Math. Struct. in Comp. Science (2019), *vol.* 29, *pp.* 1177–1216.
doi:10.1017/S0960129518000403 First published online 12 October 2018

On the unification of classical, intuitionistic and affine logics

CHUCK LIANG

Hofstra University, Hempstead, NY, U.S.A.
Email: Chuck.C.Liang@Hofstra.edu

Received 26 May 2017; revised 20 September 2018

This article presents a *unified logic* that combines classical logic, intuitionistic logic and affine linear logic (restricting contraction but not weakening). We show that this unification can be achieved semantically, syntactically and in the computational interpretation of proofs. It extends our previous work in combining classical and intuitionistic logics. Compared to linear logic, classical fragments of proofs are better isolated from non-classical fragments. We define a phase semantics for this logic that naturally extends the Kripke semantics of intuitionistic logic. We present a sequent calculus with novel structural rules, which entail a more elaborate procedure for cut elimination. Computationally, this system allows affine-linear 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 that correspond to delimited control effects.

1. Introduction

The study of structural rules forms a significant part of proof theory, in particular, the rule of contraction. The different levels of admissibility of contraction lead to various forms of logic, classical and non-classical. At a deeper level, structural rules, and restrictions thereof, have profound effects on cut elimination and consequently, on computational applications. We formulate a *unified logic* that combines classical, intuitionistic and affine-linear logics (linear logic with weakening). Such a combination must be achieved without a collapse into classical logic: non-classical connectives must retain their strengths when mixed with classical formulas. Compared to other unified systems, including LU (Girard 1993) and our own LKU (Liang and Miller 2011), this system diverges from linear logic and *polarization* in that contractions are controlled in a very different way. Instead of the exponential operators ? and !, the logic admits a restricted form of Peirce's formula, which enables contractions when certain conditions are reached. The system *Affine Control Logic* (ACL) is an extension of the PCL system presented in Liang and Miller (2013b). This article is also an extended version of Liang (2016).

Linear logic embeds classical and intuitionistic logics but 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. We seek to better isolate non-classical from classical fragments of proofs. One part of our solution relies on *focusing* (Andreoli 1992). This idea is also found in the 'stoup' of LC (Girard 1991) and related systems. When a 'positive' formula occupies the stoup the proof takes on the characteristics of an intuitionistic proof. However, the focusing approach alone is not satisfactory: a (multiplicative) implication $A \rightarrow B$ is 'negative' and cannot occupy the stoup. The second part of our solution is to replace the positive/negative polarities of focusing with a semantically motivated categorization of formulas into *red* and *green*. This approach admits a version of Peirce's formula that enables contractions *on all formulas* when the stoup is a green formula. Contractions are enabled *dynamically* during proofs compared to *statically* by the presence of the ? operator in formulas.

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 propose to replace it. Thus, much of this article is devoted to showing that ACL can stand on its own as a logic, with its own notion of model, sequent calculus, cut elimination, soundness and completeness. In particular, a large section is devoted to the cut-elimination procedure as it pertains to the system's structural rules. We also demonstrate the computational significance of this logic by defining a natural deduction system, term calculus and a logical way to *delimit* the capturing of continuations found in classical computations.

2. Syntax

We focus on propositional logic in this presentation. The addition of first-order quantifiers would be a rather standard exercise. Formulas of propositional ACL are freely composed from connectives &, $\oplus$, $\rightarrow$, $\twoheadrightarrow$, $\otimes$ and $\vee$, constants $\top$, 0 and $\perp$, and atomic formulas. The symbol $\twoheadrightarrow$ represents affine implication while $\rightarrow$ represents intuitionistic implication. Intuitionistic disjunction also requires a separate connective: $\vee$ (see Section 5 for detailed explanation). The linear constant 1 is equivalent to $\top$ in affine logic. When a formula is either atomic, 0 or $\perp$, we refer to it as a *literal.*

We use a device similar to *polarization,* but to avoid confusion, we use the term *'colours.'*

Definition 2.1. Formulas are coloured *red* or *green* as follows:[†]

— Atomic formulas are arbitrarily coloured red or green.
— $\perp$ 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.

[†] In our previous system PCL (Liang and Miller 2013b), the colours were called polarities. However, these 'polarities,' are not the same as polarities of the same names in Liang and Miller (2013a). The original semantic motivation for the meaning of $\perp$ in PCL was described in Liang and Miller (2013a) as the *second largest element of a Heyting algebra.* The models of PCL are a sub-class of the models of *PIL* as formulated in Liang and Miller (2013a). However, neither logic is a fragment of the other.

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

— $A \rightarrow B$ is green if B is green, otherwise, it is red.
— $A \rightarrowtriangle B$ 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 $\bot$ and a red 0, there are four versions of negation in ACL. They are abbreviated as follows:

$$-A = A \rightarrowtriangle \bot \qquad \neg A = A \rightarrow \bot \qquad \sim A = A \rightarrow 0 \qquad \frown A = A \rightarrowtriangle 0$$

$\neg A$ and $-A$ are logically equivalent (both are green) but may lead to different proofs. This article primarily uses $-A$ ($\neg A$ was used in Liang and Miller (2013b)).

ACL is claimed to be a unified logic not so much because of the different versions of connectives, but because of the colours. Although we use the symbols $\&$ and $\oplus$ from linear logic, here they can be classical or non-classical. Conceptually, green means classical and red means *arbitrary:* classical or non-classical. The constant $\bot$ in ACL is entirely different from its counterpart in linear logic as will become clear in the next section.

We use the letter E for an arbitrary green formula and e for a green literal (green atom or $\bot$). 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. The colouring of $\&$ and $\oplus$ is similar to the positive/negative polarization of LC (Girard 1991), but the similarity stops here. 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 colours are best explained semantically.

3. Semantics

Red and green represent two levels of provability. The syntactic inference rules that green formulas induce will appear fanciful without a proper explanation from a semantic perspective.

Definition 3.1. A *Phased Frame* is 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 $a \cdot b$ and further require that **$\mathbf{a \preceq b}$ if and only if $\mathbf{ac = b}$ for some c.**

It is important that $\preceq$ is a proper partial order and not just a preorder. Not every commutative monoid gives rise to such a structure: **inverses are not allowed.** This restriction would also be valid for any phase semantics that use a monoid of multisets with multiset-union in their completeness proofs.

Given two sets of worlds A and B, we write $AB = \{xy : x \in A,\ y \in B\}$. It always holds that $ab = ba$ and $(ab)c = a(bc)$. By inference, it also 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:

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

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

Our phased models are closer to those of Okada (2002) than to Girard's original version. The principal difference between our models and those of linear logic is two-fold. 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 the 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.

Definition 3.2. An *Ordered Phase Space* is 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 proper partial order).
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.
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 $\{x \in W \,:\, \text{for all } y \in W\backslash\{r\},\ xy \in W\backslash\{r\}\}$ (which interprets $\top \multimap \top$).

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 $cl(S) \subseteq cl(V)$.

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 $rr = r$.

Definition 3.3. A phase model on an ordered phase space is an interpretation (valuation) of formulas A as facts A^p as follows:

— Red atoms are interpreted by arbitrary facts (elements of D).
— Green atoms are interpreted by either $W\backslash\{r\}$ or to W, i.e., to either $\perp$ or $\top$.
— $\top^p \;=\; W = cl(\{r\})$.
— $\perp^p \;=\; W\backslash\{r\}$.
— $0^p \;=\; cl(\varnothing) = \bigcap D$, the smallest possible fact ($\varnothing$ is the empty set).
— $(A \otimes B)^p \;=\; cl(A^p B^p) = cl(\{xy \,:\, x \in A^p,\ y \in B^p\})$.
— $(A \multimap B)^p \;=\; \{x \in W \,:\, \text{for all } y \in A^p,\ xy \in B^p\}$.
— $(A \rightarrow B)^p \;=\; \{x \in W \,:\, \text{for all } y \in I(A^p),\ xy \in B^p\}$.
— $(A \oplus B)^p \;=\; cl(A^p \cup B^p)$.
— $(A \vee B)^p \;=\; cl(I(A^p) \cup I(B^p))$.
— $(A \,\&\, B)^p \;=\; A^p \cap B^p$.

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

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 easily holds by induction on formulas that *all green formulas valuate to $\bot^p$ or $\top^p$:*

Lemma 3.4. 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. The constant $\top$ can in fact be designated red or green: it makes little difference.

In the following, we will not distinguish between formulas A and their interpretations A^p except when there is possibility for confusion.

Note that showing $r \in A \multimap B$ ($r \in (A \multimap B)^p$) is equivalent to showing that $A \subseteq B$. Clearly, $\bot \multimap E$ is valid for all green E. Since $x \preceq xy$, the upward closure of facts entails the admissibility of weakening ($A \otimes B \multimap A$).

A consequence of interpreting $\bot$ as $W \backslash \{r\}$ is that $\mathbf{A} \oplus -\mathbf{A}$ **is valid**: if $r \notin A$ (A^p), then $A \subseteq \bot$ and therefore $r \in -A$.

A more important consequence is that Peirce's formula $((\mathbf{P} \multimap \mathbf{E}) \rightarrow \mathbf{P}) \multimap \mathbf{P}$ 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 $P \subseteq \bot$ and $\bot \subseteq E$. Then, since $rr = r$, we also have $(P \multimap E) \rightarrow P \subseteq P$.

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 '$u \models A$.'‡

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. Unlike sets of formulas, we cannot construct a Hintikka style saturation for multisets and are forced to accept arbitrary multiset unions. This means that some of these multisets will be inconsistent (derives 0). In the phase semantics of linear logic, 0 is interpreted by $W^\perp$. However, our '$\bot$' has an entirely different meaning than $\bot$ in linear logic. In a model, the smallest

‡ 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 $v \models B$.

possible 0^p is the empty set and the largest possible 0^p is $\perp^p$. Consistency with respect to both 0 and $\perp$ can only be guaranteed at the root, which has the property $rr = r$, (i.e., it is 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. 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 (Ilik et al. 2010; Veldman 1976). Furthermore, because the root world cannot be in 0^p, there is still no model for 0 (or for $\perp$).

