Intuitionistic Control Logic

Chuck Liang
Hofstra University
Hempstead, NY

Dale Miller
INRIA & LIX/Ecole Polytechnique
Palaiseau, France

March 2, 2012

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. The new constant requires a simple-yet-significant modification of intuitionistic logic both semantically and proof-theoretically. We define a Kripke-style semantics as well as a topological space interpretation in which the new constant is given a precise denotation. We define a sequent calculus and prove cut-elimination. We then formulate a natural deduction proof system with a term calculus, one that gives a direct, computational interpretation of contraction. This calculus shows that ICL is fully capable of typing programming language control constructs such as *call/cc* while maintaining intuitionistic implication as a genuine connective.

Contents

1 Introduction

It is by now well known that much constructive content can be found in classical proofs and that the Curry-Howard correspondence can be extended by accepting certain classical principles. Since Griffin ([Gri90]) showed the relationship between classical axioms such as $\neg\neg A \Rightarrow A$ and control operators, several *constructive* classical systems have been formulated, including Girard's *LC* proof system [Gir91] and Parigot's *classical deduction* system, from which is derived $\lambda\mu$-calculus [Par92]. Several variants of $\lambda\mu$-calculus have been proposed since (Herbelin and Saurin's manuscript [HS09] includes a summary of these variants). However, the isomorphism between lambda abstraction and intuitionistic implication is a very strong one. If one collapses intuitionistic logic into classical logic altogether and consider 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 is to say that the assumption A is not localized to the left disjunct. The classical implication $A \Rightarrow B$ is equivalent to the forms $\neg A \vee B$ and $\neg(A \wedge \neg B)$ (among others), each of which yields different procedural information in proofs involving them. The constructive meaning of classical logic is dependent on how we *choose* to interpret classical implication.

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

We propose here a new logic that can be described as 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 add not new axioms to intuitionistic logic but a new logical constant for *false*. Specifically, we distinguish between two symbols for *false*: 0 and $\perp$. The constant 0 will have the same meaning as *false* in intuitionistic logic. The two constants will allow us to define two forms of negation: $\sim A$ using 0 and $\neg A$ using $\perp$. For example, $A \vee \neg A$ will be provable but not $A \vee \sim A$. On the other hand, neither form of negation will be "involutive," thus preserving the strength of intuitionistic implication. In the proof theory of ICL, $\perp$ indicates points in a proof where *contraction* and *multiplicative disjunction* can be used. Furthermore, during cut-elimination, when *cut* is permuted above a contraction or the introduction of a multiplicative disjunction, a form of proof transformation similar to the *structural reductions* of $\lambda\mu$-calculus takes place. We give a Kripke-style semantics for ICL, as well as an interpretation in a topological space. The semantic presentation allows us to observe a general property concerning the *admissible rules* of ICL. We formulate a sound and complete sequent calculus that admits cut. We then define a natural deduction system with proof terms, where disjunctions are given a much more computationally meaningful interpretation than mere "injections." These results show that the computational content of non-intuitionistic proofs can be obtained without collapsing all of intuitionistic logic into classical logic.

As our choice of symbols may suggest, the original impetus for using two constants for *false* can be traced to linear logic. Several attempts to unify logics (including much of our own previous work) are based on ideas derived from linear logic and related systems of Girard. In this paper, however, there will be no discussion of "polarity," nor will any notion of "duality" be assumed to exist *a priori.* Aside from the original impetus for $\perp$, the starting point of ICL is the semantics of intuitionistic logic. In terms of Tarski's topological interpretation of intuitionistic logic, specifically in the metric space of real numbers, the constant $\perp$ can be denoted by a set consisting of all real numbers *minus a single point.*

2 Syntax

We focus only on propositional logic in this presentation. We assume that there are countably many atomic formulas. The connectives of ICL are $\wedge$, $\vee$ and $\supset$. There are also the logical constants $\top$, 0 and $\perp$.

Although there are three constants, one *true* and two versions of *false,* ICL should not be confused with a "three-valued" logic. This will be clear from its semantics: there are uncountably many "truth assignments."

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

$$\frac{\Gamma \vdash A \wedge B;\ [\Delta]}{\Gamma \vdash A;\ [\Delta]}\ \wedge E_1 \qquad \frac{\Gamma \vdash A \wedge B;\ [\Delta]}{\Gamma \vdash B;\ [\Delta]}\ \wedge E_2 \qquad \frac{\Gamma \vdash A \vee B;\ [\Delta] \quad A, \Gamma \vdash C;\ [\Delta] \quad B, \Gamma \vdash C;\ [\Delta]}{\Gamma \vdash C;\ [\Delta]}\ \vee E$$

$$\frac{\Gamma \vdash A \supset B;\ [\Delta] \quad \Gamma \vdash A;\ [\Delta]}{\Gamma \vdash B;\ [\Delta]}\ \supset E \qquad \frac{\Gamma \vdash 0;\ [\Delta]}{\Gamma \vdash A;\ [\Delta]}\ 0E$$

$$\frac{\Gamma \vdash A;\ [\Delta] \quad \Gamma \vdash B;\ [\Delta]}{\Gamma \vdash A \wedge B;\ [\Delta]}\ \wedge I \qquad \frac{\Gamma \vdash A;\ [B, \Delta]}{\Gamma \vdash A \vee B;\ [\Delta]}\ \vee I_1 \qquad \frac{\Gamma \vdash B;\ [A, \Delta]}{\Gamma \vdash A \vee B;\ [\Delta]}\ \vee I_2$$

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

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

Figure 1: The Natural Deduction System NJC, without terms

Intuitionistic Negation: $\sim A \ = \ A \supset 0$

Classical Negation: $\neg A \ = \ A \supset \bot$

We wish to present ICL using a balance of syntax and semantics. Most of the proof theory of ICL will be given after we have defined the meaning of formulas. However, for those who wish to see a proof system before semantics, we first present in Figure 1 the natural deduction system *NJC*. This system is given without terms, which are introduced in section 6. This is clearly a natural deduction system, although presented using the syntax of sequents. 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, Δ represents $\{B\} \cup \Delta$ and does not preclude the possibility that $B \in \Delta$; thus contraction is obviated in these contexts. The formula A in $\Gamma \vdash A; [\Delta]$ is called the *current formula.* There is always exactly one current formula. A formula A is provable if $\vdash A; [\,]$ is provable. The following is a sample proof:

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

Except for the *Esc* and *Con* rules (for explicit contraction of the current formula), this proof system is isomorphic to the usual natural deduction system (NJ) for intuitionistic logic. 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. Without *Esc*, the $\vee$-introduction rules are no different from the additive forms found in NJ. An NJC proof can be considered to consist of segments of intuitionistic proofs joined by *Esc*. It holds that a formula that does not contain $\bot$ as a subformula is provable if and only if it is intuitionistically provable (this will become clear in section 4, where we formulate an equivalent sequent calculus that enjoys the subformula property). A formula containing $\bot$ may also 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. ICL loses the disjunction property for formulas that contain $\bot$, but will gain much in return.

We leave further discussion of this system and other proof-theoretical properties to later sections.

3 Kripke Semantics

We give a Kripke-style semantics for ICL, in which some possible worlds may validate $\bot$, but there will be no model for $\bot$. We consider only Kripke frames that are (finitely) rooted trees: 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 $\models$ maps 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. A formula is *consistent* if it is valid in some model. Both 0 and $\bot$ are inconsistent.

Since intuitionistic models can also be assumed to have rooted frames, it is easy to show that

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

Sample Truths and Falsehoods

As expected, many *but not all* axioms of classical logic become valid in ICL if written with $\bot$.

$\mathbf{A} \vee \neg \mathbf{A}$ is valid.

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 $q \models \bot$. In fact, it holds that $\mathbf{r} \models A$ *if and only if* $\mathbf{r} \not\models \neg A$. In contrast, the purely intuitionistic formula $\sim A \vee A$ has the usual two-world countermodel, since a r-model can be interpreted as a regular model of intuitionistic logic.

$(\neg \mathbf{P} \supset \mathbf{P}) \supset \mathbf{P}$ is valid.

We refer to this formula as *our version of Peirce's law.* The well-known formula $((P \supset Q) \supset P) \supset P$ is not valid in ICL because it is not valid in intuitionistic logic. It becomes valid if Q is replaced with $\bot$. This is provable by contradiction. Assume that for some possible world u in some model, $u \models \neg P \supset P$ but that $u \not\models P$. But since $u \models P \vee \neg P$, this means that $u \models \neg P$, and thus by assumption it must be that $u \models P$.

In Section 6 we will show that the natural deduction proof of this formula does indeed allow us to simulate the *call/cc* operator.

The De Morgan laws in terms of ¬ are valid

The case of the De Morgan law that is not intuitionistically provable is $\neg(A \wedge B) \supset (\neg A \vee \neg B)$. To show its validity in ICL, it is enough to consider an arbitrary root world $\mathbf{r}$. Since $\mathbf{r} \not\models \bot$, if $\mathbf{r} \models \neg(A \wedge B)$ then $\mathbf{r} \not\models A \wedge B$, which means that $\mathbf{r} \not\models A$ or $\mathbf{r} \not\models B$. But $\mathbf{r} \models P \vee \neg P$ for any P, and thus $\mathbf{r} \models \neg A$ or $\mathbf{r} \models \neg B$. The other cases of the De Morgan laws with respect to $\vee$ and $\wedge$ are already valid intuitionistically (if one treats $\bot$ as in minimal logic), and thus are also valid in ICL.

$\neg\neg\mathbf{A} \equiv (\bot \vee \mathbf{A})$

By "$P \equiv Q$" we mean that $(P \supset Q) \wedge (Q \supset P)$ is valid. This equivalence is also provable by the definition of $\models$. The significance of this equivalence is that a double-negation, which indicates a shift to classical logic, can be replaced by a form that requires a very different proof structure (see the *Esc* inference rule in Section 2, and in later sections.)

$\neg\neg\mathbf{A} \supset \mathbf{A}$ is *not* valid.

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 $q \models \bot$, but $q \not\models \neg\neg a \supset a$.

One should not expect this axiom to be valid because *ex falso quodlibet* only holds for 0 in ICL ($0 \supset A$ is valid), but not for $\bot$. Yet another explanation is that $\supset$ is *intuitionistic implication.* We can define "classical implication" as a derived connective:

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

If we replaced $\supset$ with $\Rightarrow$, then $\neg\neg A \Rightarrow A$ becomes abbreviation for $\neg\neg\neg A \vee A$, which is valid just as $\neg A \vee A$ is valid. If we attempted to define classical negation in terms of $\Rightarrow$, we will find that both $A \Rightarrow \bot$ and $A \Rightarrow 0$ are equivalent to $\neg A$. Thus $\neg$ is correctly identified as classical negation.

Unlike in intuitionistic logic, $\neg A \vee B$ is in fact equivalent to $\neg(A \wedge \neg B)$, so it is not necessary to use a "negative" translation. Other equivalent forms of classical implication include $A \supset (B \vee \bot)$ and $A \supset \neg\neg B$. These equivalences can all be verified semantically. 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 its strength. In contrast, $\sim\sim(A \supset B)$ collapses to a classical implication by Glivenko's theorem.

$\sim\neg\mathbf{A} \supset \mathbf{A}$ is valid.

The presence of both 0 and $\bot$ allows us to consider additional forms of double-negation. The formulas $\neg\neg A \supset A$, $\sim\sim A \supset A$ and $\neg\sim A \supset A$ are all invalid. However, $\sim\neg A \supset A$, which is $((A \supset \bot) \supset 0) \supset A$, is valid (but its converse is not valid). Proof-theoretically speaking, both contraction and *ex falso quodlibet* are available in its derivation. This formula will allow us to emulate the $\mathcal{C}$ control operator (see Section 6).

The Hilbert axiom in the form $(\neg\mathbf{B} \supset \neg\mathbf{A}) \supset (\mathbf{A} \supset \mathbf{B})$ is *not* valid.

The reasons generalize the case of $\neg\neg A \supset A$. The axiom is valid if one replaces $\supset$ with $\Rightarrow$ (our definition of $A \Rightarrow B$ as $\neg A \vee B$ is therefore complete with respect to classical logic). Indeed the validity of this axiom (with $\supset$) would indicate a collapse into classical logic. ICL is a *unified* logic, where classical negation exists alongside intuitionistic implication without either destroying the other.

A General Law of Admissible Rules

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.*

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 $\mathbf{r} \not\models \bot$. 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 do not give us a general admissibility property for intuitionistic logic.

The proof of this general rule of admissibility is an advantage of the Kripke semantics, which affords the easiest proof, compared to other forms of semantics we have considered, and to proof theory.

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

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

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

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

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

Figure 2: The Sequent Calculus LJC

4 Sequent Calculus and Cut-Elimination

The sequent calculus LJC is derived from NJC by replacing the elimination rules with left-introduction rules. We retain the additive treatment of context; the notation A, Θ does not preclude the possibility that $A \in \Theta$. The right-side introduction rules of NJC are all preserved in LJC, including *Esc*, *Con* and *Id*. We give the entire set of rules in Figure 2.

The $\bot L$ rule is only needed if one wishes to apply the *Id* rule only to atomic formulas. One could consider an alternative proof system where $\bot L$ and *Esc* are replaced by the following rules:

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

That is, allow weakening on the current formula, *but only when it is* $\bot$. However, from a computational standpoint (see Section 6) this alternative does not introduce any additional *content* into proofs. It will also unnecessarily complicate the technical arguments to follow. Therefore we will retain LJC as defined.

Contractions inside Γ and Δ are obviated by the use of sets. The admissibility of weakening is formalized in the following lemma:

Lemma 3 *If* $\Gamma \vdash A; [\Delta]$ *is provable, then* $\Gamma'\Gamma \vdash A; [\Delta\Delta']$ *is also provable. Furthermore, the proofs can have the same height and structure.*

By "same structure" we mean that the two proofs are isomorphic: the same inference rules on the same principal formulas are used in each step. This lemma is proved by induction on the height of proofs.

The equivalence between LJC and NJC will follow from cut-elimination: since the rules that are unique in NJC/LJC occur on the *right* hand side of sequents, the translation between elimination rules and left-introduction rules follow the same arguments as for intuitionistic logic.

Sequent calculus rules have the subformula property (analytic tableau). With this property it can easily be shown that there are unprovable sequents, including $\vdash \bot; [\,]$.

