Affine Control Logic

Chuck Liang
Hofstra University
Hempstead, NY

July 4, 2016

Abstract

This paper extends our previous work in combining classical logic with intuitionistic logic [LM13a, LM13b] to also include affine linear logic (linear logic with weakening), resulting in a unified system that we call *Affine Control Logic.* Connectives from different logics can mix without collapse, and fragments of the logic are defined not by restricting proofs but entirely by the subformula property. Linear logic is adjusted so that contraction is not enabled by the exponential operator ? but by a restricted form of Peirce's formula. This formula, when admissible, enables contractions on both the left and right-hand sides of sequents, thus also replacing !. Classical fragments of proofs are better isolated from non-classical fragments using this technique. We define a phase semantics for this logic that finds the Kripke semantics of intuitionistic logic as a fragment. We give a cut elimination proof that requires a combination of methods not commonly found in other such proofs, thus affirming the proof-theoretic novelty of this logic. Computationally, the goal of this logic is to allow affine-linear computational interpretations of proofs to be combined with classical interpretations such as the $\lambda\mu$ calculus. We show how cut-elimination must respect the boundaries between classical and non-classical modes of proof which correspond to delimited control effects. A natural deduction system with a term calculus is defined for a fragment of this logic.

Contents

1 Introduction

The goal of this paper is to formulate a *unified logic* that combines classical, intuitionistic and affine-linear logics (restricting contraction but allowing weakening). The connectives of this logic are similar to those of linear logic but contractions are controlled in a very different way. Instead of dual exponential operators ? and !, a single principle, based on a specialized form of Peirce's formula, is used to enable contractions on the left and right-hand side of sequents when certain conditions are reached. The system Affine Control Logic (ACL) is an extension of our PCL system presented in [LM13b].

This system also descend from previous attempts at formulating unified logics, including LU [Gir93] and our own LKU [LM11]. These system were all based on linear logic. Linear logic embeds both classical and intuitionistic logics, but it is limited in its ability to *mix* them. For example, the interpretation of intuitionistic implication as $!A \multimap B$ is a crucial component of linear logic. However, this interpretation is not compatible with the fragment that interprets classical logic. Consider $?((!A \multimap B) \oplus C)$ (equivalently $?(!A \multimap B) \parr ?C$): here we are attempting to write an intuitionistic implication as a subformula of a classical disjunction. The strength of intuitionistic implication is compromised: it may be possible to use the assumption A to prove C: the intuitionistic meaning and proof structure of $!A \multimap B$ would not survive such a mixture.

Our goal is to isolate classical reasoning to fragments of proofs, thereby allowing them to coexist with non-classical fragments without collapsing one into the other. One component of this effort relies on the idea of "stoup" of LC [Gir91]. When a "positive" formula occupies the stoup the proof takes on the characteristics of an intuitionistic proof. This phenomenon can be described as a special instance of "focusing." We shall use focusing in a similar way. However, the focusing approach alone is not entirely satisfactory: a (multiplicative) implication $A \to B$ is "negative" and cannot occupy the stoup. In [LM13b] we defined *Polarized Control Logic* (PCL), which unified classical and intuitionistic logics using an alternative to positive/negative polarization. In this paper we modify PCL to obtain a new propositional logic that unifies intuitionistic logic, classical logic, and the essential elements of affine linear logic. The unification of these logics is achieved semantically, proof-theoretically, and in terms of the computational interpretation of proofs. We define a new phase semantics where *facts* are upwardly closed sets, which forms a natural extension of the Kripke semantics of intuitionistic logic. We define a sequent calculus with structural rules that require an essential element of focusing and a cut-elimination procedure (the heart of the paper) that requires new techniques because of these rules. Finally, we also define a natural deduction system with a computational interpretation that includes $\lambda\mu$-style controls while also allowing non-classical constructs to retain their strengths. The unified system is also greater than the sum of its parts, as new fragments (new logics) can be identified easily by virtue of the subformula property. We also show that the transition between the different modes of proof, classical, intuitionistic and affine, can be interpreted using delimited control operators.

Many uses of linear logic are also valid in affine logic, and some are enhanced by it. Semantically, the models of affine logic are closer to those of intuitionistic logic, and this will allow us to adopt our previous efforts in combining classical logic with intuitionistic logic [LM13a, LM13b]. Creating an alternative to linear logic is no easy task. Linear logic generalizes the principles of Gentzen in allowing cut elimination in a setting where *some but not all* formulas are subject to contraction. This is a central role of the ?/! duality, and we will need to find a new approach. Consistent with our previous work, the key is to adopt a new form of negation centered around a reinterpretation of the constant $\perp$. However adopting this negation to affine connectives is not straightforward, and will require new proof theoretic techniques and semantic interpretation.

Linear logic has offered much insight in the way that it decomposes proofs in classical and intuitionistic logics. The reader may be tempted to try to understand ACL in the same way. It is not possible, however, to directly translate ACL into linear logic (affine or proper). Already in [LM13b] we showed that proofs using the new form of negation cannot be embedded in linear logic without formulas such as $?!A$. Such formulas hopelessly destroy *focus* and cannot be used to define *synthetic connectives*. ACL further requires a structural rule that would appear delusional in terms of linear logic (imagine the dereliction rule inverted). Thus, much of this paper is devoted to showing that ACL can stand on its own as a new logic, with its own notion of model, sequent calculus, cut elimination, and soundness and completeness results. In particular, for a more than casual understanding of this logic, the main components of the cut elimination proof should be read.

This paper is an extended version of [Lia16].

2 Syntax and "Colors"

We focus on propositional logic in this presentation. The addition of first order quantifiers would be a rather standard exercise. Much more interesting is the addition of second order quantifiers, which will be discussed in Section 9. Formulas of propositional ACL are freely composed from connectives &, $\oplus$, $\rightarrow$, $\multimap$, $\otimes$ and $\vee$, constants $\top$, 0 and $\perp$, and atomic formulas. The symbol $\multimap$ represents affine implication while $\rightarrow$ represents intuitionistic implication. Although we use the symbols & and $\oplus$ from linear logic, here they can be classical or non-classical. Intuitionistic disjunction requires a separate connective: $\vee$. The linear constant 1 is equivalent to $\top$ in affine logic. The exponentials ? and ! will be replaced by the ability to mix $\otimes$ and $\multimap$ with intuitionistic and classical formulas: this will result in stronger invariants than the unrestricted use of ! and ?. Despite using the same symbol, the constant $\perp$ in ACL is entirely different from its counterpart in linear logic. The new role of $\perp$ is the most significant distinction between ACL and affine linear logic.

We use a device similar to '*polarization,* but to avoid confusion we use the term *"colors."* Formulas are colored *red* or *green.* [1] Atomic formulas are arbitrarily colored, while the coloring of other formulas is as follows:

- $\perp$ and $\top$ are green; 0 is red.
- $A \,\&\, B$ is green if both A and B are green, otherwise, it is red.
- $A \oplus B$ is green if A is green or B is green, otherwise, it is red.
- $A \rightarrow B$ is green if B is green, otherwise, it is red.
- $A \multimap B$ is green if B is green, otherwise, it is red.
- $A \otimes B$ and $A \vee B$ are always red.

Since there are two implications and two constants for false, a green $\perp$ and a red 0, there are four forms of negation in ACL. We define abbreviations for them as follows:

$$-A = A \multimap \perp \qquad \neg A = A \rightarrow \perp \qquad \sim A = A \rightarrow 0 \qquad \frown A = A \multimap 0$$

$\neg A$ and $-A$ are logically equivalent but give different proof structure. This paper is primarily concerned with $-A$.

We use the letter E for an arbitrary green formula and e for either a green atom or $\perp$. Unlike the positive/negative polarization, red and green are not "duals" of each other. For example, if E is green then $-E$ is still green: "$-$" is *not* an involutive negation. It is possible for red and green formulas to be logically equivalent. In contrast, $?X \multimap !Y$ is not provable for any X, Y in linear logic. ACL is not just a repackaging of linear logic. Conceptually, green means classical and red means *arbitrary:* classical or non-classical. The coloring of & and $\oplus$ is similar to the positive/negative polarization of LC [Gir91], but these are the only cases that intersect. The polarities of LC are consistent with those of focused proofs, which are also the subjects of some of our own work. Although the proof systems of ACL use a small element of focusing, that is not the purpose of the red/green colors. It is possible to explain positive/negative polarization purely syntactically, in terms of the invertibility of inference rules and the role that they play in controlling cut elimination. In contrast, the red and green colors represent two levels of *soundness.* The role they play in unifying logics can only be explained semantically.

[1] In our previous work these colors were also called polarities. However, these "polarities," although related, are also not the same as polarities of the same names in [LM13a]. The original semantic motivation for the meaning of $\perp$ in PCL was described in [LM13a] as the *second largest element of a Heyting algebra.* The models of PCL are a subclass of the models of *PIL* as formulated in [LM13a]. However, neither logic is a fragment of the other.

3 Semantics

The semantic tradition of linear logic emphasize the understanding of proofs, not formulas. Traditional model theory is sometimes disparaged because it is concerned primarily with truth and consistency, but not with proofs. This type of semantics, however, is exactly what we need in order to explain the difference between the red and green colors, for they define two levels of *provability.* The green $\perp$ defines a stronger level of consistency compared to the red 0. The syntactic properties that green formulas induce will appear fanciful without a proper explanation from this perspective.

Define a *Phased Frame* to be a structure $\langle W, \preceq, r, \cdot \rangle$ where $\preceq$ is a partial ordering relation on the set of *possible worlds* (or *phases*) W. This structure also forms a commutative monoid with operation $\cdot$ and unit $r \in W$. We write ab for $a \cdot b$. Given two sets of worlds A and B, $AB = \{xy : x \in A,\ y \in B\}$. It always holds that $ab = ba$ and $(ab)c = a(bc)$. We further require the following property:

- **a $\preceq$ b if and only if ac = b for some c.**

It is important that $\preceq$ is a proper partial order and not just a preorder. Thus not every commutative monoid gives rise to such a structure: **inverses are not allowed.** The models that we shall build for the completeness proof will have monoids formed by multisets of formulas, with multiset union being the monoid operation. This is also the typical way to prove completeness for phase semantics, but this *extra structure* was never important until now.

By inference, it always holds that $a \preceq ab$. Crucially, the anti-symmetry of $\preceq$ means that the unit r is unique and is the *least* element of W since $r \preceq ru = u$ for all $u \in W$. We refer to r as the *root.* The following properties are also easily inferred:

- if $a \preceq b$ then $ac \preceq bc$.
- if $a \preceq b$ and $bb = b$, then $ab = b$.

It is also worth noting that finite phased structures must contain a *top world* t with the property that $tt = t$.

Our phased models are closer to those of Okada [Oka02] than to Girard's original. The principal difference between our models and those of linear logic is twofold. First, the *facts* of the space (subsets of W that can interpret formulas) are upwardly closed sets. A set S is upwardly closed if $x \in S$ and $x \preceq y$ implies $y \in S$. This corresponds to the monotonicity property of intuitionistic Kripke models. However, unlike intuitionistic logic, not all upwardly closed sets are necessarily facts. The second, and most important difference is that, in phase semantics for linear logic $\perp$ is represented by any arbitrary set, whereas here it is fixed to be $W \backslash \{r\}$, *the upwardly closed set that consists of all worlds above the root.* The two sets W ($\top$) and $W \backslash \{r\}$ ($\perp$) form an embedded, two-element boolean algebra with nothing in between them.

Formally, let an *Ordered Phase Space* be a structure of the form $(W, \cdot, r, \preceq, D)$, where W, $\preceq$, r and $\cdot$ satisfy the requirements of a phased frame. D is a set of upwardly closed subsets of W called *facts* that is furthermore required to satisfy the following properties:

1. D contains $W \backslash \{r\}$, which are upwardly closed ($\preceq$ is a partial order, not just a preorder).
2. For any subsets A and B of W such that $B \in D$, the set $\{x \in W : \text{ for all } y \in A,\ xy \in B\}$ is also in D. This set is upwardly closed because if $x \preceq x'$ then $x' = xz$ and $xzy \in B$ since B is upwardly closed. We can call this set the *pseudo-pseudocomplement* of A relative to B.
3. D must be closed under the following *closure operator* on subsets of W: $cl(S) = \bigcap\{V \in D : S \subseteq V\}$. Upward closure is preserved by arbitrary intersections.

Note that by the first two requirements, D must also contain all of W, which is equal to $\{x \in W : \text{ for all } y \in W \backslash \{r\},\ xy \in W \backslash \{r\}\}$. Clearly r is in this set.

It holds that $S \subseteq cl(S)$ and if S is already a fact in D then $S = cl(S)$. It also holds that $cl(cl(S)) = cl(S)$, $cl(S)V \subseteq cl(SV)$ and if $S \subseteq V$ then $cl(S) \subseteq cl(V)$. We will not distinguish between formulas A and their interpretation in phase space A^p except when there is possibility for confusion.

Given $S \subseteq W$, let $I(S) = \{u \in S : uu = u\}$. These are the worlds that admit contraction. $I(W)$ is never empty since $rr = r$. A phase model on an ordered phase space is defined by a mapping from atomic formulas to facts, with the following conditions:

- Red atoms are mapped to arbitrary facts (elements of D).
- Green atoms are mapped to either $W\backslash\{r\}$ or to W, i.e., to either $\bot^p$ or $\top^p$.

The interpretation (valuation) of all formulas is then defined as follows:

- $\top$ ($\top^p$) is represented by $W = cl(\{r\})$. There is no need for both $\top$ and 1 in affine logic.
- $\bot$ is represented by $W\backslash\{r\}$
- 0 is $cl(\emptyset) = \bigcap D$, the smallest possible fact ($\emptyset$ is the empty set).
- $A \otimes B$ is $cl(AB) = cl(\{xy : \ x \in A, \ y \in B\})$
- $A \multimap B$ is $\{x \in W : \text{ for all } y \in A, \ xy \in B\}$
- $A \rightarrow B$ is $\{x \in W : \text{ for all } y \in I(A), \ xy \in B\}$
- $A \oplus B$ is $cl(A \cup B)$
- $A \vee B$ is $cl(I(A) \cup I(B))$
- $A \,\&\, B$ is $A \cap B$

A formula A is valid in a model if $r \in A^p$, i.e., $A^p = W$. A formula is valid if it is valid in all models.

It is easily shown by induction on formulas that *all green formulas valuate to* $\bot$ *or* $\top$.

Lemma 1 *For every green formula E, $E^p \neq \top^p$ if and only if $E^p = \bot^p$.*

It should now be clear why $A \otimes B$ (and $A \vee B$) is always red: we cannot guarantee that it will always valuate to $\top$ or $\bot$ even if A "and" B are green. In the syntactic proof theory of ACL, this property of $\otimes$ is reflected by the impermutability of its introduction rule with respect to certain structural rules. The constant $\top$ can be designated red or green: it makes little difference in this case.

An important consequence of interpreting $\bot$ as $W\backslash\{r\}$ is that $\mathbf{A} \oplus -\mathbf{A}$ **is valid**. Note that showing $r \in A \multimap B$ is equivalent to showing that $A \subseteq B$. Thus if $r \notin A^p$ then $A^p \subseteq \bot^p$ and therefore $r \in -A^p$. Thus $r \in A^p \cup -A^p \subseteq cl(A^p \cup -A^p)$.

