \supset B$ is interpreted by $A \rightarrow B$, which is the interior of $(\mathbb{R} - A) \cup B$.

A result of Tarski shows that, *given a finite Heyting algebra H and a dense-in-itself metric space, in this case the topology of $\mathbb{R}$, there is a dense, open subspace G of $\mathbb{R}$ and an isomorphism f from*

H into a subalgebra HG of G. If we applied this result to finite Heyting algebras with a unique second-largest element, i.e., with $\perp\!\!\!\perp$, we can deduce that $f(\perp\!\!\!\perp)$ maps to a *dense* subset of G (except in the special case when $f(\perp\!\!\!\perp)$ is empty), and that $G - f(\perp\!\!\!\perp)$ consists of a set of *isolated points.* In fact, for propositional logic, we can show this set can be assumed to consist of a single, arbitrary point. In this semantics of ICL $\perp\!\!\!\perp$ is $\mathbb{R}$ minus a single number. To be consistent with the usual habit of associating the number 1 with *true, let us choose this number to be* 1*:*

$$\perp\!\!\!\perp = \{x \in \mathbb{R} : \ x < 1 \text{ or } x > 1\}$$

A *valuation* (i.e., model) is represented by a mapping h from atomic formulas to open subsets of $\mathbb{R}$ that is extended to all formulas as follows:

1. $h(\top) = \mathbb{R}; \ \ h(0) = \emptyset; \ \ h(\bot) = \perp\!\!\!\perp$
2. $h(A \wedge B) = h(A) \cap h(B)$
3. $h(A \vee B) = h(A) \cup h(B)$
4. $h(A \supset B) = h(A) \rightarrow h(B)$

There is, however, also a degenerate case, which corresponds to r-models with frames consisting of only one element: the root $\mathbf{r}$ itself (this also refers to the special case of an empty $f(\perp\!\!\!\perp)$ above). Thus the definition of h on atomic formulas is given an secondary extension to h':

1. $h'(a) = \mathbb{R}$ if $1 \in h(a); \ \ h'(a) = \perp\!\!\!\perp$ if $1 \notin h(a)$ for atoms a
2. $h'(0) = h'(\bot) = \perp\!\!\!\perp; \ \ h'(\top) = \mathbb{R}$
3. $h'(A \vee B) = h'(A) \cup h'(B); \ \ h'(A \wedge B) = h'(A) \cap h'(B)$
4. $h'(A \supset B) = h'(A) \rightarrow h'(B)$

h' maps all formulas into the two-element boolean algebra $\{\mathbb{R}, \perp\!\!\!\perp\}$. A formula A is considered valid if for all h, $h(A) = \mathbb{R}$ and $h'(A) = \mathbb{R}$. The following properties hold for h and h':

- $1 \in h(A)$ if and only if $h(A) \rightarrow \perp\!\!\!\perp = \perp\!\!\!\perp$, and likewise for h'
- $1 \notin h(A)$ if and only if $h(A) \rightarrow \perp\!\!\!\perp = \mathbb{R}$, and likewise for h'.
- $1 \in h(A)$ if and only if $1 \notin h(\neg A)$, and likewise for h'.

The significance of these properties is that, in order to verify an formula of ICL that involves $\bot$, it is often only necessary to consider the cases $1 \in h(A)$ and $1 \notin h(A)$ (and similarly for h') for each atomic formula A in the formula. In other words we can build a kind of truth table. For example, for $h(A \vee \neg A)$, if $1 \in h(A)$ then $h(\neg A) = \perp\!\!\!\perp$, so $h(A) \cup h(\neg A) = \mathbb{R}$. If $1 \notin A$, then $h(\neg A) = \mathcal{I}((\mathbb{R} - A) \cup \perp\!\!\!\perp) = \mathcal{I}(\mathbb{R}) = \mathbb{R}$. Here, $\mathcal{I}$ is the interior operation.

This form of semantics is in fact quite different from the finite Kripke models and other structures explored here. We have included it here not just as another semantics, but because it offers a context for expanding ICL: there can be more than one "$\bot$" in this space (see Section 8.2).

For further exposition of this form of semantics, the reader may consult the longer version of this paper [11], as well as the treatise of Rasiowa and Sikorski [15] for background information. For a modern proof of the topological completeness of intuitionistic logic, see Mints [12, Chapter 9].

4.3 $\bot$ in a Cartesian Closed Category

Since neither form of negation in ICL is involutive, cartesian closed categories should not collapse into posets as they do when forcefully applied to classical logic.

Let $\mathbf{2}$ represent the category that consists of two objects a and b and three arrows: the identity arrows $id_a : a \rightarrow a$ and $id_b : b \rightarrow b$ plus a single arrow $s : a \rightarrow b$. This category is nothing but a two-element boolean algebra.

Let D be a cartesian closed category with terminal object $\mathbf{T}$ (as well as coproducts and an initial object $\mathbf{0}$). We write "$X \to Y$" to mean any arrow from X to Y. An object X is *uninhabited* if $Hom_D(\mathbf{T}, X) = \emptyset$. Let $\mathbf{D_2}$ be a functor from D to $\mathbf{2}$ defined as follows:

- $\mathbf{D_2}(X) = a$ if X is uninhabited.
- $\mathbf{D_2}(X) = b$ if $\mathbf{T} \to X$ exists in D.

Arrows are mapped by D_2 in the only possible way. In particular, if $T \to X$ exists but Y is uninhabited then it must hold that $Hom_D(X, Y) = \emptyset$. Only arrows $Y \to X$ may exist and are mapped to $s : a \to b$ in $\mathbf{2}$.

Now assume that $\mathbf{D_2}$ has a *right adjoint* $\mathbf{R_2}$, which is a functor from $\mathbf{2}$ to D (as illustrated in Figure 1). Then there is a one-to-one correspondence between arrows $X \to \mathbf{R_2}(Y)$ in D and $\mathbf{D_2}(X) \to Y$ in $\mathbf{2}$. If such an adjunction exists, then $\mathbf{R_2}(b)$ will indeed be isomorphic to the terminal object $\mathbf{T}$. However, *the analogous relationship does not hold between* $\mathbf{R_2}(a)$ *and the initial object.*

Let $\perp\!\!\!\perp = \mathbf{R_2}(a)$. As opposed to the initial object, $\perp\!\!\!\perp$ is *a terminal object in the full subcategory of uninhabited objects in* D. In other words:

▶ **Proposition 3.** *For each object* X *in a category* D *with* $\perp\!\!\!\perp$*,* X *is uninhabited if and only if there is a unique arrow* $\eta_X : X \to \perp\!\!\!\perp$.