Cut-Elimination

Along with the usual reasons, we require cut-elimination to make an important argument in the proof of semantic completeness. We need to show that in every *maximally* $\bot$-consistent set of formulas Γ, given a formula A, exactly one of A or $\neg A$ is in Γ. This argument follows from the provability of $A \vee \neg A$ and the admissibility of cut. Then the addition of any formula to Γ will result in a $\bot$-inconsistent set: these sets will represent the possible worlds properly above the root $\mathbf{r}$.

The cut rule for LJC is the following:

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

We augment the usual inductive measure for cut-elimination proofs by using a lexicographical ordering on the tuple (s, nc, h), which is associated with each cut. Each value has the following meaning, in order of importance:

1. **s:** the size of the cut formula (the number of atoms, constants and connectives)
2. **nc:** the number of explicit contractions (*Con* rules) on the the cut formula in the subproofs of the cut.
3. **h:** the sum of the heights of the two subproofs of the cut.

We note that any consecutive applications of the *Con* rule on the same formula are redundant, since we use sets to represent the right-side context inside [ ]. Thus on any branch of a proof tree it can be assumed that there is at most one *Con* rule for each formula.

Not surprisingly, many cases of the proof are the same as for intuitionistic sequent calculus (LJ and variants). Although we have verified all cases, in this presentation we focus on the cases that are different from intuitionistic logic.

We distinguish *parametric cuts,* where at least one cut formula is not principal, and *key cuts* where the cut-formula is principal in both rules. By "principal formula" we mean the formula introduced by an introduction rule, or the formula that is subject to a contraction *Con* rule, or $\perp$ in the case of *Esc*. As usual we start with the uppermost cuts, which has only cut-free subproofs.

First, we verify the parametric cases by showing that these cases are all reducible to cuts with smaller height measures. This will also imply that all cuts can be reduced to the key cases. It is noteworthy that the special rules *Esc* and *Con* do not cause any extra difficulty in the parametric cases. For example:

$$\frac{\Gamma \vdash A;\ [\Delta] \quad \dfrac{A, \Gamma' \vdash B;\ [\Delta']}{A, \Gamma' \vdash \perp;\ [B, \Delta']}\ Esc}{\Gamma\Gamma' \vdash \perp;\ [B, \Delta\Delta']}\ cut \qquad \text{permutes to} \qquad \frac{\dfrac{\Gamma \vdash A;\ [\Delta] \quad A, \Gamma' \vdash B;\ [\Delta']}{\Gamma\Gamma' \vdash B;\ [\Delta\Delta']}\ cut}{\Gamma\Gamma' \vdash \perp;\ [B, \Delta\Delta']}\ Esc$$

The other parametric cases are similar and offer no surprises. In all cases the height measure of cuts is reduced. Thus we can assume that all cuts can be reduced to key cases.

The key cases, where the cut formulas are the conclusions of introduction rules, proceed as in intuitionistic sequent calculus. An exception is the $\vee$ case, because we have chosen to give this connective a computationally more meaningful interpretation (had we chosen the additive form of $\vee$, then this case will be the same as in LJ). This case is handled similarly to the case of contraction, where the right-side cut formula is the conclusion of *Con* (see below). The fact that the principal formula persists in the premises is handled in a typical way. The extra "copies" of the principal formula in the premises are first cut away by cuts with lower height measures, followed by cuts on the subformulas of the principal formula, which reduces the "main instance." We show one representative case:

$$\frac{\dfrac{A, \Gamma \vdash B;\ [\Delta]}{\Gamma \vdash A \supset B;\ [\Delta]}\ \supset R \quad \dfrac{A \supset B, \Gamma' \vdash A;\ [\Delta'] \quad B, A \supset B, \Gamma' \vdash D;\ [\Delta']}{A \supset B, \Gamma' \vdash D;\ [\Delta']}\ \supset L}{\Gamma\Gamma' \vdash D;\ [\Delta\Delta']}\ cut$$

We have made explicit the persistence of the principal formula in the left premise. This cut reduces to:

$$\frac{\dfrac{\dfrac{\Gamma \vdash A \supset B;\ [\Delta] \quad A \supset B, \Gamma' \vdash A;\ [\Delta']}{\Gamma\Gamma' \vdash A;\ [\Delta\Delta']}\ cut \quad A, \Gamma \vdash B;\ [\Delta]}{\Gamma\Gamma' \vdash B;\ [\Delta\Delta']}\ cut \quad \dfrac{\Gamma \vdash A \supset B;\ [\Delta] \quad B, A \supset B, \Gamma' \vdash D;\ [\Delta']}{B, \Gamma\Gamma' \vdash D;\ [\Delta\Delta']}\ cut}{\Gamma\Gamma' \vdash D;\ [\Delta\Delta']}\ cut$$

The uppermost cuts have smaller height measures while the lower cuts have smaller cut formulas.

The above case is in fact no different from that of a typical intuitionistic cut-elimination proof. We now show the cases that distinguish NJC from regular intuitionistic logic. The leading case of interest is when the cut formula is $\bot$ and is the conclusion of an *Esc* rule:

$$\dfrac{\dfrac{\Gamma \vdash A;\ [\Delta]}{\Gamma \vdash \bot;\ [A,\Delta]}\,Esc \qquad \bot,\Gamma' \vdash B;\ [\Delta']}{\Gamma\Gamma' \vdash B;\ [A,\Delta\Delta']}\,cut$$

This cut is reduced as follows. We identify in the right subproof inference rules in which the instance of $\bot$ on the left-hand side is active. Since no left-introduction rule exists for $\bot$, this means that it must be an initial rule: $\bot,\Gamma_1 \vdash \bot;\ [\Delta_1]$. If no such initial rules exist for this particular instance of $\bot$, then the conclusion of the cut follows from weakening. Each such instance is replaced by a copy of the left subproof of the cut, after weakening. The transformation that takes place has the following form:

$$\dfrac{\dfrac{\Gamma \vdash A;\ [\Delta]}{\Gamma \vdash \bot;\ [A,\Delta]}\,Esc \qquad \dfrac{\dfrac{\overline{\bot,\Gamma_1 \vdash \bot;\ [\Delta_1]}}{\vdots}}{\bot,\Gamma' \vdash B;\ [\Delta']}}{\Gamma\Gamma' \vdash B;\ [A,\Delta\Delta']}\,cut \qquad \longrightarrow \qquad \dfrac{\dfrac{\dfrac{\Gamma\Gamma_1 \vdash A;\ [\Delta\Delta_1]}{\Gamma\Gamma_1 \vdash \bot;\ [A,\Delta\Delta_1]}\,Esc}{\vdots}}{\Gamma\Gamma' \vdash B;\ [A,\Delta\Delta']}$$

The provability of the new premise follows from the admissibility of weakening. That is, a parametric cut between $\Gamma \vdash \bot;[A,\Delta]$ and $\bot,\Gamma' \vdash B;[\Delta']$ can be permuted upwards until the occurrence of $\bot$ is active in a $Id/\bot L$ rule. Thus the transformed structure is a valid proof.

The other crucial case of cut elimination pertains directly to the explicit contraction rule *Con*. Any consecutive applications of *Con* on the same formula can be merged (since the [ ] context is a set). We then transform all instances of cut where the right-side cut formula is the conclusion of a *Con* rule:

$$\dfrac{\dfrac{\Gamma \vdash A;\ [A,\Delta]}{\Gamma \vdash A;\ [\Delta]}\,Con \qquad A,\Gamma' \vdash B;\ [\Delta']}{\Gamma\Gamma' \vdash B;\ [\Delta\Delta']}\,cut$$

The subproof above *Con* may contain instances of *Esc* with active formula A. In fact, these instances can be "stacked:"

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma_1 \vdash A;\ [A,\Delta_1]}{\Gamma_1 \vdash \bot;\ [A,\Delta_1]}\,Esc}{\vdots}}{\Gamma_2 \vdash A;\ [A,\Delta_2]}}{\Gamma_2 \vdash \bot;\ [A,\Delta_2]}\,Esc}{\vdots}}{\Gamma \vdash A;\ [A,\Delta]}}{\Gamma \vdash A;\ [\Delta]}\,Con$$

Note however, once A has been added to the [ ]-context (viewing the proof from the bottom), further contractions on A are superfluous, and thus we can assume that no further *Con* rules on A will appear in the subproof. Furthermore, in the uppermost instance of *Esc*, A can be deleted from the [ ]-context, since it will not be "used" again. This cut is transformed as follows (for each branch of the left subproof):

$$
\dfrac{\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{\Gamma_1 \vdash A;\ [\Delta_1] \quad A,\Gamma' \vdash B;\ [\Delta']}{\Gamma_1\Gamma' \vdash B;\ [\Delta_1\Delta']}\,cut}{\Gamma_1\Gamma' \vdash \bot;\ [B,\Delta_1\Delta']}\,Esc \\ \vdots\end{array}}{\Gamma_2\Gamma' \vdash A;\ [B,\Delta_2\Delta']} \quad A,\Gamma' \vdash B;\ [\Delta']}{\Gamma_2\Gamma' \vdash B;\ [B,\Delta_2\Delta']}\,cut}{\Gamma_2\Gamma' \vdash \bot;\ [B,\Delta_2\Delta']}\,Esc \\ \vdots\end{array}}{\Gamma\Gamma' \vdash A;\ [B,\Delta\Delta']} \quad A,\Gamma' \vdash B;\ [\Delta']}{\Gamma\Gamma' \vdash B;\ [B,\Delta\Delta']}\,cut}{\Gamma\Gamma' \vdash B;\ [\Delta\Delta']}\,Con
$$

For each new *cut* introduced, we can assume that the right-side cut formula A is not again the conclusion of *Con*, since such contractions are redundant in the original proof and can be eliminated. Thus the new cuts introduced have fewer contractions on the cut formula in their subproofs, thus reducing the inductive measure. The key case for $\vee$ is similar: we permute the cuts into points of the subproofs indicated by *Esc*. Each new cut has a smaller cut formula (also see Section 6). This completes the cut-elimination proof. $\Box$

5 Soundness and Completeness

The soundness of the proof systems with respect to the Kripke semantics is not as straightforward as one might expect. We must be careful in interpreting a sequent as a single formula. In intuitionistic logic, $(A \supset B) \vee D$ is not equivalent to $A \supset (B \vee D)$, but the corresponding equivalence does hold in classical logic. It would be a mistake to interpret the right-hand side of sequents, $A; [\Delta]$, as a disjunction over A and the formulas in Δ since we do not know if the sequent in question has an intuitionistic proof.

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 follows:

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

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.

The reader may observe that this interpretation of sequents is similar to a double-negation translation. However, the "translation" is only used for the sake of a semantic argument, and has no impact on the structure of proofs. We do not believe that there is a translation from ICL to intuitionistic logic (or even to linear logic - see Section 8) that would preserve the structure of proofs, including at least some of the structure of the cut-elimination procedure.

Theorem 4 *(Soundness) If a formula is provable then it is valid in all r-models.*

Proof By induction on the structure of LJC proofs (NJC can also be used). Given the interpretation of sequents, the cases of most of the inference rules use the same arguments as for intuitionistic logic: that is to say that they are straightforward. We show the cases that are unique to ICL.

For the *Esc* rule, if the premise is provable then by inductive hypothesis $(\neg\check{\Delta} \wedge \hat{\Gamma}) \supset A$ is valid in all models. We need to show the same for the formula $(\neg(A \vee \check{\Delta}) \wedge \hat{\Gamma}) \supset \bot$. For any world u in any model, if $u \models \neg(A \vee \check{\Delta}) \wedge \hat{\Gamma}$ then $u \models \hat{\Gamma}$ and $u \models \neg(A \vee \check{\Delta})$. This implies that if $u \models A$ then $u \models \bot$, and if $u \models \check{\Delta}$ then $u \models \bot$. The same is true for all $v \succeq u$, and thus $u \models \neg A$ and $u \models \neg\check{\Delta}$. But then $u \models A$ by the inductive hypothesis, so $u \models \bot$.

For one of the $\vee R$ ($\vee I$) rules, assume that $(\neg(A \vee \check{\Delta}) \wedge \hat{\Gamma}) \supset B$ is valid. We need to show that $(\neg\check{\Delta} \wedge \hat{\Gamma}) \supset A \vee B$ is also valid. For any possible world u, we know that $u \models A \vee \neg A$. Thus if $u \models \neg\check{\Delta} \wedge \hat{\Gamma}$ but $u \not\models A$, then $u \models \neg A$. We can derive from this that $u \models \neg(A \vee \check{\Delta})$, and thus $u \models B$ and therefore $u \models A \vee B$.

The case for the contraction *Con* rule is similar: we show the result by contradicting the assumption that $u \not\models A$, since that would imply $u \models \neg A$, which by inductive hypothesis implies that $u \models A$. □

Our completeness proof for LJC makes critical use of cut-elimination. The organization of our proof is loosely based on Fitting [Fit69]. We use the notation $\Gamma \not\vdash A;\ [\Delta]$ to indicate that the sequent in question is not provable (in which case we also say that $\Gamma \not\vdash A;\ [\Delta]$ is *consistent*). Our tableau is constructed from structures consisting of a tuple of sets of formulas (Γ, Δ, Θ), with Θ non-empty. We refer to such a structure as an *antisequent*. An antisequent is *consistent* if $\Gamma \not\vdash A; [\Delta]$ for all formulas $A \in \Theta$.

A set of formulas Γ is $\perp$-consistent if $\Gamma \not\vdash \perp; [\,]$. Our strategy is to construct a *maximally $\perp$-consistent* set to represent the root of a countermodel.

Given an *unprovable* formula A^0, we fix an enumeration of all subformulas of A^0 as $B_1, B_2, \ldots$. For each B_i we also consider its classical negation $\neg B_i$. For this purpose we define the enumeration $A_1, A_2, A_3, A_4, \ldots = B_1, \neg B_1, B_2, \neg B_2, \ldots$. We also assume that this enumeration has the property that if B_i is smaller (has fewer symbols) than B_j, the $i > j$, i.e, larger subformulas (and their negations) are enumerated first (this assumption is not strictly required but it simplifies some of the subsequent arguments). We use the following pseudo-procedure to show the existence of a *saturated* antisequent.

1. Let $\Gamma_0 = \Delta_0 = \{\}$. Let $\Theta_0 = \{A^0, \perp\}$. Let the antisequent $S_0 = (\Gamma_0, \Delta_0, \Theta_0)$. S_0 is consistent since $\vdash \perp; [\,]$ has no proof and $\vdash A; [\,]$ is assumed to have no proof.