The semantics also determine the validity of the following examples:

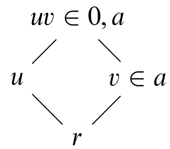
— $(P \mathrel{-\!\triangleright} P \mathrel{-\!\triangleright} E) \mathrel{-\!\triangleright} P \mathrel{-\!\triangleright} E$ (with a green E) is valid. This formula, which is related to Peirce's formula, enables contractions on the *left* despite the affine-linear $\mathrel{-\!\triangleright}$. It will play an important role in our proof theory as explained in Section 4.
— $(A \mathrel{-\!\triangleright} B) \mathrel{-\!\triangleright} (A \rightarrow B)$ is valid. This is equivalent to dereliction ($(A \multimap B) \multimap !A \multimap B$ holds in linear logic). The converse is not valid unless B is green.
— $A \rightarrow E \equiv A \mathrel{-\!\triangleright} E \equiv -A \oplus E$. Green implications collapse into classical disjunction.
— Several other important properties, also found in PCL (Liang and Miller 2013b), should also be noted. These include the fact that *none of the negations* $\neg$, $\sim$, $-$ *and* $\frown$ *are involutive*. In particular, $--A \mathrel{-\!\triangleright} A$ is not valid: our $\perp$ is not the same as the $\perp$ of linear logic. We do have that $\neg\neg E \rightarrow E$ and $--E \mathrel{-\!\triangleright} E$ are valid if E is green. It also holds as an *admissible rule* that if $--A$ is valid, then A is valid. Additionally, the non-intuitionistic De Morgan law $-(A \& B) \mathrel{-\!\triangleright} -A \oplus -B$ is valid.

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

$$\begin{array}{c} qq \in a^p, b^p, c^p \\ | \\ q \in a^p, c^p \\ | \\ r \end{array}$$

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 $r \notin a \mathrel{-\!\triangleright} b$ because $q \in a$ but $rq = q \notin b$. 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 $\perp$ ($\perp^p = \{q, qq\}$), but they do not force d, and thus $r \notin --d \mathrel{-\!\triangleright} d$ 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 colour 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 counter-model 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.

4. Sequent calculus

The core syntactic proof system of ACL is a sequent calculus that serves as a platform for proving cut elimination and completeness, and upon which the correctness of other proof systems of ACL can be based. The core of the sequent calculus are new structural rules centred around Peirce's formula in the form $(-P \to P) \multimap P$. We first motivate these rules informally before presenting the system. Peirce's formula implies the admissibility of contraction on the right-hand side of sequents. What restricts arbitrary contractions is when the contracted 'copies' can be used. We can embed Peirce's formula and its use as a contraction rule with the following kinds of inference rules:

$$\frac{-P, \Gamma \vdash P}{\Gamma \vdash P}\ lock \qquad \frac{\Gamma \vdash P \qquad \bot \vdash E}{-P, \Gamma \vdash E}\ unlock$$

When describing proofs, we usually take the perspective of bottom-up proof construction. The first rule above is valid by Peirce's formula: the copy of P is 'locked' as $-P$ on the left. However, this copy can only be 'unlocked' when the right-hand side contains a green formula E (since $\bot \multimap E$ is valid). To distinguish *unlock* from an ordinary $\multimap$-left introduction rule, we require it to be *focused:* i.e., the right premise of *unlock* is required to be proved by an initial rule ($\bot L$ rule). This small but essential element of focusing is similarly found implicitly in LC and in Jagadeesan et al. (2005).

In a context where both right- and left-side contractions can be restricted, we observe that Peirce's formula implies a counterpart to itself: $(\mathbf{P} \multimap \mathbf{P} \multimap \mathbf{E}) \multimap \mathbf{P} \multimap \mathbf{E}$. The term *counterpart* is appropriate because it justifies *left-side* contractions:

$$\frac{\Gamma, P, P \vdash E}{\Gamma, P \vdash E}\ Pr.$$

Contractions on the left-hand side are not normally permitted in affine logic. However, when the right-side formula is green, this restriction is cancelled. A green formula on the right thus *unlocks contractions left and right*. This rule, which we call 'Pr,' is much stronger than the promotion rule of linear logic. The counterpart formula is semantically valid and it can be derived syntactically using standard sequent calculus introduction rules for

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

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

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

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

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

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

Fig. 1. The sequent calculus of ACL. e is a green atom or $\bot$. a is an atom. $[\Theta : A]$ may only appear to the left of ; in sequents.

implication combined with the lock/unlock rules:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P \vdash P \qquad P \multimap E \vdash P \multimap E}{P \multimap (P \multimap E), P \vdash P \multimap E} \qquad \bot \vdash E}{-(P \multimap E), P \multimap (P \multimap E), P \vdash E}\ unlock}{-(P \multimap E), P \multimap (P \multimap E) \vdash P \multimap E}}{P \multimap (P \multimap E) \vdash P \multimap E}\ lock}{\vdash (P \multimap (P \multimap E)) \multimap P \multimap E}$$

Of course, this formula is also provable if contractions on the left are available: $((P \to P \to E) \to P \to E)$ is intuitionistically provable with $\lambda x \lambda y.xyy$. However, the proof fragment above does not use left-side contraction, but only right-side contraction as implied by Peirce's formula. This formula justifies contractions left and right. One might call it a *self-dual* principle. It replaces the dual exponential operators ? and !. The important lesson from the proof theory of linear logic is not these particular operators, but that without this kind of duality, cut elimination will fail.

The sequent calculus is found in Figure 1. Unlike the informal rules above, we use *dyadic* sequents of the form $\Gamma; \Delta \vdash A$. Here, Δ is a multiset but Γ is a set and A, Γ does not preclude the possibility that $A \in \Gamma$. In contrast A, Δ denotes the multiset union (sum) of $\{A\}$ and Δ (A is added to Δ). The semantic interpretation of a sequent $\Gamma; \Delta \vdash A$ is the same 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 $[\Theta : A]$ may only appear in the set context (Γ). It has the meaning of the formula $-(\Theta^{\otimes} \multimap A)$, but is not itself considered a formula. It cannot appear as a subformula. The special notation distinguishes it for focusing: it can only be principal in *Unlock*. See Corollary 7.1 of Section 7 for the correctness of this focused treatment. We often refer to

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

the single formula on the right as the *stoup*. A formula A is provable if $;\vdash A$ is provable. Weakening holds as an admissible rule (see Lemma 4.1 below).

The *Lock* rule allows contraction not only on the stoup formula, but also on part of the multiset on the left. This is required for cut elimination as will become clear in Section 6. Since we interpret $;\Theta \vdash A$ as $\Theta^{\otimes} \rightarrowtriangle A$, this rule can still be seen as a single right-side contraction, and is semantically sound. Computationally speaking, *Lock* saves not only a copy of the current continuation but also a part of its *operating environment*.

Except for $\bot L$ and $\vee R$, the introduction rules are rather standard for the dyadic representation (e.g., see Hodas and Miller (1994)). The restriction to a green *literal* (green atom or $\bot$) in the *Unlock*, Pr and $\bot L$ rules can in fact be relaxed to any green formula (see Lemma 4.3). The restriction simplifies cut elimination (Section 6). It also imposes a normal form for proofs: contractions, enabled by *Unlock* and Pr, are only allowed after right-introduction rules have been exhausted in bottom-up proof construction. This normal form is similar to focusing, but it is derived from *colours* instead of polarities.

Although *Lock* can be applied at any point, the effect of contraction only appears above *Unlock*. Formulas without green subformulas can only have non-classical proofs.

As sample proofs, we show a version of the excluded middle, $A\oplus -A \;\; (A\oplus(A\rightarrowtriangle\bot))$, and a version of the double-negation axiom, $\sim\!-A\rightarrowtriangle A \;\; (((A\rightarrowtriangle\bot)\to 0)\rightarrowtriangle A)$.

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

The sequent $;A\vdash A$ follows from the Id rule if A is atomic, but this can be generalized to any formula (see Lemma 4.2). 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.

It is possible to derive a left-side contraction using the *Lock* and *Unlock* rules:

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

That is, contraction inside the multiset context also becomes valid when a green e is found in the stoup. However, such derivations cannot replace the Pr rule because of the dyadic representation of sequents (using both sets and multisets). The Pr rule is needed to prove formulas, such as $(A\to -A)\rightarrowtriangle -A$. To demonstrate the use of Pr (and Dr), we

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

prove the equivalence between $\neg A$ $(A \to \bot)$ and $-A$ $(A \mathrel{-\!\triangleright} \bot)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{;A \vdash A}{A; \vdash A}\,Dr \qquad ;\bot \vdash \bot}{A; \neg A \vdash \bot}\,{\to}L}{;\neg A, A \vdash \bot}\,Pr}{;\neg A \vdash -A}\,{-\!\triangleright}R \qquad\qquad \dfrac{\dfrac{\dfrac{;A \vdash A}{A; \vdash A}\,Dr \qquad ;\bot \vdash \bot}{A; -A \vdash \bot}\,{-\!\triangleright}L}{;-A \vdash \neg A}\,{\to}R$$

The Pr rule is needed because of the restriction of empty multiset in the $\to L$ rule. In contrast, $!A \multimap \bot$ and $A \multimap \bot$ are not equivalent in linear logic because the linear $\bot$ has no properties to justify an equivalent Pr, which in terms of linear logic would look like an inverse dereliction:

$$\dfrac{!A, \Delta \vdash \bot}{A, \Delta \vdash \bot}$$

The key difference between green formulas and that of the ? operator (! on the left) 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 literal in the stoup effectively switches the proof into a 'classical mode:' contractions become unlocked on *all* formulas, left and right. This means that we do not have to keep ? on all the formulas that may *at some point* require contraction. The term *dynamic* is appropriate because in proving a formula, such as a red $A \oplus B$ that may contain green subformulas, we do not know before the proof is constructed whether contractions can be made on certain formulas; it depends on whether a green literal is encountered in the stoup. It is the *proof*, as opposed to just the *formula,* that determines if contractions are allowed. Classical reasoning is 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: $((P \mathrel{-\!\triangleright} E) \to P) \mathrel{-\!\triangleright} P$, only E needs to be green whereas in linear logic 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 $P \mathrel{-\!\triangleright} E$ 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.