An even more important consequence is that Peirce's formula in the form $((\mathbf{P} \multimap \mathbf{E}) \rightarrow \mathbf{P}) \multimap \mathbf{P}$ is valid as long as $\mathbf{E}$ is green. $\mathbf{P}$ can be arbitrary (the occurrence of $\rightarrow$ is stronger than $\multimap$ in that position). If $r \in P$ ($r \in P^p$) the result is obvious since then $(P \multimap E) \rightarrow P \subseteq P = W$ because P is upwardly closed. If $r \notin P$, then $r \in (P \multimap E)$ since $\bot \subseteq E$ (all green formulas valuate to $\bot$ or $\top$). Then, since $rr = r$, we also have $(P \multimap E) \rightarrow P \subseteq P$.

The syntactic consequences of the validity of this version of Peirce's formula are profound. It means that, upon encountering a green formula as the current or *stoup* formula in a proof, contraction is enabled, *not just on the green formula itself but on all formulas at that point in the proof.*

The closure operator is not needed in all the cases of A^p. In the cases of $\otimes$, $\oplus$ and 0, the sets defined are already upwardly closed even without applying the closure operator cl. For example, AB is upwardly closed if either A or B is a fact: if $xy \in AB$ with $x \in A$ and $y \in B$, and $xy \preceq z$, then $xyc = z$ for some c with $x \in A$ and $yc \in B$ because $y \preceq yc$ and B is upwardly closed; thus $z = xyc \in AB$. Similarly, the union of two upwardly closed sets remains upwardly closed. It would be simpler to allow all upwardly closed sets to be facts, but completeness would be lost. The cases that require the cl operator above correspond to the connectives with *non-invertible* right introduction rules in our sequent calculus. If all upwardly closed sets are facts, or if ACL is restricted to those connectives that do not require the closure operator cl, then these phase models are perhaps better seen as Kripke models: $u \in A^p$ can be read as "$u \models A$."[2].

The constant 0 is not necessarily interpreted by the empty set, which is to be expected in phase semantics. The completeness proofs of such semantics typically define a set of multisets of formulas and multiset union as the monoid operation. This means that we cannot guarantee that these multisets will be consistent (does not derive 0), because the union of two consistent multisets may become inconsistent. In the phase semantics of linear logic, 0 is interpreted by $W^\perp$. However, our "$\perp$" has an entirely different meaning than $\perp$ in linear logic. Consistency can only be guaranteed at the root, which has the property $rr = r$, (i.e., it can be a *set* as opposed to a multiset). In models with a non-empty 0^p, the empty set, which is upwardly closed, is *not* a fact.

It easily holds that $0^p \subseteq A^p$ for all formulas A. The largest possible 0^p is $\perp^p$ and the smallest possible $\perp^p$ is the empty set, but these cases only occur if D is just a two-element boolean algebra. In terms of Kripke style semantics, the fact that 0^p may not be empty means that there will be possible worlds that "force" 0. Such kinds of Kripke models are not unknown [Vel76, ILH10]. Furthermore, because the root world cannot be in 0^p, there is still no model for 0 (or for $\perp$).

For another example of reasoning with this semantics, one can check that $(A \otimes B) \multimap A$ is valid, confirming the admissibility of weakening as follows. We need to show that $r \in (A \otimes B) \multimap A$. This means showing that $(A \otimes B)^p \subseteq A^p$. By A^p is upwardly closed and $x \preceq xy$, so $A^pB^p \subseteq A^p$. Thus $(A \otimes B)^p = cl(A^pB^p) \subseteq cl(A^p)$. But $cl(A^p) = A^p$ since A^p is a fact.

The validity of Peirce's formula forms the core of ACL. In the form $((P \multimap E) \rightarrow P) \multimap P$, or $(-P \rightarrow P) \multimap P$, it implies the admissibility of contraction on the right-hand side of sequents. In syntactic proof theory it is possible to embed this principle as an inference rule:

$$\frac{-P \vdash P}{\vdash P}$$

Peirce's formula also implies the admissibility of a counterpart to itself: $(\mathbf{P} \multimap \mathbf{P} \multimap \mathbf{E}) \multimap \mathbf{P} \multimap \mathbf{E}$. This formula implies the admissibility of *left-side* contractions

$$\frac{P, P \vdash E}{P \vdash E}$$

This counterpart of Peirce's formula (disregarding colors) is provable in intuitionistic logic (if $\multimap$ is replaced by $\rightarrow$, by $\lambda x \lambda y.xyy$), while Pierce's formula is not. What is important to note however, is that *this formula can also be proved with contraction on the right.* Using standard sequent calculus rules:

$$\dfrac{\dfrac{\dfrac{\dfrac{P \vdash P \qquad P \rightarrow Q \vdash P \rightarrow Q, Q}{P, P \rightarrow (P \rightarrow Q) \vdash P \rightarrow Q, Q}}{P \rightarrow (P \rightarrow Q) \vdash P \rightarrow Q, P \rightarrow Q}}{P \rightarrow (P \rightarrow Q) \vdash P \rightarrow Q}}{\vdash (P \rightarrow (P \rightarrow Q)) \rightarrow P \rightarrow Q}$$

The proof uses weakening (on Q) and contraction on the right, *but not on the left.* The admissibility of Peirce formula is stronger, and implies the admissibility of contraction on both the left and right-hand sides of sequents. One might call it a *self dual* principle. It replaces the duality of ? and !, where one is used for contractions on the right while the other one is used on the left.

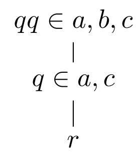
The semantics also determine the validity of the following examples

- $A \multimap (A \otimes A)$ is not valid, but $A \rightarrow (A \otimes A)$ is valid (because of the $vv = v$ assumption). This shows that linearity is present and distinguishable.

[2]Under this interpretation, $u \models A \rightarrow B$ holds iff *for all* v, $vv = v$ *and* $v \models A$ *implies* $uv \models B$. Under the global assumption that $vv = v$, i.e, $I(W) = W$, we can show that this condition is equivalent to the traditional Kripke model definition of intuitionistic implication: *for all* $v \succeq u$, *if* $v \models A$ *then* $v \models B$. The argument uses the properties noted above: if $u \models A \rightarrow B$, and if $v \models A$ for $v \succeq u$, then with the assumption that $vv = v$ it follows that $uv \preceq vv = v \preceq uv$, and so $v \models B$.

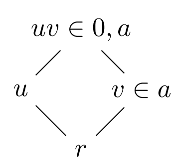
- $(A \rightarrowtriangle B) \rightarrowtriangle (A \rightarrow B)$ is valid. This is dereliction. The converse is not valid unless B is green. $A \otimes B \rightarrowtriangle A \,\&\, B$ is also valid (because of weakening).
- $[A \otimes (A \rightarrow B)] \rightarrowtriangle B$ is not valid, but $[A \otimes (A \rightarrow B)] \rightarrow B$ is valid. This is consistent with the linear logic translation that $A \rightarrow B$ is $!A \multimap B$. However, $!(A \otimes B) \multimap !A$ holds if weakening is always available.
- $(A \,\&\, B) \rightarrowtriangle (A \otimes B)$ is not valid but $(A \,\&\, B) \rightarrow (A \otimes B)$ is valid.
- $(A \rightarrowtriangle B) \rightarrowtriangle (-A \oplus B)$ is valid, but the converse $(-A \oplus B) \rightarrowtriangle (A \rightarrowtriangle B)$ is only valid if B is green. $A \rightarrowtriangle E$ and $A \rightarrow E$ are equivalent when E is green.
- Several other important properties, mostly inherited from PCL [LM13b], should also be noted. These include the fact that *none of the negations $\neg$, $\sim$, $-$, and $\frown$ are involutive.* In particular, $--A \rightarrowtriangle A$ is not valid: our $\bot$ is not the same as the $\bot$ of linear logic. We do have that $\neg\neg E \rightarrow E$ and $--E \rightarrowtriangle E$ are valid if E is green. It also holds as an *admissible rule* that if $--A$ is valid, then A is valid. Additionally, the De Morgan law $-(A \,\&\, B) \rightarrowtriangle -A \oplus -B$ is valid: the others cases are already intuitionistically valid: i.e., they do not require $\bot$ to be green.

The following model, with three distinct worlds r, q and qq, verifies several of the examples above:



Here it is assumed that $qq = qqq = qqqq$. All upwardly closed sets in this model are facts. This means that $cl(S) = S$ for all upwardly closed S: this is an intuitionistic Kripke frame, but with $q \neq qq$. If $u \in Q^p$ we will say that *"u forces Q."* The interpretation of the atoms a, b, c are that $a^p = c^p = \{q, qq\}$ and $b^p = \{qq\}$. In other words q forces a, c, and qq forces a, b, c. For example, r forces $a \rightarrow b$ since the only world above r that has the property $uu = u$ is qq. But $r \notin a \rightarrowtriangle b$ because $q \in a$ but $rq = q \notin b$. Another example: $q \in a \,\&\, c$ but $q \notin a \otimes c$ because $qq \neq q$. The same model also shows that $\neg$ and $-$ are not involutive negations in ACL: let d be a red atom that is not forced at any of the worlds. Then all worlds above r forces $\neg\neg d$ and $--d$ because they force $\bot$ ($\bot^p = \{q, qq\}$), but they do not force d, and thus $r \notin --d \rightarrowtriangle d$ and $r \notin \neg\neg d \rightarrow d$. The same model also plays the part of an intuitionistic Kripke model and shows that $b \oplus \sim b$ is not valid, and that $\sim\sim b \rightarrow b$ is not valid (regardless of the color of b), since $r \in \sim\sim b$ but $r \notin b$.

Our semantics preserve the advantage of Kripke semantics in the existence of small but effective countermodels. However, it should be noted that the monoid's closure property also diverges from what is typically expected in Kripke semantics. The countermodel for $\sim a \oplus \sim\sim a$ requires a top world that "forces" 0:



In this model $0^p = \{uv\}$ and $a^p = \{v, uv\}$. It can be assumed that $I(W) = W$. The empty set is not a fact since 0^p must be the smallest fact. In the intuitionistic Kripke countermodel the top world is not needed, but it is rather unavoidable if the frame is a monoid.

It is important to recognize that *the meaning of affine and intuitionistic (red) formulas do not necessarily collapse when mixed with classical (green) formulas.* As an example of this property, the green formula $E \oplus \sim E$ is still not valid because the subformula $\sim E = E \rightarrow 0$ is an intuitionistic implication. The root world is the only classically consistent world (i.e., consistent with respect to $\bot$) The validity of red formulas and subformulas are thus determined by more than just the root. Another example is that $E \rightarrowtriangle E \otimes E$ is not valid even with the green E.

$$\frac{[\Theta : A], \Gamma; \Delta\Theta \vdash A}{\Gamma; \Delta\Theta \vdash A}\ Lock \qquad \frac{\Gamma; \Delta\Theta \vdash A}{[\Theta : A], \Gamma; \Delta \vdash e}\ Unlock \qquad \frac{\Gamma; \Delta, A \vdash B}{A, \Gamma; \Delta \vdash B}\ Dr \qquad \frac{A, \Gamma; \Delta \vdash e}{\Gamma; \Delta, A \vdash e}\ Pr$$

$$\frac{\Gamma; \Delta, A \vdash B}{\Gamma; \Delta \vdash A \multimap B} \multimap R \quad \frac{\Gamma; \Delta_1 \vdash A \quad \Gamma; \Delta_2, B \vdash C}{\Gamma; \Delta_1\Delta_2, A \multimap B \vdash C} \multimap L \quad \frac{A, \Gamma; \Delta \vdash B}{\Gamma; \Delta \vdash A \to B} \to R \quad \frac{\Gamma; \vdash A \quad \Gamma; \Delta, B \vdash C}{\Gamma; \Delta, A \to B \vdash C} \to L$$

$$\frac{\Gamma; \Delta_1 \vdash A \quad \Gamma; \Delta_2 \vdash B}{\Gamma; \Delta_1\Delta_2 \vdash A \otimes B} \otimes R \quad \frac{\Gamma; \Delta, A, B \vdash C}{\Gamma; \Delta, A \otimes B \vdash C} \otimes L \quad \frac{\Gamma; \Delta \vdash A_i}{\Gamma; \Delta \vdash A_1 \oplus A_2} \oplus R \quad \frac{\Gamma; \Delta, A \vdash C \quad \Gamma; \Delta, B \vdash C}{\Gamma; \Delta, A \oplus B \vdash C} \oplus L$$

$$\frac{\Gamma; \Delta \vdash A \quad \Gamma; \Delta \vdash B}{\Gamma; \Delta \vdash A \,\&\, B} \,\&R \quad \frac{\Gamma; \Delta, A_i \vdash C}{\Gamma; \Delta, A_1 \,\&\, A_2 \vdash C} \,\&L \quad \frac{\Gamma; \vdash A_i}{\Gamma; \Delta \vdash A_1 \vee A_2} \vee R \quad \frac{A, \Gamma; \Delta \vdash C \quad B, \Gamma; \Delta \vdash C}{\Gamma; \Delta, A \vee B \vdash C} \vee L$$

$$\frac{}{\Gamma; \Delta, a \vdash a}\ Id \qquad \frac{}{\Gamma; \Delta \vdash \top}\ \top R \qquad \frac{}{\Gamma; \Delta, 0 \vdash A}\ 0L \qquad \frac{}{\Gamma; \Delta, \bot \vdash e}\ \bot L$$

Figure 1: The Unified Sequent Calculus LAC. e must be a green atom or the constant $\bot$

4 A Single Conclusion Sequent Calculus

Various choices can be made in designing a proof system for ACL. The Kripke-like semantics suggests a system similar to the Beth-Fitting intuitionistic tableaux [Fit69], which is often written as a multiple-conclusion version of intuitionistic sequent calculus. This is the approach we used in PCL because, when converted to a natural deduction form, it offered more opportunities to assign computational meaning to proofs. We in fact showed how the proof system of PCL naturally suggested a form of dynamic scoping for continuation variables, and devised an abstract machine to realize this interpretation. While these options are also available for ACL, for purely proof-theoretical study we first choose a simpler, single-conclusion sequent calculus. While multiple conclusions offer more flexibility, a single conclusion version offers a few more invariants that will be useful in proving cut elimination.

The sequent calculus LAC is found in Figure 1. Here, Δ is a multiset but Γ is a set and A, Γ does not preclude the possibility that $A \in \Gamma$. Weakening and contraction in Γ, and weakening in Δ, hold as admissible rules. The semantic interpretation of a sequent $\Gamma; \Delta \vdash A$ is as for the formula $\Gamma^{\&} \to (\Delta^{\otimes} \multimap A)$, where $\Gamma^{\&}$ is the &-conjunction over formulas in Γ and $\Delta^{\otimes}$ is the $\otimes$-conjunction over formulas in Δ. An empty Γ or Δ means $\top$. Elements $[\Theta : A]$ are treated as any other formula in Γ, and has the same meaning as $-(\Theta^{\otimes} \multimap A)$. The special notation is used so that this formula can only be principal as part of *Unlock*. *Unlock* is a focused version of the $\to L$ rule combined with the $\bot L$ and $\otimes L$ rules. (see Corollary 5 of Section 6 for further explanation). Intuitively, from a computational perspective, one can regard $[\Theta : A]$ as a form of *closure: Lock* saves not only a copy of the current continuation but also a part of its *operating environment.* The continuation is no longer stateless.

Although *Lock* can be applied at any point in a proof, the effect of contraction is only available when *Unlock* can be applied. Thus if a formula has no green subformulas, it can only have a non-classical proof. We informally refer to the singleton formula on the right-hand side as the *stoup.* The stoup is never empty. A formula A is provable if $; \vdash A$ is provable.