2. For each $i > 0$ and formula A_i, if $(A_i, \Gamma_{i-1}, \Delta_0, \Theta_0)$ is consistent, let $\Gamma_i = A_i, \Gamma_{i-1}$; otherwise, let $\Gamma_i = \Gamma_{i-1}$. Let $\Gamma^* = \bigcup \Gamma_i$.

3. For each $i > 0$ and subformula A_i, if $(\Gamma^*, A_i, \Delta_{i-1}, \Theta_0)$ is consistent, let $\Delta_i = A_i, \Delta_{i-1}$; otherwise, let $\Delta_i = \Delta_{i-1}$. Let $\Delta^* = \bigcup \Delta_i$.

4. For each $i > 0$ and subformula A_i, if $(\Gamma^*, \Delta^*, A_i, \Theta_{i-1})$ is consistent, let $\Theta_i = A_i, \Theta_{i-1}$; otherwise, let $\Theta_i = \Theta_{i-1}$. Let $\Theta^* = \bigcup \Theta_i$.

5. Let $R = S_0^* = (\Gamma^*, \Delta^*, \Theta^*)$

It can be shown that R satisfies the following properties:

Lemma 5 *For S_0^* as defined,*

1. *S_0^* remains consistent.*

2. *If $A \wedge B \in \Theta^*$, then either $A \in \Theta$ or $B \in \Theta^*$,* since at least one premise of $\wedge R$ with $A \wedge B$ as principal formula must be consistent.

3. *If $A \vee B \in \Theta^*$ then both $A \in \Theta^*$ and $B \in \Theta^*$.*

4. *If $A \wedge B \in \Gamma^*$ then $A \in \Gamma^*$ and $B \in \Gamma^*$.*

5. *If $A \vee B \in \Gamma^*$ then either $A \in \Gamma^*$ or $B \in \Gamma^*$*

6. *If $A \supset B \in \Gamma^*$ then either $A \in \Theta^*$ or $B \in \Gamma^*$.*

7. *A formula F is in at most one of the sets Γ^* and Θ^*,* by virtue of the *Id* rule.

8. *$\Delta^* = \Theta^*$.* Since $\perp \in \Theta^*$ and the antisequent is consistent, the premise of all instances of *Esc*-rules must also be consistent (unprovable), thus $F \in \Delta^*$ implies $F \in \Theta^*$. Conversely, by virtue of the *Con* rule, if $F \in \Theta$ then $F \in \Delta$.

9. ***Exactly one* of B_i and $\neg B_i$ is always in Γ^*, for each subformula B_i of A^0.**

   Both cannot be in Γ^* because $\bot \in \Theta_0$. Since $\Gamma \vdash B_i \vee \neg B_i; [\Delta]$ is always provable, if $B_i \vee \neg B_i, \Gamma \vdash A^0; [\Delta]$ is provable, then *by cut-elimination* $\Gamma \vdash A^0; [\Delta]$ is also provable. Thus the consistency of $\Gamma \not\vdash A^0; [\Delta]$ implies the consistency of $B_i \vee \neg B_i, \Gamma \not\vdash A^0; [\Delta]$, and therefore either $B_i, \Gamma \not\vdash A^0; [\Delta]$ or $\neg B_i, \Gamma \not\vdash A^0; [\Delta]$.

The last property is most crucial and is why we first gave a cut-elimination proof. It means that for any unprovable formula A^0, there is a saturation that is *maximally* $\bot$-consistent. In addition, this saturation have the properties of a Hintikka set, up to $A \supset B \in \Theta^*$.

Note that the saturation S^* can be defined for any consistent antisequent S in exactly the same way. However, above R the content of Δ^* would be meaningless since Γ^* would become $\bot$-inconsistent. All the Hintikka-set properties of Lemma 5 are retained except for properties 8 and 9, which only apply to R.

The countermodel for A^0 is formed by defining a sequence of saturated , consistent antisequents $S_0^*, S_1^*, S_2^* \ldots$ as follows:

1. Given $S_i^* = (\Gamma_i, \Delta_i, \Theta_i)$, enumerate all formulas of the form $A \supset B \in \Theta_i$ as $G_1, \ldots G_n$.
2. For each $G_j = A \supset B$, let $S_{i+j} = (\{A\} \cup \Gamma_i, \Delta_i, \{B\})$ and form the saturation S_{i+j}^*, but ignoring Δ_i ($\Delta_{i+1} = \Delta_i$). This defines $S_{i+1}^*, \ldots S_{i+n}^*$.
3. Now enumerate all formulas of the form $A \supset B \in \Theta_{i+1}$ as $G_1', \ldots, G_k'$ and form $S_{i+n+1}^*, \ldots, S_{i+n+k}^*$ as above. Repeat exhaustively.

For propositional logic, by the subformula property the number of possible antisequents consisting of the subformulas of a designated formula is clearly bounded, therefore giving rise to a finite sequence $S_0^*, \ldots, S_m^*$. This sequence forms a *saturated tableau.*

Above R, this argument is no different from regular intuitionistic logic: the consistency of $\Gamma \not\vdash A \supset B; [\Delta]$ requires forming a new possible world above the one represented by Γ. The key difference here is that, since Γ^* in $S_0^* = (\Gamma^*, \Delta^*, \Theta^*)$ is *maximally* $\bot$-consistent, either $A, \Gamma^* = \Gamma^*$ or A, Γ^* is $\bot$-inconsistent. Furthermore, if $A, \Gamma^* = \Gamma^*$, then it must also hold that $\{B\}^* = \Theta^*$, since $\Gamma^* \not\vdash B;\ [\Delta^*]$ is assumed consistent. This means that, in fact, $S_j = S_0$ if $G_j = A \supset B$ and $A, \Gamma^* = \Gamma^*$. The countermodel is formalized as follows:

Given $S_0^*, S_1^*, S_2^* \ldots S_m^*$ as defined above for an unprovable formula A^0, form the r-model $\langle \mathbf{W}, \mathbf{r}, \preceq, \models \rangle$ by setting $\mathbf{W} = S_0^*, \ldots, S_m^*$ and $\mathbf{r} = S_0^*$; let $S_i^* \preceq S_j^*$ if $\Gamma_i \subseteq \Gamma_j$ and set $S_i^* \models a$ if $a \in \Gamma_i$ for atomic formulas a. The $\models$ relation is then extended to all formulas as defined.

Lemma 6 *The structure $\langle \mathbf{W}, \mathbf{r}, \preceq, \models \rangle$ formed from the saturated tableau $S_0^*, \ldots, S_m^*$ is a r-model, and the following holds for each $S_i^* = (\Gamma_i, \Delta_i, \Theta_i)$*

1. *if $A \in \Theta_i$ then $S_i^* \not\models A$*
2. *if $A \in \Gamma_i$ then $S_i^* \models A$*

Proof That the structure is indeed a r-model follows easily from the above definitions and arguments. In particular, because the enumeration of the subformulas of A^0 and their negations, $A_1, A_2, \ldots$, is *fixed,* when $\Gamma_i = \Gamma_j$ it follows that $\Theta_i = \Theta_j$, and thus $\preceq$ is a partial ordering relation.

The proof of the two listed properties is by a simultaneous induction on the structure of formulas. We detail the important cases.

For an atomic formula a, if $a \in \Gamma_i$ then $S_i^* \models a$ by definition; if $a \in \Theta_i$ then $a \notin \Gamma_i$ for else the antisequent would be inconsistent, and thus $S_i^* \not\models a$.

For $A \supset B \in \Gamma_i$, we also have that $A \supset B \in \Gamma_j$ for all $\Gamma_j \supseteq \Gamma_i$. But then (by lemma 5) either $A \in \Theta_j$ or $B \in \Gamma_j$ so by inductive hypothesis $S_j^* \not\models A$ or $S_j^* \models B$, for all $S_j^* \succeq S_i^*$. Thus by definition of $\models$, $S_i^* \models A \supset B$. On the other hand, if $A \supset B \in \Theta_i$ then there exists a $S_j^* \succeq S_i^*$ with $A \in \Gamma_j$ and $B \in \Theta_j$. Thus by inductive hypothesis $S_j^* \models A$ but $S_j^* \not\models B$ and thus $S_i^* \not\models A \supset B$ by definition of $\models$. The other connectives are similar.

If $\bot \in \Gamma_i$, it cannot be the case that $\Gamma_i = \Gamma_0$ since $\bot \in \Theta_0$ and S_0^* is consistent. Thus $S_i^* \succ S_0^*$ (properly above S_0^*) and by definition of $\models$, $S_i^* \models \bot$. If $\bot \in \Theta_i$, then it must be the case that $\Theta_i = \Theta_0$, since otherwise Γ_i is $\bot$-inconsistent (because either $B_i \in \Gamma_0$ or $\neg B_i \in \Gamma_0$). But by definition of $\models$ for $\mathbf{r}$, $S_0^* \not\models \bot$.

□

Completeness in contrapositive form follows directly from the lemma: given an unprovable formula A^0, in the model $\langle \mathbf{W}, \mathbf{r}, \preceq, \models \rangle$ as defined, $\mathbf{r} \not\models A^0$ since $\mathbf{r} = S_0^*$ and $A^0 \in \Theta_0$.

Theorem 7 *(Completeness) If a formula is valid in all r-models then it is provable.*

Because the use of contraction is controlled, we can also derive from completeness and the subformula property of cut-free LJC proofs that (propositional) ICL remains decidable. Starting from an end-sequent, one enumerates all possible proof fragments of height one. Each applicable instance of an inference rule will form a branch in a tree. For each branch we then enumerate all possible proof fragments of height two, and repeat exhaustively. A branch can be terminated if an initial rule is encountered. Repeat sequents on a branch are disregarded. Since we use *sets,* the number of unique sequents formed from subformulas of a given formula is finite. In particular every branch of the tree is finite and thus by König's Lemma, the tree is also finite.

Corollary 8 *ICL in its current propositional form is decidable.*

6 Natural Deduction and $\lambda\gamma$-Calculus

In this section we reformulate the natural deduction system NJC with a proof-term calculus that extends λ-calculus. The basis of this term system is Parigot's $\lambda\mu$-calculus [Par92] and its many variants, notably those found in [dG94], [OS97], and [AH03]. Much of the syntax found here also borrows from $\lambda\mu$ calculus. However, the system is different enough so that we refer to it as $\lambda\gamma$-calculus. The main distinction is that the logical interpretation of terms and reductions are based on ICL and not classical logic. Additionally, contraction and disjunctions are given more computationally meaningful interpretations.

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

- λ-variables x, y, ... and γ-variables a, b, ...
- λ-abstraction $\lambda x.t$ and γ-abstraction $\gamma a.s$
- injective abstractions $\omega^\ell a.s$ and $\omega^r a.s$
- application $(s\ t)$
- pairs (s,t) and projections $(s)_\ell$ and $(s)_r$.
- escapee: $[a]t$

The annotated version of NJC is found in Figure 3. We have chosen a multiplicative treatment of contexts in this presentation because it is more reasonable in practice, 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. In an escaped term $[a]t$, a is a γ-variable that labels the term t. We assume that all bound variables are renamed whenever necessary. Since formulas are now indexed, contractions inside the left/right contexts are nolonger obviated if two instances of the same formula have different indices. However, we note that the following lemma is provable by a simple induction:

Lemma 9 *If* $s : A^x, A^y, \Gamma \vdash B; [C^d, C^e, \Delta]$ *is provable, then* $s[x/y][d/e] : A^x, \Gamma \vdash B; [C^d, \Delta]$ *is also provable;*

$$\frac{s : \Gamma \vdash A;\ [\Delta] \quad t : \Gamma' \vdash B;\ [\Delta']}{(s,t) : \Gamma\Gamma' \vdash A \wedge B;\ [\Delta\Delta']}\ \wedge I \qquad \frac{s : \Gamma \vdash A;\ [B^d, \Delta]}{\omega^\ell d.s : \Gamma \vdash A \vee B;\ [\Delta]}\ \vee I_1 \qquad \frac{s : \Gamma \vdash B;\ [A^d, \Delta]}{\omega^r d.s : \Gamma \vdash A \vee B;\ [\Delta]}\ \vee I_2$$

$$\frac{t : A^x, \Gamma \vdash B;\ [\Delta]}{(\lambda x.t) : \Gamma \vdash A \supset B;\ [\Delta]}\ \supset I \qquad \frac{s : \Gamma \vdash A \wedge B;\ [\Delta]}{(s)_\ell : \Gamma \vdash A;\ [\Delta]}\ \wedge E_1 \qquad \frac{s : \Gamma \vdash A \wedge B;\ [\Delta]}{(s)_r : \Gamma \vdash B;\ [\Delta]}\ \wedge E_2$$

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

$$\frac{t : \Gamma \vdash A \supset B;\ [\Delta] \quad s : \Gamma' \vdash A;\ [\Delta']}{(t\ s) : \Gamma\Gamma' \vdash B;\ [\Delta\Delta']}\ \supset E \qquad \frac{s : \Gamma \vdash 0;\ [\Delta]}{\mathit{abort}\ s : \Gamma \vdash A;\ [\Delta]}\ 0E \qquad \frac{}{\mathit{exit} : \Gamma \vdash \top;\ [\Delta]}\ \top I$$

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

Figure 3: NJC with terms

We use $s[t/q]$ to represent capture-avoiding substitution in the usual sense, for both γ and λ variables q.

A sample $\lambda\gamma$ proof term is $\omega^\ell d.\lambda x.[d]x$, which proves $\neg A \vee A$. For a less trivial example, we give a proof for the non-intuitionistic case of the De Morgan laws: $\neg(A \wedge B) \supset (\neg A \vee \neg B)$.