In fact, the unique arrow is given by a natural transformation η that is part of the adjunction. Also implied by the property is that $\perp\!\!\!\perp$ *is uninhabited* and *there is at most one arrow* $A \to \perp\!\!\!\perp$ *for any* A.

There is no arrow $\perp\!\!\!\perp^{(\perp\!\!\!\perp^A)} \to A$ (which corresponds to $\neg\neg A \supset A$), since $\perp\!\!\!\perp$ is *not* the initial object except in degenerate cases.

We shall only consider those categories that contain $\perp\!\!\!\perp$. Examples of these categories include any finite Heyting algebra with a second-largest element (see Figure 1). Here, it is clear that $R_2(a)$ must be the second largest element. These algebras correspond to Kripke frames with a unique root, and thus to the Kripke r-models of ICL. This is enough to prove the completeness of this categorical characterization of ICL in so far as *provability* is concerned. However, we are of course interested in a categorical interpretation not just of validity but of *proofs.* Since there is at most one arrow to $\perp\!\!\!\perp$ from any object, there is a collapse of all proofs of formulas $\neg A$. However, there are formulas that contain $\perp$ but do not suffer the collapse. For example, $A \vee \perp$ is logically equivalent to $\neg\neg A$. While there is at most one proof of $\neg\neg A$, there can still be many proofs of A, and therefore of $A \vee \perp$.

4.4 The Representation of Proofs

From the essential property of $\perp\!\!\!\perp$ we are tempted to conclude that arrows such as $\mathbf{T} \to A + \perp\!\!\!\perp^A$ *"exist"* using an argument that starts by assuming that "either $\mathbf{T} \to A$ exists or A is uninhabited." While this argument is not constructive, we can isolate the classical assumption to a single constant and still derive a meaningful semantics of proofs *modulo* that constant. Within the proof theory of ICL, this corresponds to using an *axiom* to capture this constant.

We can attempt to construct an arrow $\mathbf{T} \to B$ *under the assumption* that B is uninhabited. Critically, in our brand of categories "B is uninhabited" is equivalent to "$B \to \perp\!\!\!\perp$ exists" (equivalently $\mathbf{T} \to \perp\!\!\!\perp^B$ exists). Review the *functional completeness* result from [9]: for any cartesian (closed) category K we may assume an indeterminate arrow $x : \mathbf{T} \to A$ and form the *polynomial category* $K[x]$. Then given any $f(x) : \mathbf{T} \to C$ in $K[x]$, there is a unique arrow $h : \mathbf{T} \to C^A$ such that $app(h, x) = f(x)$. In other words, $h = \lambda x.f(x)$.

We can also assume an indeterminate arrow $d : T \to \perp\!\!\!\perp^B$. The essential property of $\perp\!\!\!\perp$, combined with functional completeness, means we can argue that *if* $\mathbf{T} \to B$ *exists in the polynomial category* $K[d]$*, then* $\mathbf{T} \to B$ *exists in* K. This argument is a constructive interpretation of Proposi-

tion 3. In proof theory, one might implement this kind of inference as one of two sets of rules:

$$\frac{[\neg A], \Gamma \vdash A}{\Gamma \vdash A}, \qquad \text{or as} \qquad \frac{\Gamma \vdash A;\ [A, \Delta]}{\Gamma \vdash A;\ [\Delta]} \quad \text{and} \quad \frac{\Gamma \vdash A;\ [\Delta]}{\Gamma \vdash \bot;\ [A, \Delta]}$$

Clearly these rules capture some form of *contraction.* The brackets [ ] designate a special *context* for the special assumptions $\neg B$, which are distinguished from other assumptions that may appear in a purely intuitionistic context (as in $\neg A \vdash \neg A$). While the first rule appears closer to our conceptual explanation, the second pair of rules have better proof-theoretic characteristics, including the subformula property. The "cut-elimination" procedure is also clearer with the second version.

We have put "cut-elimination" in quotes because it is relatively easy to see that the first inference rule can be considered a *cut* against *our version of Peirce's formula:*

$$\frac{\dfrac{s :\ \neg A^d, \Gamma \vdash A}{\lambda d.s :\ \Gamma \vdash \neg A \supset A}\ \supset I \qquad \gamma :\vdash (\neg A \supset A) \supset A}{\gamma(\lambda d.s) :\ \Gamma \vdash A}\ \supset E\ (\mathbf{cut})$$

Gentzen-style *Hauptsatz* cannot eliminate cuts against axioms (unless the axiom is never used). Such proofs will include the constant γ. Technically there is a γ_X for each type/object X. In categorical terms, the special property of $\perp\!\!\!\perp$ tells us that an arrow $\gamma_P : P^{(\perp\!\!\!\perp^P)} \to P$ is at least *"not non-existent."* Depending on whether P is inhabited or uninhabited, we can construct γ_P from

$$P^{(\perp\!\!\!\perp^P)} \to \mathbf{T} \to P \text{ (inhabited)} \quad \text{or} \quad P^{(\perp\!\!\!\perp^P)} \to P^{(\perp\!\!\!\perp^P)} \times \mathbf{T} \to P^{(\perp\!\!\!\perp^P)} \times \perp\!\!\!\perp^P \to P \text{ (uninhabited).}$$

A more refined semantics in which γ can be argued to exist in a stronger sense would be interesting but not necessarily a better match for our proof theory and its computational interpretation. What we have is a *semantics of proofs modulo* γ that matches the extent of our proof theory. The proof theory cannot eliminate the above cut and, therefore, has the same limit as its semantics.

If this proof theory is regarded as defective, then it should be noted that the right-side contraction rule of Gentzen's LK can be given a similar interpretation. Although we cannot eliminate the cut, we can still permute other cuts around it. The second choice in inference rules rephrases this process as a familiar one: *the permutation of cut above contraction.* We can still speak of cut-elimination in a technical sense. When we formulate rewriting rules for "$\lambda\gamma$-terms," the γ, being a persistent cut/contraction, will not disappear after reduction in contrast to the λ-binder after β-reduction. Those who are familiar with the $\lambda\mu$-calculus should recognize this phenomenon: the μ binder behaves similarly. Our proof-terms will yield computational meaning by giving us the control operators.