Many linear logic proof systems use dual contexts on the left (and sometimes right) hand side in sequents. If one examined the part of LAC that only unifies intuitionistic logic with affine-linear logic, then this sequent calculus is similar to that of Lolli [HM94], Forum [Mil96], as well as several other systems including LU [Gir93], DILL [Bar96], and Andreoli's focusing sequent calculus [And92], where the practice likely originated.

The following are sample proofs. The first is a proof of a version of the excluded middle, $A \oplus -A$ $\;(A \oplus A \multimap$

$\bot$), and the second is that of a version of the double-negation axiom, $\sim\!-A \rightarrowtriangle A$ $(((A \rightarrowtriangle \bot) \to 0) \rightarrowtriangle A)$.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{;A \vdash A}\,Id}{;A \vdash A \oplus -A}\,\oplus R}{[: A \oplus -A]; A \vdash \bot}\,Unlock}{[: A \oplus -A]; \vdash -A}\,\rightarrowtriangle R}{\dfrac{[: A \oplus -A]; \vdash A \oplus -A}{;\vdash A \oplus -A}\,Lock}\,\oplus R
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{;A \vdash A}\,Id}{[: A]; A \vdash \bot}\,Unlock}{[: A]; \vdash -A}\,\rightarrowtriangle R \quad \dfrac{}{[: A]; 0 \vdash A}\,0L}{[: A]; \sim\!-A \vdash A}\,\to L}{;\sim\!-A \vdash A}\,Lock}{;\vdash \ \sim\!-A \rightarrowtriangle A}\,\rightarrowtriangle R
$$

The proofs will fail if $\bot$ was replaced with a red formula, such as 0 ($A\oplus \sim\!A$ remains unprovable). On the other hand, if 0 was replaced with $\bot$ in the proof of $\sim\!-A \rightarrowtriangle A$, then that proof will also fail, unless A is green ($\bot \rightarrowtriangle A$ holds only for green A). None of the negations of ACL are involutive without conditions, but the negations can be mixed to give the desired computational effect (i.e., the $\mathcal{C}$ control operator). A slight adjustment to the proof of $\sim\!-A \rightarrowtriangle A$ also proves a version of Peirce's formula, $(-A \to A) \rightarrowtriangle A$: replace $0L$ with an Id rule.

Given the semantic validity of the Peirce-like formula $(-A \to A) \rightarrowtriangle A$, it would be valid to design structural rules of the following forms:

$$
\dfrac{-A, \Gamma; \Delta \vdash A}{\Gamma; \Delta \vdash A}\,Lock \qquad \dfrac{\Gamma; \Delta \vdash A}{-A, \Gamma; \Delta \vdash e}\,Unlock
$$

Here, the Unlock rule can be seen as just a special case of $\to L$, since $\bot \to e$ is valid for any green e. Indeed these simplified rules are enough for the examples above. However, they are not enough for cut-elimination. The most crucial case of cut-reduction is permutation of cut above a contraction. In particular, consider:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\Gamma_1; \Delta_1 \vdash A}{-A, \Gamma_1; \Delta_1 \vdash e}\,Unlock}{\vdots}}{\dfrac{-A, \Gamma; \Delta \vdash A}{\Gamma; \Delta \vdash A}\,Lock} \qquad \Gamma; \Delta', A \vdash B}{\Gamma; \Delta\Delta' \vdash B}\,cut
$$

Multiple, *structural* cuts are needed to cut the extra copies of A that appear as $-A$ on the left-hand side, which may be unlocked multiple times when green formulas are encountered on the right-hand side. These multiple cuts will entail not only the need to contract copies of B at the end, but also copies of the multiset Δ', which is not normally subject to contraction. For example, the proof of $-A, \Gamma; \Delta \vdash A$ may require a $\otimes R$ rule that splits the context Δ, but which *copies* $-A$ to each premise: these copies will spawn multiple copies of Δ' in the resulting proof after cuts are applied to the subproofs. Thus *Lock* must be generalized to contract more than just the right-hand side (stoup) formula. The given rules for *Lock* and *Unlock* in Figure 1 subsume the simpler cases since we can choose Θ to be empty (thus representing the formula $\top$). Clearly the generalized *Lock* still represents instances of $(-P \to P) \rightarrowtriangle P$, with P replaced by $\Theta^{\otimes} \rightarrowtriangle A$: the rule is semantically sound. The *Unlock* rule optionally retains $[\Theta : A]$ inside Γ in the premise.

An example of a formula that requires not just the stoup formula to be copied by Lock is the following:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{;b \vdash b}{b; \vdash b}\,Dr \qquad ;a \vdash a}{b; b \multimap a : a}\multimap L}{[b \multimap a : a], b; \vdash e}\,Unlock}{[b \multimap a : a]; \vdash b \to e}\to R \qquad ;e \vdash e}{[b \multimap a : a]; (b \to e) \multimap e \vdash e}\multimap L}{[b \multimap a : a], (b \to e) \multimap e; \vdash e}\,Dr \qquad ;b \multimap a, a \vdash a}{[b \multimap a : a], (b \to e) \multimap e; b \multimap a, e \to a \vdash a}\to L}{(b \to e) \multimap e; b \multimap a, e \to a \vdash a}\,Lock}{; \vdash (b \multimap a) \multimap ((b \to e) \multimap e) \to (e \to a) \multimap a}\to/\multimap R^*
$$

Here, e is a green atom and a is a red one (b is arbitrary). This mixture of intuitionistic and affine implication requires that the premise $b \multimap a$ be copied because of the requirements of $\to L$.

The Pr rule corresponds to the counterpart to the Peirce-like formula: $(A \to -A) \multimap -A$. It is in fact possible to derive a rule similar to Pr using the generalized *Lock* and *Unlock* rules:

$$
\dfrac{\dfrac{\Gamma; \Delta, A, A \vdash e}{[A : e], \Gamma; \Delta, A \vdash e}\,Unlock}{\Gamma; \Delta, A \vdash e}\,Lock
$$

That is, contraction inside the multiset context also becomes valid when a green e is found in the stoup. The generalized *Lock* rule captures not only the Peirce-like formula for right-side contraction, but also for left-side contraction. This property is crucial for cut elimination to succeed. The Pr rule is not technically equivalent to this derived rule because of our *dyadic* representation of sequents (using both sets and multisets). Thus Pr is kept as a separate inference rule. It is needed to prove formulas such as $(A \to -A) \multimap -A$.

The key contrast between the role of green formulas and that of the ? operator in linear logic can be described as *dynamic versus static* approaches to allowing contraction. Once we place a ? before a formula, it can be contracted anywhere. However, $?A$ only enables contraction on itself. In contrast, the presence of a green formula in the stoup effectively switches the proof into a "classical mode:" contractions become unlocked on *all* formulas, left and right. Conceptually, this means that we do not have to keep ? on all the formulas that may *at some point* require contraction. Subformula occurrences of green formulas mean that it is possible *but not necessary* for a proof to include classical fragments. They determine *where in the proof*, as opposed to *on which formulas,* are contractions allowed. Classical reasoning is thus localized inside segments of proofs. Compared to the example of Section 1, although $(A \to B) \oplus E$ is green (if E is green), A cannot escape its scope unless B is also green: intuitionistic implication survives the mixture with classical logic. In proving a formula such as Peirce's: $((P \multimap E) \to P) \multimap P$, only E needs to be green whereas in linear logic, clearly more than one ? would be needed. In ACL, there is no restriction on the formula P: no ? is required for it to be contracted. Only the inner $P \multimap E$ becomes a classical implication: the others keep their strengths in the sense that the proof segment below *Unlock* stays non-classical, and *must* stay as such.

It should also not be assumed that the presence of a green e in the stoup cancels the meaning of all non-classical connectives and constants. For example, while $\neg\neg E \to E$ is provable, $\sim\sim E \to E$ ($(E \to 0) \to 0) \to E$ is still not provable. The constant 0, being red, cannot enable a contraction on E. Also, it does not hold that $(E_1 \,\&\, E_2) \multimap (E_1 \otimes E_2)$ even when E_1 and E_2 are both green ($A \otimes B$ is always red). It would be entirely incorrect to suppose that the entire subproof above a sequent with e in the stoup becomes classical. Once the green e vacates the stoup, by an *Unlock* for example, the classical mode is canceled. *The contracted formulas do not lose their non-classical strength.* Only in the "purely classical" fragment, where all atoms are green and 0 and $\otimes$ are not used, does LAC become classical logic.

4.1 Fragments of ACL

Coloring information is used in LAC to enforce soundness: classical versus non-classical soundness. Without the common restriction on $Unlock$, Pr and $\bot L$, it is easy to see that LAC degenerates into another proof system for classical logic, with some redundant symbols and rules.

We enumerate some important fragments of ACL, all determined by the subformula property. However, ACL is more than the sum of these fragments because connectives can mix without restriction.

- **Purely Negative ACL.** Restrict to &, $\to$, $\multimap$, $\bot$ and $\top$. The semantic interpretations of these connectives do not require the closure operator cl. This fragment is the core of ACL. It can be given a simpler, Kripke style semantics. It is already possible to have control operators in this fragment, without becoming entirely classical.

- **Affine-Linear and Classical Logic.** Do not use $\to$.

- **Classical Logic.** Color all atoms green and restrict to &, $\oplus$, $\to$, $\bot$ and $\top$. In this fragment, $\neg$ becomes an involutive negation. The color restriction in the $Unlock$, Pr and $\bot L$ rules becomes meaningless.

- **Intuitionistic Logic.** Color all atoms red and restrict to &, $\vee$, $\to$, 0. All formulas are red. $\top$ can be replaced by $0 \multimap 0$, or simply be considered red. All proofs, even partial proofs, are intuitionistic once useless *Lock*s are discarded.

4.2 Note on Intuitionistic Disjunction

The connective $\vee$ is included in ACL for the sake of intuitionistic completeness *without a classical collapse.* With $\oplus$, all the propositional intuitionistic axioms are provable *except* $(A \to C) \to (B \to C) \to (A \oplus B) \to C$ where C is red. Using a green C will mean a collapse of $\to$ into classical implication. However, including $\vee$ as a connective also has consequences. Note that the $\vee R$ rule folds in a weakening: elsewhere weakening can be pushed to the initial rules. $A \vee B$ is similar to $!A \oplus !B$ in linear logic, which requires an empty linear context. In the affine case, the context must be weakened away. It is also possible to simulate $!A$ as $A \vee 0$, and consequently, intuitionistic implication becomes equivalent to $(A \vee 0) \multimap B$. It holds that $(A \vee B) \multimap (A \oplus B)$. $A \vee \neg A$ is also provable. Other fragments of ACL become possible with this new connective, including our previous effort, polarized control logic (PCL), which does contain $\vee$ but not $\oplus$, $\multimap$ and $\otimes$. However, PCL models are equivalent to purely intuitionistic models where $I(W) = W$, thus the difference between $\oplus$ and $\vee$ disappears.

Our framework allows other connectives to be considered as well. In particular, we can define a "purely intuitionistic conjunction" with semantic interpretation $(A \wedge B)^p = cl(I(A^p)I(B^p))$, and the following introduction rules

$$\frac{\Gamma; \vdash A \qquad \Gamma; \vdash B}{\Gamma; \Delta \vdash A \wedge B} \wedge R \qquad \frac{A, B, \Gamma; \Delta \vdash C}{\Gamma; \Delta, A \wedge B \vdash C} \wedge L$$

Like $\vee$ and $\otimes$, the color of $\wedge$ is always red. In the purely non-linear fragment without $\otimes$ and $\multimap$, $\wedge$ has the same provability properties as & (and is therefore not required for intuitionistic completeness). In that case, the affine-linear context can be required to contain at most one formula, and therefore serves the purpose of a *left-side stoup* in proofs. The inclusion of both & and $\wedge$ may also be appropriate in a focused sequent calculus, since & is negative (synchronous on the left) and $\wedge$ is positive (see Section A).

5 Cut Elimination

The proof-theoretic novelty of a logical system depends on the details of its cut-elimination proof. A system that only repackages old ideas is likely to have a cut-elimination proof that surprises no one: there will be nothing not found in other such proofs. For example, the polarities of a focused proof system are significant in this respect because of their meaningful effect on cut-elimination: the subproof that contains the synchronous cut formula is "attractive." Thus the importance of our cut-elimination proof is far more than that cuts are admissible, for it establishes that ACL is truly a new logical system that requires its own, unique proof theory. We should also mention at this point, that our coloring scheme does not cause cut-elimination to become confluent as in a focused system, for in the worst case (with all green formulas), the system degenerates into classical LK. ACL is designed with properties other than focusing in mind. The problem of confluent cut reduction is orthogonal: it can still be achieved by adding focusing (i.e., positive versus negative polarization) and we describe this process in depth in Section A.

In order to consider cut-elimination carefully, let us regard *modus ponens* in the following forms.

$$\frac{\ldots \vdash A \otimes (A \multimap B)}{\ldots \vdash B} \qquad \frac{\ldots \vdash A \,\&\, (A \rightarrow B)}{\ldots \vdash B}$$

Other forms, that use other combinations of $\multimap$, $\rightarrow$, & and $\otimes$ are not generally valid, at least not without restrictions. This analysis implies that the cut rules relative to LAC should be in two forms:

$$\frac{\Gamma; \Delta_1 \vdash A \quad \Gamma; \Delta_2, A \vdash B}{\Gamma; \Delta_1\Delta_2 \vdash B}\ cut_1 \qquad \frac{\Gamma; \vdash A \quad A, \Gamma; \Delta \vdash B}{\Gamma; \Delta \vdash B}\ cut_2$$

Lessons from linear logic may suggest that cut_1 (and cut_2) cannot be admissible, for a contraction in the form of *Lock* is possible on the cut formula A in the left subproof. We do not have the ! operator and the Pr rule is not an exact analogue. This suggests that the non-contractable context Δ_2 in cut_1 (and Δ in cut_2) should be empty, lest we run into the following kind of scenario:

$$\frac{\dfrac{\vdash A, A}{\vdash A} \quad A, \Delta \vdash}{\Delta \vdash}\ cut \qquad \longrightarrow \qquad \frac{\dfrac{\vdash A, A \quad A, \Delta \vdash}{\Delta \vdash A}\ cut \quad A, \Delta \vdash}{\Delta\Delta \vdash}\ cut$$

In linear logic the situation is avoided because A must be $?A$, and (left-side) promotion of $?A$ necessitates a contractable context ($!\Delta$): thus the cut is permuted to the point of promotion in the right subproof. Here, however, we must permute the cut up to the points where the contractions are unlocked, in the *left* subproof. This is a type of what Parigot calls *structural reduction* as found in $\lambda\mu$-calculus: it enables the capture of continuations in a computational interpretation. The cut-elimination procedure for ACL is more delicate than that of linear logic because its structural rules are more delicate. The semantics clearly show that the cuts are sound as written. Although this is no guarantee that cut elimination will work, it gives us considerable confidence to proceed.

By the admissibility of both weakening and contraction in the Γ context, we could have written the rules multiplicatively with respect to Γ as well, but that would only confuse the issue here. However, we will not hesitate to split Γ into $\Gamma_1\Gamma_2$ whenever we find it to be more convenient.

Formulas $[\Theta : A]$ cannot be cut formulas because they cannot appear on the right-hand side. Formulas inside the [ ] are not subject to cut rules. See Corollary 5 for further clarification.

With a minor exception in the case of the $\bot L$ rule, the reduction of cut with respect to the introduction rules is standard: our introduction rules are no different from those found elsewhere, such as in the logic programming language Lolli [HM94]. The structural rules dominate the cut-elimination proof. The proof is by simultaneous induction on both cut_1 and cut_2. The inductive measure is the lexicographical ordering consisting of the size of the cut formula, followed by the number of *Lock* rules on the cut formula above the cut, followed by the number of Pr rules on the cut formula above the cut, followed by the number of Dr rules on the cut formula above the cut, followed by the heights of subproofs.