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{x : \neg(A \wedge B)^x \vdash \neg(A \wedge B);\ [\,] \qquad \cfrac{y : \neg(A \wedge B)^x, A^y, B^z \vdash A;\ [\,] \quad z : \neg(A \wedge B)^x, A^y, B^z \vdash B;\ [\,]}{(y,z) : \neg(A \wedge B)^x, A^y, B^z \vdash A \wedge B;\ [\,]}\ \wedge I}{x\ (y,z) : \neg(A \wedge B)^x, A^y, B^z \vdash \bot;\ [\,]}\ \supset E}{\lambda z.x\ (y,z) : \neg(A \wedge B)^x, A^y \vdash \neg B;\ [\,]}\ \supset I}{[d]\lambda z.x\ (y,z) : \neg(A \wedge B)^x, A^y \vdash \bot;\ [\neg B^d]}\ Esc}{\lambda y.[d]\lambda z.x\ (y,z) : \neg(A \wedge B)^x \vdash \neg A;\ [\neg B^d]}\ \supset I}{\omega^\ell d.\lambda y.[d]\lambda z.x\ (y,z) : \neg(A \wedge B)^x \vdash \neg A \vee \neg B;\ [\,]}\ \vee I_1}{\lambda x.\omega^\ell d.\lambda y.[d]\lambda z.x\ (y,z) :\ \vdash \neg(A \wedge B) \supset (\neg A \vee \neg B);\ [\,]}\ \supset I$$

The interpretation of disjunctions using a form of abstraction is not so unusual when one considers its similarity to implication: both are "multiplicative disjunctions." Our $\omega^{\ell/r}$-binders are similar to the one of [RPW00], though there are important differences (see Section 8). 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 a left/right injection operator. Intuitively, the computational meaning of a (non-vacuous) ω-abstraction can be thought of as that of *coroutines,* with terms $[d]r$ representing *yield.* Alternatively, one can think of $\omega^\ell d.s$ of type $A \vee B$ as a procedure of type A that can *throw an exception* of type B (and similarly for $\omega^r d.s$), with $[d]r$ representing a throw operation.

Contractions are represented in terms as γ-abstractions of the form $\gamma a.s$ where a is a γ-variable.

Choices in Term Reduction Strategies

To specify term reduction rules, a choice needs to be made in regard to the cut-elimination procedure when right-side contractions are involved. Consider again a right-side cut formula that is subject to contraction:

$$\cfrac{\cfrac{\Gamma \vdash A;\ [A, \Delta]}{\Gamma \vdash A;\ [\Delta]}\ Con \qquad A, \Gamma' \vdash B;\ [\Delta']}{\Gamma\Gamma' \vdash B;\ [\Delta\Delta']}\ cut$$

The cut-elimination procedure for LJC defined in Section 4 is consistent with a call-by-value reduction strategy (the uppermost cuts are reduced first), while normalization in natural deduction does not fix an evaluation strategy. Additionally, in NJC it is not always necessary to permute a cut above a contraction (Con). In sequent calculus, a cut on a compound formula is always decomposed into cuts on subformulas. However, in natural deduction there are no left-introduction rules and thus no "key cases." The redex formed by $\supset E$ is always permuted up to an initial rule, which implies a wholesale substitution of the left subproof into the right subproof, with the original contraction intact. As a result, there are two valid ways to reduce this cut. In the terminology of Parigot, this conflict can be explained in terms of *structural reduction* versus *logical reduction.* The mixture of these strategies must be careful to preserve confluence.

The *cut* rule is also admissible in NJC and can be annotated with terms, as it is a *modus ponens* in thin disguise:

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

It is also possible to define a secondary cut rule:

$$\frac{\Gamma \vdash A; [B, \Delta] \quad B, \Gamma' \vdash C; [\Delta']}{\Gamma\Gamma' \vdash A;\ [C, \Delta\Delta']}\ \mathit{cut}_2$$

The admissibility of cut_2 is by translation to *cut*. However, this type of cut requires a form of *substitution* different from that of β-reduction. Substitution must take place *inside* the *left* subproof of cut_2, at points indicated by Esc, which are represented by terms of the form $[d]w$. That is,

$$\cfrac{\cfrac{\cfrac{\cfrac{r : \Gamma_1 \vdash B;\ [\Delta_1]}{[d]r : \Gamma_1 \vdash \bot;\ [B^d, \Delta_1]}\ Esc}{\vdots}}{t : \Gamma \vdash A;\ [B^d, \Delta]} \qquad s : B^x, \Gamma' \vdash C;\ [\Delta']}{u : \Gamma\Gamma' \vdash A;\ [C^d, \Delta\Delta']}\ \mathit{cut}_2$$

is equivalent to[1]

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

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 substitution operation used for *structural reduction* in $\lambda\mu$-calculus.

Now consider the permutation of cut above contraction in light of these annotations:

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

The above cut reduces to the following, with corresponding effect on the terms:

$$\cfrac{\cfrac{\cfrac{t : \Gamma \vdash A;\ [A^d, \Delta] \quad s : A^x, \Gamma' \vdash B;\ [\Delta']}{t\{[d](\lambda x.s)\ w/[d]w\} : \Gamma\Gamma' \vdash A;\ [B^d, \Delta\Delta']}\ \mathit{cut}_2 \qquad s : A^x, \Gamma' \vdash B;\ [\Delta']}{((\lambda x.s)\ t\{[d](\lambda x.s)\ w/[d]w\}) : \Gamma\Gamma' \vdash B;\ [B^d, \Delta\Delta']}\ \mathit{cut}}{\gamma d.((\lambda x.s)\ t\{[d](\lambda x.s)\ w/[d]w\}) : \Gamma\Gamma' \vdash B;\ [\Delta\Delta']}\ Con$$

This transformation directly suggests a reduction rule of the form

$$(\lambda x.s)\ (\gamma a.t) \ \longrightarrow\ \gamma a.((\lambda x.s)\ t\{[a]((\lambda x.s)\ w)/[a]w\}) \quad (\gamma\text{-reduction})$$

[1] We have annotated the conclusion of cut_2 with the same term as the translated form because we do not wish to introduce another form of *redex,* which would unnecessarily complicate the term rewriting system.

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. $(u,v)\ (\omega^{\ell} d.t) \longrightarrow \gamma d.(u\ t\{[d](v\ w)/[d]w\})$;
   $(u,v)\ (\omega^{r} d.t) \longrightarrow \gamma d.(v\ t\{[d](u\ w)/[d]w\})$ (ω-reduction)
4. $(u,v)\ \gamma d.t \longrightarrow \gamma d.(u,v)\ t\{[d](u,v)w/[d]w\}$ ($\omega\gamma$-reduction)
5. $(u,v)_{\ell} \longrightarrow u$; $(u,v)_r \longrightarrow v$. (projections)
6. $(\gamma d.s)_{\ell} \longrightarrow \gamma d.s_{\ell}\{[d]w_{\ell}/[d]w\}$; $(\gamma d.s)_r \longrightarrow \gamma d.s_r\{[d]w_r/[d]w\}$. ($\gamma$-projections)
7. $(\omega^{r} a.s)\ t \longrightarrow \omega^{r} a.s\{[a](r\ t)/[a]r\}$; $(\omega^{\ell} a.s)\ t \longrightarrow \omega^{\ell} a.s\{[a](r\ t)/[a]r\}$. ($\omega\mu$-reduction)
8. $(abort\ s)\ t \longrightarrow (abort\ s)$. (aborted reduction)
9. $\gamma a.s \longrightarrow s$ when a does not appear free in s. (vacuous contraction)
10. $\gamma a.\gamma b.s \longrightarrow \gamma a.s[a/b]$. ($\gamma$-renaming)
11. $[d]\gamma a.s \longrightarrow [d]s[d/a]$. ($\mu$-renaming)

Figure 4: Term Reduction Rules

However, this rule clashes with β-reduction in the usual sense. Consider the term $(\lambda x.s)\ ((\lambda y.y)\ \gamma a.u)$, which will likely reduce in divergent directions. One way to solve this problem, which has been used by others in similar contexts (including [OS97]), 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, the range of possibilities that the structure of ICL proofs suggest, without fixing a reduction strategy. We make a choice similar to that of the original $\lambda\mu$-calculus, which is to defer to β-reduction in this situation. Instead, redexes of the form $(\gamma d.s)\ t$ are allowed, leading to what we refer to as "$\mu\gamma$-reduction."

First, we list the entire set of reductions rules in Figure 4. These rules define cut-elimination for NJC. Each rule either eliminates a redex/cut by substitution, or reduces the cut to either cuts on smaller formulas (as in $\vee E$) or pushes the redex inside a γ (Con). We discuss these rules below.

Disjunctions in ICL can represent at least the same level of computational content found in $\lambda\mu$-calculus and its variants. When NJC is presented without terms, it may appear that Con can be derived from a combination of $\vee I$ and $\vee E$. With terms, however, it becomes evident that a non-additive $\vee E$ embeds a contraction, which should be considered a primitive operation. A non-intuitionistic proof of $A \vee B$ may contain subproofs of A as well as those of B: a contraction will be needed in $\vee$-elimination to form a proof of $C; [\Delta]$ from a proof of $C; [C, \Delta]$. The γ binder is needed in addition to $\omega^{\ell/r}$ lest we confuse A with $A \vee A$.

In NJC, a $\vee$-elimination "redex" is represented by the application of a pair of λ-terms to a $\omega^{\ell/r}$-abstraction, with the following reduction rule:

$$(\lambda x.s, \lambda y.t)\ (\omega^{\ell} d.u) \longrightarrow \gamma d.(\lambda x.s)\ t\{[d](\lambda y.t)\ w/[d]w\} \quad (\omega\text{-reduction})$$

and similarly for $\omega^{r} d.u$. It corresponds to the normalization/cut-elimination case for $\vee E$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{r : \Gamma_1 \vdash B;\ [\Delta_1]}{[d]r : \Gamma_1 \vdash \bot;\ [B^d, \Delta_1]}\ Esc}{\vdots}}{u : \Gamma \vdash A;\ [B^d, \Delta]}}{\omega^{\ell} d.u : \Gamma \vdash A \vee B;\ [\Delta]}\ \vee I_1 \qquad s : A^x, \Gamma \vdash C;\ [\Delta] \qquad t : B^y, \Gamma \vdash C;\ [\Delta]}{(\lambda x.s, \lambda y.t)\ \omega^{\ell} d.u : \Gamma \vdash C;\ [\Delta]}\ \vee E$$

Here, subproofs of the form $[d]w$ can occur multiple times in the proof u. This proof reduces to:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{r : \Gamma_1 \vdash B;\ [\Delta_1] \qquad t : B^y, \Gamma \vdash C;\ [\Delta]}{(\lambda y.t)\ r : \Gamma_1\Gamma \vdash C;\ [\Delta_1\Delta]}\ cut}{[d](\lambda y.t)\ r : \Gamma_1\Gamma \vdash \bot;\ [C^d, \Delta_1\Delta]}\ Esc}{\vdots}}{u\{[d](\lambda y.t)w/[d]w\} : \Gamma \vdash A;\ [C^d, \Delta]} \qquad s : A^x, \Gamma \vdash C;\ [\Delta]}{\dfrac{(\lambda x.s)\ u\{[d](\lambda y.t)w/[d]w\} : \Gamma \vdash C;\ [C^d, \Delta]}{\gamma d.(\lambda x.s)\ u\{[d](\lambda y.t)w/[d]w\} : \Gamma \vdash C;\ [\Delta]}\ Con}\ cut$$

After further cut-reduction, the term reduces to $\gamma d.s[u\{([d]t[w/y])/[d]w\}/x]$.

The $\mu\gamma$ rule applies a γ-abstraction to a term. This rule represents the permutation of *cut* above *Con* when the cut formula is of the form $A \supset B$ (i.e., a $\supset E$ beneath a *Con*), as well as when it is of the form $A \vee B$ (for $\omega\mu$-reduction) and 0 (for aborted reduction). The cases when the redex above *Con* is a $\vee$-elimination or $\wedge$-elimination are represented by the $\omega\gamma$ rule and the γ-projection rules respectively. All rules are implied to have requirements regarding capture-avoiding substitution (e.g. in the $\mu\gamma$ rule d is not free in t).

The structural reduction rule in $\lambda\mu$-calculus is

$$(\mu a.s)\ t \ \longrightarrow\ \gamma a.s\{[a](w\ t)/[a]w\}$$

The difference between this "μ-reduction" and our "$\mu\gamma$-reduction" can be explained by the following types of inferences (leaving out the contexts Γ, Δ for intuitive clarity):

$$\dfrac{A \supset B; [A \supset B] \qquad A}{B; [B]}\ \mu\gamma \qquad\qquad \dfrac{A, B \supset C \qquad B}{A, C}\ \mu$$

We cannot adopt the $\lambda\mu$ reduction rule directly because γ-abstractions correspond to contractions. Although the original reduction rule of $\lambda\mu$-calculus appears to be more flexible, it is something that can be recovered with another form of ω-reduction.

The $\omega\mu$ reduction rules are the closest counterpart to structural reduction as defined by Parigot, but has a different logical interpretation. The rules correspond to inferences of the following forms:

$$\dfrac{\dfrac{s : \Gamma \vdash A;\ [(B \supset C)^d, \Delta]}{\omega^\ell d.s : \Gamma \vdash A \vee (B \supset C);\ [\Delta]}\ \vee I_1 \qquad t : \Gamma' \vdash B; [\Delta']}{(\omega^\ell d.s)\ t : \Gamma\Gamma' \vdash A \vee C;\ [\Delta\Delta']} \qquad \dfrac{\dfrac{s : \Gamma \vdash B;\ [(A \supset C)^d, \Delta]}{\omega^r d.s : \Gamma \vdash (A \supset C) \vee B;\ [\Delta]}\ \vee I_2 \qquad t : \Gamma' \vdash A; [\Delta']}{(\omega^r d.s)\ t : \Gamma\Gamma' \vdash C \vee B;\ [\Delta\Delta']}$$

These rules, and the corresponding $\omega\mu$ reduction rule, are also admissible by cut-elimination. For example, the rule for ω^ℓ corresponds to:

$$\dfrac{\dfrac{s : \Gamma \vdash A; [(B \supset C)^d; \Delta] \qquad \dfrac{x : (B \supset C)^x, \Gamma' \vdash B \supset C; [\Delta'] \quad t : (B \supset C)^x, \Gamma' \vdash B; [\Delta']}{(x\ t) : (B \supset C)^x, \Gamma' \vdash C; [\Delta']}\ \supset E}{s\{[d](\lambda x.x\ t)\ w/[d]w\} : \Gamma\Gamma' \vdash A;\ [C^d, \Delta\Delta']}\ cut_2}{\omega^\ell d.s\{[d](\lambda x.x\ t)\ w/[d]w\} : \Gamma\Gamma' \vdash A \vee C;\ [\Delta\Delta']}\ \vee I_1$$

Here, x is not free in t and $s\{[d](\lambda x.x\ t)\ w/[d]w\}$ will of course reduce to $s\{[d]w\ t/[d]w\}$. We show later in this section how these reductions can be regarded as consistent with $\vee$-elimination.

The μ-renaming rule is equivalent to a rule found in $\lambda\mu$-calculus. In the context of NJC this rule 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 (Lemma 9).

Projections are a special kind of "application", and require rules similar to γ-reduction, since all terms may be prefixed by γ.