In the following, we will ignore the difference between the arrows $A \to B$ and the corresponding arrows $\mathbf{T} \to B^A$. However, we shall write $[d]$ for the arrow $B \to \perp\!\!\!\perp$ that corresponds canonically to $d : T \to \perp\!\!\!\perp^B$. We choose this syntax to more clearly show the correspondence with $\lambda\mu$ calculus.

For example, we can prove our version of the excluded middle, $A + \perp\!\!\!\perp^A$, by assuming the arrow $[d] : (A + \perp\!\!\!\perp^A) \to \perp\!\!\!\perp$, then constructing $\mathbf{T} \to (A + \perp\!\!\!\perp^A)$. Now the formula $\neg(A \vee \neg A) \supset (A \vee \neg A)$ is provable in *intuitionistic* logic (with $\neg$ interpreted as minimal negation), and the proof uses both a left- and a right-injection on the two "copies" of $A \vee \neg A$. The proof is isomorphic to a classical proof of $A \vee \neg A$ using contraction (such a proof is found in the next section). Let ω^ℓ and ω^r be the injections arrows. Composing $A \xrightarrow{\omega^\ell} (A + \perp\!\!\!\perp^A) \xrightarrow{[d]} \perp\!\!\!\perp$ gives us $[d]\omega^\ell : A \to \perp\!\!\!\perp$, from which we can obtain $\omega^r(\lambda x.[d]\omega^\ell(x)) : \mathbf{T} \to A + \perp\!\!\!\perp^A$. This arrow is a "polynomial" over the indeterminate d. We can use λ-abstraction to form an arrow $\lambda d.\omega^r(\lambda x.[d]\omega^\ell(x)) : \perp\!\!\!\perp^{(A+\perp\!\!\!\perp^A)} \to (A + \perp\!\!\!\perp^A)$. Now we apply γ (for type $A \vee \neg A$): the *proof term* of $A \vee \neg A$ can thus be represented by

$$\gamma_{(A+\perp\!\!\!\perp^A)}(\lambda d.\omega^r(\lambda x.[d]\omega^\ell(x))) : \mathbf{T} \to A + \perp\!\!\!\perp^A$$

In the following sections we will shorten the syntax and just write $\gamma d.\omega^r(\lambda x.[d]\omega^\ell(x))$.

$$\frac{A, B, \Gamma \vdash C;\ [\Delta]}{A \wedge B, \Gamma \vdash C;\ [\Delta]}\ \wedge L \qquad \frac{A, \Gamma \vdash C;\ [\Delta] \quad B, \Gamma \vdash C;\ [\Delta]}{A \vee B, \Gamma \vdash C;\ [\Delta]}\ \vee L$$

$$\frac{\Gamma \vdash A;\ [\Delta] \quad B, \Gamma \vdash C;\ [\Delta]}{A \supset B, \Gamma \vdash C;\ [\Delta]}\ \supset L \qquad \frac{}{0, \Gamma \vdash A;\ [\Delta]}\ 0L \qquad \frac{}{\bot, \Gamma \vdash \bot;\ [\Delta]}\ \bot L$$

$$\frac{\Gamma \vdash A;\ [\Delta] \quad \Gamma \vdash B;\ [\Delta]}{\Gamma \vdash A \wedge B;\ [\Delta]}\ \wedge R \qquad \frac{\Gamma \vdash A;\ [\Delta]}{\Gamma \vdash A \vee B;\ [\Delta]}\ \vee R_1 \qquad \frac{\Gamma \vdash B;\ [\Delta]}{\Gamma \vdash A \vee B;\ [\Delta]}\ \vee R_2$$

$$\frac{A, \Gamma \vdash B;\ [\Delta]}{\Gamma \vdash A \supset B;\ [\Delta]}\ \supset R \qquad \frac{}{\Gamma \vdash \top;\ [\Delta]}\ \top R \qquad \frac{}{A, \Gamma \vdash A;\ [\Delta]}\ Id$$

$$\frac{\Gamma \vdash A;\ [A, \Delta]}{\Gamma \vdash A;\ [\Delta]}\ Con \qquad \frac{\Gamma \vdash A;\ [\Delta]}{\Gamma \vdash \bot;\ [A, \Delta]}\ Esc$$

Figure 2 The Sequent Calculus LJC

5 Sequent Calculus and Cut Elimination

The first proof system we present for ICL is motivated by our semantic completeness proof, which uses the tableaux method. However, our proof system has the appearance of a sequent calculus, given in Figure 2. In a sequent $\Gamma \vdash A; [\Delta]$, the *sets* Γ and Δ represent the left and right-side *contexts,* for which weakening can be shown to be admissible. The notation B, Θ represents $\{B\} \cup \Theta$ and does not preclude the possibility that $B \in \Theta$; thus contraction is obviated in these contexts. The formula A in $\Gamma \vdash A; [\Delta]$ is called the *control formula.* There is always exactly one control formula. A formula A is provable if $\vdash A; [\,]$ is provable. The following is a sample proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A;\ [\,]}\ Id}{A \vdash A \vee \neg A;\ [\,]}\ \vee I_1}{A \vdash \bot;\ [A \vee \neg A]}\ Esc}{\vdash \neg A;\ [A \vee \neg A]}\ \supset I}{\vdash A \vee \neg A;\ [A \vee \neg A]}\ \vee I_2}{\vdash A \vee \neg A;\ [\,]}\ Con$$

Note the correspondence with the proof term of Section 4.4: Esc represents the application of the indeterminate $[d]$ and Con represents γ. Except for the Esc and Con rules, LJC is a typical intuitionistic sequent calculus. Formulas inside the $[\Delta]$ context play no role in provability without the Esc (escape) rule. In fact, without Esc a proof is still entirely intuitionistic since the Con rule would become useless. A LJC proof can be considered to consist of segments of intuitionistic proofs joined by Esc. It is immediate that a formula that does not contain $\bot$ as a subformula can only have an intuitionistic proof. A formula containing $\bot$ may still have an intuitionistic proof if the proof does not use Esc: in such a case $\bot$ will have the same meaning as *false* in minimal logic.

The $\bot L$ rule is only needed if one wishes to apply the Id rule only to atomic formulas.