We detail the permutation of cut above *Lock*, *Unlock*, Pr and Dr below.

The case of Lock

This case of the cut-elimination procedure demonstrates the presence of *structural* as opposed to *logical* cuts. The scenario for permuting cut_2 above $Lock$ is the following

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma_1;\Delta_1 \vdash A}{[:A],\Gamma_1;\Delta_1 \vdash e_1}\,Unlock}{\vdots}}{[:A],\Gamma;\vdash A}}{\Gamma;\vdash A}\,Lock \qquad A,\Gamma';\Delta' \vdash B}{\Gamma\Gamma';\Delta' \vdash B}\,cut_2$$

The figure means to convey that there could be multiple $Unlocks$ above the left subproof, possibly stacked. Note that although the affine-linear context is initially empty in the left subproof (regarding the proof from the bottom), it may become non-empty when $Unlock$ is applied. This cut is reduced as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma_1;\Delta_1 \vdash A}{\Gamma_1\Delta_1;\vdash A}\,Dr^* \qquad A,\Gamma';\Delta' \vdash B}{\Gamma'\Gamma_1\Delta_1;\Delta' \vdash B}\,cut_2}{[\Delta':B],\Gamma'\Gamma_1\Delta_1;\vdash e_1}\,Unlock}{[\Delta':B],\Gamma'\Gamma_1;\Delta_1 \vdash e_1}\,Pr^*}{\vdots}}{[\Delta':B],\Gamma\Gamma';\vdash A} \qquad A,\Gamma';\Delta' \vdash B}{\dfrac{[\Delta':B],\Gamma\Gamma';\Delta' \vdash B}{\Gamma\Gamma';\Delta' \vdash B}\,Lock}\,cut_2$$

The illustration generalizes to cuts on multiple branches and to stacked cuts, in which case $[\Delta' : B]$ is not removed from Γ until the topmost occurrence of $Unlock$. Naturally we need not be concerned with duplication inside the intuitionistic (Γ) context: we chose a multiplicative presentation of these contexts here for clarity.

The key observation here is that the contractability of Δ_1 is not determined statically by the formulas, not by ! or ?, but dynamically, by the color of the right-side formulas. When $Unlock$ is available, so is Pr, which allows Δ_1 to be contracted.

The case for cut_1 is a simpler version of the cut_2 case because it does not require the Pr rule. Computationally, $Lock$ reflects the capturing of a continuation ($(A \to B)$ or $(A \twoheadrightarrow B)$) and applying it to multiple places in a term: this type of control operation is known to be able to lead to non-termination. The correct *strategy* is needed to guarantee termination.

For both cut_1 and cut_2, there is another case where the cut formula is inside the affine-linear multiset being locked. For example:

$$\dfrac{\Gamma;\Delta \vdash A \qquad \dfrac{\dfrac{\dfrac{\dfrac{\Gamma_1';\Delta_1'\Theta, A \vdash B}{[\Theta, A : B],\Gamma_1';\Delta_1' \vdash e_1}\,Unlock}{\vdots}}{[\Theta, A : B],\Gamma';\Delta'\Theta, A \vdash B}}{\Gamma';\Delta'\Theta, A \vdash B}\,Lock}{\Gamma\Gamma';\Delta\Delta'\Theta \vdash B}\,cut_1$$

Such a case reduces to

$$
\dfrac{\dfrac{\Gamma;\Delta\vdash A \qquad \dfrac{\dfrac{\vdots}{\dfrac{\dfrac{\dfrac{\Gamma;\Delta\vdash A \qquad \Gamma'_1;\Delta'_1\Theta,A\vdash B}{\Gamma\Gamma'_1;\Delta\Delta'_1\Theta\vdash B}\,cut_1}{[\Delta\Theta:B],\Gamma\Gamma'_1;\Delta'_1\vdash e_1}\,Unlock}{}}}{[\Delta\Theta:B],\Gamma\Gamma';\Delta'\Theta,A\vdash B}}{[\Delta\Theta:B],\Gamma\Gamma';\Delta\Delta'\Theta\vdash B}\,cut_1}{\Gamma\Gamma';\Delta\Delta'\Theta\vdash B}\,Lock
$$

So we have merely replaced locking A with locking Δ.

The case of Dr

The potential problem here is the effect of Dr on the applicability of cut_2, which requires an empty affine-linear context:

$$
\dfrac{\dfrac{\Gamma;B\vdash A}{B,\Gamma;\vdash A}\,Dr \qquad \dfrac{\dfrac{\dfrac{\Gamma_1;\Delta_1,A\vdash C_1}{A,\Gamma_1;\Delta_1\vdash C_1}\,Dr}{\vdots}}{A,\Gamma';\Delta\vdash C}}{B,\Gamma\Gamma';\Delta\vdash C}\,cut_2
$$

The notation $\vdots$ represents multiple instances of Dr on the cut formula A in the right subproof. A can be ignored above the topmost Dr (weakened). This cut is reduced as follows:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\Gamma;B\vdash A \quad \Gamma_1;\Delta_1,A\vdash C_1}{\Gamma\Gamma_1;\Delta_1,B\vdash C_1}\,cut_1}{B,\Gamma\Gamma_1;\Delta_1\vdash C_1}\,Dr}{\vdots}}{B,\Gamma\Gamma';\Delta\vdash C}
$$

The permutation of Dr above cut_1 is relatively trivial. Since Dr corresponds to dereliction in linear logic, this cut-permutation technique is also found in several other cut-elimination proofs. But the cases of *Lock*, *Pr* and *Unlock* are unique to ACL.

The case of Unlock

This case also appears in PCL, but was argued for a multiple-conclusion proof system. It concerns permuting cut above *Unlock* (and to a lesser extent $\bot E$). We have restricted e in *Unlock* and $\bot E$ to be a green atom or $\bot$. In fact this restriction can be relaxed to allow any formula E: one can check that this relaxation preserves semantic soundness since green formulas are characterized by exactly the same semantic properties as green atoms. The restriction was used for two reasons. First, technically speaking, "sequent calculus" should not look beyond the top-level form of a formula to determine which inference rule applies. The second reason is that it allows us to write a simpler cut elimination proof.

Let us temporarily refer to the relaxed version of *Unlock* as *Unlock'*. A crucial case of cut-elimination for LAC is the following:

$$
\dfrac{\dfrac{\Gamma;\Delta_1\Theta\vdash A}{[\Theta:A],\Gamma;\Delta_1\vdash E}\,Unlock' \qquad \Gamma;\Delta_2,E\vdash B}{[\Theta:A],\Gamma;\Delta_1\Delta_2\vdash B}\,cut_1
$$

A similar case occurs with the $\bot L$ rule and with cut_2.

Cut-elimination in sequent calculus usually calls for the cut to be permuted parametrically above each inference rule to reach a "key case" where the cut formula in both sequents are the principal formulas of introduction rules. That strategy clearly would not work here. However, if E was a green atom, then the cut can only be permuted parametrically above the right-side subproof (the *"attractive"* subproof) until an Id rule is reached, at which point the cut is eliminated by substitution as in natural deduction. If the cut formula e is $\bot$, then we similarly permute the cut upwards until the right-proof branch reaches $\bot L$, at which point the right-side formula in the conclusion of $\bot L$ must be green (some $\Gamma; \Delta_2', \bot \vdash e'$), which means the cut can then be replaced by:

$$\dfrac{\dfrac{\Gamma; \Delta_1\Theta \vdash A}{\Gamma; \Delta_1\Delta_2'\Theta \vdash A}\ (\mathit{weakening})}{[\Theta : A], \Gamma; \Delta_1\Delta_2' \vdash e'}\ \mathit{Unlock}$$

We can also generalize cut-elimination to allow the unrestricted versions of Unlock (and $\bot E$ and Pr) by showing that their uses can be permuted to atomic cases:

- *In a cut-free proof, Unlock' can be replaced by Unlock.*

This is proved by showing that an Unlock' rule on a green formula E can always be permuted to Unlock' on its subformulas. We show the most interest case of the transformation

$$\dfrac{\Gamma; \Delta\Theta \vdash A}{[\Theta : A], \Gamma; \Delta \vdash B \mathrel{-\!\!\triangleright} E}\ \mathit{Unlock}' \quad \longrightarrow \quad \dfrac{\dfrac{\dfrac{\Gamma; \Delta\Theta \vdash A}{\Gamma; \Delta\Theta, B \vdash A}\ (\mathit{weakening})}{[\Theta : A], \Gamma; \Delta, B \vdash E}\ \mathit{Unlock}'}{[\Theta : A], \Gamma; \Delta \vdash B \mathrel{-\!\!\triangleright} E}\ \mathrel{-\!\!\triangleright} r$$

The admissibility of weakening in Δ is thus crucial for the cut-elimination argument.

Results analogous to the equivalence between Unlock and Unlock' also hold for $\bot L$ and for the Pr rule (Lemma 2 below).

The case of Pr

The methods used in the previous cases are combined in the case of permuting cut above the Pr rule. There are two principal scenarios to consider:

$$\dfrac{\dfrac{A, \Gamma; \Delta \vdash e}{\Gamma; \Delta, A \vdash e}\ Pr \qquad \Gamma; \Delta', e \vdash R}{\Gamma; \Delta\Delta' \vdash R}\ \mathit{cut}_1 \qquad \text{and} \qquad \dfrac{\Gamma; \Delta' \vdash A \qquad \dfrac{\dfrac{\vdots}{\dfrac{\dfrac{\Gamma_1; \Delta_1, A \vdash D}{A, \Gamma_1; \Delta_1 \vdash D}\ Dr}{} \atop A, \Gamma; \Delta \vdash e}}{\Gamma; \Delta, A \vdash e}\ Pr}{\Gamma; \Delta\Delta' \vdash e}\ \mathit{cut}_1$$

The first (left) case is solved by restricting e to be a green atom or $\bot$, just as in the case for Unlock, then showing that the restriction can be relaxed to any green formula using a separate set of permutations. One example should suffice to convince:

$$\dfrac{A, \Gamma; \Delta \vdash B \oplus E}{\Gamma; \Delta, A \vdash B \oplus E}\ Pr \quad \longrightarrow \quad \dfrac{\dfrac{\dfrac{\dfrac{A, \Gamma; \Delta \vdash B \oplus E}{[: B \oplus E], A, \Gamma; \Delta \vdash E}\ \mathit{Unlock}}{[: B \oplus E], \Gamma; \Delta, A \vdash E}\ Pr}{[: B \oplus E], \Gamma; \Delta, A \vdash B \oplus E}\ \oplus R}{\Gamma; \Delta, A \vdash B \oplus E}\ \mathit{Lock}$$

A green (classical) disjunction is only "additive" in a superficial sense. This treatment of $B \oplus E$ in fact suggests the following, alternative introduction rule for green disjunctions:

$$\dfrac{[: B], \Gamma; \Delta \vdash E}{\Gamma; \Delta \vdash B \oplus E}\ \oplus ER$$

The rule combines a *Lock* with a $\oplus$-introduction that selects E. It also anticipates another $\oplus$-introduction after *Unlock*, this time selecting B. The rule is sound and complete as it can emulate the original introduction (with weakening). However, this rule is evidently multiplicative. Since all other green formulas (safe for atoms) are already *asynchronous,* this rule suggests a simple normal form for LAC proofs: *all green formulas on the right can be decomposed immediately down to atoms, or to* $\bot$.

The argument is similar in the cases of &, $\rightarrow$ and $\multimap$ (weakening is required in the cases of $\rightarrow$ and $\multimap$). However, had we naively designated $B \otimes C$ to be green if B "and" C are green, then such a permutation cannot be made (the context is split below Pr). Semantically, $B \otimes C$ is always red because we cannot guarantee that it will be valid above the root even when B and C are both green. This rather abstract semantic explanation is represented syntactically in the non-permutability of $\otimes$ above Pr (and *Unlock*).

The fact that the restriction on *Unlock*, Pr and $\bot L$ can be all be relaxed is an important property of ACL, since the relaxed forms will be used in the completeness proof and in the natural deduction system.

Lemma 2 *The restriction to e being a green atom or* $\bot$ *in the Unlock,* Pr *and* $\bot L$ *rules can be relaxed to allow any green formula* E.

The other case of permuting cut with respect to Pr is relatively simple since the Pr rule can be duplicated beneath to remove the additional copies of Δ'. That is, we permute the cut above instances of Dr on A: if there are no such instances then the result follows from weakening. Otherwise we have:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma;\Delta' \vdash A \quad \Gamma_1;\Delta_1, A \vdash D}{\Gamma\Gamma_1;\Delta_1\Delta' \vdash D}\ cut_1}{\Delta'\Gamma\Gamma_1;\Delta_1 \vdash D}\ Dr^*}{\vdots}}{\Delta'\Gamma;\Delta \vdash e}}{\Gamma;\Delta\Delta' \vdash e}\ Pr^*$$

The arguments for cut_2 are the same as for cut_1.

Theorem 3 cut_1 *and* cut_2 *are admissible in LAC.*

5.1 Results Related to Cut Elimination

Another result, relatively easy to prove but which is crucial for completeness, is *initial elimination.*

Theorem 4 $;A \vdash A$ *is provable for any formula* A

The *Unlock* rule is a special case of $\multimap L$, but we wish to keep the effect integral (a *focused* effect):

Corollary 5 $[\Theta : A], \Gamma; \Delta \vdash B$ *is provable if and only if* $-(\Theta^{\otimes} \multimap A), \Gamma; \Delta \vdash B$ *is provable*

The forward direction (soundness of focusing) follows because *Unlock* can be emulated with $\rightarrow L$, $\bot L$, $\multimap R$ and $\otimes L$. The reverse direction (completeness of focusing) follows from cut elimination and initial elimination because we can show that

$$[\Theta : A]; \vdash -(\Theta^{\otimes} \multimap A)$$

is provable. This corollary is also critical for completeness.

The most important use of focusing in the *Unlock* rule is that the right premise of the implicit $\rightarrow L$ must be the conclusion of an initial rule ($\bot L$). This represents a non-trivial use of focusing.

Another relatively obvious but important result is the following:

Proposition 6 *If a formula is provable with an atom b colored red, then the same formula is provable with b colored green.*

This holds because a green atom can only lead to more proofs. The consequences of this lemma are significant. Combined with cut elimination and initial elimination, it allows us to show that LAC has the *substitution property.*

Theorem 7 *The substitution property for LAC holds as follows:*

1. *If a formula A is provable with an atom b colored red, then $A[C/b]$ is also provable for any formula C.*
2. *If a formula A is provable with an atom e colored green, then $A[E/e]$ is also provable for any green formula E.*

Part 2 of this theorem follows from Lemma 2.

The other important consequence of Theorem 7 is that, were we to extend ACL to include *second order* propositional quantifiers, then the colors of bound variables are not in question: *universally quantified propositional variables are red, while existentially quantified ones are green.* See Section 9 for further discussion.

We observe that, since ACL is intended as a unified logic, in the worst case, cut-elimination could become as uncontrolled as in classical LK: in particular when all atoms are green. However, such uncontrolled segments are localized in proofs. Cut elimination involving the *Unlock* and the *Pr* rules are all by substitution as in natural deduction, without duplicating structure in the subproofs that end in these rules.