Finally, the rule for *abort* is also justified in the proof theory 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 $\gamma d.(abort\ s)\{[d]w\ t/[d]w\}\ t$, which reduces to $\gamma d.abort\ s\{[d]w\ t/[d]w\}$. It is subject only to structural reduction: the outer application to t is absorbed.

We define a term t to be of type A if $t : \vdash A; [\,]$ is provable. Clearly only closed terms are typed.

Subject reduction is a consequence of the fact that the reduction rules follow cut-elimination and other valid proof transformations, as we have already shown:

Proposition 10 *(Subject Reduction) If t has type A and $t \longrightarrow t'$, then t' also has type A.*

The Computational Content of Contraction and Disjunction

An important proof term is that of our version of Peirce's formula, $((P \supset \bot) \supset P) \supset P$:

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

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 $\mu\gamma$-reduction. Cut-elimination in the presence of a contraction requires reductions inside the $[\,]$ context *as well as* outside. This term can still emulate the *call/cc* construct of programming languages. Call this term $\mathcal{K}$, then $(\mathcal{K}\ M\ k_1\ k_2)$ reduces to a term of the form $\gamma d.(M\ \lambda y.[d](y\ k_1\ k_2))\ k_1\ k_2$. For example, given the term context $E[z] = (z\ k_1\ k_2)$, $E[\mathcal{K}M]$ reduces to $\gamma d.E[M(\lambda y.[d]E[y])]$, which emulates the behavior of *call/cc* (see [dG94] for further analysis of $\lambda\mu$-based systems and control operators).

In contrast to *call/cc,* the $\mathcal{C}$ operator of Felleisen et al. [FFKD87] 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 unprovable, but we can consider proofs of $\sim\neg A \supset A$ and $\neg\neg A \supset (A \vee \bot)$. Each of these formulas presents a solution for emulating $\mathcal{C}$ in NJC. The following term proves $\sim\neg A \supset A$:

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

If this proof term is labeled $\mathcal{C}_1$, then $\mathcal{C}_1 M$, when applied to a term t, is only subject to structural reduction inside the *abort* subterm, by virtue of the rule $(abort\ s)\ t \longrightarrow (abort\ s)$. 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). If $t : \vdash A; [\,]$ is provable in NJC then t must be closed, thus all valid formulas of ICL are proved by closed terms.

The other approach to emulating $\mathcal{C}$ is to make direct use of the enriched interpretation of disjunction. The formula $\neg\neg A \supset (A \vee \bot)$ has the following proof (shortened to avoid repetition):

$$\dfrac{\dfrac{\dfrac{x : \neg\neg A^x \vdash \neg\neg A; [\,] \quad \lambda y.[d]y : \neg\neg A^x \vdash \neg A;\ [A^d]}{(x\ \lambda y.[d]y) : \neg\neg A^x \vdash \bot;\ [A^d]} \supset E}{\omega^r d.(x\ \lambda y.[d]y) : \neg\neg A^x \vdash A \vee \bot;\ [\,]} \vee I_2}{\lambda x.\omega^r d.(x\ \lambda y.[d]y) :\ \vdash \neg\neg A \supset (A \vee \bot);\ [\,]} \supset I$$

Refer to this term as $\mathcal{C}_2$. The $\omega\mu$ reduction rule gives this term the expected behavior of the control operator. $\mathcal{C}_2 M$ reduces to $\omega^r d.M(\lambda y.[d]y)$ and $\mathcal{C}_2 Mt$ reduces to $\omega^r a.M(\lambda y.[a]y\ t)$.

One might argue that the $\omega\mu$ reduction rules were added specifically to accommodate the $\mathcal{C}$ operator. However, with the enriched computational interpretation of disjunction some of the benefits of both *call/cc* and $\mathcal{C}$ can be obtained directly. For example, exception handling can be modeled by an $\vee$-elimination $(\lambda x.(x\ s),\ \lambda y.u)\ (\omega^\ell d.\lambda z.t)$. Here, x is not free in s (thus reversing the application) and $\omega^\ell d.\lambda z.t$ is of type $(A \supset C) \vee B$, i.e., *a procedure of type $A \supset C$ that could throw an exception of type B.* The term s (input) has type A and the *exception handler* $\lambda y.u$ has type $B \supset C$. The redex thus has the intuitive meaning of

try $(\lambda z.t)s$ *catch exception* e *with* $\lambda y.u$.

The term reduces to $\gamma d.t\{[d]\lambda(y.u)e/[d]e\}[s/z]$ and has type C. A vacuous γ would indicate that no exceptions were thrown, in which case the term reduces to $t[s/z]$. Surely this is a more intuitive representation of exception handling compared to using *call/cc.*

In $\lambda\mu$-calculus and related systems, the $\mathcal{C}$ operator is usually typed as $\neg\neg P \Rightarrow P$, with "$\Rightarrow$" representing a *classical* implication. It is here that the approach of ICL distinguishes itself. The formula $\neg\neg P \supset P$ remains unprovable in ICL because $\supset$ represents *intuitionistic* implication. $\neg\neg P \Rightarrow P$ is provable, but what of its computational content? The formula $A \supset (B \vee \bot)$ is logically equivalent in ICL to $\neg A \vee B$, and thus can be considered equivalent to classical implication. But that, of course, would be a very fortuitous interpretation of classical implication. With the definition of $A \Rightarrow B$ as $\neg A \vee B$, $\neg\neg P \Rightarrow P$ becomes $\neg\neg\neg P \vee P$, and we get a different, albeit similar proof term: $\omega^\ell d.\lambda x.x(\lambda y.[d]y)$. The computational content of this term is different from that of $\mathcal{C}$. With $A \Rightarrow B$ defined as $\neg(A \wedge \neg B)$, which is suggested by some double-negation translations of classical logic, the proof of $\neg\neg P \Rightarrow P$ is a pure λ-term, which does not enable structural reduction. Moreover, since $\neg\neg A \equiv (A \vee \bot)$, if we had chosen to define classical implication as $A \supset \neg\neg B$, then the proof of $\neg\neg P \Rightarrow P$ becomes merely $\lambda x.x$, losing virtually all computational content.

In short, the computational interpretation of classical logic is dependent on how we *choose* to interpret classical implication, which does not have the naturally computational properties of its intuitionistic counterpart.

Confluence

To prove confluence, we follow the commonly used Tait-Martin Löf strategy of first defining a parallel, reflexive reduction relation $\stackrel{\|}{\longrightarrow}$ as follows:

- $s \stackrel{\|}{\longrightarrow} s$.

  The following rules all have the assumption $\mathbf{s} \stackrel{\|}{\longrightarrow} \mathbf{s}'$:

- $\lambda x.s \stackrel{\|}{\longrightarrow} \lambda x.s'$; $\quad \gamma d.s \stackrel{\|}{\longrightarrow} \gamma d.s'$.

- $\omega^\ell d.s \stackrel{\|}{\longrightarrow} \omega^\ell d.s'$; $\quad \omega^r d.s \stackrel{\|}{\longrightarrow} \omega^r d.s'$.

- $[d]s \stackrel{\|}{\longrightarrow} [d]s'$; $\quad \gamma a.s \stackrel{\|}{\longrightarrow} s'$, where a is not free in s.

- $\gamma a \gamma b.s \stackrel{\|}{\longrightarrow} \gamma a.s'[a/b]$; $\quad [a]\gamma b.s \stackrel{\|}{\longrightarrow} [a]s'[a/b]$.

- $(s,t)_\ell \overset{\|}{\longrightarrow} s'$; $(t,s)_r \overset{\|}{\longrightarrow} s'$; $abort\ s \overset{\|}{\longrightarrow} abort\ s'$.
- $(\gamma a.s)_\ell \overset{\|}{\longrightarrow} \gamma a.s'_\ell\{[a]w_\ell/[a]w\}$; $(\gamma a.s)_r \overset{\|}{\longrightarrow} \gamma a.s'_r\{[a]w_r/[a]w\}$.
- $(\gamma a.(\gamma b.s)_\ell) \overset{\|}{\longrightarrow} \gamma a.s'[a/b]_\ell\{[a]w_\ell/[a]w\}$; $(\gamma a.(\gamma b.s)_r) \overset{\|}{\longrightarrow} \gamma a.s'[a/b]_r\{[a]w_r/[a]w\}$.

  The following rules have the assumptions $\mathbf{s} \overset{\|}{\longrightarrow} \mathbf{s'}$ and $\mathbf{t} \overset{\|}{\longrightarrow} \mathbf{t'}$:
- $(s\ t) \overset{\|}{\longrightarrow} (s'\ t')$; $(s,t) \overset{\|}{\longrightarrow} (s',t')$; $(abort\ s)\ t \overset{\|}{\longrightarrow} (abort\ s')$
- $(\lambda x.s)\ t \overset{\|}{\longrightarrow} s'[t'/x]$; $(\gamma d.s)\ t \overset{\|}{\longrightarrow} \gamma d.(s'\{[d]w\ t'/[d]w\}\ t')$
- $\gamma a.((\gamma b.s)\ t) \overset{\|}{\longrightarrow} \gamma a.(s'[a/b]\{[a]w\ t'/[a]w\}\ t')$
- $(\omega^\ell d.s)\ t \overset{\|}{\longrightarrow} \omega^\ell d.s'\{[d]w\ t'/[d]w\}$; $(\omega^r d.s)\ t \overset{\|}{\longrightarrow} \omega^r d.s'\{[d]w\ t'/[d]w\}$.

  The following rules have the assumptions $\mathbf{s} \overset{\|}{\longrightarrow} \mathbf{s'}$, $\mathbf{t} \overset{\|}{\longrightarrow} \mathbf{t'}$ and $\mathbf{u} \overset{\|}{\longrightarrow} \mathbf{u'}$:
- $(s,t)\ \omega^\ell d.u \overset{\|}{\longrightarrow} \gamma d.(s'\ u'\{[d]t'\ w/[d]w\})$; $(s,t)\ \omega^r d.u \overset{\|}{\longrightarrow} \gamma d.(t'\ u'\{[d]s'\ w/[d]w\})$
- $(s,t)\ \gamma d.u \overset{\|}{\longrightarrow} \gamma d.((s',t')\ u'\{[d](s',t')w/[d]w\})$
- $\gamma a.((s,t)\ \gamma b.u) \overset{\|}{\longrightarrow} \gamma a.(s',t')u'[a/b]\{[a](s',t')w/[a]w\}$, where a is not free in (s,t).

The following properties of $\overset{\|}{\longrightarrow}$ are provable by induction.

1. If $s \overset{\|}{\longrightarrow} s'$ and x is free in s' then x is free in s.
2. If $s \overset{\|}{\longrightarrow} s'$ and $t \overset{\|}{\longrightarrow} t'$ then $s[t/x] \overset{\|}{\longrightarrow} s'[t'/x]$.
3. If $s \overset{\|}{\longrightarrow} s'$ and $t \overset{\|}{\longrightarrow} t'$ then $s\{[d]t/[d]w\} \overset{\|}{\longrightarrow} s'\{[d]t'/[d]w\}$

Proposition 11 *If $s \overset{\|}{\longrightarrow} s_1$ and $s \overset{\|}{\longrightarrow} s_2$, then for some term t, $s_1 \overset{\|}{\longrightarrow} t$ and $s_2 \overset{\|}{\longrightarrow} t$.*

To prove the above "diamond property" for $\overset{\|}{\longrightarrow}$, we argue by (simultaneous) induction on the structure of terms. The base case of variables x, a reduces only to themselves. Most of the cases follow directly from the inductive hypotheses. Some slightly non-trivial cases arise from "critical pairs," where more than one (non-reflexive) rule is applicable to a term. Many of these cases involve the renaming rules and terms of the form $\gamma d.s$. These cases follow from the property concerning free variables noted above, and from the renaming of bound-variables, which is always considered implicit.

A minor difficulty arises in the case of the γ-renaming rule, which eliminates consecutive contractions. For example, such a case occurs when either the $\mu\gamma$ rule or the γ-renaming rule can be applied. That is, $(\gamma a\gamma b.s)\ t \overset{\|}{\longrightarrow} (\gamma a.s'[a/b])\ t'$ by γ-renaming, and $(\gamma a\gamma b.s)\ t \overset{\|}{\longrightarrow} \gamma a.((\gamma b.s')\{[a]w\ t'/[a]w\}\ t')$ by $\mu\gamma$-reduction. But then

$$(\gamma a.s'[a/b])\ t' \overset{\|}{\longrightarrow} \gamma a.(s'[a/b]\{[a]w\ t'/[a]w\}\ t'), \quad \text{and}$$

$$\gamma a.((\gamma b.s')\{[a]w\ t'/[a]w\}\ t') \overset{\|}{\longrightarrow} \gamma a\gamma b.(s'\{[a]w\ t'/[a]w\}\{[b]r\ t'/[b]r\}\ t') \overset{\|}{\longrightarrow} \gamma a.(s'[a/b]\{[a]w\ t'/[a]w\}\ t').$$

This is a reduction of more than one step. To enforce that the diamond property holds for $\overset{\|}{\longrightarrow}$, we have therefore included the rule

$$\gamma a.((\gamma b.s)\ t) \overset{\|}{\longrightarrow} \gamma a.(s'[a/b]\{[a]w\ t'/[a]w\}\ t')$$

Other cases are similar. The confluence of $\longrightarrow$ follows from Proposition 11.

The Permutation of Contractions (Somewhat tentative)

The $\omega\mu$ reduction rules were specifically used to provide the equivalent of μ-reduction in $\lambda\mu$ calculus. The logical inference, from $(A \supset B) \vee C$ and A to $B \vee C$, is clearly sound, and the reduction rule correspond to the cut-elimination procedure. However, it is also possible to use a $\vee E$ rule for this deduction, which will lead to a different kind of term rewriting rule. To reconcile $\omega\mu$ reduction with ω reduction, we need to first examine the relationship between the Con rule and the $\vee$-introduction rules.

We have considered Con as a primitive rule, and included it in our cut-elimination procedure, to show that an alternative interpretation of disjunctions is possible. The injective abstractions ω^ℓ and ω^r can be replaced by the usual injection operations, labeled s^ℓ and s^r, and the $\vee I$ rules will revert to their usual additive forms:

$$\frac{s : \Gamma \vdash A_1;\ [\Delta]}{s^\ell : \Gamma \vdash A_1 \vee A_2;\ [\Delta]}\ \vee I_1^a \qquad \frac{s : \Gamma \vdash A_2;\ [\Delta]}{s^r : \Gamma \vdash A_1 \vee A_2;\ [\Delta]}\ \vee I_2^a$$

The proof term for the excluded middle $\neg A \vee A$ would then be $\gamma d.(\lambda x.[d]x^r)^\ell$.