The presence of a green e in the stoup does not cancel the strength of all non-classical (red) formulas. For example, while $\neg\neg E \to E$ is provable, $\sim\sim E \to E$ $(((E \to 0) \to 0) \to E)$ is not. The constant 0, being red, cannot unlock a contraction on E. It is incorrect to suppose that the entire subproof above a sequent with e in the stoup becomes classical. Once the green e vacates the stoup classical structural rules are no longer available.

4.1. *Basic properties*

We now establish some essential properties of the sequent calculus, all of which are simpler than cut elimination (Section 6). Weakening in the sequent calculus occur in the initial rules and in the $\vee R$ rule. Weakening also holds as an admissible rule for all proofs:

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

Lemma 4.1. If there is a proof of $\Gamma;\Delta\vdash A$, then there is also a proof of $\Gamma'\Gamma;\Delta'\Delta\vdash A$.

This lemma is provable by induction on the height of proofs.

A property that is almost as important as cut elimination is *initial elimination:*

Theorem 4.2. *$\Gamma;\Delta,A\vdash A$ is provable for any formula A.*

Proof. By induction on the formula A. For positive connectives $\oplus$, $\vee$, $\otimes$, apply (from the bottom-up) the left introduction rule followed by the right introduction rule. For the other connectives, apply the right introduction rule first. The only other interesting point about this proof is confirming that the set and multiset contexts are used appropriately by the introduction rules. Thus, the intuitionistic connectives are worth showing:

$$\dfrac{\dfrac{\dfrac{\Gamma;A\vdash A}{A,\Gamma;\vdash A}\,Dr \qquad A,\Gamma;\Delta,B\vdash B}{A,\Gamma;\Delta,A\to B\vdash B}\,\to L}{\Gamma;\Delta,A\to B\vdash A\to B}\,\to R \qquad\qquad \dfrac{\dfrac{\dfrac{\Gamma;A\vdash A}{A,\Gamma;\vdash A}\,Dr}{A,\Gamma;\Delta\vdash A\vee B}\,\vee R \qquad \dfrac{\dfrac{\Gamma;B\vdash B}{B,\Gamma;\vdash B}\,Dr}{B,\Gamma;\Delta\vdash A\vee B}\,\vee R}{\Gamma;\Delta,A\vee B\vdash A\vee B}\,\vee L$$

Dr is the only structural rule used in the proof, and only in these cases. □

Another result, similar to initial elimination, relaxes the restriction to a green literal in the $Unlock$, Pr and $\bot L$ rules:

Lemma 4.3. 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.

Proof. The proof is by simultaneous induction for all rules, on the form of the green formula on the right-hand side, showing that each such unrestricted rule applied to the green formula can be permuted to be on green subformulas. The special case, where the green formula is $\top$ is trivial. Let us refer to the unrestricted versions of these rules as $Unlock'$, Pr' and $\bot L'$. The following representative cases should suffice to convince:

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

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

$$\dfrac{}{\Gamma;\Delta,\bot\vdash E_1\,\&\,E_2}\,\bot L' \quad\longrightarrow\quad \dfrac{\dfrac{}{\Gamma;\Delta\vdash E_1}\,\bot L' \qquad \dfrac{}{\Gamma;\Delta\vdash E_2}\,\bot L'}{\Gamma;\Delta,\bot\vdash E_1\,\&\,E_2}\,\&R$$

In the cases for Pr, the non-literal green formula is first locked (regarding proofs bottom-up), then Pr is applied on green subformulas, and finally the original green formula is recovered via $Unlock$. The cases for $Unlock$ and $\bot L$ always reduce to applying the same rule to green subformulas. □

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

Note that $E_1 \otimes E_2$ is still red, and indeed such a permutation would fail even with both subformulas green.

We chose not to write the sequent calculus with the relaxed rules because, consistent with the atomic Id rule, inference rules in sequent calculi should only depend on the top-level structure of formulas. More significantly perhaps, the relaxed rules would have required the above permutations to be a part of the cut-elimination proof.

We now establish the *substitution property*. First observe:

Proposition 4.4. *If a formula is provable with an atom b coloured red, then the same formula is provable with b coloured green.*

This holds because a green atom can only lead to more proofs, i.e., any inference rule that can be applied with b red can also be applied with b green.

Lemma 4.5. The substitution property for ACL holds as follows:

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

Proof. Substitution of a formula for an atom easily generalizes to sequents. The proof generalizes the statement to be for sequents, then both parts are proved by induction on the structure of proofs. Part 1 uses initial elimination (Lemma 4.2). In fact, it is fairly obvious that the only part of any proof that needs to be changed is the Id rule, which is the only rule that requires its principal formula to be an arbitrary atom. Part 2 uses initial elimination and Lemma 4.3 to modify the $\bot L$, *Unlock* and Pr rules. □

This lemma means that red atoms represent arbitrary formulas. ACL can be extended to a *second order* propositional logic, where *universally quantified propositional variables are coloured red, while existentially quantified variables are green.* One proves $\forall A.A \to A$ and $\exists E.\neg\neg E \to E$. A detailed exposition of this extension is left to future work.

5. Fragments of ACL

The subformula property of the (cut-free) sequent calculus means that a fragment of ACL can be defined by simply restricting the use of connectives, constants and the colours of atoms. Some of the more significant fragments are as follows:

— *Intuitionistic Logic*: Colour all atoms red and restrict to $\&, \vee, \to, 0$. All formulas are red. $\top$ can be replaced by $0 \to 0$ (or simply be considered red).
  The completeness of this fragment with respect to intuitionistic logic is verified by proving each of the propositional intuitionistic axioms (which can be found in Moschovakis (2015)). The rule of *Modus Ponens* follows from cut elimination (proved in Section 6). The soundness of this fragment with respect to intuitionistic logic follows from two observations. First, all *Lock* rules are useless because there is no *Unlock* without green formulas. In particular, the disjunction property holds for $A \vee B$ when A and B do not contain green subformulas. Second, although the Dr rule is used, it

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

does not affect soundness with respect to intuitionistic logic, in which contractions are always valid on the left. In terms of intuitionistic sequent calculus, Dr is a null operation. Pr cannot appear. All other inference rules that can appear in cut-free proofs of this fragment are admissible in intuitionistic sequent calculus.

— *Classical Logic*: Colour all atoms green and restrict to $\&, \oplus, \rightarrow, \bot$ and $\top$. All formulas are green and the restrictions on $Unlock$, Pr and $\bot L$ become meaningless (given Lemma 4.3). Classical axioms including $\neg\neg E \rightarrow E$ are provable. There is no need for $\vee$, although the introduction rules of all connectives are sound with respect to classical logic.

— *Intuitionistic Linear Affine Logic:* Colour all atoms red and restrict to $\&, \oplus, \otimes, \multimap, \vee, \rightarrow, 0$. Formulas $!A$ can be emulated by $A \vee 0$.

The subformula property also means that cut elimination is preserved inside each fragment. The fact that each of the above logics can be identified as a self-contained fragment justifies ACL as a unified logic. However, ACL is more than the sum of these parts. Any set of restrictions on formulas defines a new logical system, including:

— *Purely Negative Fragment:* Restrict to $\&, \rightarrow, \multimap, \bot$ and $\top$. The semantic interpretations of these connectives do not require the closure operator cl. They can be given a Kripke style semantics.

— *Affine-Linear Logic plus Classical Logic*: Do not use $\rightarrow$ and $\vee$.

— *Non-Linear Fragment*: Do not use $\multimap$ and $\otimes$.

5.1. *Notes concerning intuitionistic disjunction and conjunction*

The connective $\vee$ is included in ACL for the sake of intuitionistic completeness *without a classical collapse*. With $\oplus$, all propositional intuitionistic axioms are provable *except for* $(A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow (A \oplus B) \rightarrow C$, where C is red. A green C would collapse $\rightarrow$ into classical implication (thus the classical fragment can use $\oplus$). However, including $\vee$ as a connective has other 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 in fact $A \rightarrow B \equiv (A \vee 0) \multimap B$. $A \vee \neg A$ is also provable: technically this formula is red because $\bot \multimap (A \vee \neg A)$ is not provable. However, $\bot \rightarrow (A \vee \neg A)$ is provable, which gives this formula some of the properties of green formulas. Technically, our previous effort in combining intuitionistic and classical logics, PCL, is not a fragment of ACL. PCL contains $\vee$ but not $\oplus, \multimap$ or $\otimes$. However, PCL models are equivalent to purely intuitionistic models, where $I(W) = W$, thus the difference between $\oplus$ and $\vee$ disappears.

The connective $\&$ is used in both the classical and intuitionistic fragments. However, this usage imposes certain constraints on proofs. For example, in a proof of $a \rightarrow c, a \& b; \vdash c$ (with c red) the introduction of $a \& b$ must be above that of $a \rightarrow c$ because of the restriction in the $\rightarrow L$ rule. It is possible to define a *purely intuitionistic conjunction* $\wedge$ with semantic

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

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

Fig. 2. The cut rules of ACL.

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 colour 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. Specifically, $A\& B;\vdash A\wedge B$ and $A\wedge B;\vdash A\& B$ are both provable. In this setting, 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 (asynchronous on the right) and $\wedge$ is positive (asynchronous on the left).

6. Cut elimination

The importance of the cut-elimination proof is more than just showing the admissibility of cuts, for it establishes ACL as a new logical system that requires its own, rather unique proof theory. The principal cut rules of ACL are found in Figure 2. The rules cut_1 and cut_2 are distinguished by the set or multiset context in which the cut formula appears on the left-hand side. These rules are enough to prove completeness and the usual properties expected from cut elimination. However, the proof itself is for the generalized rules found in Figure 3. These rules are called *mix* (Gentzen 1935) as they can remove multiple copies of the cut formula. The rules marked *dmix* represent *delayed cuts:* the cut formula is locked inside some $[\Theta:B]$ and cannot be removed until unlocked. Two of these rules are also *multicuts* as they have three subproofs (with the middle subproof delineated notationally by braces): they allow us to describe how cuts are permuted above multiple as opposed to individual inference rules. The abbreviated notation $[\Theta_i:A]_{i=1}^n,\Gamma;\Delta\vdash A$ indicates *zero or more* instances of $[\Theta_i:A]$ in Γ. The notation also does not preclude the possibility of other such elements in Γ, with or without copies of the cut formula. Thus, the mix_1 rule subsumes cut_1. Since $A^?$ represents the optional presence of A, mix_2 subsumes cut_2. We still refer to these rules as *'cut rules'* and the formula being removed is still the *cut formula*. We always display the subproof above a cut with the cut formula on the right-hand side of $\vdash$ as the left-most subproof. The subproof with the cut formula on the left is always displayed as the right-most subproof.