5.1 Cut Elimination

The cut rules for LJC are the following:

$$\frac{\Gamma \vdash A;\ [\Delta] \quad A, \Gamma' \vdash B;\ [\Delta']}{\Gamma\Gamma' \vdash B;\ [\Delta\Delta']}\ cut \qquad \frac{\Gamma \vdash A; [B, \Delta] \quad B, \Gamma' \vdash C; [\Delta']}{\Gamma\Gamma' \vdash A;\ [C, \Delta\Delta']}\ cut_2$$

The proof of cut-elimination can be found in a longer version of this paper online [11]. For this presentation, cut-elimination in the context of natural deduction, in Section 6, is more relevant.

5.2 Soundness and Completeness

The Soundness of LJC with respect to the Kripke semantics is proved by induction on the structure of proofs. However, one must be careful with the semantic interpretation of a sequent.

Let $\hat{\Gamma}$ represent the $\wedge$-conjunction over all formulas in Γ. If Γ is empty, then $\hat{\Gamma}$ represents $\top$. Let $\check{\Delta}$ represent the $\vee$-disjunction over all formulas in Δ. If Δ is empty, then $\check{\Delta}$ represents 0. The semantic interpretation of a sequent as a formula is defined as:

$$M \models (\Gamma \vdash A; [\Delta]) \text{ if and only if } M \models (\neg\check{\Delta} \wedge \hat{\Gamma}) \supset A.$$

This interpretation is adequate for soundness, for an *end-sequent* $\vdash A; [\,]$ is interpreted as $((0 \supset \bot) \wedge \top) \supset A$, which is equivalent to A.

Our completeness proof for LJC makes critical use of cut-elimination. Given an unprovable formula, we construct a countermodel by saturation. We say that (Γ, Δ) are $\bot$-consistent if $\Gamma \vdash \bot; [\Delta]$ is not provable. The idea is to generate a *maximally* $\bot$*-consistent* saturation to represent the root world $\mathbf{r}$ of the countermodel. Since $A \vee \neg A$ is always provable, by cut-elimination it follows that the saturation must contain *exactly one* of A or $\neg A$ for each formula A. Thus, any proper extension to the saturation will result in $\bot$-inconsistent sets. This is exactly what we want: $\mathbf{r} \not\models \bot$ but $q \models \bot$ for all $q \succ \mathbf{r}$. See [11] for further details and [4] for background on this style of completeness proof.

From cut-elimination and completeness it also follows that (propositional) ICL is decidable.

6 Natural Deduction and the Lambda-Gamma Calculus

In this section we present the natural deduction system NJC with a proof-term calculus that extends λ-calculus. While the presentation here does not strictly assume knowledge of the $\lambda\mu$ calculus, some important background for this section can be found in Section 4.4. The basis of this term system is Parigot's $\lambda\mu$-calculus [14] and its many variants, notably those found in [1], [2] and [13]. We have chosen a syntax close to $\lambda\mu$, although the name γ seem appropriate given our formulations in Section 4.4. We first present the implicational fragment, then discuss disjunction (and conjunction) separately in Section 8.1.

We define a $\lambda\gamma$-term as one of the following forms:

- λ-variables $x, y, \ldots$ and γ-variables $a, b, \ldots$
- λ-abstraction $\lambda x.t$ and γ-abstraction $\gamma a.s$
- applications $(s\ t)$, and *escapees* $[a]t$

In terms of Section 4.4, there should only one class of variables since $\gamma d.X$ is really $\gamma(\lambda d.X)$. For convenience, however, λ-variables represent indeterminate morphisms $\mathbf{T} \to A$ while γ-variables represent indeterminates $A \to \perp\!\!\!\perp$. The implicational fragment of NJC is found in Figure 3. The equivalence between NJC and LJC follows from cut-elimination in LJC. We have chosen a multiplicative treatment of contexts in this presentation, although contexts are still represented with sets. Terms are associated with entire subproofs (not with individual formulas). Formulas in the left-context Γ are *indexed* by unique λ-variables and formulas in the right-context $[\Delta]$ are indexed by unique γ-variables. A term $[a]t$ is really just $(a\ t)$, but a is a γ-variable. We assume that all bound variables are renamed whenever necessary. Contractions inside the Γ and Δ contexts are still admissible, but may require variable renaming.

The *cut* rules are also admissible in NJC and can be annotated with terms as follows:

$$\frac{t : \Gamma \vdash A;\ [\Delta] \quad s : A^x, \Gamma' \vdash B;\ [\Delta']}{(\lambda x.s)\ t : \Gamma\Gamma' \vdash B;\ [\Delta\Delta']}\ cut \qquad \frac{t : \Gamma \vdash A;\ [B^d, \Delta] \quad s : B^x, \Gamma' \vdash C;\ [\Delta']}{t\{[d](\lambda x.s)w/[d]w\} : \Gamma\Gamma' \vdash A;\ [C^d, \Delta\Delta']}\ cut_2$$

$$\frac{t : A^x, \Gamma \vdash B;\ [\Delta]}{(\lambda x.t) : \Gamma \vdash A \supset B;\ [\Delta]}\ \supset I \qquad \frac{t : \Gamma \vdash A \supset B;\ [\Delta] \quad s : \Gamma' \vdash A;\ [\Delta']}{(t\ s) : \Gamma\Gamma' \vdash B;\ [\Delta\Delta']}\ \supset E$$

$$\frac{s : \Gamma \vdash 0;\ [\Delta]}{\mathit{abort}\ s : \Gamma \vdash A;\ [\Delta]}\ 0E \qquad \frac{}{\mathit{exit} : \Gamma \vdash \top;\ [\Delta]}\ \top I \qquad \frac{}{x : A^x, \Gamma \vdash A;\ [\Delta]}\ Id$$

$$\frac{t : \Gamma \vdash A;\ [\Delta]}{[d]t : \Gamma \vdash \bot;\ [A^d, \Delta]}\ Esc \qquad \frac{u : \Gamma \vdash A;\ [A^d, \Delta]}{\gamma d.u : \Gamma \vdash A;\ [\Delta]}\ Con$$