6 Soundness and Completeness

The soundness of LAC inference rules is argued by induction on the structure of proofs. In particular, *Lock* is sound by the validity of the version of Peirce's formula $(-P \to P) \twoheadrightarrow P$, and Pr is sound because of its counterpart $(P \to -P) \twoheadrightarrow -P$. The *Unlock* rule is just a synchronized instance of $\to L$. The other rules can be checked to be sound case by case. For example, the soundness of the $\oplus L$ rule holds since if $A^p \subseteq C^p$ and $B^p \subseteq C^p$ then $A^p \cup B^p \subseteq C^p$ and thus $cl(A^p \cup B^p) \subseteq cl(C^p) = C^p$.

The completeness proof of ACL differs from other phase semantic completeness proofs principally in the following ways. Because of the meaning and central role of $\perp$, the unit/root of the monoid that we build is not the empty set or multiset. Instead, it is a maximally consistent set with the characteristics of Hintikka sets. Also, instead of constructing a canonical model of all proofs, we build a *countermodel* for a formula that's assumed to be unprovable. In addition, our completeness proof differs from others in that it *requires* cut-elimination, for otherwise there is no mention of the *Lock* rule in the proof.

Assuming that a formula A is not provable, we show the existence of a countermodel as follows. A set or multiset Θ is said to be *consistent* with respect to a formula P if P is not derivable from it. The root world of the model will be a set that's maximally consistent with respect to A and to $\perp$. In the following we write $\Gamma; \Delta \nvdash A$ to mean $\Gamma; \Delta \vdash A$ *is not provable.*

Lemma 8 *If* $;\nvdash A$ *then* $;\nvdash A \oplus \perp$.

This is a non-trivial lemma since A may be red. First, it is clear that $\perp$ has no cut-free proof. Then we show the contrapositive of the lemma. The essential argument is that, first, we show if $A \oplus \perp$ is provable then it should follow from $;\vdash A$ or from $[: A \oplus \perp]; \vdash A$. In the second case, we show that a proof of $[: A]; \vdash A$ can also be constructed: when $A \oplus \perp$ is unlocked there must be a green subformula e of A in the stoup, which means that if $\perp$ becomes derivable from the left-hand side at this point, then so is e (by cut with $;\perp \vdash e$). We can therefore continue to emulate the proof of $A \oplus \perp$ to construct a proof of A. Since $A \oplus \perp$ is provably equivalent to $- - A$, this lemma shows that *if* $- - A$ *is provable then* A *is also provable.* This admissibility result is also easily verified semantically.

Lemma 9 *If* $B \oplus C, \Gamma; \nvdash A \oplus \perp$ *then either* $B, \Gamma; \nvdash A \oplus \perp$ *or* $C, \Gamma; \nvdash A \oplus \perp$

This lemma holds because $A \oplus \bot$ is green. By Lemma 2 we can assume that the relaxed form of PR can be used in proofs. If $B \oplus C, \Gamma; \not\vdash A \oplus \bot$ then $\Gamma; B \oplus C \not\vdash A \oplus \bot$ by the DR rule (arguing the contrapositive). But then by the $\oplus L$ rule either $\Gamma; B \not\vdash A \oplus \bot$ or $\Gamma; C \not\vdash A \oplus \bot$. Thus by the (relaxed) PR rule either $B, \Gamma; \not\vdash A \oplus \bot$ or $C, \Gamma; \not\vdash A \oplus \bot$. This is the only place where the Pr rule is needed in the completeness proof.

Define a *proxy subformula* B of a formula P to be either a subformula of P or a formula $\Delta^{\otimes} \multimap B$ where B and every $D \in \Delta$ are subformulas of P. The *Lock* rule is implicitly applied to proxy subformulas.

For the purpose of the completeness proof, we extend the notion of the provability of $\Gamma; \Delta \vdash B$ to allow Γ to be an infinite set. Such a sequent is provable if $\Gamma'; \Delta \vdash B$ is provable for some finite subset Γ' of Γ.

Now we construct a countermodel CA as follows:

1. A possible world in W consists of a set Γ and a multiset Δ of formulas that we simply write as $\Gamma\Delta$. Let Γ^{∞} represent a *multiset* such that, for each distinct formula A in Γ, there are countably infinite many occurrences of A in Γ^{∞} (and nothing else). This device type casts a set into a multiset and simplifies some arguments. Δ will always be a finite multiset so if A occurs in both Δ and Γ, then it is absorbed in $\Gamma^{\infty}\Delta$. The partial ordering is defined as $\Gamma\Delta \preceq \Gamma'\Delta'$ iff $\Gamma^{\infty}\Delta \subseteq \Gamma'^{\infty}\Delta'$ where $\subseteq$ here is the multi-subset relation. The monoid operation is defined to be $\Gamma\Delta \cdot \Gamma'\Delta' = \Gamma\Gamma'\Delta\Delta'$;

2. Construct the root world $r = \Gamma_r$ as follows. Enumerate all proxy subformulas B of A and their negations $-B$. Then construct Γ_r to be a maximally consistent set with respect to $A \oplus \bot$ by inserting each B or $-B$ into Γ_r as long as Γ_r remains $A \oplus \bot$-consistent (by "inserting" we of course mean a hypothetical construction to show that such a saturation exists). By Corollary 5, inserting $-(\Delta^{\otimes} \multimap C)$ is equivalent to inserting $[\Delta : C]$. Two other properties are assured:

   (a) It cannot be the case that B and $-B$ are both in Γ_r as that would mean that $\bot$ and thus $A \oplus \bot$ are derivable from Γ_r. Since Γ_r is $\bot$-consistent, it must also be 0-consistent.

   (b) If $\Gamma; \not\vdash A \oplus \bot$, then $B \oplus -B, \Gamma; \not\vdash A \oplus \bot$ because $; \vdash B \oplus -B$ is provable and cut is admissible. By Lemma 9, this means that in a maximally consistent saturation *exactly one of either B or $-B$ will be inserted into* Γ_r. With Γ_r thus saturated, it follows that any proper addition to Γ_r (limited to the proxy subformulas of A and their negations) will render it $\bot$-inconsistent. In other words, either $\Gamma_r C = \Gamma_r$ or $\Gamma_r; C \vdash \bot$ becomes provable. This is the most critical use of cut elimination in the completeness proof. It confirms that the *Lock* rule, which is required to prove $B \oplus -B$ but is otherwise not directly referred to in this proof, is required for completeness.

3. The worlds W consist of all pairs $\Gamma\Delta$ of proxy subformulas and their negations such that $\Gamma_r \subseteq \Gamma$. *Furthermore, we can assume that the number of formulas in $\Gamma \backslash \Gamma_r$ is finite.* This assumption is important.

   It is easily verified that Γ_r satisfy the requirements of being the root. $I(W)$ corresponds to those worlds where the proper multiset Δ is empty.

   The rest of the proof mostly emulates Okada.

4. For any formula A, let $Pr(A) = \{\Gamma\Delta : \ \Gamma; \Delta \vdash A \textit{ is provable}\}$. By the admissibility of weakening, $Pr(A)$ is upwardly closed. The set of facts D of the model are restricted to be those subsets of W that are equivalent to $\bigcap Pr(A_i)$ where A_i ranges over an arbitrary collection of formulas $A_0, \ldots A_i, \ldots$. Clearly we have $\top^p \in D$ since $Pr(\top) = W$ and $\bot^p \in D$ since $Pr(\bot) = W \backslash \{r\}$. D is certainly closed under the *cl* operator as defined. Furthermore, if $B \in D$ then $\{x : \textit{for all } y \in A,\ xy \in B\} \in D$. Assume that $B = \bigcap Pr(C_i)$ and $\Gamma\Delta$ is in this set. Then for any $\Gamma_r\Gamma'\Delta' \in A$, we have that $\Gamma\Gamma'; \Delta\Delta' \vdash C_i$ is provable for all C_i. Since we can assume that Γ' and Δ' are finite sets and multisets, this means that $\Gamma; \Delta \vdash \Gamma'^{\&} \to \Delta'^{\otimes} \multimap C_i$ is provable. Thus $\Gamma\Delta \in \bigcap Pr(\Gamma'^{\&} \to \Delta'^{\otimes} \multimap C_i)$ for all C_i, therefore qualifying as a fact. Thus all the conditions required of facts are satisfied.

5. The valuation of atomic formulas is defined to be

$$a^p = Pr(a) = \{\Gamma\Delta : \ \Gamma; \Delta \vdash a \textit{ is provable}\}$$

Naturally, green atoms are mapped to $\bot^p$ or $\top^p$ since all $\Gamma\Delta$ above Γ_r derives $\bot$ and therefore all green formulas (by cut). The fact $0^p = \bigcap D$ is $Pr(0) = \{\Gamma\Delta : \Gamma;\Delta \vdash 0 \textit{ is provable}\}$. Clearly this is the smallest fact since (by cut) $Pr(0) \subseteq Pr(B)$ for all formulas B. 0^p is not empty if 0 is a subformula of the formula that's assumed to be unprovable.

6. We can show that $B^p = Pr(B)$ for all formulas B. However, for completeness it is only necessary to show that $\Gamma_r B \in B^p$ and $B^p \subseteq Pr(B)$. This is proved by mutual induction on the structure of B. The cases for atoms and constants are trivial. We show a selection of representative cases for the connectives.

   For $\Gamma_r A \otimes B \in (A \otimes B)^p = cl(A^p B^p)$: by inductive hypothesis $\Gamma_r A \in A^p$, $\Gamma_r B \in B^p$ thus $\Gamma_r AB \in A^p B^p \subseteq cl(A^p B^p) = \bigcap\{F \in D : A^p B^p \subseteq F\}$. But each F in $cl(A^p B^p)$ is of the form $\bigcap Pr(C_i)$ for some collection of formulas C_i. If $\Gamma_r AB \in \bigcap Pr(C_i)$ then $\Gamma_r; A, B \vdash C_i$ is provable and by $\otimes L$ so is $\Gamma_r; A \otimes B \vdash C_i$. Thus $\Gamma_r A \otimes B$ is in each such $Pr(C_i)$ and therefore in $cl(A^p B^p)$.

   Notice here we can apply cut elimination to show that in fact $Pr(A \otimes B) \subseteq (A \otimes B)^p$, but this is not necessary.

   For $(A \otimes B)^p = cl(A^p B^p) \subseteq Pr(A \otimes B)$: by inductive hypothesis $A^p \subseteq Pr(A)$, $B^p \subseteq Pr(B)$ so $A^p B^p \subseteq Pr(A)Pr(B)$. By the $\otimes R$ rule, $Pr(A)Pr(B) \subseteq Pr(A \otimes B)$. Since $(A \otimes B)^p$ is the intersection of all facts that contain $A^p B^p$ and $Pr(A \otimes B)$ is a fact, it holds that $(A \otimes B)^p \subseteq Pr(A \otimes B)$.

   For $\Gamma_r A \vee B \in (A \vee B)^P = cl(I(A^p) \cup I(B^p)$: by inductive hypothesis $\Gamma_r, A \in A^p$, $\Gamma_r, B \in B^p$ but since A^p, B^p are facts of the form $PR(C_i)$, $\Gamma_r A^\infty \in I(A^p)$ and $\Gamma_r B^\infty \in I(B^p)$ (by the DR rule). So $\Gamma_r A^\infty \in cl(I(A^p) \cup I(B^p))$ and likewise for $\Gamma_r B^\infty$. But $cl(I(A^p) \cup I(B^p))$ is a fact, which is some $PR(D_i)$. Thus by the $\vee L$ rule, $\Gamma_r, A \vee B; \vdash D_i$ also holds, so $\Gamma_r A \vee B \in cl(I(A^p) \cup I(B^p)$.

   For $(A \vee B)^p = cl(I(A^p) \cup I(B^p)) \subset PR(A \vee B)$: inductive hypotheses give that $A^p \subset PR(A)$ and $B^p \subset PR(B)$. Thus $I(A^p) \subset PR(A)$ and $I(B^p) \subset PR(B)$. This means that if $\Gamma \in I(A^p)$ then $\Gamma; \vdash A$ is provable and likewise if $\Gamma \in I(B^p)$. By the $\vee R$ rule, $PR(A \vee B)$ contains $I(A^p) \cup I(B^p)$. Now $cl(I(A^p) \cup I(B^p))$ is the intersection of all facts that contains $I(A^p) \cup I(B^p)$ and $PR(A \vee B)$ is a fact, thus if $\Gamma\Delta \in cl(I(A^p) \cup I(B^p))$ then $\Gamma\Delta \in PR(A \vee B)$. A subtlety here is that nothing is assumed for Δ: it does not have to be empty in the closure. The argument will fail if weakening is not embedded into the $\vee R$ rule.

   Mutual induction is required in the case of implication. For $\Gamma_r A \multimap B \in (A \multimap B)^p$ we need to show that if $\Gamma\Delta \in A^p$ then $\Gamma\Delta, A \multimap B \in B^p$. By inductive hypothesis $A^p \subseteq Pr(A)$ and $\Gamma_r B \in B^p = \bigcap Pr(C_i)$ where C_i ranges over a collection of formulas. Thus if $\Gamma\Delta \in A^p$ then $\Gamma; \Delta \vdash A$ is provable and $\Gamma_r; B \vdash C_i$ is provable. By the $\multimap L$ rule this means that $\Gamma; \Delta, A \multimap B \vdash C_i$ is provable and so $\Gamma\Delta, A \multimap B \in B^p$ as well.

   For $(A \multimap B)^p \subseteq Pr(A \multimap B)$, by inductive hypothesis $\Gamma_r A \in A^p$ and $B^p \subseteq Pr(B)$. Thus if $\Gamma\Delta \in (A \multimap B)^p$ then $\Gamma\Delta, A \in B^p \subseteq Pr(B)$. So by the $\multimap R$ rule we have that $\Gamma\Delta \in Pr(A \multimap B)$.

   For the implication $\rightarrow$, if $A \in A^p$ then $A^\infty \in A^p \cap I(W)$ (by the DR rule), and if $\Gamma\Delta \in A^p \cap I(W)$ then Δ must be empty. With these observations similar arguments as for $\multimap$ can then be applied.

7. Completeness then follows since $\Gamma_r; \nvdash A$ and thus $\Gamma_r \notin Pr(A)$, so $\Gamma_r \notin A^p$. That is, the unit/root of the monoid is not found inside A^p (in terms of Kripke models, $\Gamma_r \not\models A$).