Conceptually speaking, structural reduction, as described by Parigot, is difficult to define in sequent calculus in a manner precise enough to prove termination. Cut elimination for sequent calculi tend to describe reductions *locally,* as the permutation of cuts above single inference rules. This misses the correlation between *Lock* and *Unlock*, which spans across multiple rules. For example, consider an attempted generalization of cut_2:

$$\frac{[\Theta:A],\Gamma;\vdash A \qquad A,\Gamma';\Delta'\vdash B}{[\Theta\Delta':B],\Gamma\Gamma';\Delta'\vdash B}\ cut_2'$$

Such a cut is not generally admissible, with any reasonable conclusion, if Θ is an arbitrary, non-empty multiset. In writing such a rule, we are disregarding that, in subproofs of actual *proofs* (of

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

$$\frac{[:A]^?, \Gamma; \Delta \vdash A \qquad [\Theta'_j, A\!:\!D_j]_{j=1}^m, \Gamma'; \Delta', A \vdash C}{[\Delta\Theta'_j\!:\!D_j]_{j=1}^m, \Gamma\Gamma'; \Delta\Delta' \vdash C}\; mix_1$$

$$\frac{\dfrac{}{[:A]^?, \Gamma; \Delta \vdash A}\,(*) \qquad \{[\Theta_i\!:\!A]_{i=1}^n, \Gamma''; \Delta'' \vdash A\} \qquad \dfrac{}{[\Theta'_j, A\!:\!D_j]_{j=1}^m, \Gamma'; \Delta', A \vdash C}\,(*L_A)}{[\Theta_i\Delta'\!:\!C]_{i=1}^n [\Delta\Theta'_j\!:\!D_j]_{j=1}^m, \Gamma\Gamma'\Gamma''; \Delta''\Delta' \vdash C}\; smix_1$$

$$\frac{\dfrac{}{[:A]^?, \Gamma; \Delta \vdash A}\,(*) \qquad \{[\Theta_i\!:\!A]_{i=1}^n, \Gamma''; \Delta'' \vdash B\} \qquad \dfrac{}{[\Theta'_j, A\!:\!D_j]_{j=1}^m, \Gamma'; \Delta', A \vdash C}\,(*L_A)}{[\Theta_i\Delta'\!:\!C]_{i=1}^n [\Delta\Theta'_j\!:\!D_j]_{j=1}^m, \Gamma\Gamma'\Gamma''; \Delta'' \vdash B}\; dmix_1$$

$$\frac{[:A]^?, \Gamma; \Delta \vdash A \qquad [\Theta'_j, A\!:\!D_j]_{j=1}^m, \Gamma'; \Delta' \vdash C}{[\Delta\Theta'_j\!:\!D_j]_{j=1}^m, \Gamma\Gamma'; \Delta' \vdash C}\; dmix^\ell$$

$$\frac{[:A], \Gamma; \Delta \vdash B \qquad [\Theta'_j, A^+\!:\!D_j]_{j=1}^m, A^?, \Gamma'; \Delta', A^* \vdash C}{[\Delta'\!:\!C], [\Theta'_j\!:\!D_j]_{j=1}^m, \Gamma\Gamma'; \Delta \vdash B}\; dmix^r$$

$$\frac{[:A]^?, \Gamma; \vdash A \qquad [\Theta'_j, A^+\!:\!D_j]_{j=1}^m, A^?, \Gamma'; \Delta', A^* \vdash C}{[\Theta'_j\!:\!D_j]_{j=1}^m, \Gamma\Gamma'; \Delta' \vdash C}\; mix_2$$

$(*)$ = $Lock$, $Unlock$, Pr or right-introduction rule; $(*L_A)$ = left-introduction rule on A;
$.^*$ = arbitrary occurrences, $.^+$ = at least one occurrence. $^n = n$ occurrences, $n, m \geq 0$.
$[\Delta\Theta_i : B_i]_{i=1}^m = [\Delta\Theta_1 : B_1] \ldots [\Delta\Theta_m : B_m], \quad [\Delta\Theta_j : A]_{j=1}^m = [\Delta\Theta_1 : A] \ldots [\Delta\Theta_m : A]$

Fig. 3. Additional cut rules of ACL.

formulas), Θ is not arbitrary but must correspond to a *Lock*. This is the dynamic captured by the multicuts *smix*$_1$ and *dmix*$_1$: the *middle subproof* of these cuts can have [Θ : *A*] with non-empty Θ because they correspond to *Lock* rules in the leftmost subproof. The restrictions to the cuts of Figure 3 are necessary to define a delicate permutation strategy. To be clear, the restrictions do not mean that there cannot be locked formulas in sequents (that conclude subproofs) subject to cut, nor that the cut formula cannot be subject to *Lock* and *Unlock* rules in the subproofs above the cut. The cuts we show to be admissible are more than enough to prove the properties of Section 7, in particular Corollary 7.1, which shows that locked formulas and their focused treatment can be replaced by ordinary formulas and introduction rules.

The cut-elimination proof uses the following properties and definitions.

Definition 6.1. Given an inference rule (excluding cuts), the formula *affected* by the rule is the principal formula of an introduction rule, the formula moved between the set and multiset by *Dr* or *Pr*, a formula inside [Θ :*A*] (either in Θ or *A*) subject to a *Lock* or *Unlock*, both instances of the atomic formula in *Id*, any formula in the multiset context in the conclusion of ∨*R*, and the green literal in the stoup in *Unlock*, *Pr* and ⊥*L*.

When a formula is not affected by an inference rule, it is *parametric* to the rule.

Since the cut (mix) rules select which instances of the cut formula to remove, it is important to define what we mean by the occurrences of a formula in a proof.

Definition 6.2. Given a proof ending in a sequent $\Gamma;\Delta \vdash B$ and either a formula A in Γ or an instance of a formula A in Δ, B, the *occurrences of this formula in the proof* is traced from the sequent as follows. Attach an index to the formula: $^{x}\!A$. If the inference rule above the sequent does not affect $^{x}\!A$, then the index stays attached in the sequents above the rule. If the inference rule affecting $^{x}\!A$ is Pr, Dr, $Lock$ or $Unlock$, then all copies of $^{x}\!A$ created by the rule will be annotated with the same index. The index is discarded along with the formula when it is subject to weakening or becomes the principal formula of an introduction rule. *Furthermore, in the sequents immediately above a cut, all instances of the formula to be cut are given the same index.*

The index does not define a new kind of formula but is a proof annotation, a proof term. It will not be shown in subsequent proofs to avoid notational clutter. However, this assumed device will allow us to refer unambiguously to, for example, *the number of Unlock rules that affect the cut formula in a proof.*

In typical cut-elimination proofs for sequent calculi, permuting cuts above introduction rules that are parametric to the cut formula are rather straightforward (as they are for cut_1 and cut_2). However, the mix rules also change the contents of locked formulas in $[\Theta : A]$. The permutation of the mix rules above the introduction rules of Figure 1 becomes difficult. We solve this problem by deriving several admissible transformations on cut-free proofs. For convenience, we use the following notation.

$$\frac{\Gamma;\Delta \vdash A}{\Gamma;\Delta, B \vdash A}\ (W^{\bullet}).$$

The notation does not represent a new inference rule. The intended meaning is that the transformation (in this case weakening) is applied to the cut-free proof above the line, resulting in the cut-free proof with the conclusion below the line. Such transformations are never applied above cuts, but they may be required beneath cuts that have been permuted above introduction rules. The validity of the following transformations are easily observed:

Proposition 6.3. *The following transformations on Unlock are valid*

1.
$$\frac{\Gamma;\Delta\Theta \vdash B}{[\Theta : B], \Gamma;\Delta \vdash e}\ \mathit{Unlock} \quad \longrightarrow \quad \dfrac{\dfrac{\Gamma;\Delta\Theta \vdash B}{\Gamma;\Delta\Theta \vdash B \oplus C}\ \oplus R}{[\Theta : B \oplus C], \Gamma;\Delta \vdash e}\ \mathit{Unlock}$$

2.
$$\frac{\Gamma;\Delta\Theta, A \vdash B}{[\Theta, A : B], \Gamma;\Delta \vdash e}\ \mathit{Unlock} \quad \longrightarrow \quad \dfrac{\dfrac{\Gamma;\Delta\Theta, A \vdash B}{\Gamma;\Delta\Theta \vdash A \rightarrowtriangle B}\ \rightarrowtriangle R}{[\Theta : A \rightarrowtriangle B], \Gamma;\Delta \vdash e}\ \mathit{Unlock}$$

3.
$$\frac{A, \Gamma;\Delta\Theta \vdash B}{[\Theta : B], A, \Gamma;\Delta \vdash e}\ \mathit{Unlock} \quad \longrightarrow \quad \dfrac{\dfrac{A, \Gamma;\Delta\Theta \vdash B}{\Gamma;\Delta\Theta \vdash A \rightarrow B}\ \rightarrow R}{[\Theta : A \rightarrow B], \Gamma;\Delta \vdash e}\ \mathit{Unlock}$$