Figure 3 NJC with Terms: the Implicational Fragment

The admissibility of cut_2 is by translation to cut:

$$\dfrac{\dfrac{\dfrac{r : \Gamma_1 \vdash B;\ [\Delta_1] \quad s : B^x, \Gamma' \vdash C;\ [\Delta']}{(\lambda x.s)r : \Gamma_1\Gamma' \vdash C;\ [\Delta_1\Delta']}\ cut}{[d](\lambda x.s)r : \Gamma_1\Gamma' \vdash \bot;\ [C^d, \Delta_1\Delta']}\ Esc}{\vdots}$$
$$\frac{\vdots}{t\{[d](\lambda x.s)w/[d]w\} : \Gamma\Gamma' \vdash A;\ [C^d, \Delta\Delta']}\ cut_2$$

There may be many instances of Esc with associated terms $[d]w$ inside the subproof t. cut_2 requires a form of substitution different from that of β-reduction. Substitution must take place *inside* the *left* subproof. The substitution $t\{[a]v/[a]w\}$ represents *"inductively replacing in* t *all occurrences of subterms of the form* $[a]w$ *with* $[a]v$*:"* this is the same structural substitution operation found in $\lambda\mu$-calculus [14].

Now consider the permutation of cut above contraction in the following case:

$$\frac{\dfrac{t : \Gamma \vdash A;\ [A^d, \Delta]}{\gamma d.t : \Gamma \vdash A;\ [\Delta]}\ Con \qquad s : A^x, \Gamma' \vdash B;\ [\Delta']}{(\lambda y.s)\ (\gamma d.t) : \Gamma\Gamma' \vdash B;\ [\Delta\Delta']}\ cut$$

The redex $(\lambda x.s)\ (\gamma d.t)$ represents such a cut. There are two ways to reduce this cut: either β-reduction, or a combination of β-reduction and structural reduction (cut both instances of A above Con). To preserve confluence, some choice must be made. One possibility (adopted often in similar systems, including [13]), is to require a call-by-value reduction strategy. This may in fact be the most reasonable choice, but our goal is to explore, in this presentation, a larger range of possibilities that the structure of ICL proofs suggest. We choose a similar approach to that of the original $\lambda\mu$-calculus, which is to defer to β-reduction in the above situation. The cut_2 form of reduction, however, will reappear in the context of $\vee$-elimination in Section 8.1. Only redexes of the form $(\gamma d.s)t$ require structural reduction. These redexes correspond to proofs of the form

$$\frac{\dfrac{s : \Gamma \vdash A \supset B;\ [(A \supset B)^d, \Delta]}{\gamma d.s : \Gamma \vdash A \supset B;\ [\Delta]}\ Con \qquad t : \Gamma' \vdash A;\ [\Delta']}{(\gamma d.s)\ t : \Gamma\Gamma' \vdash B;\ [\Delta\Delta']}\ \supset E$$

Such a proof is transformed by reducing the outer redex, represented by $A \supset B$ outside of $[\ ]$, as well as all redexes inside the subproof s indicated by subterms $[d]w$. The reductions rules are listed in Figure 4: they define cut-elimination for NJC. All rules are implied to have requirements regarding capture-avoiding substitution (e.g. in the $\mu\gamma$ rule d is not free in t).

1. $(\lambda x.s)\ t \longrightarrow s[t/x]$. ($\beta$-reduction)
2. $(\gamma d.s)\ t \longrightarrow \gamma d.(s\{[d](w\ t)/[d]w\}\ t)$. ($\mu\gamma$-reduction)
3. $abort(s)\ t \longrightarrow abort(s)$. (aborted reduction)
4. $\gamma a.s \longrightarrow s$ when a does not appear free in s. (vacuous contraction)
5. $\gamma a.\gamma b.s \longrightarrow \gamma a.s[a/b]$. ($\gamma$-renaming)
6. $[d]\gamma a.s \longrightarrow [d]s[d/a]$. ($\mu$-renaming)

Figure 4 Term Reduction Rules for the Implicational Fragment

The rule for $abort$ is justified since if 0-elimination proves $A \supset B$ then certainly 0-elimination proves B. $abort$ can be considered to be a constant of type $0 \supset A$ (which has proof $\lambda x.abort\ x$). Note that $(\gamma d.abort\ s)\ t$ reduces to $\gamma d.abort\ s\{[d]w\ t/[d]w\}$: the outer application to t is absorbed.

The μ-renaming rule is also found in $\lambda\mu$-calculus. In the context of NJC it corresponds to the elimination of a redundant contraction, because the active formula A in an Esc rule can always persist in the context Δ:

$$\dfrac{\dfrac{s : \Gamma \vdash A;\ [A^b, \Delta]}{\gamma b.s : \Gamma \vdash A;\ [\Delta]}\ Con}{[d]\gamma b.s : \Gamma \vdash \bot;\ [A^d, \Delta]}\ Esc \qquad \text{is converted to} \qquad \dfrac{s[d/b] : \Gamma \vdash A;\ [A^d, \Delta]}{[d]s[d/b] : \Gamma \vdash \bot;\ [A^d, \Delta]}\ Esc$$

Likewise, the γ-renaming rule eliminates consecutive contractions, which are redundant since contractions *inside* the $[\Delta]$ context are always admissible. Another rule that should be classified with the renaming rules is the elimination of a vacuous γ binder. With this rule, the term for a purely intuitionistic proof will reduce to a λ-term.

Subject Reduction, Strong Normalization and Confluence

Subject reduction is a consequence of the fact that the reduction rules follow cut-elimination and other valid proof transformations. Strong normalization is proved using the reducibility method of Tait and Girard. Our proof follows closely the proof (for the simply typed version) found in [6] because we found it to be readily adaptable. Much of this proof, including the reducibility criteria, can be used virtually without modification. The proof of confluence likewise follows the standard approach of Tait-Martin-Löf, by first defining a parallel, reflexive reduction relation. Further details of these proofs are found in [11].