Theorem 10 *A formula is provable in LAC if and only if it is valid in ACL.*

One might expect the completeness proof to shed light on the question of decidability for ACL. Propositional affine linear logic is decidable [Kop95], and in [Laf97] Lafont gave a phase model proof. The models of ACL are consistent with those of affine linear logic (facts are ideals). Although the model CA of the completeness proof is infinite, it is possible to construct a quotient model by defining a congruence relation. However, we are not able to duplicate Lafont's arguments further because the quotient model is not finitely

$$\frac{s : [B_1^{y_1} \ldots B_n^{y_n} : A]_e^d, \Gamma; B_1^{x_1} \ldots B_n^{x_n}, \Delta \vdash A}{\gamma d.s\{x_1/y_1 \ldots x_n/y_n\} : \Gamma; B_1^{x_1} \ldots B_n^{x_n}, \Delta \vdash A} \; Lock_e \qquad \frac{t : \Gamma; \Delta\Theta \vdash A}{[d]t : [\Theta : A]_e^d, \Gamma; \Delta \vdash e} \; Unlock_e$$

$$\frac{t : \Gamma; \Delta, A^y \vdash B}{t\{x/y\} : A^x, \Gamma; \Delta; \vdash B} \; Dr \qquad \frac{t : \Gamma\Delta; \vdash e}{!t : \Gamma; \Delta \vdash e} \; Pr$$

$$\frac{s : \Gamma; \Delta, A^x \vdash B}{\lambda x.s : \Gamma; \Delta \vdash A \multimap B} \multimap I \qquad \frac{s : \Gamma; \Delta_1 \vdash A \multimap B \quad t : \Gamma; \Delta_2 \vdash A}{(s\ t) : \Gamma; \Delta_1\Delta_2 \vdash B} \multimap E$$

$$\frac{s : A^x, \Gamma; \Delta \vdash B}{\lambda^! x.s : \Gamma; \Delta \vdash A \to B} \to I \qquad \frac{s : \Gamma; \Delta \vdash A \to B \quad t : \Gamma; \vdash A}{(s\_t) : \Gamma; \Delta \vdash B} \to E$$

$$\frac{s : \Gamma; \Delta_1 \vdash A \quad t : \Gamma; \Delta_2 \vdash B}{(s; t) : \Gamma; \Delta_1\Delta_2 \vdash A \otimes B} \otimes I \qquad \frac{s : \Gamma; \Delta_1 \vdash A \otimes B \quad t : \Gamma; \Delta_2, A^x, B^y \vdash C}{let\ (x; y) = s\ in\ t : \Gamma; \Delta_1\Delta_2 \vdash C} \otimes E$$

$$\frac{t : \Gamma; \Delta \vdash \bot}{\mathcal{B}(t) : \Gamma; \Delta \vdash e} \bot E\ (break) \quad \frac{t : \Gamma; \Delta \vdash 0}{\mathcal{A}(t) : \Gamma; \Delta \vdash A} 0E\ (abort) \quad \frac{}{. : \Gamma; \Delta \vdash \top} \top I \quad \frac{}{x : \Gamma; \Delta, A^x \vdash A} Id$$

Figure 2: The Natural Deduction System NAC. e must be a green atom or $\bot$ in $Lock_e$, $Unlock_e$, Pr and $\bot E$

generated. This is because there can be infinitely many *proxy* subformulas of a formula. To prove decidability along these lines we would need to show that the number of possible formulas subject to *Lock* in a proof is finite. It would be somewhat of a surprise, however, if ACL is not decidable since every model of ACL is also a model of affine linear logic. For the time being, we leave the question of propositional decidability to future work.

7 Natural Deduction and Computation

This section demonstrates the computational significance of ACL by defining a natural deduction system NAC (Figure 2) with proof terms. Although terms can also be associated with sequent calculus, we choose natural deduction for its simpler syntax, without the need for explicit substitutions. We restrict to the connectives $\to$, $\multimap$, $\otimes$ and the two forms of *false*. We prefer to associate a proof term with an entire subproof, as in [Par92], and not just the stoup formula.

We have modified the *Lock* rule based on the more general Peirce's formula

$$((P \multimap e) \to P) \multimap P$$

so e is not necessarily $\bot$. However, we still require that it's atomic. Although it would be valid to allow any green formula (by Lemma 2), retaining this normal form significantly simplifies term structure and reduction rules. Here, $[\Theta : A]_e$ has the meaning of the formula $(\Theta^{\otimes} \multimap A) \multimap e$. The new $Lock_e$ rule only superficially violates the subformula property. It is useless without $Unlock_e$, which can only be applied if e is a subformula of the end sequent. Instances of $Lock_e$ where e is not such a subformula can be discarded. A formula locked using $Lock_\bot$ can be unlocked by any green e by the validity of $\bot \multimap e$. The original, un-subscripted versions of $Lock/Unlock$ are still valid and are equivalent to $Lock_\bot/Unlock_\bot$. The generalized rules are not required for completeness but are more useful in that they allow us to use the green formulas more meaningfully as types.

All formulas except the stoup are indexed. The notation $[\Theta : A]_e^d$ indicates that the index variable d is associated with the entire intended formula $(\Theta^{\otimes} \multimap A) \multimap e$. Such boxed formulas are indexed by γ-

$$
\cfrac{x : [:P]^d_e; ((P \multimap e) \to P)^x \vdash (P \multimap e) \to P \qquad \cfrac{\cfrac{\cfrac{}{y :\ ; P^y \vdash P}\ Id}{[d]y : [:P]^d_e; P^y \vdash e}\ Unlock_e}{\lambda y.[d]y : [:P]^d_e; \vdash P \multimap e}\ \multimap I}{\cfrac{\cfrac{(x\_\lambda y.[d]y) : [:P]^d_e; ((P \multimap e) \to P)^x \vdash P}{\gamma d.(x\_\lambda y.[d]y) :\ ; ((P \multimap e) \to P)^x \vdash P}\ Lock_e}{\mathcal{K} = \lambda x.\gamma d.(x\_\lambda y.[d]y) :\ ; \vdash ((P \multimap e) \to P) \multimap P}\ \multimap I}\ \to E
$$

$$
\cfrac{x :\ ; \frown\frown A^x \vdash \frown\frown A \qquad \cfrac{\cfrac{z :\ ; \frown e^z \vdash \frown e \qquad \cfrac{y : A^y \vdash A}{[d]y : [:A]^d_e; A^y \vdash e}\ Unlock_e}{z\ [d]y :\ [:A]^d_e; \frown e^z, A^y \vdash 0}\ \multimap E}{\lambda y.(z\ [d]y) :\ [:A]^d_e; \frown e^z \vdash \frown A}\ \multimap I}{\cfrac{\cfrac{\cfrac{(x\ \lambda y.(z\ [d]y)) :\ [:A]^d_e; \frown e^z, \frown\frown A^x \vdash 0}{\mathcal{A}(x\ \lambda y.(z\ [d]y)) :\ [:A]^d_e; \frown e^z, \frown\frown A^x \vdash A}\ 0E}{\gamma d.\mathcal{A}(x\ \lambda y.(z\ [d]y)) :\ \frown e^z, \frown\frown A^x \vdash A}\ Lock_e}{\mathcal{C}_1 = \lambda z \lambda x.\gamma d.\mathcal{A}(x\ \lambda y.(z\ [d]y)) :\ ; \vdash \frown e \multimap (\frown\frown A \multimap A)}\ \multimap I^*}\ \multimap E
$$

Figure 3: Sample proofs: Peirce's Formula and Use-Once Control Operator

variables while others are indexed by λ-*variables.* We assume that variables are always distinguishable and are renamed to avoid clash when necessary. In particular, renamings are used in the $Lock_e$ and $Unlock_e$ rules (notation $\{x/y\}$ represents substitution). In practice, we can also consider a version of *Lock* that always copies the entire affine context, which would remain complete by the admissibility of weakening. However, adopting such a rule would make some of the subsequent examples syntactically clumsy, and thus we allow *Lock* to be more selective.

There are two types of lambda abstraction: λ and $\lambda^!$, that correspond to $\multimap$ and $\to$ respectively. There are also two types of application: $(s\ t)$ and (s_t): these correspond, respectively, to $A \otimes (A \multimap B) \multimap B$ and $A \to (A \to B) \multimap B$, the two forms of *Modus Ponens* that are possible. It is not valid to mix $\otimes$ with $\to$ and still deduce B without severe restrictions. One potential problem with linear lambda terms is how to type terms such as $\lambda x.((\lambda f.\lambda y.f\ (f\ y))\ x)$: here, x appears once before reduction but twice afterwards. In (intuitionistic) linear logic there is only one $\multimap$ and a ! operator that can be placed anywhere, offering few invariants. The solution to this problem in our unified logic is rather obvious: the term $\lambda x.((\lambda^! f.\lambda y.f\ \ (f\ \ y))\_x)$ cannot be assigned a *red* type, because of the context restriction on $\to$ elimination. The term $\lambda x.((\lambda^! f.\lambda y.f\ \ (f\ \ y))\ x)$ is not typable at all.

The rule *Dr* carry no meaning except for variable renaming: this is the only computational content of left-side contraction. The rule *Pr*, however, is more significant as it can affect the permutation of cuts (see next section). We have modified the *Pr* rule so that it affects all formulas in the affine linear context. All free variables inside the scope of ! may appear more than once. However, this does not mean a complete classical collapse, for red subformulas of green formulas will retain their non-classical strength: in $!\lambda x.t$, x can still appear only once in t unless it is inside the scope of another ! in t. The proof of $(P \multimap -P) \multimap -P$, for example, is $\lambda x \lambda y.!xyy$. We can preserve the original version of *Pr* using terms such as $!x.t$, to indicate the singleton formula that *Pr* affects. However, x must still be considered free in $!x.t$, which will have an effect on substitutions (beta-reduction) that's disproportionate to its usefulness. Thus we chose the simpler representation. ! is not affected by substitution. Note that terms such as $!x.\lambda y.s$ are not possible because of the continued restriction on e being atomic in the *Pr* rule. This normal form allows us to avoid having to modify the definition of β-reduction.

Terms $\gamma d.s$ represent contraction and are equivalent to $\mu d.[d]s$ in classical $\lambda\mu$ calculus. The fact that γ locks in the sequence $\mu d.[d]\ldots$ doesn't mean that we cannot derive the more general $\mathcal{C}$ control operator

[FFKD87] (compared to *call/cc*). We can prove the purely classical $\neg\neg E \rightarrow E$, or the hybrid $((A \rightarrowtriangle \bot) \rightarrow 0) \rightarrowtriangle A$. Given that there are two implications, two constants for false, and two colors, there are 64 versions of the double negation axiom that can be considered in the unified logic.

Since we use a single-conclusion system, weakening on the right can only take the form of $0E$ (*abort*), $\bot E$ (*break*) and $Unlock_\bot$. The $\lambda\mu$ notation $[d]t$ is equivalent to (dt), or to $\mathcal{B}(dt)$ in the case of $Unlock_\bot$[3]. It is also possible to design a multiple-conclusion proof system, based on the multiple-conclusion version of intuitionistic sequent calculus (i.e., the Beth-Fitting intuitionistic tableau). In such a context an additional, *intuitionistic* version of μ would be needed, and γ can then lock a formula on the right that's *not* in the stoup, which means that γ can be seen as a non-binding operator. We showed in [LM13b] how such an interpretation, along with an appropriate abstract machine, can formulate dynamically scoped continuation jumps. However, that subject is orthogonal to the main aims of the present paper.

Figure 3 displays two sample proofs. The first is for our version of Peirce's formula. The second is new: $\frown e \rightarrowtriangle (\frown\frown A \rightarrowtriangle A)$. Here, A is any formula, red or green. The assumption $\frown e = e \rightarrowtriangle 0$ causes a collapse into classical logic since it implies that 0, and therefore all formulas, have the characteristics of green formulas. However it is a *one-time only* assumption: the collapse is momentary. In order for this *use-once* control operator to have its usual effect, a permission "token" in the form $\frown e$ (or $\sim e$) needs to be consumed. A similar proof derives a *call-once/cc* operator, this time with no restriction on Q:

$$\lambda x \lambda z.\gamma d.x(\lambda y.\mathcal{A}(z\ [d]y)) : ((P \rightarrowtriangle Q) \rightarrowtriangle P) \rightarrowtriangle \frown e \rightarrowtriangle P$$

Term Reduction Rules

We list below a relatively conservative set of reduction rules for our proof terms.

- $(\lambda x.s)\ t \ \longrightarrow\ s[t/x], \qquad (\lambda x.s) \cdot t \ \longrightarrow\ s[t/x]$
- $(\gamma d.s)\ t \ \longrightarrow\ \gamma d.s\{[d](w\ t)/d[w]\}\ t, \qquad (\gamma d.s) \cdot t \ \longrightarrow\ \gamma d.s\{[d](w \cdot t)/d[w]\} \cdot t$
- $\mathcal{A}(s)\ t/\mathcal{A}(s) \cdot t \ \longrightarrow \mathcal{A}(s), \qquad \mathcal{B}(s)\ t/\mathcal{B}(s) \cdot t \ \longrightarrow \mathcal{B}(s)$
- $let\ (x;y) = (u;v)\ in\ t \ \longrightarrow\ t[u/x, v/y]$
- $\gamma a \gamma b.s \ \longrightarrow\ \gamma a.s[a/b]$
- $[d]\gamma a.s \ \longrightarrow\ [d]s[d/a]$
- $\gamma d.s \ \longrightarrow\ s$ when d is not free in s

The first four sets of rules are reduction rules while the last three are *renaming rules,* which eliminate redundant *Lock*s (redundant contractions).

Theorem 11 *The term reduction rules satisfy subject reduction and are strongly normalizing.*

Subject reduction is shown by checking case by case that the rules represent valid proof transformations. All of these rules have equivalents in classical $\lambda\mu$ calculus. All typable ACL terms are typable in classical logic: there are just *fewer* valid reductions because of the stronger type system. For example, the case for $let\ (x;y) = (u;v)\ in\ t$ is just a special case of $(\lambda x \lambda y.t)\ u\ v$. Every reduction path here corresponds to a reduction path in $\lambda\mu$ calculus. The presence of Dr and the treatment of indices in the *Lock* and *Unlock* rules represent nothing more than α-conversion, which does not affect normalization. The ! operators representing Pr likewise do not affect normalization. Thus there is no question that this system is strongly normalizing given that the result is known for classically typed $\lambda\mu$ terms.

[3] See [dG94] for a clear explanation of $\lambda\mu$-calculus and control operators

8 Structural Rules and Delimited Control

The transitions between different modes of proof, in the form of the structural rules of LAC and NAC, have the effect on cut-elimination that of delimited control operators. This correspondence is consistent with the recent work of Ilik [Ili12], which shows that delimited control behavior can be seen as resulting from the transition between non-classical and classical modes of proof. In this section we explore two extensions of the proof representation presented above that captures forms of delimited control.

8.1 Delimited Abort

The manner in which coloring information determines how cut is reduced with respect to the *Unlock* ($Unlock_\bot$) and $\bot E$ rules leads to an interesting computational effect. Consider

$$\dfrac{s : \Gamma; \Delta \vdash e_1 \to e_2 \quad \dfrac{t : \Gamma; \Delta' \vdash A}{[d]t :\ [\Delta' : A]^d, \Gamma; \vdash e_1}\ Unlock}{(s_{\_}[d]t) :\ [\Delta' : A]^d, \Gamma; \Delta \vdash e_2}\ \to E$$

With e_1 and e_2 both green, there are two ways to reduce this cut. The first is by usual β-reduction, once s has been reduced to a lambda-term. A second possibility is to reduce to the following:

$$\dfrac{t : \Gamma; \Delta\Delta' \vdash A}{[d]t :\ [\Delta' : A]^d, \Gamma; \Delta \vdash e_2}\ Unlock$$

With weakening, the same t still proves the premise. The same choice exists for $\multimap$. The context formed by s is discarded. However, if e_2 was not green, then the only choice is β-reduction. A similar situation exists if the last rule of s is also *Unlock* (assuming the relaxed version of *Unlock*). A term ($[d]s\ [d]t$) can be reduced to either $[d]s$ or $[d]t$ and determinism would require a specific evaluation strategy.