4. *If* $\Gamma;\Delta' \vdash C$ *is provable*
$$\frac{\Gamma;\Delta\Theta \vdash B}{[\Theta : B], \Gamma;\Delta \vdash e}\ \mathit{Unlock} \quad \longrightarrow \quad \dfrac{\dfrac{\dfrac{\Gamma;\Delta\Theta \vdash B}{\Gamma;\Delta\Delta'\Theta \vdash B}\ (W^{\bullet}) \quad \dfrac{\Gamma;\Delta' \vdash C}{\Gamma;\Delta\Delta'\Theta \vdash C}\ (W^{\bullet})}{\Gamma;\Delta\Delta'\Theta \vdash B \,\&\, C}\ \&R}{[\Theta\Delta' : B \,\&\, C], \Gamma;\Delta \vdash e}\ \mathit{Unlock}$$

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

5. *If* $\Gamma;\Delta_2\vdash C$ *is provable*

$$\dfrac{\Gamma;\Delta\Theta\vdash B}{[\Theta:B],\Gamma;\Delta\vdash e}\,\mathit{Unlock}\quad\longrightarrow\quad\dfrac{\dfrac{\Gamma;\Delta\Theta\vdash B\quad\Gamma;\Delta_2\vdash C}{\Gamma;\Delta\Delta_2\Theta\vdash B\otimes C}\,\otimes R}{[\Theta\Delta_2:B\otimes C],\Gamma;\Delta\vdash e}\,\mathit{Unlock}$$

6. *If* $\Gamma;\Delta_2,C\vdash D$ *and* $\Gamma;\Delta_1\vdash B$ *are provable*

$$\dfrac{\Gamma;\Delta\Theta\vdash B}{[\Theta:B],\Gamma;\Delta\vdash e}\,\mathit{Unlock}\quad\longrightarrow\quad\dfrac{\dfrac{\Gamma;\Delta\Theta\vdash B\quad\Gamma;\Delta_2,C\vdash D}{\Gamma;\Theta\Delta\Delta_2,B\rightarrowtriangle C\vdash D}\,\rightarrowtriangle L}{[\Theta\Delta_2,B\rightarrowtriangle C:D],\Gamma;\Delta\vdash e}\,\mathit{Unlock}$$

$$\dfrac{\Gamma;\Delta\Theta,C\vdash D}{[\Theta,C:D],\Gamma;\Delta\vdash e}\,\mathit{Unlock}\quad\longrightarrow\quad\dfrac{\dfrac{\Gamma;\Delta_1\vdash B\quad\Gamma;\Theta\Delta,C\vdash D}{\Gamma;\Theta\Delta\Delta_1,B\rightarrowtriangle C\vdash D}\,\rightarrowtriangle L}{[\Theta\Delta_1,B\rightarrowtriangle C:D],\Gamma;\Delta\vdash e}\,\mathit{Unlock}$$

7. *If* $\Gamma;\vdash B$ *and* $\Gamma;\Delta',C\vdash D$ *are provable*

$$\dfrac{\Gamma;\Delta\Theta\vdash B}{[\Theta:B],\Gamma;\Delta\vdash e}\,\mathit{Unlock}\quad\longrightarrow\quad\dfrac{\dfrac{\Gamma;\vdash B\quad\dfrac{\Gamma;\Delta',C\vdash D}{\Gamma;\Delta\Theta\Delta',C\vdash D}\,(W^\bullet)}{\Gamma;\Delta\Theta\Delta',B\to C\vdash D}\,\to L}{[\Theta\Delta',B\to C:D],\Gamma;\Delta\vdash e}\,\mathit{Unlock}$$

$$\dfrac{\Gamma;\Delta\Theta,C\vdash D}{[\Theta,C:D],\Gamma;\Delta\vdash e}\,\mathit{Unlock}\quad\longrightarrow\quad\dfrac{\dfrac{\Gamma;\vdash B\quad\dfrac{\Gamma;\Delta',C\vdash D}{\Gamma;\Theta\Delta\Delta',C\vdash D}\,(W^\bullet)}{\Gamma;\Theta\Delta\Delta',B\to C\vdash D}\,\to L}{[\Theta\Delta',B\to C:D],\Gamma;\Delta\vdash e}\,\mathit{Unlock}$$

8.

$$\dfrac{\Gamma;\Delta\Theta,C\vdash D}{[\Theta,C:D],\Gamma;\Delta\vdash e}\,\mathit{Unlock}\quad\longrightarrow\quad\dfrac{\dfrac{\Gamma;\Delta\Theta,C\vdash D}{\Gamma;\Delta\Theta,B\,\&\,C\vdash D}\,\&L}{[\Theta,B\,\&\,C:D],\Gamma;\Delta\vdash e}\,\mathit{Unlock}$$

9.

$$\dfrac{\Gamma;\Delta\Theta,B,C\vdash D}{[\Theta,B,C:D],\Gamma;\Delta\vdash e}\,\mathit{Unlock}\quad\longrightarrow\quad\dfrac{\dfrac{\Gamma;\Delta\Theta,B,C\vdash D}{\Gamma;\Delta\Theta,B\otimes C\vdash D}\,\otimes L}{[\Theta,B\otimes C:D],\Gamma;\Delta\vdash e}\,\mathit{Unlock}$$

10. *If* $\Gamma;\Delta',B\vdash D$ *is provable*

$$\dfrac{\Gamma;\Delta\Theta,C\vdash D}{[\Theta,C:D],\Gamma;\Delta\vdash e}\,\mathit{Unlock}\quad\longrightarrow\quad\dfrac{\dfrac{\dfrac{\Gamma;\Delta',B\vdash D}{\Gamma;\Delta\Theta\Delta',B\vdash D}\,(W^\bullet)\quad\dfrac{\Gamma;\Delta\Theta,C\vdash D}{\Gamma;\Delta\Theta\Delta',C\vdash D}\,(W^\bullet)}{\Gamma;\Delta\Theta\Delta',B\oplus C\vdash D}\,\oplus L}{[\Theta\Delta',B\oplus C:D],\Gamma;\Delta\vdash e}\,\mathit{Unlock}$$

11. *Weakening is also valid on locked formulas*

$$\dfrac{\Gamma;\Delta\Theta\vdash B}{[\Theta:B],\Gamma;\Delta\vdash e}\,\mathit{Unlock}\quad\longrightarrow\quad\dfrac{\dfrac{\Gamma;\Delta\Theta\vdash B}{\Gamma;\Delta\Theta\Theta'\vdash B}\,(W^\bullet)}{[\Theta\Theta':B],\Gamma;\Delta\vdash e}\,\mathit{Unlock}$$

12. *Redundant locked formulas can be removed.*

$$\dfrac{A,\Gamma;\Delta\Theta,A\vdash D}{[\Theta,A:D],A,\Gamma;\Delta\vdash e}\,\mathit{Unlock}\quad\longrightarrow\quad\dfrac{\dfrac{A,\Gamma;\Delta\Theta,A\vdash D}{A,\Gamma;\Delta\Theta\vdash D}\,Dr^*}{[\Theta:D],A,\Gamma;\Delta\vdash e}\,\mathit{Unlock}$$

Locked formulas inside $[\Theta:A]$ have no effect on proofs beneath *Unlock*, and thus transforming each *Unlock* instance results in modified proofs. Each transformation can be applied repeatedly

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

$$\frac{[\Theta_i, A\!:\!B_i]_{i=1}^n, A, \Gamma; \Delta \vdash C}{[\Theta_i\!:\!B_i]_{i=1}^n, A, \Gamma; \Delta \vdash C}\,(Dr^\bullet) \qquad \frac{[\Theta_i\!:\!B_i]_{i=1}^n, \Gamma; \Delta \vdash A}{[\Theta_i\Theta_i'\!:\!B_i]_{i=1}^n, \Gamma; \Delta \vdash A}\,(W_{[]}^\bullet)$$

$$\frac{[\Theta_i\!:\!B_k]_{i=1}^n, \Gamma; \Delta \vdash B_k\ ,\ {\scriptstyle k\in\{1,2\}}}{[\Theta_i\!:\!B_1 \oplus B_2]_{i=1}^n, \Gamma; \Delta \vdash B_1 \oplus B_2}\,(\oplus R^\bullet) \quad \frac{[\Theta_i\!:\!A]_{i=1}^n, \Gamma; \Delta \vdash A \quad [\Theta_i\!:\!B]_{i=1}^n, \Gamma; \Delta \vdash B}{[\Theta_i\!:\!A \,\&\, B]_{i=1}^n, \Gamma; \Delta \vdash A \,\&\, B}\,(\&R^\bullet)$$

$$\frac{[\Theta_i, A\!:\!B]_{i=1}^n, \Gamma; \Delta, A \vdash B}{[\Theta_i\!:\!A \rightarrowtriangle B]_{i=1}^n, \Gamma; \Delta \vdash A \rightarrowtriangle B}\,(\rightarrowtriangle R^\bullet) \quad \frac{[\Theta_i\Delta_1\!:\!A]_{i=1}^n, \Gamma; \Delta_1 \vdash A \quad \Gamma; \Delta_2 \vdash B}{[\Theta_i\Delta_1\Delta_2\!:\!A \otimes B]_{i=1}^n, \Gamma; \Delta_1\Delta_2 \vdash A \otimes B}\,(\otimes R_1^\bullet)$$

$$\frac{[\Theta_i\!:\!B]_{i=1}^n, A, \Gamma; \Delta \vdash B}{[\Theta_i\!:\!A \to B]_{i=1}^n, \Gamma; \Delta \vdash A \to B}\,(\to R^\bullet) \quad \frac{\Gamma; \Delta_1 \vdash A \quad [\Theta_i\Delta_2\!:\!B]_{i=1}^n, \Gamma; \Delta_2 \vdash B}{[\Theta_i\Delta_1\Delta_2\!:\!A \otimes B]_{i=1}^n, \Gamma; \Delta_1\Delta_2 \vdash A \otimes B}\,(\otimes R_2^\bullet)$$

$$\frac{[\Theta_i, B_i\!:\!C]_{i=1}^n, \Gamma; \Delta, B_i \vdash C}{[\Theta_i, B_1 \,\&\, B_2\!:\!C]_{i=1}^n, \Gamma; \Delta, B_1 \,\&\, B_2 \vdash C}\,(\&L^\bullet) \quad \frac{[\Theta_i\!:\!A]_{i=1}^n, \Gamma; \vdash A \qquad \Gamma; \Delta', B \vdash C}{[\Theta_i\Delta', A \rightarrowtriangle B\!:\!C]_{i=1}^n, \Gamma; \Delta', A \rightarrowtriangle B \vdash C}\,(\to L_1^\bullet)$$