7 The Computational Content of Contraction

An important proof term is that of our version of Peirce's formula:

$$\dfrac{\dfrac{x : (\neg P \supset P)^x \vdash \neg P \supset P; [\,]\quad \dfrac{\dfrac{\dfrac{\dfrac{}{y : (\neg P \supset P)^x, P^y \vdash P;\ [\,]}\ Id}{[d]y : (\neg P \supset P)^x, P^y \vdash \bot;\ [P^d]}\ Esc}{\lambda y.[d]y : (\neg P \supset P)^x \vdash \neg P;\ [P^d]}\ \supset I}{(x\ \lambda y.[d]y) : (\neg P \supset P)^x \vdash P;\ [P^d]}\ \supset E}{\gamma d.(x\ \lambda y.[d]y) : (\neg P \supset P)^x \vdash P;\ [\,]}\ Con}{\lambda x.\gamma d.(x\ \lambda y.[d]y) : \vdash (\neg P \supset P) \supset P;\ [\,]}\ \supset I$$

This term is different from what corresponds to Peirce's formula in $\lambda\mu$-calculus and its variants in that it does not require a second $[d]$ to label the entire subterm under γd, for that is obviated by the fact that γ represents contraction. Cut-elimination in the presence of a contraction requires reductions inside the $[\,]$ context *as well as* outside.

$$\frac{s : \Gamma \vdash A;\ [\Delta] \quad t : \Gamma' \vdash B;\ [\Delta']}{(s,t) : \Gamma\Gamma' \vdash A \wedge B;\ [\Delta\Delta']}\ \wedge I \qquad \frac{s : \Gamma \vdash A \wedge B;\ [\Delta]}{(s)_\ell : \Gamma \vdash A;\ [\Delta]}\ \wedge E_1 \qquad \frac{s : \Gamma \vdash A \wedge B;\ [\Delta]}{(s)_r : \Gamma \vdash B;\ [\Delta]}\ \wedge E_2$$

$$\frac{s : \Gamma \vdash A;\ [B^d, \Delta]}{\omega^\ell d.s : \Gamma \vdash A \vee B;\ [\Delta]}\ \vee I_1 \qquad \frac{s : \Gamma \vdash B;\ [A^d, \Delta]}{\omega^r d.s : \Gamma \vdash A \vee B;\ [\Delta]}\ \vee I_2$$

$$\frac{v : \Gamma_1 \vdash A \vee B;\ [\Delta_1] \quad s : A^x, \Gamma_2 \vdash C;\ [\Delta_2] \quad t : B^y, \Gamma_3 \vdash C;\ [\Delta_3]}{(\lambda x.s, \lambda y.t)\ v : \Gamma_1\Gamma_2\Gamma_3 \vdash C;\ [\Delta_1\Delta_2\Delta_3]}\ \vee E$$

Figure 5 NJC Rules for Disjunction and Conjunction

Now, $\gamma d.X$ is in fact $\gamma(\lambda d.X)$. The above proof term can be seen as $\lambda x.\gamma(\lambda d.(x\ \lambda y.(d\ y)))$, which is just the eta-expanded form of γ. This is no surprise given the analysis of Section 4.4.

Call this term $\mathcal{K}$, then $(\mathcal{K}\ M\ k_1\ k_2)$ reduces to $\gamma d.(M\ \lambda y.[d](y\ k_1\ k_2))\ k_1\ k_2$. For example, given the term context $E[z] = (z\ k_1\ k_2)$, $E[\mathcal{K}M]$ reduces to $\gamma d.E[M(\lambda y.[d]E[y])]$: this emulates the behavior of *call/cc* (see [2] for further analysis of $\lambda\mu$-based systems and control operators).

In contrast to *call/cc,* the $\mathcal{C}$ operator of Felleisen et al. [3] has a different behavior, and has been given the classical type $\neg\neg A \Rightarrow A$. The ICL formulas $\neg\neg A \supset A$ and $\sim\sim A \supset A$ are both unprovable, but we can consider a proof of $\sim\neg A \supset A$:

$$\dfrac{\dfrac{x :\sim\neg A^x \vdash \sim\neg A;\,[\,] \quad \dfrac{\dfrac{\dfrac{\dfrac{}{y :\sim\neg A^x, A^y \vdash A;\,[\,]}\,Id}{[d]y :\sim\neg A^x, A^y \vdash \bot;\ [A^d]}\,Esc}{\lambda y.[d]y :\sim\neg A^x \vdash \neg A;\ [A^d]}\,\supset I}{x\ \lambda y.[d]y :\sim\neg A^x \vdash 0;\ [A^d]}\,\supset E}{abort\ (x\ \lambda y.[d]y) :\sim\neg A^x \vdash A;\ [A^d]}\,0E}{\gamma d.abort\ (x\ \lambda y.[d]y) :\sim\neg A^x \vdash A;\ [\,]}\,Con}{\lambda x.\gamma d.abort\ (x\ \lambda y.[d]y) : \vdash \sim\neg A \supset A;\ [\,]}\,\supset I$$

Call this term $\mathcal{C}_1$, then $\mathcal{C}_1 M$, when applied to a term t, is only subject to structural reduction inside the *abort* subterm. We note that $\mathcal{C}_1 M = \mathcal{K}(\lambda k.abort(M\ k))$ and that $\mathcal{C}_1(\lambda z.M) = abort(M)$ for z not free in M (the γd in this term will be vacuous). Compare $\mathcal{C}_1$ to the version of $\mathcal{C}$ in the original $\lambda\mu$-calculus: $\lambda x.\mu\alpha.[\varphi](x\ \lambda y.\mu\delta.[\alpha]y)$. Here $abort$ replaces the free variable φ (which, like $abort$, persists in the term after reduction). All valid formulas of ICL are proved by closed terms.

8 Extending ICL

The materials presented in the previous sections are self-contained and complete. In this section we discuss two important directions of continuing work that extend the computational content of proofs.

8.1 The Computational Content of Disjunction

The computational potential of our approach to combining classical and intuitionistic logics is not limited to the implicational fragment. It is possible to add a conjunction and as well as a (non-additive) disjunction. We extend the syntax of terms to include *injective abstractions* $\omega^\ell a.s$ and $\omega^r a.s$, as well as the usual pairs (s,t) and projections $(s)_\ell$ and $(s)_r$. NJC is extended to include the rules of Figure 5. The $\omega^{\ell/r}$ binders are generalizations of injection operators. The interpretation of disjunctions using a form of abstraction is not so unusual when one considers its similarity to

implication: both can be seen as non-additive disjunctions (but $\vee$ is not "multiplicative" either)[1]. A $\vee$-introduction rule discharges a formula from the right context just as $\supset$-introduction discharges a formula from the left context. A vacuous $\omega^{\ell/r}$ binder degrades to an injection. These $\vee$-introduction rules are equivalent to the additive forms given the Con rule (the technical proofs of [11] were in fact executed for the non-additive version). The rewrite rules of $\lambda\gamma$ are extended to include