We prefer the more computationally meaningful interpretation of disjunctions using ω^ℓ and ω^r. However, having both an explicit contraction rule and a non-additive $\vee$ will surely generate alternative proofs. We can generalize the relationship between Con and the $\vee I$ rules by considering the following equivalence between proofs:

$$\cfrac{\cfrac{t : \Gamma \vdash B;\ [C^a, C \vee B^d, \Delta]}{\omega^r a.t : \Gamma \vdash C \vee B;\ [C \vee B^d, \Delta]}\ \vee I_2}{\gamma d.\omega^r a.t : \Gamma \vdash C \vee B;\ [\Delta]}\ Con \qquad \equiv \qquad \cfrac{\cfrac{t' : \Gamma \vdash B;\ [B^d, C^a, \Delta]}{\gamma d.t' : \Gamma \vdash B;\ [C^a, \Delta]}\ Con}{\omega^r a.\gamma d.t' : \Gamma \vdash C \vee B;\ [\Delta]}\ \vee I_2$$

Now the proof term t could have subproofs of the following forms, all marked by Esc rules:

$$\frac{u : \Gamma_1 \vdash C;\ [C^a, C \vee B^d, \Delta_1]}{[a]u : \Gamma_1 \vdash \bot;\ [C^a, C \vee B^d, \Delta_1]} \qquad \cfrac{\cfrac{v : \Gamma_2 \vdash C;\ [B^e, C^a, \Delta_2]}{\omega^\ell e.v : \Gamma_2 \vdash C \vee B;\ [C^a, \Delta_2]}\ \vee I_1}{[d]\omega^\ell e.v : \Gamma_2 \vdash \bot;\ [C^a, C \vee B^d, \Delta_2]} \qquad \cfrac{\cfrac{w : \Gamma_3 \vdash B;\ [C^a, \Delta_3]}{\omega^r a.w : \Gamma_3 \vdash C \vee B;\ [C^a, \Delta_3]}\ \vee I_2}{[d]\omega^r a.w : \Gamma_3 \vdash \bot;\ [C^a, C \vee B^d, \Delta_3]}$$

These are replaced in t', respectively, by the following subproofs:

$$\frac{u' : \Gamma_1 \vdash C;\ [C^a, B^d, \Delta_1]}{[a]u' : \Gamma_1 \vdash \bot;\ [B^d, C^a, \Delta_1]} \qquad \frac{v[d/e] : \Gamma_2 \vdash C;\ [B^d, C^a, \Delta_2]}{[a]v[d/e] : \Gamma_2 \vdash \bot;\ [B^d, C^a, \Delta_2]} \qquad \frac{w : \Gamma_3 \vdash B;\ [C^a, \Delta_3]}{[d]w : \Gamma_3 \vdash \bot;\ [B^d, C^a, \Delta_3]}$$

where u' relates to u by applying the transformation recursively. A similar equivalence can be shown if $\vee I_1$ was used instead of $\vee I_2$. The transformed proof using t' pushes γ inward pass ω-binders.

Note that if in the transformed t' no subproof of the form $[d]w$ exists, then the swapped γ-binder becomes vacuous and can be eliminated. In light of this fact, consider the following reduction:

$$\cfrac{s : \Gamma \vdash (A \supset B) \vee C;\ [\Delta] \qquad \cfrac{\cfrac{y : (A \supset B)^y, \vdash A \supset B; [\,] \quad t : (A \supset B)^y, \Gamma \vdash A;\ [\Delta]}{(y\ t) : (A \supset B)^y, \Gamma \vdash B;\ [\Delta]}\ \supset E}{\omega^\ell e.(y\ t) : (A \supset B)^y, \Gamma \vdash B \vee C;\ [\Delta]}\ \vee I_1 \qquad \lambda x.\omega^r a.x : C^x, \Gamma \vdash B \vee C;\ [\Delta]}{(\lambda y.\omega^\ell e.(y\ t), \lambda x.\omega^r a.x)\ s : \Gamma \vdash B \vee C;\ [\Delta]}\ \vee E$$

Assume that $s = \omega^r d.u$. By the ω-reduction rule as defined, this proof term reduces to

$$\gamma d.\omega^r a.u\{[d]\omega^\ell e.(r\ t)/[d]r\}$$

Both ω-binders in this term are vacuous. Thus by the transformation of t to t' described above (specifically for the subproof $[d]\omega^\ell e.v$), this term can be replaced with:

$$\omega^r a.\gamma d.u\{[a](r\ t)/[d]r\} \ \equiv\ \omega^r a.u\{[a](r\ t)/[d]r\}$$

In this sense the $\omega\mu$ reductions are consistent with $\vee E$ and ω reduction. In particular, in regard to the $\mathcal{C}$ operator, we have the following equivalence:

$$\mathcal{C}Mt \ \longrightarrow\ \omega^r a.M(\lambda y.[a]y\ t) \ \equiv\ \gamma d.\omega^r a.M(\lambda z.[d]\omega^\ell e.(z\ t[z/y])) \ \equiv\ (\lambda y.\omega^\ell e.(y\ t), \lambda x.\omega^r a.x)\ (\mathcal{C}M)$$

In similar manner, it is also possible to push γ into pairs $(A \wedge B)$, and even into λ-abstractions: that is, permute a *Con* rule on $A \supset B$ into a *Con* rule on B. It is not possible to push a contraction into an application $(a\ b)$ as that would undo cut-elimination.

The permutation of contractions over other introduction rules is related to the phenomenon of *focusing.* In a focused proof system, introduction rules are applied in uninterrupted phases, with structural rules only applicable in between these phases. In the context of the $\lambda\gamma$-term calculus, focusing could provide a valuable tool for program transformation. However, we shall leave that problem to another occasion.

7 A Topological Semantics

Here we define an alternative, topological space semantics for propositional ICL. This semantics has the advantage of simplicity. Most of the non-intuitionistic axioms of ICL can be verified in this semantics using a method close to the construction of truth tables. Only a very minimal background in topology is required to understand this semantics.

To those familiar with the translation of Kripke models into Heyting algebras by forming a lattice of upwardly closed subsets of the Kripke frame, 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 $\perp$. However, 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 topology of a *dense-in-itself* metric space, such as that of the real line $\mathbb{R}$. This form of semantics for non-classical logics is due to Tarski and McKinsey [Tar38, MT46].

Our topological space is a structure $(\mathbb{R}, \mathcal{O})$, where $\mathcal{O}$ is a set of open sets, which consists of unions over countable collections of disjoint, open intervals of $\mathbb{R}$ (countable by the density of the rationals in $\mathbb{R}$). $\mathcal{O}$ contains $\mathbb{R}$ and $\mathbf{0}$, which is the empty set, and is closed under arbitrary unions and pairwise intersections. We define a Heyting algebra $\mathcal{HR}$ from the topology as $\langle \mathcal{O}, \sqsubseteq, \sqcup, \sqcap, \rightarrow, \mathbf{0} \rangle$, where $\sqsubseteq$, $\sqcup$ (join) and $\sqcap$ (meet) are the usual set-theoretic relations and the relative pseudo-complement $a \rightarrow b$ is the *interior* of $(\mathbb{R} - a) \cup b$, written $\mathcal{I}((\mathbb{R} - a) \cup b)$. We use a different set of symbols for the set-theoretic operators so as to distinguish formulas of the algebra from general statements concerning sets. Here, "$-$" is set subtraction.

In this universal topology for ICL, the denotation of $\perp$, written $\perp\!\!\!\perp$, 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. The results here will obviously generalize to any other choice. Then $\perp\!\!\!\perp$ is defined as

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

This is clearly an open set. Furthermore, the only other open set containing $\perp\!\!\!\perp$ is $\mathbb{R}$ itself. The pair $\perp\!\!\!\perp$ and $\mathbb{R}$ effectively forms an embedded, two-element boolean algebra, which is all that is needed to interpret classical propositional logic. Since it holds that $\mathbb{R} \rightarrow \perp\!\!\!\perp = \perp\!\!\!\perp$ and $\perp\!\!\!\perp \rightarrow \perp\!\!\!\perp = \mathbb{R}$, the boolean complement operation in this algebra is defined by $x \rightarrow \perp\!\!\!\perp$. The top element of this algebra coincides with the top element of the host Heyting algebra (there are two *false* but only one *true*).

A *valuation* (i.e., model) is represented by a mapping h from atomic formulas into $\mathcal{HR}$ that is extended to all formulas as follows:

1. $h(\top) = \mathbb{R}$; $h(0) = \mathbf{0}$; $h(\perp) = \perp\!\!\!\perp$
2. $h(A \wedge B) = h(A) \sqcap h(B)$
3. $h(A \vee B) = h(A) \sqcup h(B)$
4. $h(A \supset B) = h(A) \rightarrow h(B)$

In particular $h(\neg A) = h(A) \rightarrow \perp\!\!\!\perp$ and $h({\sim} A) = h(A) \rightarrow \mathbf{0}$.

There is, however, a degenerate case, which corresponds to r-models with frames consisting of only one element: the root $\mathbf{r}$ itself. Thus h is given an secondary extension to h', which is defined as follows:

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. $h'(A \vee B) = h'(A) \sqcup h'(B)$; $h'(A \wedge B) = h'(A) \sqcap h'(B)$
4. $h'(A \supset B) = h'(A) \to h'(B)$

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 $h'(A) = \mathbb{R}$. In particular, $A \supset B$ is valid under h if $h(A) \subseteq h(B)$ and $h'(A) \subseteq h'(B)$.

As an example of the need for h', it holds that $\perp\!\!\!\perp \to \mathbf{0} = \mathbf{0}$ (because $\perp\!\!\!\perp$ is *dense*), which means that $(\bot \supset 0) \supset 0$ would be considered valid if only interpreted under h. But $h'(\bot \supset 0) = \perp\!\!\!\perp \to \perp\!\!\!\perp = \mathbb{R}$, so $h'((\bot \supset 0) \supset 0) = \mathbb{R} \to \mathbf{0} \neq \mathbb{R}$. h' represents the special case when $\bot \equiv 0$.

The following illuminating properties hold for h and h':

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

One might suspect that the mapping $h'(A)$ is equivalent to $h(\neg\neg A)$. It is important that they are not equivalent. For sake of argument, let us define $h^2(A) = h(\neg\neg A) = (h(A) \to \perp\!\!\!\perp) \to \perp\!\!\!\perp$. h^2 also maps all formulas to either $\perp\!\!\!\perp$ or $\mathbb{R}$. The following properties can be established for h^2:

- $h^2(\top) = \mathbb{R}$; $h^2(\bot) = h^2(0) = \perp\!\!\!\perp$
- $h^2(A \wedge B) = h^2(A) \sqcap h^2(B)$
- $h^2(A \vee B) = h^2(A) \sqcup h^2(B)$

However, the homomorphic properties of h^2 do *not* extend to $\to$. A perfect homomorphism would indicate that intuitionistic implication collapses into a classical one when interpreted under the double-negation $\neg\neg$. This would compromise our main goal of finding a meaningful combination of intuitionistic and classical logics in a way that does not destroy either. In contrast, h' is *defined* to be a homomorphism on the two-element boolean algebra. As it corresponds to Kripke models with a single element, it is supposed to represent those situations where intuitionistic implication does collapse into a classical one.

Compare $h^2(A \supset B)$ (equivalently $(h(A \supset B) \to \perp\!\!\!\perp) \to \perp\!\!\!\perp$) and $h(A \Rightarrow B) = h(\neg A \vee B)$ in a situation where $1 \notin h(A)$ and $1 \notin h(B)$. Let $h(B)$ be the open interval $\{x : 2 < x \text{ and } x < 3\}$. and $h(A) = \{x : 0 < x \text{ and } x < 1\}$. Then $1 \in \mathbb{R} - h(A)$ but $1 \notin \mathcal{I}((\mathbb{R} - h(A)) \cup h(B)) = h(A) \to h(B)$. This scenario is possible when 1 is a *limit point* (accumulation point) of $h(A)$ without being inside $h(A)$, and thus may be excluded by the interior operation $\mathcal{I}$. Under this valuation, $h^2(A \supset B) = \perp\!\!\!\perp$ but $h(\neg A \vee B) = \mathbb{R}$. The same countermodel shows that $h^2(A \supset B) \neq h^2(A) \to h^2(B)$.

Correctness of the Topological Semantics

The soundness of the proof systems LJC/NJC with respect to this semantics can be shown by induction on the structure of proofs, much as in the soundness proof for the Kripke semantics. We concentrate on completeness, which relies on known results for intuitionistic logic.

Given an unprovable formula A^0, by Kripke completeness there is a r-model in which $\mathbf{r} \not\models A^0$. Translate the r-model into a finite Heyting algebra H using the well known method of forming a lattice of upwardly closed sets, with the valuation $v(A) = \{u \in \mathcal{W} : u \models A\}$. Then $v(\perp)$ is the second-largest element of the Heyting lattice. A result of Tarski [Tar38] implies that, *given H and a dense-in-itself metric space, in this case $\mathcal{HR}$, the topology of $\mathbb{R}$, there is a dense, open subspace G of $\mathcal{HR}$ and an isomorphism f from H into a (Heyting) subalgebra HG of G.* The top element of HG is G and the mapping $s(U) = G \cap U$ is a homomorphism from $\mathcal{HR}$ onto G.

Let $\perp^f = f(v(\perp))$, which is therefore the second-largest element in HG. If $\perp^f = \mathbf{0}$, then HG is a two-element (boolean) algebra and the valuation v is then easily shown to be isomorphic to a valuation h' on $\mathcal{HR}$ as defined earlier (which mapped all formulas into the two-element algebra consisting of $\perp\!\!\!\perp$ and $\mathbb{R}$).