$$\frac{[\Theta_i, A, B\!:\!C]_{i=1}^n, \Gamma; \Delta, A, B \vdash C}{[\Theta_i, A \otimes B\!:\!C]_{i=1}^n, \Gamma; \Delta, A \otimes B \vdash C}\,(\otimes L^\bullet) \quad \frac{\Gamma; \vdash A \quad [\Theta_i\Delta', B\!:\!C]_{i=1}^n, \Gamma; \Delta', B \vdash C}{[\Theta_i\Delta', A \to B\!:\!C]_{i=1}^n, \Gamma; \Delta', A \to B \vdash C}\,(\to L_2^\bullet)$$

$$\frac{[\Theta_i\Delta_1\!:\!A]_{i=1}^n, \Gamma; \Delta_1 \vdash A \qquad \Gamma; \Delta_2, B \vdash C}{[\Theta_i\Delta_1\Delta_2, A \rightarrowtriangle B\!:\!C]_{i=1}^n, \Gamma; \Delta_1\Delta_2, A \rightarrowtriangle B \vdash C}\,(\rightarrowtriangle L_1^\bullet)$$

$$\frac{\Gamma; \Delta_1 \vdash A \qquad [\Theta_i\Delta_2, B\!:\!C]_{i=1}^n, \Gamma; \Delta_2, B \vdash C}{[\Theta_i\Delta_1\Delta_2, A \rightarrowtriangle B\!:\!C]_{i=1}^n, \Gamma; \Delta_1\Delta_2, A \rightarrowtriangle B \vdash C}\,(\rightarrowtriangle L_2^\bullet)$$

Fig. 4. Transformations on cut-free proofs. Each proof of a sequent above a labelled rule can be transformed to a proof of the sequent beneath that rule.

to locked formulas for which it is applicable. Figure 4 in turn defines a set of admissible transformations on cut-free proofs that correspond to the introduction rules of Figure 1, plus some other transformations that will be useful.

Lemma 6.4. The transformations on cut-free proofs of Figure 4 are admissible.

Proof. By induction on the structure of proofs, applying the appropriate transformation of Proposition 6.3 to each instance of *Unlock*. For example, transformation 6.3, along with the corresponding $\&R$ rule of Figure 1, is used to derive ($\&R^\bullet$). □

The $\vee L$ and $\vee R$ rules do not require transformation as they affect the multiset differently.

To be clear, the cut-elimination theorem is proved for the original rules of Figure 1. The transformations are only applied to cut-free proofs. To simplify the main proof, the following properties correspond to cases where the cut formula is a constant

Lemma 6.5.

1. If $\Gamma;\Delta \vdash \bot$ is provable, then $\Gamma;\Delta \vdash e$ is also provable for any green literal e.
2. If $\Gamma;\Delta \vdash 0$ is provable, then $\Gamma;\Delta \vdash A$ is also provable for any formula A.
3. If $\top, \Gamma;\Delta \vdash A$ or $\Gamma;\Delta, \top \vdash A$ is provable, then $\Gamma;\Delta \vdash A$ is provable.

Each of these properties is proved by induction on the height of proofs.

The overall strategy of the cut-elimination proof is similar to other such proofs in that there is a double induction: first we show that all cuts with cut-free subproofs (i.e., proofs where only the final rule is a cut) can be eliminated. Then, we show that, starting from the topmost cuts, all cuts can be eliminated from a proof. The following inductive measure is used in the first of these results.

Definition 6.6. The measure of a proof that ends in a cut rule of Figure 3 is the lexicographical ordering on the tuple $(\mathcal{S}, \mathcal{U}, \mathcal{P}, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3)$, consisting of, in order of precedence:

1. $\mathcal{S}$: the size of the cut formula.
2. $\mathcal{U}$: the number of *Unlock* rules that affect the cut formula in the leftmost and rightmost subproofs above the cut (mix).
3. $\mathcal{P}$: the number of *Pr* rules that affect the cut formula in the leftmost and rightmost subproofs above the cut.
4. $\mathcal{H}_1$: the height of the leftmost subproof above the cut rule with the cut formula on the right-hand side of the final sequent.
5. $\mathcal{H}_2$: the height of the rightmost subproof above the cut with the cut formula on the left-hand side of the final sequent.
6. $\mathcal{H}_3$: for $smix_1$ and $dmix_1$, this value is the height of the middle subproof; for other cuts, this value is $\mathcal{H}_1 + 1$.

Lemma 6.7. All cuts with cut-free subproofs are admissible.

Proof. By induction, simultaneously, for the mix rules of Figure 3 (which subsume cut_1 and cut_2) using the inductive measure of Definition 6.6. Cuts are permuted above inference rules until the cut formula is principal on both left and right, reaching a 'key case:' it is only here that the cut formula is removed, replaced by cuts on its subformulas. If no key case is reached the cut is eliminated by weakening. The cases of literal cut formulas follow from Lemma 6.5. Although this strategy is similar to other cut-elimination proofs, the details of how to reach the key cases are rather unique to ACL.

Permutations of mix$_1$: The algorithm for permuting mix_1 is as follows:

1. Permute the cut above inference rules in the left subproof until the left subproof ends in an initial rule, *Lock*, *Unlock*, *Pr*, or right-introduction rule on the cut formula. The only exception is a *Lock* on just the stoup (creating $[:A]$).
2. Permute the cut above inference rules in the right subproof until an initial rule, or when the cut formula is the principal formula of a left-introduction rule.
3. If no initial rule is reached by the above permutations, permute mix_1 to a $smix_1$.

Now in more detail

Stage 1: First, permuting mix_1 over an *Unlock* or *Pr* rule that ends the left subproof is not possible, nor necessary:

$$\dfrac{\dfrac{[\Theta : B], \Gamma; \Delta\Theta \vdash B}{[\Theta : B], \Gamma; \Delta \vdash e}\ Unlock \qquad [\Theta'_j, e : B_j]_{j=1}^{m}, \Gamma'; \Delta', e \vdash C}{[\Theta : B], [\Delta\Theta'_j : B_j]_{j=1}^{m}, \Gamma\Gamma'; \Delta'\Delta \vdash C}\ mix_1$$

Even if $B = e$, the cut cannot be permuted over this *Unlock* because C *can be red,* which would prevent *Unlock* from being replicated beneath the cut. The cut must be permuted into the right subproof. Since e is a literal, the permutation will end at an initial rule in the right subproof (with e on the left of $\vdash$): here the cut is eliminated by weakening. The same applies to *Pr* and to $\bot L$. Permuting mix_1 over a *Lock* that only creates $[:A]$ is trivial , as the conclusion of mix_1 is not affected by the presence of $[:A]$. Permuting mix_1 above *Dr* that ends the left subproof is as follows:

$$\dfrac{\dfrac{[:A]^?, B, \Gamma; \Delta, B \vdash A}{[:A]^?, B, \Gamma; \Delta \vdash A}\ Dr \qquad [\Theta'_j, A : D_j]_{j=1}^{m}, \Gamma'; \Delta', A \vdash C}{[\Delta\Theta'_j : D_j]_{j=1}^{m}, B, \Gamma\Gamma'; \Delta\Delta' \vdash C}\ mix_1$$

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

$$\downarrow$$

$$\dfrac{\dfrac{\dfrac{[:A]^?, B, \Gamma; \Delta, B \vdash A \qquad [\Theta'_j, A : D_j]_{j=1}^m, \Gamma'; \Delta', A \vdash C}{[B, \Delta\Theta'_j : D_j]_{j=1}^m, B, \Gamma\Gamma'; \Delta\Delta', B \vdash C}\; mix_1}{[\Delta\Theta'_j : D_j]_{j=1}^m, B, \Gamma\Gamma'; \Delta\Delta', B \vdash C}\; (Dr^\bullet)}{[\Delta\Theta'_j : D_j]_{j=1}^m, B, \Gamma\Gamma'; \Delta\Delta' \vdash C}\; Dr$$

The inductive measure is reduced by $\mathcal{H}_1$. The $Dr^\bullet$ transformation is applied to the cut-free proof obtained from the inductive hypothesis and its effect on the structure of proofs is immaterial to the argument. In displaying permutations of cut, we explicitly repeat formulas in the set context (B in this case) in the premises of rules that affect them, so as to always take into account the possibility of contraction, which the set enables.

Other permutations into the left subproof are over left-introduction rules that do not affect the cut formula. The most difficult case is

$$\dfrac{\dfrac{[:A], \Gamma; \Delta_1 \vdash B \qquad [:A], \Gamma; \Delta_2, C \vdash A}{[:A], \Gamma; \Delta_1\Delta_2, B \rightarrowtriangle C \vdash A}\; \rightarrowtriangle L \qquad [\Theta'_j, A : D_j]_{j=1}^m, \Gamma'; \Delta', A \vdash D}{[\Theta'_j\Delta_1\Delta_2, B \rightarrowtriangle C : D_j]_{j=1}^m, \Gamma\Gamma'; \Delta_1\Delta_2\Delta', B \rightarrowtriangle C \vdash D}\; mix_1$$

Form subproofs:

$$\dfrac{[:A], \Gamma; \Delta_1 \vdash B \qquad [\Theta'_j, A : D_j]_{j=1}^m, \Gamma'; \Delta', A \vdash D}{[\Delta' : D], [\Theta'_j : D_j]_{j=1}^m, \Gamma\Gamma'; \Delta_1 \vdash B}\; dmix^r$$

$$\dfrac{[:A], \Gamma; \Delta_2, C \vdash A \qquad [\Theta'_j, A : D_j]_{j=1}^m, \Gamma'; \Delta', A \vdash D}{[\Theta'_j\Delta_2, C : D_j]_{j=1}^m, \Gamma\Gamma'; \Delta_2\Delta', C \vdash D}\; mix_1$$