- $(u, v)\ (\omega^{\ell} d.t) \longrightarrow \gamma d.(u\ t\{[d](v\ w)/[d]w\})$;
  $(u, v)\ (\omega^{r} d.t) \longrightarrow \gamma d.(v\ t\{[d](u\ w)/[d]w\})$ (ω-reduction)
- $(u, v)\ \gamma d.t \longrightarrow \gamma d.(u, v)\ t\{[d](u, v)w/[d]w\}$ ($\omega\gamma$-reduction)
- $(u, v)_{\ell} \longrightarrow u$; $(u, v)_{r} \longrightarrow v$. (projections)
- $(\gamma d.s)_{\ell} \longrightarrow \gamma d.s_{\ell}\{[d]w_{\ell}/[d]w\}$; $(\gamma d.s)_{r} \longrightarrow \gamma d.s_{r}\{[d]w_{r}/[d]w\}$. ($\gamma$-projections)

Since a pair may appear in the scope of γ, projection must be defined for such pairs as special instances of structural substitution. Likewise, an ω-binder may also appear inside γ, hence the $\omega\gamma$ rule. The critical ω-reduction rules correspond to the cut-elimination case for $\vee$:

$$\dfrac{\dfrac{u : \Gamma \vdash A;\ [B^d, \Delta]}{\omega^{\ell} d.u : \Gamma \vdash A \vee B;\ [\Delta]}\ \vee I_1 \qquad s : A^x, \Gamma \vdash C;\ [\Delta] \qquad t : B^y, \Gamma \vdash C;\ [\Delta]}{(\lambda x.s, \lambda y.t)\ \omega^{\ell} d.u : \Gamma \vdash C;\ [\Delta]}\ \vee E$$

This proof reduces to:

$$\dfrac{\dfrac{\dfrac{u : \Gamma \vdash A;\ [B^d, \Delta] \qquad t : B^y, \Gamma \vdash C;\ [\Delta]}{u\{[d](\lambda y.t)w/[d]w\} : \Gamma \vdash A;\ [C^d, \Delta]}\ cut_2 \qquad s : A^x, \Gamma \vdash C;\ [\Delta]}{(\lambda x.s)\ u\{[d](\lambda y.t)w/[d]w\} : \Gamma \vdash C;\ [C^d, \Delta]}\ cut}{\gamma d.(\lambda x.s)\ u\{[d](\lambda y.t)w/[d]w\} : \Gamma \vdash C;\ [\Delta]}\ Con$$

One possible interpretation of this type of $\vee$-elimination, $(\lambda x.(x\ s),\ \lambda y.u)\ (\omega^{\ell} d.t)$, is the execution of a procedure that could throw an exception, with the second lambda term representing an *exception handler*. Expressions $[d]e$ throw exceptions.

The extended NJC is equivalent to LJC and is thus sound and complete, although some results such as strong normalization have yet to be extended to the additional rules.

8.2 Adding Multiple Controls

Non-additive disjunction in ICL can potentially extend the proofs-as-programs paradigm beyond the use of control operators. However, the fact that all expressions of the form $[d]e$ have type $\bot$ limits the types that reasonable programs are expected to have[2]. Though some of the theoretical elegance of ICL would be lost, it may be useful to have more than one version of $\bot$. The topological semantics of Section 4.2 provides a framework for this extension. Tarski's result that allowed us to map finite Heyting algebras to the topology of $\mathbb{R}$ in fact indicate that $\perp\!\!\!\perp$, as the second-largest point in a finite algebra, can be represented by other *dense open sets* (except the special case where the finite algebra has but two elements). Such a set consists of a countable collection of disjoint open intervals (countable by the density of the rationals in $\mathbb{R}$). Its complement, which is a closed set, consist of a countable set of isolated points (with an empty interior).

Let $\mathbb{R}_1 = \mathbb{R} - \{1\}$. Let $\mathbb{I}$ be the set of all integers and let $\mathbb{I}_1$ represent $\mathbb{I} - \{1\}$. Let $\mathbb{R}_{\mathbb{I}}$ represent $\mathbb{R} - \mathbb{I}$: this is a dense open set that consists of all intervals $\ldots (-1 : 0), (0 : 1), (1 : 2), \ldots$. Here,

[1] Our $\omega^{\ell/r}$-binders are similar to the one of [16]: while their $\vee$-introduction rule also involves an abstraction (but is invertible/multiplicative), their $\vee$-elimination carries a completely different computational meaning. Similar rules for introducing disjunctions have also been given in [8, Chapter 7].

[2] this was pointed out to us by Hugo Herbelin.

$(a : b)$ is $\{x \in \mathbb{R} : a < x < b\}$. Note that if $\mathbb{R}_\mathbb{I} \subseteq B$, then B is also an open set. Between $\mathbb{R}_1$ and $\mathbb{R}_\mathbb{I}$ one finds a *boolean lattice of dense open sets,* each set corresponds to a subset of $\mathbb{I}_1$. Call this "lattice of bottoms" $\mathcal{B}$. For each $B \in \mathcal{B}$, $A \cup (A \to B) = \mathbb{R}$ for any open set A.

In this version of ICL, $\bot$ *is now a connective of one argument.* Formulas $\bot^A$ are interpreted by

$$h(\bot^A) = (\mathbb{R}_1 - h(A)) \cup \mathbb{R}_\mathbb{I}$$

The special case of a two-element algebra is still handled by h', with $h'(\bot^A) = h'(0) = \mathbb{R}_1$: h' does not require separate inference rules. The following properties hold:

- $h(\bot^0) = \mathbb{R}_1$; $\quad h(\bot^\top) = \mathbb{R}_\mathbb{I}$
- $h(\bot^{(A \cup B)}) = h(\bot^A) \cap h(\bot^B)$; $\quad h(\bot^{(A \cap B)}) = h(\bot^A) \cup h(\bot^B)$; $\quad h(\bot^{\bot^{\bot^A}}) = h(\bot^A)$
- If $h(A) \subseteq h(B)$, then $h(\bot^B) \subseteq h(\bot^A)$ (the converse does not hold).