The main case is when $\perp^f \neq \mathbf{0}$. Because $\perp^f$ is second-largest in HG, it holds that $\perp^f$ is dense in that $\mathcal{I}(G - \perp^f) = \mathbf{0}$ (i.e., $\perp^f \to \mathbf{0} = \mathbf{0}$ in HG). Since $\perp^f$ is open, it consists of a collection of open intervals (a, b) (sets $\{x : a < x < b\}$). Then $G - \perp^f$, the complement of $\perp^f$ in G, must be *closed,* which means that it consists of a collection of closed intervals $[c, d]$ (sets $\{x : c \leq x \leq d\}$). But if any such closed interval has a non-empty interior ($a \neq b$), that would contradict the fact that $G - \perp^f$ has an empty interior. Thus $G - \perp^f$ consists of a collection of isolated points[2]. Let $\tau \in G - \perp^f$ be one of these points (recall that our choice of $\tau = 1$ was entirely arbitrary; we can map each number x to $x - \tau + 1$, which is clearly an isomorphism). Then for $D = (G - \perp^f) - \{\tau\}$, D is also a collection of isolated points, and thus closed. *This means that $G - D = \perp^f \cup \{\tau\}$ must be open.* Let $G' = \perp^f \cup \{\tau\}$: this is also a subspace of $\mathcal{HR}$ and $s'(V) = V \cap G'$ is a homomorphism from $\mathcal{HG}$ onto the topology of G'. We define an isomorphism g from HG onto a subalgebra HG' of G': let $g(G) = G'$ and let $g(V) = V$ if $V \neq G$. It is easy to show that g is indeed an isomorphism given the following observation: since $\perp^f$ is larger than all elements except G in HG and G' in HG', it holds that if $A \cup B = G$ then either $A = G$ or $B = G$, and likewise in G'.

The rest of the proof proceeds as in the completeness proof for intuitionistic logic, a thorough exposition of which can be found in Rasiowa and Sikorski [RS63]. We give the main arguments below.

Let $g' = g \circ f \circ v$. Then g' is a valuation on HG' isomorphic to the valuation v on the finite Heyting algebra H. Define the valuation h on $\mathcal{HR}$ by: $h(a) = S$ for some open subset S of $\mathbb{R}$ such that $S \cap G' = g'(a)$ for atomic formulas a. Let $h(\top) = \mathbb{R}$, $h(0) = \mathbf{0}$, and $h(\perp) = \mathbb{R} - \{\tau\}$. Note that $h(\perp) \cap G' = \perp^f$. In fact, the property $h(F) \cap G' = g'(F)$ holds for all formulas F because $s'(V) = G' \cap V$ is a homomorphism. Given an unprovable formula A^0, it holds that $v(A^0) \neq v(\top)$, so $g'(A^0) \neq G'$. Thus $h(A^0) \cap G' \neq G'$, which means that $h(A^0) \neq \mathbb{R}$. Therefore a countermodel as a valuation on the finite Heyting algebra H can be mapped to a countermodel in the topology of $\mathbb{R}$. $\square$

The fact that $G - \perp^f$ must consist of a set of isolated points also hints at how this semantics can be extended to include first-order quantifiers (at least over a domain as large as $\mathbb{R}$). The Kripke-style semantics will lose its elegance since the *domain* of possible worlds cannot stay constant for intuitionistic logic, and hence we can no longer isolate classical reasoning to an single *root.* In the topological setting, however, we can take as the denotation of $\perp$ the set $\mathbb{R} - \mathbb{I}$: the reals minus the integers. That is, $\perp\!\!\!\perp$ will consist of an enumerable set of open intervals $(0, 1)$, $(1, 2)$, etc. The open sets above $\perp\!\!\!\perp$ correspond one-to-one with subsets of $\mathbb{I}$, and hence form an infinite boolean algebra. This boolean algebra is also closed under the same $\sqcup$, $\sqcap$ and $\to$ operations as in the host Heyting algebra. The quantifiers $\exists$ and $\forall$ can then be defined by infinite joins and meets respectively. Classical versions of these quantifiers can be defined from formulas such as $\exists x.(A \vee \perp)$. Axioms such as $A \vee \neg A$ will still hold under this version of $\perp\!\!\!\perp$: in this case, for every integer $x \notin h(A)$, it holds that $x \in \mathcal{I}((\mathbb{R} - h(A)) \cup \perp\!\!\!\perp)$. The proof-theoretic extension to first-order quantifiers is trivial and does not affect cut-elimination in any significant way. In future work, we will likely bypass the first-order quantifiers and consider directly a *second order propositional ICL.*

8 Related Work and Conclusion

$\lambda\mu$-Calculus and Variants

[2] In the sense that each $x \in G - \perp^f$ is a *boundary* point of $G - \perp^f$.

We believe that ICL offers a better explanation of the extension of lambda calculus to control operators compared to the collapse of all of intuitionistic logic into classical logic. We have certainly benefited from the study of several variants of $\lambda\mu$-calculus. However, in addition to preserving intuitionistic implication, we believe that the proof theory of ICL and the $\lambda\gamma$-calculus derived from it offer a clearer formulation of control operators and other programming language constructs. The correspondance between proofs and types is more natural and straightforward. In particular, although contraction is present (or admissible) in other systems, they do not give a direct, computational interpretation of an explicit contraction rule (other than as the renaming of index variables).

The naming rules of the original $\lambda\mu$-calculus can leave free variables in proofs of axioms, such as for $\neg\neg A \Rightarrow A$, a fact that has been widely criticized. The version of $\lambda\mu$ of Ong and Stewart [OS97] does not have this problem. In fact, there are clearly similarities between this version and ours, especially in light of our interpretation of sequents in Section 5. Their "$\perp$-intro" rule is in fact very close to our *Esc* rule (but of course their "$\perp$" is the regular *false* of classical logic). However, their rule for "$\perp$-elim" has no counterpart in our system:

$$\frac{\Gamma; \Delta, B^{\beta} \vdash s : \perp}{\Gamma; \Delta \vdash \mu\beta^{B}.s : B} \ \perp\!-elim$$

It appears that "$\perp$" is playing *two different roles* in this system. Because B^{β} in the premise of $\perp$-elim can be introduced by weakening, this rule implements *ex falso quodlibet* ($0E$ in our system). However, it is also only by using this rule that a μ binder can be introduced into terms. This means that their proof of Peirce's formula, for example, must use $\perp$-elim, a fact that is not consistent with the subformula property. Their proof of Peirce's formula somewhat resembles a NK proof with the "*RAA*" rule, which is precisely the flaw of NK. A better explanation for the provability of this formula, one that preserves the subformula property, is that it requires *contraction,* not *ex falso quodlibet.* Yet, the computational content of Peirce's formula is not attributed to contraction but to $\perp$-elim in this system. With $\lambda\gamma$, it is clear that the central reason for the provability of our version of Peirce's formula is contraction. The γ binder is introduced precisely by contraction on the current formula. Our *Con* rule has a form opposite to that of the *passivate* rule of $\lambda\mu$ (also called the *decide* rule by others): in the *passivate/decide* rule, the contracted formula does not become the current formula.

Important similarities are also found between ICL and the *Minimal Classical Logic* (MC) of Ariola and Herbelin [AH03]. By removing *ex falso quodlibet* from full classical logic, they show that Peirce's formula can still be proved, and that MC axioms are all proved by closed $\lambda\mu$-terms. There is a property for MC that parallels our Proposition 1: *if $\perp$ does not occur in A then A is provable in MC iff A is provable in classical logic.* The same property holds for ICL's version of $\perp$, but with respect to *intuitionistic logic.* The proof theory of MC also shares similarities with NJC. Their version of $\lambda\mu$ is one of the few that does not ignore *abort*. However, there is still only one *false* in MC, written using the symbol $\perp$. MC does not prove $\neg\neg A \Rightarrow A$. Consequently, they cannot formulate the $\mathcal{C}$ operator within MC. Their solution is to adopt a *top-level continuation constant* called *top*, which then recovers full classical logic. By using *two* forms of *false*, we are able to consider the formula $\sim\neg A \supset A$, which is valid in ICL. Our solution is therefore rather straightforward, and without a collapse into classical logic.

Disjunction is ignored by most of the literature on λ-calculus. Unfortunately, it is also often ignored by those on $\lambda\mu$-calculus despite the fact that a multiplicative disjunction carries a different kind of computational content compared to an additive one. When disjunctions are delt with, they often only appear in the additive form, with the following exception:

Proof terms for multiplicative, classical disjunction also appear as abstractions in [RPW00]. They used ν to represent the abstraction, but without distinguishing between ν^{ℓ} and ν^{r}. While their $\vee$-introduction rule is similar to ours, their treatment of $\vee$-elimination is entirely different:

$$\frac{\Gamma \vdash t : A, B^{\beta}, \Delta}{\Gamma \vdash \nu\beta.t : A \vee B, \Delta} \ \vee I \qquad \frac{\Gamma \vdash t : A \vee B, \Delta}{\Gamma \vdash \langle\beta\rangle t : A, B^{\beta}, \Delta} \ \vee E$$

A new kind of named term, $\langle\beta\rangle t$ (as opposed to $[\beta]t$), marks a $\vee$-elimination. Evidently, $\langle\beta\rangle\nu\beta.t$ should reduce to t during normalization, and is thus a renaming step. The computational content of their disjunction is thus entirely different from ours. It is not clear what kind of programming language constructs can be represented by their rules for $\vee$, except in terms of *proof search* in the logic programming sense.

The proof theory of ICL allows disjunction to be treated either additively or multiplicatively depending on whether $\perp$ is a subformula *"in the right place."* The $\vee$ in Girard's LC system [Gir91] can also behave additively or multiplicatively. However, unlike LC, we do not need to examine subformulas recursively in order to decide which inference rule to use. Our ω^ℓ and ω^r binders naturally become *injection* operators when vacuous, *i.e.,* when the disjunction is in fact intuitionistic. Since our $\vee$ can be additive and thus have *non-invertible* introduction rules, we obviously cannot use the $\vee$-elimination rule of [RPW00]. Our $\vee$-elimination rule is the same as in standard natural deduction when the $\omega^{\ell/r}$ binder is vacuous; when it is not vacuous, both β-reduction and structural reduction must take place, leading to a more computationally meaningful interpretation of $\vee$-elimination.

As we have observed in Section 5, the semantic interpretation of a sequent such as $\vdash A; [B]$ is the formula $\neg B \supset A$, which is similar to a double-negation. One might therefore dismiss our treatment of $A \vee B$ as nothing other than $\neg B \supset A$. This is not correct: the equivalence does not hold in general. For example, $\neg A \vee A$ under this naive translation is $\neg A \supset \neg A$, which is surely provable. But $A \vee \neg A$ is translated as $\neg\neg A \supset A$, which is not provable. We cannot translate a disjunction unless we know if it's provable, as well as how it is to be proved.

The Unified Logic LU

In terms of Girard's LC, $\perp$ is "negative." By associating "polarities" with formulas, one can identify the portions of a classical proof that are in fact intuitionistic, thus revealing the constructive content of classical proofs. However, LC does not contain intuitionistic implication (and neither does its generalization *polarized linear logic*). In a system called LU, Girard attempted to extend his polarized framework to include classical, intuitionistic and linear logics. However, this system was only partially successful. In the denoational semantics of LU (based on the coherent space semantics of linear logic), the distributivity of $\wedge$ over $\vee$ fails to hold for all possible polarity combinations. This failure is reflected in the proof theory in that admissible cuts require restrictions: in particular, when intuitionistic implication is agreesively mixed with formulas of other polarities, cuts cannot always be permuted above contractions.

We know of two approaches to improving LU. The first approach, which formed the basis of most of our earlier attempts to unify logics, was to add even more polarity information to connectives and formulas, enough so that cuts can always be permuted above contractions. In an earlier effort, we in fact proposed a system containing six polarities (compared to three in LU) and eighteen connectives.

The other way of avoiding the problems of LU is to adopt a vastly simpler system. This is the approach of ICL. The semantic structure of ICL is identical to that of intuitionistic logic: it is a distributive lattice. Cut-elimination is virtually guaranteed for any sound and complete proof system[3].

Linear Logic

In terms of linear logic, while the provability of ICL formulas might be easily captured by a translation, their *proofs* could not. Formulas inside the [ ] context are subject to structural rules but not to introduction rules until the control formula is $\perp$. One might experiment with linear forms such as $?!A$, but such experiments are not likely to succeed since "?!" cannot be part of a *synthetic connective* (it destroys focusing). Linear logic extended with *subexponentials,* i.e., extra pairs of exponentials that might preserve focusing in some circumstances, may be able to capture the proof theory of ICL in a fully adequate way.

PIL

[3]This is not a "deep" observation. A relative pseudo-complement $a \rightarrow b$ evaluates to true if $a \leq b$ in the lattice. For a proof system with possibly multiple conclusions, the admissibility of cut says that if $a \leq (b \sqcup c)$ and $(b \sqcap d) \leq e$, then $(a \sqcap d) \leq (c \sqcup e)$ (here, a and d represent finite meets while c and e represent finite joins). This follows easily from distributivity and associativity, which hold in all pseudo-complemented lattices.

Despite some essential similarities, ICL is *not* a fragment of our slightly earlier work, PIL [LM11]. The semantics of the connectives, and of $\perp$ are different. In terms of the "Kripke hybrid" models of PIL, the root world of a r-models is the only *classical* world, while all worlds above the root are *imaginary,* in that they force $\perp$. The study of the semantics of PIL lead to the development of ICL as a separate logic.

We have devoted a large part of this paper to a traditional semantic interpretation of ICL. An important value of *semantics* is that they give us the ability to reason about a logic in an entirely different way from the syntactic proof theory (if it parallels the proof theory, then clearly this is less true). For example, we are able to semantically derive not only formulas that are valid but also observe ones that are not. The value of the traditional Kripke semantics is clearly demonstrated in that we can easily observe a *general property of admissible rules* that is not obvious from the proof theory. The first insights into the possibility of this logic came from considering where to place Girard's $\perp$ in the metric space of real numbers, which Tarski showed to be capable of interpreting intuitionistic logic. We do however, intend to explore other forms of semantics including extensions of Kripke lambda models as well as categorical approaches. Since neither form of negation in ICL is "involutive," we have reason to believe that cartesian closed categories should not degenerate into boolean algebras as they do for classical logic.

References

[AH03] 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.

[dG94] 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.

[FFKD87] M. Felleisen, D. Friedman, E. Kohlbecker, and B. Duba. A syntactic theory of sequential control. *Theoretical Computer Science*, 52(3):205–237, 1987.

[Fit69] Melvin C. Fitting. *Intuitionistic Logic Model Theory and Forcing.* North-Holland, 1969.

[Gir91] Jean-Yves Girard. A new constructive logic: classical logic. *Math. Structures in Comp. Science*, 1:255–296, 1991.

[Gri90] Timothy Griffin. The formulae-as-types notion of control. In *17th Annual ACM Symp. on Principles of Programming Languages*, pages 47–57, 1990.

[GTL89] Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and Types.* Cambridge University Press, 1989.

[HS09] Hugo Herbelin and Alexis Saurin. $\lambda\mu$-calculus and $\Lambda\mu$-calculus: a capital difference. Manuscript, 2009.