In contrast to a term $\mathcal{A}(t)$, which uses 0-elimination, the "break" generated by a $[d]t$ or $\mathcal{B}(t)$ cannot escape the entire program context but is thrown upwards towards the nearest red context, i.e., the red continuation skips to where the break occurs. We can write a special case of $\multimap$ elimination (similarly for $\to$-elimination) marking a switch between green and red contexts (R is red):

$$\dfrac{u : \Gamma; \Delta \vdash e \multimap R \quad v : \Gamma; \Delta' \vdash e}{(u\ \sharp v) :\ \Gamma; \Delta\Delta' \vdash R}\ \multimap \sharp E$$

This rule should be applied in place of the usual $\multimap E$ in all such cases. Reduction, restricted to simple λ terms plus $\mathcal{B}(t)$ (which subsumes *Unlock* where $[d]s \equiv \mathcal{B}(d\ s)$), can be redefined by the following rules, in order of precedence:

$s\ \mathcal{B}(t) = \mathcal{B}(t)$;
$(\lambda x.s)\ t = match\ t\ with$
  $\mid \sharp u \Rightarrow s[u/x]$
  $\mid v \Rightarrow s[v/x]$

The "delimiter" $\sharp$ is a type annotation that indicates a transition from green to red; it has no meaning independently of such a context. The delimiter is dropped after substitution, which gives this reset/prompt marker a dynamic behavior. We can prove that the usual congruent closure of this reduction relation preserves types (subject reduction) regardless of evaluation order. Instances of subterms ($s\ t$) where s is of type $E \multimap R$ are not well-typed: they must be in the form ($s\ \sharp t$). For example, $(\lambda x.g\ \sharp(f_2\ x))\ \sharp(f_1\ \mathcal{B}(u))$ reduces to $(g\ \sharp\mathcal{B}(u))$. Both f_1 and f_2 are aborted. Here, f_2 must be of some type $E \multimap E'$ and g of type $E' \multimap R$. Reducing to a term that contains $(f_2\ \sharp\mathcal{B}(u))$ is not type-sound. This behavior is dynamic because one cannot determine which $\sharp$ will stop the abort without reducing the term. In practice, finer control can be gained by casting a green type E into a red equivalent such as $E \otimes \top$ (use a "dummy" term of type $E \multimap E \otimes \top$), which would then allow $\sharp$ to appear.

The delimited abort operation can be used to model exception handling, with $\sharp$ representing the *catching* of an exception, which is likewise dynamically scoped in languages such as ML or Java.

8.2 Capturing Delimited Continuations

Besides *Unlock* and $\bot E$, the other rule that is sensitive to coloring information is Pr, which cancels the non-contractable affine context when the stoup is green. Consider the following sample scenario:

$$
\dfrac{\lambda x.u : \Gamma;\vdash e \multimap R \qquad \dfrac{\lambda y.f : \Gamma;\vdash e' \multimap e \qquad \dfrac{\dfrac{\dfrac{\begin{array}{c}\dfrac{t : \Gamma_1;\Delta_1 \vdash e'}{[:e']^k_e, \Gamma_1;\Delta_1 \vdash e}\,Unlock_e \\ \vdots \end{array}}{[:e']^k_e, \Gamma\Delta;\vdash e'}}{s : \Gamma\Delta;\vdash e'}\,Lock_e}{!s : \Gamma;\Delta \vdash e'}\,Pr}{(\lambda y.f)\ !s : \Gamma;\Delta \vdash e}\,\multimap E}{\lambda x.u\ \sharp((\lambda y.f)\ !s) : \Gamma;\Delta \vdash R}\,\multimap \sharp E
$$

With R red but e, e' green, it is possible to permute the cut with proof $\lambda y.f$ above the Pr, and then above the *Unlock* (unlike the generic cut elimination procedure of Section 5). However, the only way to cut with $\lambda x.u$ is to substitute the right subproof into u (β-reduction), as R will not be able to duplicate Pr:

$$
\dfrac{\lambda x.u : \Gamma;\vdash e \multimap R \qquad \dfrac{\dfrac{\begin{array}{c}\dfrac{\lambda y.f : \Gamma;\vdash e' \multimap e \quad t : \Gamma_1;\Delta_1 \vdash e'}{(\lambda y.f)t : \Gamma_1\Gamma;\Delta_1 \vdash e}\,\multimap E \\ \vdots \end{array}}{\Gamma_1\Gamma\Delta;\vdash e}}{!s\{(\lambda y.f)w/[d]w\}\Gamma_1\Gamma;\Delta \vdash e}\,Pr}{\Gamma;\Delta \vdash R}\,\multimap \sharp E
$$

The transitional type $e \multimap R$ again signals delimited control.

To construct a system that allows the direct-style capture of delimited continuation, we need to first fix a call-by-value like reduction strategy (as in [OS97]). Terms $\gamma d.t$ are not considered values and thus terms of the form $(\lambda x.s)\ \gamma d.t$ are not reduced by β: rather the cut is permuted to instances of $[d]w$ inside t. Our scheme differs from the usual definition of call-by-value in that whether terms $!t$ are considered values depend on the presence of $\sharp$, which is itself not a term constructor. To be more precise, we define the following categories of terms and evaluation contexts:

Values : $V \ ::= \ x \ | \ \lambda x.T$

Pre-values : $P \ ::= \ V \ | \ \gamma d.T \ | \ !P$

Terms: $T \ ::= \ V \ | \ \gamma d.t \ | \ !T \ | \ [d]T \ | \ (T\ T) \ | \ (T\ \sharp T)$

The same $\multimap \sharp E$ rule is used. Both the $\sharp$ and ! symbols are needed, since if u is not of type $e \multimap R$, then no delimitation is required despite the !. It is only when $\sharp$ and ! appears together that evaluation will be affected. It is not possible to combine the introduction of $\sharp$ and ! into a single rule as it would not be preserved under substitution.

The definition of evaluation context E must distinguish between redexes of the form $(\lambda x.s)\ !t$, which should move the Pr rule beneath the cut (to $!(\lambda x.s)t$), and situations such as $(\lambda x.s)\ \sharp !T$, in which case T may require further evaluation.

- $E \ ::= \ [\ ] \ | \ E\ T \ | \ P\ E \ | \ E\ \sharp T \ | \ P\ \sharp E \ | \ !E$

$P\ \sharp !_{\bar{z}} E$. Evaluation is defined by the following rules, in which $(\sharp)$ represents the possible (but consistent) presence of the $\sharp$ symbol.

- $E[P_1\ !P_2] \ \longrightarrow \ E[!(P_1\ P_2)]$ (permutation of cut above Pr)

- $E[P\ (\sharp)\gamma d.t] \ \longrightarrow\ E[\gamma d.P\ (\sharp)t\{[d]P\ (\sharp)w/[d]w\}]$ (capture of evaluation context)
- $E[(\gamma d.t)\ (\sharp)V] \ \longrightarrow\ E[\gamma d.t\{[d]w\ (\sharp)V/[d]w\}\ V]$ (permutation of cut above *Lock*)
- $E[(\gamma d.t)\ \sharp!P] \ \longrightarrow\ E[\gamma d.t\{[d](w\ \sharp!P)/[d]w\}\ V]$ ($\lambda\mu$-style structural reduction)
- $E[(\lambda x.T)\ \sharp!P] \ \longrightarrow\ E[s\{!P/x\}]$ (beta-reduction)
- $E[(\lambda x.T)\ (\sharp)V] \ \longrightarrow\ E[s\{V/x\}]$ (beta-reduction)

Define a *redex* to be any term of one of the forms r found in an evaluation rule $E[r] \longrightarrow s$. We have the following:

Lemma 12 *Every closed term T is either a pre-value P or of the form $E[r]$ where r is a redex. Furthermore, E and r are unique.*

This lemma is proved by induction on the structure of closed terms. We detail the most important (and representative) cases:

Assume that T is of the form $(T_1\ T_2)$. We have the following mutually exclusive cases:

1. T_1 is not of the form P. Then by inductive hypothesis, $T_1 = E'[r]$. So let $E = E'\ T_2$ and so $T = E[r]$. E is uniquely determined if E' is.
2. T is of the form $(P\ T_2)$ but T_2 is not a pre-value. Again by inductive hypothesis $T_2 = E'[r]$ so let $E = P\ E'$.
3. Assume T is of the form $(P_1\ P_2)$. Let $E = []$. We know that P_1 cannot be of the form $!_z.P'$ since the Pr rule cannot be applied to an arrow type. We therefore have the following possibilities:
   (a) $P_1 = \lambda x.T$. Then the redex r is either $(\lambda x.T)\ V$, $(\lambda x.T)\ \gamma d.t$, or $(\lambda x.T)\ !P$.
   (b) $P_1 = \gamma d.T$. The redex is either $(\gamma d.T)\ !P$, $(\gamma d.T)\ (\gamma f.S)$ or $(\gamma d.T)\ V$.

This lemma shows both the *progress* of evaluation and that evaluation is deterministic. Of course we can also show that evaluation is type sound (subject reduction).

The type of delimited control operator that γ implements is dynamic, since in the captured term $[d]Pw$, which can be generalized into a continuation context, is not itself delimited. We shall not attempt any termination results as it is known that the dynamic control/prompt operations can lead to non-terminating behavior under call-by-value (see [KY08]), even in typed settings. That does not contradict cut-elimination, since β-reduction is still possible. However, the full power of delimited control operators are only revealed in a direct style, call-by-value setting where terms such as $\lambda y.f$ are captured as part of the continuation (as opposed to applied immediately as in call-by-name). Under such a setting, delimitation is *logically necessitated* by the green/red distinction of ACL.

9 The Colors of Second Order Bound Variables

The addition of first-order quantifiers to ACL would be a rather standard exercise. More significant would be addition of second-order quantifiers. One problem that has faced polarized systems, including LC and focusing systems, has been how to assign polarities to second order formulas, specifically to propositional variables that are bound by $\forall$ and $\exists$. One might consider two versions of each quantifier, which restricts also the polarity of formulas that can instantiate them. Another approach might be to keep bound variables unpolarized. None of these approaches is satisfactory.

In contrast, the red and green colors of ACL do not represent a "duality" but rather *two levels of provability.* Proposition 6 states that provability is always preserved by replacing red atoms with green ones and substitution property of ACL (Theorem 7) clearly indicate that red atoms have the characteristics of universally quantified propositional variables: one proves $\forall A.A \rightarrow A$. On the other hand, one can only prove $\exists A.\neg\neg A \rightarrow A$. It is therefore tempting to regard existentially quantified variables as green atoms. This uniform scheme of assigning colors was first proposed in [LM13b] but not carefully verified. The interpretation of $\exists$ is the closure over a possibly infinite union of facts: it can hardly be anything else. Its introduction rules are also those that are expected. In particular, the right introduction rule can still instantiate the bound variable with any term while the left introduction rule must replace the bound variable with an *arbitrary* variable, which must be red. Such rules are required for soundness. However, regarding existentially quantified variables as green is consistent with the properties that we expect of green formulas, including the properties established by Lemma 1 and Lemma 2. Formally, we specify that:

- In $\forall X.P$, the free occurrences of X in P are red;
- In $\exists X.P$, the free occurrences of X in P are green.
- $\forall X.P$ is green if P is green, otherwise it is red.
- $\exists X.P$ is green if P is green, otherwise, it is red.

The inference rules for $\forall$ and $\exists$, in the context of LAC, are as follows

$$\frac{\Gamma; \Delta \vdash A[B/X]}{\Gamma; \Delta \vdash \exists X.A}\ \exists R \qquad \frac{\Gamma; \Delta, A \vdash P}{\Gamma; \Delta, \exists Y.A \vdash P}\ \exists L,\ Y \text{ red} \qquad \frac{\Gamma; \Delta \vdash A}{\Gamma; \Delta \vdash \forall Y.A}\ \forall R,\ Y \text{ red} \qquad \frac{\Gamma; \Delta, A[B/X] \vdash P}{\Gamma; \Delta, \forall X.A \vdash P}\ \forall L$$

The usual restrictions on free occurrences of Y apply. We can still universally quantify over green formulas using $\forall X.(\bot \rightarrow X) \rightarrow P$: although $\bot \rightarrow X$ is not technically green, it implies equivalence with green formulas. No complementary form exists, however, to restrict $\exists$ to quantify over only red formulas. That would mean *requiring something to be anything:* a self-contradiction.

Semantically, (again following [Oka02]) we consider non-standard second order phase models where facts D is a proper subset of the facts of a standard model that remains closed over the interpretation of all connectives and constants, including:

- $(\forall X.A)^p \ = \ \bigcap_{d \in D} A[X^p = d]^p$
- $(\exists X.A)^p \ = \ cl(\bigcup_{d \in D} A[X^p = d]^p)$

Here, by $A[X^p = d]^p$ we mean the interpretation of A under the assumption that propositional variable (atom) X valuates to fact d. We can extend the property established by Lemma 1 to green instances of these formulas. In particular, we note that

- if $\forall X.A$ is green then $A[C/X]$ is also green for any formula C.
- if $\exists X.A$ is green and B is green, then $A[B/X]$ is also green.

This property is easily established by induction on A. Clearly all green $\exists X.A$ valuates to $\top$ or $\bot$ since $A[\bot/X]$ does. The permutation property of Lemma 2 also holds under this extended coloring scheme. For example, $\exists X.(X \oplus B)$ is green, and can activate the generalized PR rule by the following permutation:

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\Gamma, A; \Delta \vdash \exists X.X \oplus B}{[: \exists X.X \oplus B], \Gamma, A; \Delta \vdash \bot}\ Unlock}{[: \exists X.X \oplus B], \Gamma; \Delta, A \vdash \bot}\ Pr}{[: \exists X.X \oplus B], \Gamma; \Delta, A \vdash \bot \oplus B}\ \oplus R}{[: \exists X.X \oplus B], \Gamma; \Delta, A \vdash \exists X.X \oplus B}\ \exists R}{\Gamma; \Delta, A \vdash \exists X.X \oplus B}\ Lock$$

For these reasons the green coloring of existentially quantified propositional variables is valid, despite some awkwardness regarding the $\exists L$ rule. The point being that if X is not considered green then we would not be able to recognize that formulas such as $\exists X.X \oplus B$ have the properties of green formulas.

10 Conclusion

Let us summarize the important components of ACL as follows.

1. A phase semantics that approximate the Kripke semantics of intuitionistic logic. Facts are upwardly closed sets. The constant $\bot$ is the second-largest fact, with more attributes than its counterpart in linear logic.

2. The coloring of formulas into green (classical) and red (possibly non-classical). The colors of arbitrary formulas reduce to the colors of atoms and constants, in particular to the green $\bot$.

3. The validity of the Peirce-like formula $(-P \to P) \multimap P$, which implies its "weaker" half, $(P \to -P) \multimap -P$. They enable contractions on arbitrary formulas when $\bot$, or any green formula, is encountered as the current, or stoup formula in a proof. This "self-dual" principle replaces ? and ! in allowing restrictions on contraction to coexist with cut elimination.

4. A sound and complete sequent calculus that enables contractions dynamically. The classical effect of green formulas is localized in proof segments.

5. The cut-elimination procedure for this proof reveals the importance of new structural rules such as *Unlock* and *Pr*, and how red/green coloring impacts cut-elimination. It also includes $\lambda\mu$-style *structural* reductions.

6. From these elements we derive a computational interpretation of natural deduction proofs that allows intuitionistic and affine-linear lambda terms to coexist with control operators such as *call/cc*.

The logical interpretation of the computational content of proofs that use contractions on the right-hand side thus does not require a collapse into classical logic.

References

[And92] Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *J. of Logic and Computation*, 2(3):297–347, 1992.

[Bar96] A. Barber. Dual intuitionistic linear logic. Technical Report ECS-LFCS-96-347, 1996.

[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.

[Gen35] Gerhard Gentzen. Investigations into logical deduction. In M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland, Amsterdam, 1935.

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

[Gir93] Jean-Yves Girard. On the unity of logic. *Annals of Pure and Applied Logic*, 59:201–217, 1993.

[Gir11] Jean-Yves Girard. *The Blind Spot.* European Mathematical Society Publishing House, 2011.

[HM94] Joshua Hodas and Dale Miller. Logic programming in a fragment of intuitionistic linear logic. *Information and Computation*, 110(2):327–365, 1994.