The inductive measures of these cuts are reduced by $\mathcal{H}_1$. Now compose,

$$\dfrac{\dfrac{\dfrac{[\Delta' : D], [\Theta'_j : D_j]_{j=1}^m, \Gamma\Gamma'; \Delta_1 \vdash B \qquad [\Theta'_j\Delta_2, C : D_j]_{j=1}^m, \Gamma\Gamma'; \Delta_2\Delta', C \vdash D}{[\Delta' : D], [\Theta'_j : D_j]_{j=1}^m [\Theta'_j\Delta_1\Delta_2, B \rightarrowtriangle C : D_j]_{j=1}^m, \Gamma\Gamma'; \Delta_1\Delta_2\Delta', B \rightarrowtriangle C \vdash D}\; (\rightarrowtriangle L_2^\bullet)}{[\Delta' : D], [\Theta'_j\Delta_1\Delta_2, B \rightarrowtriangle C : D_j]_{j=1}^m, \Gamma\Gamma'; \Delta_1\Delta_2\Delta', B \rightarrowtriangle C \vdash D}\; (W_{[]}^\bullet)}{[\Theta'_j\Delta_1\Delta_2, B \rightarrowtriangle C : D_j]_{j=1}^m, \Gamma\Gamma'; \Delta_1\Delta_2\Delta', B \rightarrowtriangle C \vdash D}\; Lock$$

The weakening transformation was applied to $[\Theta'_j : D_j]_{j=1}^m$, merging it with $[\Theta'_j\Delta_1\Delta_2, B \rightarrowtriangle C : D_j]_{j=1}^m$. The other cases are simpler. The $dmix^r$ rule is required for $\rightarrowtriangle L$ and $\rightarrow L$, along with *Lock*. In other cases, the appropriate transformation of Figure 4 is applied, except for $\vee L$, which is applied directly, since it does not affect the multiset.

Stage 2: With the left subproof ending in *Unlock*, *Pr*, *Lock* or a right-introduction rule, mix_1 is permuted into the right subproof until an initial rule or when the cut formula is principal to a left-introduction. We must show that this permutation is possible for all structural rules as well as introduction rules that do not affect the cut formula. A *Lock* that ends the right subproof is permuted over directly, whether it affects the cut formula or not, as it changes little as to what needs to be cut:

$$\dfrac{[:A]^?, \Gamma; \Delta \vdash A \qquad \dfrac{[\Theta', A : C], [\Theta'_j, A : B_j]_{j=1}^m, \Gamma'; \Delta'\Theta', A \vdash C}{[\Theta'_j, A : B_j]_{j=1}^m, \Gamma'; \Delta'\Theta', A \vdash C}\; Lock}{[\Delta\Theta'_j : B_j]_{j=1}^m, \Gamma\Gamma'; \Delta\Delta'\Theta' \vdash C}\; mix_1$$

$$\downarrow$$

$$\cfrac{\cfrac{[{:}A]^?,\Gamma;\Delta\vdash A \qquad [\Theta',A:C],[\Theta'_j,A:B_j]_{j=1}^m,\Gamma';\Delta'\Theta',A\vdash C}{[\Delta\Theta':C],[\Delta\Theta'_j:B_j]_{j=1}^m,\Gamma\Gamma';\Delta\Delta'\Theta'\vdash C}\,mix_1}{[\Delta\Theta'_j:B_j]_{j=1}^m,\Gamma\Gamma';\Delta\Delta'\Theta'\vdash C}\,Lock$$

A *Dr* rule or an *Unlock* or *Pr* rule that does not affect the cut formula is likewise permuted over in a straightforward manner. The most important case is when an additional copy of the cut formula is unlocked: here mix_1 must permute to a mix_2:

$$\cfrac{[{:}A]^?,\Gamma;\Delta\vdash A \qquad \cfrac{[\Theta'_j,A:B_j]_{j=1}^m,\Gamma';\Delta'\Theta'_k,A,A\vdash B_k}{[\Theta'_j,A:B_j]_{j=1}^m,\Gamma';\Delta',A\vdash e}\,Unlock\ (k\leqslant m)}{[\Delta\Theta'_j:B_j]_{j=1}^m,\Gamma\Gamma';\Delta'\Delta\vdash e}\,mix_1$$

$$\downarrow$$

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{[{:}A]^?,\Gamma;\Delta\vdash A}{[{:}A]^?,\Gamma\Delta;\vdash A}\,Dr^* \qquad [\Theta'_j,A:B_j]_{j=1}^m,\Gamma';\Delta'\Theta'_k,A,A\vdash B_k}{[\Theta'_j:B_j]_{j=1}^m,\Gamma\Gamma'\Delta;\Delta'\Theta'_k\vdash B_k}\,mix_2}{[\Theta'_j:B_j]_{j=1}^m,\Gamma\Gamma'\Delta;\Delta'\vdash e}\,Unlock}{[\Theta'_j:B_j]_{j=1}^m,\Gamma\Gamma';\Delta\Delta'\vdash e}\,Pr^*}{[\Delta\Theta'_j:B_j]_{j=1}^m,\Gamma\Gamma';\Delta\Delta'\vdash e}\,(W^{\bullet}_{[]})$$

Here, Dr^* and Pr^* represent repeated applications of these rules. In the *Unlock* step, $[\Theta'_k:B_k]$ was unlocked, which already exists in $[\Theta'_j:B_j]_{j=1}^m$. The inductive measure is reduced by $\mathcal{U}$, the number of *Unlock* rules that affect the cut formula above the cut, which has precedence over the height values. The presence of the green literal e, which allows *Pr* to cancel *Dr*, made this permutation possible. The case for *Pr* is similar

$$\cfrac{[{:}A]^?,\Gamma;\Delta\vdash A \qquad \cfrac{[\Theta'_j,A:B_j]_{j=1}^m,A,\Gamma';\Delta'\vdash e}{[\Theta'_j,A:B_j]_{j=1}^m,\Gamma';\Delta',A\vdash e}\,Pr}{[\Delta\Theta'_j:B_j]_{j=1}^m,\Gamma\Gamma';\Delta\Delta'\vdash e}\,mix_1$$

$$\downarrow$$

$$\cfrac{\cfrac{\cfrac{\cfrac{[{:}A]^?,\Gamma;\Delta\vdash A}{[{:}A]^?,\Gamma\Delta;\vdash A}\,Dr^* \qquad [\Theta'_j,A:B_j]_{j=1}^m,A,\Gamma';\Delta'\vdash e}{[\Theta'_j:B_j]_{j=1}^m,\Gamma\Gamma'\Delta;\Delta'\vdash e}\,mix_2}{[\Delta\Theta'_j:B_j]_{j=1}^m,\Gamma\Gamma'\Delta;\Delta'\vdash e}\,(W^{\bullet}_{[]})}{[\Delta\Theta'_j:B_j]_{j=1}^m,\Gamma\Gamma';\Delta\Delta'\vdash e}\,Pr^*$$

The inductive measure is reduced by $\mathcal{P}$, which has precedence over $\mathcal{H}_1$.

Permutation over introduction rules that do not affect the cut formula are straightforward, and do not require the transformations of Figure 4 (because Δ' is not locked in the conclusion). For $\multimap L$ and $\to L$, $dmix^\ell$ replaces $dmix_r$ for the subproof that does not contain the cut formula. Right-introduction rules may have to be permuted over as well:

$$\cfrac{[{:}A]^?,\Gamma;\Delta\vdash A \qquad \cfrac{[\Theta'_j,A:D_j]_{j=1}^m,B,\Gamma';\Delta',A\vdash C}{[\Theta'_j,A:D_j]_{j=1}^m,\Gamma';\Delta',A\vdash B{\to}C}\,{\to}R}{[\Delta\Theta'_j:D_j]_{j=1}^m,\Gamma\Gamma';\Delta\Delta',A\vdash B\to C}\,mix_1$$

$$\downarrow$$

$$\dfrac{\dfrac{[:A]^?,\Gamma;\Delta\vdash A \quad [\Theta'_j, A:D_j]_{j=1}^m, B,\Gamma';\Delta',A\vdash C}{[\Delta\Theta'_j:D_j]_{j=1}^m, B,\Gamma\Gamma';\Delta\Delta',A\vdash C}\,mix_1}{[\Delta\Theta'_j:D_j]_{j=1}^m,\Gamma\Gamma';\Delta\Delta',A\vdash B\to C}\,{\to}R$$

There is one rule that stands out from the monotony: $\vee R$ may 'affect' the cut formula in a different way:

$$\dfrac{[:A]^?,\Gamma;\Delta\vdash A \quad \dfrac{[\Theta'_j,A:D_j]_{j=1}^m,\Gamma';\vdash B}{[\Theta'_j,A:D_j]_{j=1}^m,\Gamma';\Delta',A\vdash B\vee C}\,\vee R}{[\Delta\Theta'_j:D_j]_{j=1}^m,\Gamma\Gamma';\Delta\Delta'\vdash B\vee C}\,mix_1$$

$$\downarrow$$

$$\dfrac{\dfrac{[:A]^?,\Gamma;\Delta\vdash A\quad [\Theta'_j,A:D_j]_{j=1}^m,\Gamma';\vdash B}{[\Delta\Theta'_j:D_j]_{j=1}^m,\Gamma\Gamma';\vdash B}\,dmix^\ell}{[\Delta\Theta'_j:D_j]_{j=1}^m,\Gamma\Gamma';\Delta\Delta'\vdash B\vee C}\,\vee R$$

Stage 3: Stage 2 continues until either an initial rule or a left-introduction rule on the cut formula is reached in the right subproof. Note that, if the left subproof ends in *Unlock*, *Pr* or $\perp L$, then stage 2 will also end when the required green literal is affected by an initial rule in the right subproof. Thus, we can assume for stage 3 that the cut formula A is not a literal. Now mix_1 permutes to the multicut $smix_1$:

$$\dfrac{\dfrac{}{[:A]^?,\Gamma;\Delta\vdash A}\,(*)\qquad \dfrac{}{[\Theta'_j,A:D_j]_{j=1}^m,\Gamma';\Delta',A\vdash C}\,(*L_A)}{[\Delta\Theta'_j:D_j]_{j=1}^m,\Gamma\Gamma';\Delta\Delta'\vdash C}\,mix_1$$

$$\downarrow$$

$$\dfrac{\dfrac{\dfrac{}{[:A]^?,\Gamma;\Delta\vdash A}\,(*)\quad \{[:A]^?,\Gamma;\Delta\vdash A\}\quad \dfrac{}{[\Theta'_j,A:D_j]_{j=1}^m,\Gamma';\Delta',A\vdash C}\,(*L_A)}{[\Delta':C]^?,[\Delta\Theta'_j:D_j]_{j=1}^m,\Gamma\Gamma';\Delta\Delta'\vdash C}\,smix_1}{[\Delta\Theta'_j:D_j]_{j=1}^m,\Gamma\Gamma';\Delta\Delta'\vdash C}\,Lock^?$$

The middle subproof is initially identical to the left subproof. Conceptually, we need to 'remember' how the left-subproof ended before the cut is permuted above additional *Lock* rules. The inductive measure technically reduces by $\mathcal{H}_3$, which was $\mathcal{H}_1+1$ for mix_1: this is just a convenient way to say that *smix* will immediately reduce the inductive measure by one of the other values. The inference shown subsumes the cases of whether $[:A]$ is present. If not present, the inference ends without the final *Lock*.