[LM11] Chuck Liang and Dale Miller. Polarized intuitionistic logic. Submitted, 2011.

[MT46] J. McKinsey and A. Tarski. On closed elements in closure algebras. *Annals of Mathematics*, 47:122–162, 1946.

[OS97] 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.

[Par92] 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.

[RPW00] E. Ritter, D. Pym, and L. Wallen. Proof-terms for classical and intuitionistic resolution. *Journal of Logic and Computation*, 10(2):173–207, 2000.

[RS63] Helena Rasiowa and Roman Sikorski. *The Mathematics of Metamathematics.* Panstwowe Wydawnictwo Naukowe, Warsaw, 1963.

[Tar38] A. Tarski. Der aussagenkalkül und die topologie. *Fundamenta Mathematicae*, 31:103–134, 1938.

A Strong Normalization for the Implicational Fragment

In this temporary appendix, we prove strong normalization for the fragment of $\lambda\gamma$-calculus with only $\supset$ as connective. The other connectives can be addressed at a later point. The reduced set of rewrite rules are:

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. $abort^{A\supset B}(s)\ t \ \longrightarrow\ abort^{B}(s)$. (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)

This subset is still capable of representing the control operators $\mathcal{K}$ and $\mathcal{C}$, as we have shown. We have annotated the *abort* operator with a type to avoid confusion, although this modification is not critical.

The strong normalization proof follows the reducibility method of Tait, and specifically the proof in [GTL89] for the simply typed case, because we found it to be readily adaptable. The reducibility approach will also allow us to consider a *second-order propositional ICL* in the future. Much of the proof in [GTL89] carries over virtually without modification.

For additional clarity in the arguments, we formally define the structural substitution operation as follows:

1. $x\{[d](w\ t)/[d]w\} = x$ for λ-variable x.
2. $([e]s)\{[d](w\ t)/[d]w\} = [e]s\{[d](w\ t)/[d]w\}$ for $e \neq d$.
3. $([d]s)\{[d](w\ t)/[d]w\} = [d](s\{[d](w\ t)/[d]w\}\ t)$
4. $(s\ t)\{[d](w\ t)/[d]w\} = (s\{[d](w\ t)/[d]w\}\ t\{[d](w\ t)/[d]w\})$
5. $(\lambda x.s)\{[d](w\ t)/[d]w\} = \lambda x.s\{[d](w\ t)/[d]w\}$ for x not free in t
6. $(\gamma a.s)\{[d](w\ t)/[d]w\} = \gamma a.s\{[d](w\ t)/[d]w\}$ for $a \neq d$ and a not free in t
7. $abort^{A}(s)\{[d](w\ t)/[d]w\} = abort^{A}(s\{[d](w\ t)/[d]w\})$

Reducibility Sets

In this section we generalize the notion of the "type" of a proof term to include terms with free variables. A term t such that $t : \Gamma \vdash A; [\Delta]$ is provable is considered to have type A *relative to* Γ, Δ, although we will often drop the "relative" clause for convenience. In the following we use the syntax $t : A$ to mean *"a term t typable by A."* We also note that the Subject Reduction result in fact holds for this more general notion of typability. Structural substitution involving well-typed terms also preserves types, which easily follows from the fact that all terms $[d]w$ are of type $\bot$.

To each formula (type) F, define a set of "reducible" terms RED_F by induction on type F:

- For atomic F, as well as for the cases of $\bot$, 0 and $\top$, $s : F \in RED_F$ *iff s is strongly normalizing (SN).*
- for $A \supset B$, $s : A \supset B \in RED_{A\supset B}$ *iff for all $t \in RED_A$, $(s\ t) \in RED_B$.*

Often, if $t : A \in RED_A$ without ambiguity we just say that "t is reducible." This definition of reducibility is identical to the case of pure lambda terms.

Properties of Reducibility

A term is referred to as *neutral* if it is *not* of the form $\lambda x.s$, $abort^{A \supset B}(s)$, or $\gamma d.s$ of some $\supset$ type. We do not consider $abort^{A}(s)$ to be an application term and have used the special notation to distinguish it. All application terms $(s\ t)$ are neutral. We use the notation $A \longrightarrow_1 B$ to mean that B is obtained from A by exactly one reduction step.

The three reducibility properties are likewise identical to the intuitionistic/lambda calculus case:

CR 1 : if $s \in RED_A$ then s is strongly normalizing

CR 2 : if $s \in RED_A$ and $s \longrightarrow t$, then $t \in RED_A$

CR 3 : if s is neutral and if $s \longrightarrow_1 t$ implies $t \in RED_A$, then $s \in RED_A$.

A vacuous case of **CR 3** implies that if a neutral term is normal then it is reducible.

Lemma 12 *Every set RED_A satisfies the conditions* **CR 1***,* **CR 2** *and* **CR 3***.*

The proof of this lemma is in fact still identical to the intuitionistic case. It proceeds by induction on the type A. This may be surprising, so for example, for **CR 3** in the atomic case, if $s \longrightarrow_1 t$ and t is reducible, then by subject reduction t is also a term of atomic type, which by definition means that t is SN. But if s always reduces to an SN term, then s is also SN. This argument makes no reference to the form of the term: it includes, for example, the case when s of the form $[d]s'$ of type $\perp$. As another sample case, for **CR 2** in the $A = B \supset C$ case, if $s \in RED_{B \supset C}$ and $s \longrightarrow t$, then for arbitrary $u \in RED_B$, $(s\ u) \longrightarrow (t\ u)$. But $(s\ u)$ is reducible because s is reducible, so by inductive hypothesis, $(t\ u)$ is also reducible and therefore t is reducible. The additional cases required by the new terms only appear in the subject reduction proof.

All Well-Typed Terms are Reducible

The following series of lemmas are used to show that all terms are reducible, from which the main result follows.

Given a strongly normalizing (SN) term s, let $m(s)$ be the sum of the lengths of all terminating reduction paths of s. By König's Lemma, $m(s)$ is finite.

Lemma 13 *If for all $t \in RED_A$, $s[t/x] \in RED_B$, then $\lambda x.s \in RED_{A \supset B}$.*

The proof of this lemma is likewise still identical to intuitionistic/lambda calculus case. Since all reducible terms satisfy **CR 1**, the proof simply uses an induction on $m(s)$.

Now we finally come to results that are required because of the extra terms.

Lemma 14 *If s is reducible then $[d]s$ is reducible.*

Proof The term $[d]s$ can only be of type $\perp$. Since s is reducible, by **CR 1**, s is SN. $[d]s$ reduces to $[d]s'$ if s reduces to s'. The only other possible reduction is when $s = \gamma a.t$ and $[d]\gamma a.t$ reduces to $[d]t[d/a]$ by renaming. In both cases, since s is SN, $[d]s$ must also be SN, and is therefore reducible by definition of $RED_\perp$. $\Box$

A similar property can be proved for the *abort* operator.

Lemma 15 *If s is reducible then $abort^F(s)$ is reducible.*

Proof This lemma is proved first by induction on the type F, then by a secondary induction on the lengths of reductions. If F is atomic or constant, then by inductive hypothesis s is SN. Thus $abort^F(s)$ is SN and therefore reducible by definition.

If the term is $abort^{A \supset B}(s)$, we need to show that $abort^{A \supset B}(s)\ t$ is reducible for any $t \in RED_A$. By inductive hypothesis on s and by assumption on t, s and t are both SN. $abort^{A \supset B}(s)\ t$ reduces in one step to either $abort^B(s)$, $abort^{A\supset B}(s')\ t$, or $abort^{A\supset B}(s)\ t'$ where $s \longrightarrow_1 s'$ or $t \longrightarrow_1 t'$. In the first case the result follows from inductive hypothesis on type B. In the other cases, we argue by induction on the measure $m(s)+m(t)$, which bounds the sum of the lengths of reductions of s and t. We either have that $(abort^{A\supset B}(s)\ t) \longrightarrow_1 (abort^{A\supset B}(s')\ t)$ with $s \longrightarrow_1 s'$, or $(abort^{A\supset B}(s)\ t) \longrightarrow_1 (abort^{A\supset B}(s)\ t')$ with $t \longrightarrow_1 t'$. The inductive hypothesis on the smaller measure implies that these terms are reducible. Since $(abort^{A\supset B}(s)\ t)$ is neutral, by **CR 3** it is therefore reducible and thus $abort^{A\supset B}(s)$ is reducible. □

The corresponding property for γ-abstractions requires a more general form. We use the notation $\bar{t}$ to represent a vector $t_1 \dots t_m$ of *zero or more* reducible terms.

Lemma 16 *If $(s\{[d](w\ \bar{t})/[d]w\})\bar{t}$ is reducible, assuming it is typable, for any free γ-variables d in s and for any $\bar{t}$ with no free occurrences of d, then $\gamma d.(s\{[d](w\ \bar{t})/[d]w\}\ \bar{t})$ is reducible if it is typable.*

Proof For convenience, let θ represent the structural substitution $\{[d](w\ \bar{t})/[d]w\}$. This proof is by induction on the type of $\gamma d.(s\theta\ \bar{t})$.

In the case of atomic or constant type, since $(s\theta\ \bar{t})$ is assumed reducible, by **CR 1** it is SN. As a neutral term, $\gamma d.(s\theta\ \bar{t})$ reduces in one step to either $\gamma d.s'$ (this case includes γ-renaming), or to $(s\theta\ \bar{t})$ if γd is vacuous. In either case, $\gamma d.(s\theta\ \bar{t})$ is SN because $(s\theta\ \bar{t})$ is SN. Thus by **CR 3** it is reducible.

In the case of type $A \supset B$, we need to show that $(\gamma d.s\theta\ \bar{t})u$ is reducible for any reducible term $u \in RED_A$. This term reduces in one step to the following forms:

- $(\gamma d.s\theta\ \bar{t})\ u \longrightarrow_1 \gamma d.((s\theta\ \bar{t})\{[d](w\ u)/[d]w\}\ u) = \gamma d.(s\theta\{[d](w\ u)/[d]w\}\ \bar{t}u)$

  The equality holds because d is not free in $\bar{t}$. But $(s\theta\{[d](w\ u)/[d]w\}\ \bar{t}u)$ is $(s\{[d](w\ \bar{t}u)/[d]w\}\ \bar{t}u)$. The statement of the lemma assumes that this term is also reducible. This term can only be of type B, so by inductive hypothesis, $\gamma d.(s\{[d](w\ \bar{t}u)/[d]w\}\ \bar{t}u)$ is reducible.

- $(\gamma d.s\theta\ \bar{t})\ u \longrightarrow_1 (\gamma d.s')\ u$ or $(\gamma d.s\theta\ \bar{t})\ u \longrightarrow_1 (\gamma d.s\theta\ \bar{t})\ u'$ where $s \longrightarrow_1 s'$ or $u \longrightarrow_1 u'$. In these cases, we show that the result of $\longrightarrow_1$ is reducible by another induction on $m(s\theta\ \bar{t}) + m(u)$. Both $(s\theta\ \bar{t})$ and u are assumed reducible, so by **CR 1** are both SN. The base case follows from the vacuous form of **CR 3**. The inductive cases are trivial.

- $(\gamma d.s\theta\ \bar{t})$ could be subject to a renaming rule. This case is likewise handled by induction on the lengths of the assumed finite reductions of $(s\theta\ \bar{t})$ and u.

In all cases, the neutral term $(\gamma d.s\theta\ \bar{t})u$ reduces in one step to reducible terms, and by **CR 3** is thus reducible. Therefore, $\gamma d.(s\theta\ \bar{t})$ is reducible. □

The above lemmas culminates in an relatively easy proof of the main theorem:

Theorem 17 *Let r be any term with free λ-variables included in $x_1, \dots x_k$ and free γ-variables included in $a_1, \dots a_n$. Let $q_1, \dots, q_k$ be any reducible terms and let $\bar{t_i}$ be any zero or more reducible terms $t_i^1 \dots t_i^m$ for each $1 \leq i \leq n$. If $r[q_1/x_1, \dots, q_k/x_k]\{[a_1](w\ \bar{t_1})/[a_1]w\} \dots \{[a_n](w\ \bar{t_n})/[a_n]w\}$ is typable then it is reducible.*

Proof The proof is by induction on r. For convenience, let σ represent the substitution $[q_1/x_1, \dots, q_k/x_k]$ and let $\bar{\theta}$ represent the (composed) structural substitutions $\{[a_1](w\ \bar{t_1})/[a_1]w\} \dots \{[a_n](w\ \bar{t_n})/[a_n]w\}$.

The substitution σ is for the case of $r = \lambda y.s$, which again proceeds exactly as in the proof of [GTL89] (using Lemma 13), as is the case for application terms. In the other cases the substitution σ plays no role.

The cases of $[d]s$ and $abort^F(s)$ follow from lemmas 14 and 15 respectively.

In the case of $r = \gamma d.s$, we need to show that $\gamma d.s\sigma\bar{\theta}$ is reducible. This means showing that $(\gamma d.s\bar{\theta})u$ is reducible for any reducible u. We can assume that d is not free in u by the usual variable convention. This neutral term reduces in one step to several possible forms, one of which is $\gamma d.(s\sigma\bar{\theta}\{[d](w\ u)/[d]w\}\ u)$. *By inductive hypothesis, the term $s\sigma\bar{\theta}\{[d](w\ u)/[d]w\}$ is reducible.* Therefore $(s\sigma\bar{\theta}\{[d](w\ u)/[d]w\}\ u)$ is reducible. The inductive hypothesis in fact says that for *any* vector $\bar{u}$, $s\sigma\bar{\theta}\{[d](w\ \bar{u})/[d]w\}$ is reducible and thus $(s\sigma\bar{\theta}\{[d](w\ \bar{u})/[d]w\}\ \bar{u})$ is reducible (if it is typable). Thus by Lemma 16 on the term $s\sigma\bar{\theta}$ (which replaces s in the statement of the lemma), $\gamma d.(s\sigma\bar{\theta}\{[d](w\ u)/[d]w\}\ u)$ is reducible.

In case the neutral term $(\gamma d.s\sigma\{[d](w\,\bar{t})/[d]w\})u$ reduces in one step to other forms, we argue by induction on the reduction lengths of the assumed SN terms u and $s\sigma\{[d](w\,\bar{t})/[d]w\}$ as we had in the proof of Lemma 16. $\Box$

The above theorem proves that all well-typed terms are reducible by setting q_i to x_i and each $\bar{t_i}$ to be of zero length. Thus by **CR 1**, *all well-typed terms are strongly normalizing.* $\Box$