Since $\bot$ is involutive within $\mathcal{B}$, we adopt the following syntactic identities: $\bot^{(A \wedge B)} = \bot^A \vee \bot^B$, $\bot^{(A \vee B)} = \bot^A \wedge \bot^B$, and $\bot^{\bot^{\bot^A}} = \bot^A$. However, $\bot^{(A \supset B)}$ cannot be decomposed syntactically. The non-intuitionistic inference rules are now as follows:

$$\frac{\Gamma \vdash A;\ [A, \Delta]}{\Gamma \vdash A;\ [\Delta]}\ Con \qquad \frac{\Gamma \vdash A;\ [\Delta]}{\Gamma \vdash \bot^B;\ [A, \Delta]}\ Esc \qquad \frac{\Gamma, B \vdash A;\ [\Delta]}{\Gamma, \bot^A \vdash \bot^B;\ [\Delta]}\ Swap$$

One might expect the Esc rule to change more than it has. Recall that a formula A inside $[\Delta]$ is equivalent to a $\neg A$ on the left side of the sequent. To keep the subformula property, we now assume that "$\neg A$" always represents $A \supset \bot^A$. Under this assumption, there is a hidden second premise of Esc that is equivalent to $\neg A, \Gamma, \bot^A \vdash \bot^B$. However, this premise follows from the given premise by weakening and $Swap$, and is therefore redundant. Because Esc is virtually unchanged, all the proofs of ICL are still valid *using any* $\bot^B$ *in place of* $\bot$. With the restriction $B = A$ in the Esc rule, the proof terms will also be the same.

Cut-elimination extends to the $Swap$ rule. We are continuing the study of this logic.

9 Conclusion: ICL and Linear Logic

We end this paper by acknowledging an obvious fact: the original impetus for using two constants for *false* came from linear logic. The first insights into the existence of ICL came from considering where to place $\bot$ in context of the semantics of intuitionistic logic, specifically in the metric space of reals. However, it is *not* correct to suppose that ICL can be translated into linear logic by using any translation of intuitionistic logic, then just translate $\bot$ to $\bot$. The linear formula $A \oplus (!A \multimap \bot)$, which naively translates $A \vee \neg A$, has no proof. A polarized translation in the style of LC *might* work, at least for $\wedge$, $\vee$ and atomic negation. Say that $\bot$ is *"negative,"* $\neg a$ is negative for atomic a, and $A \vee B$ is negative if A or B is negative (similarly for $\wedge$). Then translate negative $A \vee B$ using $\parr$ instead of $\oplus$. However, it is not at all clear if this translation can be extended to $\supset$ in general. It is unlikely: consider a translation of $(\neg P \supset P) \supset P$ that allows $\supset$ to stay intuitionistic (equivalent to $!A \multimap B$). In terms of emulating ICL *proofs,* formulas inside the $[\Delta]$ context are subject to structural rules but not to introduction rules until the control formula is $\bot$. Linear formulas $?!A$ can give this behavior, and we can emulate an ICL proof once we know when formulas must be treated this way. However, formulas containing "?!" cannot form *synthetic connectives* (it destroys *focus*), and therefore cannot be used for a direct translation of ICL formulas into linear logic.

The proof systems for ICL in fact bare strong similarities to LC. The $\vee$ in LC can also behave additively or multiplicatively. The formula inside the *stoup* can never be removed unless it is negative. But LC does not accommodate intuitionistic implication. Girard's attempt to extend LC to include intuitionistic logic in a system called LU was not entirely successful, as he acknowledges.

In a slightly prior work [10] we introduced another system that extends LC to include intuitionistic logic. That system is also based on polarization. ICL is *not* a fragment of any of these systems. It is not based on any notion of polarity or duality that is assumed to exist *a priori.* Both semantically and proof-theoretically, it inherits the machinery of intuitionistic logic.

Acknowledgments. The authors wish to thank François Lamarche for valuable comments on Sections 4.3 and 4.4, as well as Hugo Herbelin, Alexis Saurin, Stéphane Lengrand and others for discussions after talks given on this work.

References

1 Zena M. Ariola and Hugo Herbelin. Minimal classical logic and control operators. In *Thirtieth International Colloquium on Automata, Languages and Programming, ICALP*, pages 871–885. Springer-Verlag, LNCS vol 2719, 2003.

2 Philippe de Groote. On the relation between lambda-mu calculus and the syntactic theory of sequential control. In *Logic Programming and Automated Reasoning, 5th international conference LPAR'94*, pages 31–43, 1994.

3 M. Felleisen, D. Friedman, E. Kohlbecker, and B. Duba. A syntactic theory of sequential control. *Theoretical Computer Science*, 52(3):205–237, 1987.

4 Melvin C. Fitting. *Intuitionistic Logic Model Theory and Forcing*. North-Holland, 1969.

5 Jean-Yves Girard. A new constructive logic: classical logic. *Math. Structures in Comp. Science*, 1:255–296, 1991.

6 Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and Types*. Cambridge University Press, 1989.

7 Timothy Griffin. The formulae-as-types notion of control. In *17th Annual ACM Symp. on Principles of Programming Languages*, pages 47–57, 1990.

8 Anders Starcke Henriksen. *Adversarial Models for Cooperative Interactions*. PhD thesis, University of Copenhagen, December 2011.

9 J. Lambek and P. J. Scott. *Introduction to Higher Order Categorical Logic*. Cambridge University Press, 1986.

10 Chuck Liang and Dale Miller. Polarized intuitionistic logic. Submitted, 2011.

11 Chuck Liang and Dale Miller. Intuitionistic control logic. Manuscript available online: **http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/iclpaper.pdf**, 2012.

12 Grigori Mints. *A Short Introduction to Intuitionistic Logic*. Kluwer Academica-Plenum Publishers, New York, 2000.

13 C.H. Luke Ong and Charles Stewart. A Curry-Howard foundation for functional computation with control. In *Symposium on Principles of Programming Languages*, pages 215–227, 1997.

14 Michel Parigot. $\lambda\mu$-calculus: An algorithmic interpretation of classical natural deduction. In *LPAR: Logic Programming and Automated Reasoning, International Conference*, volume 624 of *LNCS*, pages 190–201. Springer, 1992.

15 Helena Rasiowa and Roman Sikorski. *The Mathematics of Metamathematics*. Panstwowe Wydawnictwo Naukowe, Warsaw, 1963.

16 E. Ritter, D. Pym, and L. Wallen. Proof-terms for classical and intuitionistic resolution. *Journal of Logic and Computation*, 10(2):173–207, 2000.