[ILH10] Danko Ilik, Gyesik Lee, and Hugo Herbelin. Kripke models for classical logic. *Annals of Pure and Applied Logic*, 161(11):1367–1378, 2010.

[Ili12] Danko Ilik. Delimited control operators prove double-negation shift. *Annals of Pure and Applied Logic*, 163(11):1549–1559, 2012.

[Kop95] A. P. Kopylov. Propositional linear logic with weakening is decidable. In *Symposium on Logic in Computer Science*, pages 496–504. IEEE, 1995.

[KY08] Yukiyoshi Kameyama and Takuo Yonezawa. Typed dynamic control operators for delimited continuations. In *Symposium on Functional and Logic Programming*, pages 239–254, 2008.

[Laf97] Y. Lafont. The finite model property for various fragments of linear logic. *Journal of Symbolic Logic*, 62:1202–1208, 1997.

[Lia16] Chuck Liang. Unified semantics and proof system for classical, intuitionistic and affine logics. In *Symposium on Logic in Computer Science (LICS)*, July 2016.

[LM11] Chuck Liang and Dale Miller. A focused approach to combining logics. *Annals of Pure and Applied Logic*, 162(9):679–697, 2011.

[LM13a] Chuck Liang and Dale Miller. Kripke semantics and proof systems for combining intuitionistic logic and classical logic. *Annals of Pure and Applied Logic*, 164(2):86–111, February 2013.

[LM13b] Chuck Liang and Dale Miller. Unifying classical and intuitionistic logics for computational control. In Orna Kupferman, editor, *28th Symp. on Logic in Computer Science*, pages 283–292, 2013.

[Mil96] Dale Miller. Forum: A multiple-conclusion specification logic. *Theoretical Computer Science*, 165(1):201–232, September 1996.

[Oka02] Mitsuhiro Okada. A uniform semantic proof for cut elimination and completeness of various first and higher order logics. *Theoretical Computer Science*, 281(1-2):471–498, 2002.

[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.

[Vel76] Wim Veldman. An intuitionistic completeness theorem for intuitionistic predicate logic. *Journal of Symbolic Logic*, 41(1):159–166, 1976.

A Focused Sequent Calculus

Similarities do exist between the positive/negative polarization of LC and focused proof systems and the red/green coloring of ACL. The semantics of ACL clearly show how green means classical and red means non-classical (in terms of restrictions to structural rules). Positive formulas also exhibit some of the characteristics of red formulas, especially in that proofs cannot use classical structural rules when positive formulas occupy the stoup. The stoup is already an important part of ACL proof theory that relies on the "polarization" of green atoms and $\bot$ as negatives, which allows us to use an important element of focusing in the *Unlock* rule. However, not all red formulas are positive and not all negative formulas are green. This analysis indicates that positive/negative polarization is not enough to distinguish between classical and non-classical proofs, and that the red/green colors offer a more refined classification. However, focusing has proven to be a useful normal form in other ways, such as in deterministic proof search and in defining evaluation strategies (call-by-value versus call-by-name). In this section we propose a focused sequent calculus for ACL. By doing so, we will also be explicating the precise relationship between the red/green colors and positive/negative polarities.

Polarization

1. $\bot$ is negative, $\top$ is negative, 0 is positive (but doesn't matter).
2. Green atoms are negative. The reason is that green atoms a are in fact embeddable as $a \to \bot$ where a is not green.

   $\bot$ and green atoms must be negative also because of the *Unlock* rule, which is already focused.
3. Red atoms can be negative or positive.
4. All Green formulas are considered asynchronous on the right, this includes formulas $B \oplus E$, by virtual of the multiplicative connective that incorporates *Lock*. $B \oplus E$ is considered to be negative when appearing on the right-hand side of sequents. Left-side occurrences are regarded as positive.
5. Both $\to$ and $\multimap$ are negative: asynchronous on the right and synchronous on the left.
6. $\otimes$ is always positive and synchronous on the right, asynchronous on the left.
7. Red & is asynchronous on the right, but synchronous on the left, while red $\oplus$ is synch on the right and asynch on the left.

A *negative* literal is a negatively polarized atom or constant. These include green atoms and $\bot$.

Focusing in linear, intuitionistic and classical logics permute the application of of structural rules to take place *in between* the different positive phases. In ACL contractions can take place anywhere in the form of Lock. It may appear, therefore that ACL requires contraction on subformulas inside a synchronous phase. Fortunately, this complication is avoided due to two factors. First, the affine-linear context can be saved by *Lock*, and secondly, unused assumptions can always be weakened. It is therefore possible to see that contraction on subformulas of a positive formula can always be replaced by contraction on the overall formula, although the resulting proof may become larger (a smart theorem prover will apply appropriate sharing). For example,

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma'; \Delta\Theta_1 \vdash A}{[\Theta_1 : A], \Gamma'; \Delta \vdash e}\ Unlock}{\vdots}}{[\Theta_1 : A], \Gamma; \Theta_1 \vdash A}}{\Gamma; \Theta_1 \vdash A}\ Lock \qquad \Gamma; \Theta_2 \vdash B$$
$$\dfrac{\Gamma; \Theta_1 \vdash A \qquad \Gamma; \Theta_2 \vdash B}{\Gamma; \Theta_1\Theta_2 \vdash A \otimes B}\ \otimes R$$

Structural Rules

$$\frac{[\Omega : Q], \Gamma; \Omega \vdash Q}{\Gamma; \Omega \vdash Q}\ Lock \qquad \frac{\Gamma; \Omega\Omega' \Delta \vdash Q}{[\Omega : Q], \Gamma; \Omega' \vdash e}\ Unlock \qquad \frac{M, \Gamma; \Omega \vdash e}{\Gamma; \Omega, M \vdash e}\ Pr$$

Decision Rules

$$\frac{\Gamma; \Omega, \Downarrow N \vdash Q}{\Gamma; \Omega, N \vdash Q}\ D_1 \qquad \frac{\Gamma; \Omega, \Downarrow A \vdash Q}{A, \Gamma; \Omega \vdash Q}\ D_2 \qquad \frac{\Gamma; \Omega \vdash \Downarrow P}{\Gamma; \Omega \vdash P}\ D_3$$

Reaction Rules

$$\frac{\Gamma; \Omega, P \vdash B}{\Gamma; \Omega, \Downarrow P \vdash B}\ R_\ell \qquad \frac{\Gamma; \Omega \vdash N}{\Gamma; \Omega \vdash \Downarrow N}\ R_r \qquad \frac{\Gamma; \vdash \Downarrow^\bullet P_2}{\Gamma; \Omega \vdash \Downarrow P_2}\ Lat \qquad \frac{\Gamma; \vdash \bar{P}_2}{\Gamma; \vdash \Downarrow^\bullet \bar{P}_2}\ R_2$$

Asynchronous Phase

$$\frac{\Gamma; \Delta, A \vdash B}{\Gamma; \Delta \vdash A \twoheadrightarrow B}\ \twoheadrightarrow R \qquad \frac{A, \Gamma; \Delta \vdash B}{\Gamma; \Delta \vdash A \to B}\ \to R \qquad \frac{A, \Gamma; \Delta \vdash C \quad B, \Gamma; \Delta \vdash C}{\Gamma; \Delta, A \vee B \vdash C}\ \vee L \qquad \frac{A, B, \Gamma; \Delta \vdash C}{\Gamma; \Delta, A \wedge B \vdash C}\ \wedge L$$

$$\frac{\Gamma; \Delta \vdash A \quad \Gamma; \Delta \vdash B}{\Gamma; \Delta \vdash A \,\&\, B}\ \& R \qquad \frac{\Gamma; \Delta, A \vdash C \quad \Gamma; \Delta, B \vdash C}{\Gamma; \Delta, A \oplus B \vdash C}\ \oplus L \qquad \frac{\Gamma; \Delta, A, B \vdash C}{\Gamma; \Delta, A \otimes B \vdash C}\ \otimes L$$

$$\frac{[\Delta : B], \Gamma; \Delta \vdash E}{\Gamma; \Delta \vdash E \oplus B}\ \oplus ER_1 \qquad \frac{[\Delta : B], \Gamma; \Delta \vdash E}{\Gamma; \Delta \vdash B \oplus E}\ \oplus ER_2 \qquad \frac{}{\Gamma; \Delta \vdash \top}\ \top R \qquad \frac{}{\Gamma; \Delta, 0 \vdash A}\ 0L$$

Synchronous Phase One and Phase Two

$$\frac{\Gamma; \Omega_1 \vdash \Downarrow A \quad \Gamma; \Omega_2 \vdash \Downarrow B}{\Gamma; \Omega_1\Omega_2 \vdash \Downarrow A \otimes B}\ \otimes R \qquad \frac{\Gamma; \Omega_1 \vdash \Downarrow A \quad \Gamma; \Omega_2, \Downarrow B \vdash C}{\Gamma; \Omega_1\Omega_2, \Downarrow A \twoheadrightarrow B \vdash C}\ \twoheadrightarrow L \qquad \frac{\Gamma; \vdash \Downarrow^\bullet A \quad \Gamma; \Omega, \Downarrow B \vdash C}{\Gamma; \Omega, \Downarrow A \to B \vdash C}\ \to L$$

$$\frac{\Gamma; \Omega, \Downarrow A_i \vdash C}{\Gamma; \Omega, \Downarrow A_1 \,\&\, A_2 \vdash C}\ \& L \qquad \frac{\Gamma; \Omega \vdash \Downarrow R_i}{\Gamma; \Omega \vdash \Downarrow R_1 \oplus R_2}\ \oplus R \qquad \frac{}{\Gamma; \Omega, \Downarrow n \vdash n}\ Id_1 \qquad \frac{}{\Gamma; \Omega, p \vdash \Downarrow p}\ Id_3 \qquad \frac{}{\Gamma; \Omega, \Downarrow \bot \vdash e}\ \bot L$$

$$\frac{\Gamma; \vdash \Downarrow^\bullet A_i}{\Gamma; \vdash \Downarrow^\bullet A_1 \vee A_2}\ \vee R \qquad \frac{\Gamma; \vdash \Downarrow^\bullet A \quad \Gamma; \vdash \Downarrow^\bullet B}{\Gamma; \vdash \Downarrow^\bullet A \wedge B}\ \wedge R \qquad \frac{}{p, \Gamma; \vdash \Downarrow^\bullet p}\ Id_2$$

Syntactic Categories:
P : positive formula (level 1 or 2); Q : positive formula or negative literal
N : negative formula; M : negative formula or positive literal
R : red formula; E : green formula
e : green atom or $\bot$
p : positive atom; n : negative atom
P_2: positive-2 formula ($\vee$, $\wedge$) or positive atom; $\bar{P}_2$: non-positive-2 formula
Ω, Ω' : all negative formulas and positive literals
Γ, Δ, A, B, C : arbitrary formulas

Figure 4: Focused Sequent Calculus FAC.

can be replaced by

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma';\Delta\Theta_1 \vdash A \qquad \Gamma';\Delta\Theta_2 \vdash B}{\Gamma';\Delta\Theta_1\Theta_2 \vdash A\otimes B}\otimes R}{[\Theta_1\Theta_2 : A\otimes B],\Gamma';\Delta \vdash e}\mathit{Unlock}}{\vdots}}{[\Theta_1\Theta_2 : A\otimes B],\Gamma;\Theta_1 \vdash A} \qquad [\Theta_1\Theta_2 : A\otimes B],\Gamma;\Theta_2 \vdash B}{\dfrac{[\Theta_1\Theta_2 : A\otimes B],\Gamma;\Theta_1\Theta_2 \vdash A\otimes B}{\Gamma;\Theta_1\Theta_2 \vdash A\otimes B}\mathit{Lock}}\otimes R$$

It can always be assumed that $\Gamma \subseteq \Gamma'$. The ability to save the affine context therefore grant to these assumptions a similar property, bounded by the appearance of green literals.

Extending focus to the intuitionistic disjunction $\vee$ presents further challenges. Here, some intuition from linear logic will help us understand what is needed. If we see $A \vee B$ as $!A \oplus !B$, then it is clear that focus need not stop with $(!A \oplus !B) \oplus C$ by the associativity of $\oplus$. But $!A \oplus !(B \oplus C)$ clearly means that focus must stop because of the !. However, focus on $A \vee B \vee C$ should clearly be possible. What this suggests is that the positive polarity should be divided into two levels, with $\vee$ called a "positive-2" formula. There are thus two focusing arrows for the two levels of positive formulas. It is also possible to understand this principle independently of linear logic by considering $A \vee (B \oplus C)$ as a possible *synthetic connective*. It is easy to see what the (unfocused) introduction rules of such a connective must be:

$$\dfrac{\Gamma; \vdash A_i}{\Gamma; \vdash A_1 \vee (A_2 \oplus A_3)} \vee\oplus R \qquad \dfrac{A,\Gamma;\Delta \vdash D \quad \Gamma;\Delta, B \vdash D \quad \Gamma;\Delta, C \vdash D}{\Gamma;\Delta, A \vee (B \oplus C) \vdash D} \vee\oplus L$$

But one would not be able to prove *initial elimination* using these introduction rules:

$$; A \vee (B \oplus C) \vdash A \vee (B \oplus C)$$

However, the same exercise will show that $A \oplus (B \vee C)$ *can* be considered a synthetic connective. This preservation of focus is enabled by the *lateral* reaction rule *Lat*. The new Id_2 rule allows a level-2 focus phase to finish without a meaningless DR.

We have extended the positive-2 connectives to include the positive intuitionistic conjunction $\wedge$, as found in the focused sequent calculus LJF.

A sequent of the form $\Gamma; \Omega \vdash Q$ is "neutral" in that the only asynchronous (positive on left), non-literal formulas in the sequent are in Γ. Such a formula can be selected by D_2 (which replaces D_r, and immediately cause a reaction that terminates the $\Downarrow$ stage and decompose the asynchronous formula. Such a scenario is also found in the focused sequent calculus of full linear logic [And92], with formula such as $?(A \,\&\, B)$.

A "phase" of a focused proof in FAC runs as follows:

1. All asynchronous connectives are decomposed eagerly. These include all green formulas on the right, so at the end of this phase if the right-hand side is green it must be a green atom or $\perp$.

2. The structural rules *Pr* and *Unlock* are applied, *Pr* first. The formulas unlocked may trigger more asynchronous decompositions.

3. When no asynchronous formulas are left in the affine-linear context (Ω) or on the right-hand side, a formula is selected for focus by one of the decide rules. Since D_2 may select an asynchronous formula, it may cause an immediate *release* by R_l, which will trigger more asynchronous decompositions. The selection of a synchronous formula will begin a focus phase indicted by $\Downarrow$.

4. Focus may laterally transition to positive-2 formulas, but may not transition back.

5. The focusing phase stops when an asynchronous subformula is encountered, via the reaction rules.

As in Andreoli-inspired focused systems, focusing on atoms necessitates an initial rule. This aspect of focusing is already incorporated in the *Unlock* rule. The *Unlock* rule is in fact no longer needed: it can be replaced by a regular combination of D_ℓ and $\otimes L$, followed by asynchronous decompositions of $\multimap R$ and $\otimes L$. However, we have kept *Unlock* as a separate rule as a convenience. It makes proofs slightly more deterministic since the affine context is split in the appropriate way.

The correctness of the focused sequent calculus can be proved directly by verifying the permutablity of synchronous introductions rules with respect to each other, and the invertibility of asynchronous introduction rules.

Theorem 13 $; \vdash A$ *is provable in FAC if and only if it is provable in LAC.*

Cut Elimination in FAC directly is possible. In fact, the proof is the essentially the same as for the unfocused calculus, since all the interesting cases concern the structural rules.