Permutations of $\mathbf{smix_1}$: This multicut will finally be able to permute over arbitrary *Lock* rules that create copies of the right-side cut formula. The permutation is entirely over the middle subproof. It will permute to an initial rule or a key case, at which point $dmix^r$ and $dmix^\ell$ are used to remove additional copies of the cut formula before the cut is reduced to cuts on subformulas. *Lock* is permuted over as follows:

$$\dfrac{\dfrac{}{[:A]^?,\Gamma;\Delta\vdash A}\,(*)\quad \left\{\dfrac{[\Theta:A],[\Theta_i:A]_{i=1}^n,\Gamma'';\Delta''\Theta\vdash A}{[\Theta_i:A]_{i=1}^n,\Gamma'';\Delta''\Theta\vdash A}\,Lock\right\}\quad \dfrac{}{[\Theta'_j,A:D_j]_{j=1}^m,\Gamma';\Delta',A\vdash C}\,(*L_A)}{[\Theta_i\Delta':C]_{i=1}^n[\Delta\Theta'_j:D_j]_{j=1}^m,\Gamma\Gamma''\Gamma';\Delta''\Delta'\Theta\vdash C}\,smix_1$$

$$\downarrow$$

$$\dfrac{\dfrac{\dfrac{}{[:A]^?,\Gamma;\Delta\vdash A}(*)\quad \{[\Theta:A],[\Theta_i:A]^n_{i=1},\Gamma'';\Delta''\Theta\vdash A\}\quad \dfrac{}{[\Theta'_j,A:D_j]^m_{j=1},\Gamma';\Delta',A\vdash C}(*L_A)}{[\Theta\Delta':C],[\Theta_i\Delta':C]^n_{i=1}[\Delta\Theta'_j:D_j]^m_{j=1},\Gamma\Gamma''\Gamma';\Delta''\Delta'\Theta\vdash C}\,smix_1}{[\Theta_i\Delta':C]^n_{i=1}[\Delta\Theta'_j:D_j]^m_{j=1},\Gamma\Gamma''\Gamma';\Delta''\Delta'\Theta\vdash C}\,Lock$$

The Dr rule is likewise permuted over in a straightforward manner. The inductive measure is always reduced by $\mathcal{H}_3$. Since A cannot be a literal there is no possibility of $Unlock$ or Pr. Left-introduction rules are permuted over in similar manner to mix_1: we show one representative case

$$\dfrac{\dfrac{}{[:A]^?,\Gamma;\Delta\vdash A}(*)\quad \left\{\dfrac{[\Theta_i:A]^n_{i=1},\Gamma'';\Delta'',P,Q\vdash A}{[\Theta_i:A]^n_{i=1},\Gamma'';\Delta'',P\otimes Q\vdash A}\otimes L\right\}\quad \dfrac{}{[\Theta'_j,A:D_j]^m_{j=1},\Gamma';\Delta',A\vdash C}(*L_A)}{[\Theta_i\Delta':C]^n_{i=1}[\Delta\Theta'_j,P\otimes Q:D_j]^m_{j=1},\Gamma\Gamma''\Gamma';\Delta''\Delta',P\otimes Q\vdash C}\,smix_1$$

$$\downarrow$$

$$\dfrac{\dfrac{\dfrac{}{[:A]^?,\Gamma;\Delta\vdash A}(*)\quad \{[\Theta_i:A]^n_{i=1},\Gamma'';\Delta'',P,Q\vdash A\}\quad \dfrac{}{[\Theta'_j,A:D_j]^m_{j=1},\Gamma';\Delta',A\vdash C}(*L_A)}{[\Theta_i\Delta':C]^n_{i=1}[\Delta\Theta'_j,P,Q:D_j]^m_{j=1},\Gamma\Gamma''\Gamma';\Delta''\Delta',P,Q\vdash C}\,smix_1}{[\Theta_i\Delta':C]^n_{i=1}[\Delta\Theta'_j,P\otimes Q:D_j]^m_{j=1},\Gamma\Gamma''\Gamma';\Delta''\Delta',P\otimes Q\vdash C}(\otimes L^\bullet)$$

Although other cases may require more steps, they follow the same pattern here and as demonstrated for mix_1: the cut permutes above the introduction rule, which is then replicated beneath the cut, using the appropriate transformation of Figure 4 if necessary. In the cases of $\multimap L$ and $\rightarrow L$, $dmix_1$ replaces $dmix^r$.

Key cases: When the middle subproof ends in a right-introduction rule, we have reached a key case, since the right subproof is assumed to end in a left-introduction rule on the cut formula. We demonstrate the case for $A = P\rightarrow Q$, which is representative in that it covers all of the techniques required to reduce key cases:

$$\dfrac{..\quad \left\{\dfrac{[\Theta_i:A]^n_{i=1},P,\Gamma'';\Delta''\vdash Q}{[\Theta_i:A]^n_{i=1},\Gamma'';\Delta''\vdash P\rightarrow Q}\rightarrow R\right\}\quad \dfrac{[\Theta'_j,A:D_j]^m_{j=1},\Gamma';\vdash P\quad [\Theta'_j,A:D_j]^m_{j=1},\Gamma';\Delta',Q\vdash C}{[\Theta'_j,A:D_j]^m_{j=1},\Gamma';\Delta',P\rightarrow Q\vdash C}\rightarrow L}{[\Theta_i\Delta':C]^n_{i=1}[\Delta\Theta'_j:D_j]^m_{j=1},\Gamma\Gamma'\Gamma'';\Delta''\Delta'\vdash C}\,smix_1$$

Here .. represents a subproof ending in $[:P\rightarrow Q],\Gamma;\Delta\vdash P\rightarrow Q$ for lack of space (and $A = P\rightarrow Q$). First make cuts to eliminate locked copies of the cut formula:

$$\dfrac{[:P\rightarrow Q],\Gamma;\Delta\vdash P\rightarrow Q\quad \{[\Theta_i:A]^n_{i=1},P,\Gamma'';\Delta''\vdash Q\}\quad [\Theta'_j,A:D_j]^m_{j=1},\Gamma';\Delta',P\rightarrow Q\vdash C}{[\Theta_i\Delta':C]^n_{i=1}[\Delta\Theta'_j:D_j]^m_{j=1},P,\Gamma\Gamma'\Gamma'';\Delta''\vdash Q}\,dmix_1$$

$$\dfrac{[:P\rightarrow Q],\Gamma;\Delta\vdash P\rightarrow Q\qquad [\Theta'_j,A:D_j]^m_{j=1},\Gamma';\vdash P}{[\Delta\Theta'_j:D_j]^m_{j=1},\Gamma\Gamma';\vdash P}\,dmix^\ell$$

$$\dfrac{[:P\rightarrow Q],\Gamma;\Delta\vdash P\rightarrow Q\qquad [\Theta'_j,A:D_j]^m_{j=1},\Gamma';\Delta',Q\vdash C}{[\Delta\Theta'_j:D_j]^m_{j=1},\Gamma\Gamma';\Delta',Q\vdash C}\,dmix^\ell$$

For the first subproof with $dmix_1$, the inductive measure is reduced by $\mathcal{H}_3$. For the subproofs with $dmix^\ell$, the inductive measure is reduced by $\mathcal{H}_2$, which has precedence over $\mathcal{H}_3$.

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

Now construct

$$\frac{[\Delta\Theta'_j :D_j]_{j=1}^m, \Gamma\Gamma'; \vdash P \qquad [\Theta_i\Delta' :C]_{i=1}^n [\Delta\Theta'_j :D_j]_{j=1}^m, P, \Gamma\Gamma'\Gamma''; \Delta'' \vdash Q}{[\Theta_i\Delta' :C]_{i=1}^n [\Delta\Theta'_j :D_j]_{j=1}^m, \Gamma\Gamma'\Gamma''; \Delta'' \vdash Q}\ mix_2$$

and finally,

$$\frac{[\Theta_i\Delta' :C]_{i=1}^n [\Delta\Theta'_j :D_j]_{j=1}^m, \Gamma\Gamma'\Gamma''; \Delta'' \vdash Q \qquad [\Delta\Theta'_j :D_j]_{j=1}^m, \Gamma\Gamma'; \Delta', Q \vdash C}{[\Theta_i\Delta' :C]_{i=1}^n [\Delta\Theta'_j :D_j]_{j=1}^m, \Gamma\Gamma'\Gamma''; \Delta''\Delta' \vdash C}\ mix_1$$

The inductive measures of the final cuts are reduced by $\mathcal{S}$, the size of the cut formula. The final cuts are in fact just cut_2 and cut_1 as all locked copies of the cut formulas have been removed. The last step, therefore, follows the pattern of key cases in other cut-elimination proofs: the argument is standard. The other key cases are similar.

Permutations for dmix$^\ell$, dmix^r and dmix$_1$: These cuts always permute in one direction until they reach an *Unlock* on the cut formula. $dmix^\ell$ permutes into the right subproof until such an *Unlock*, at which point it permutes to a mix_1. $dmix^r$ permutes into the left subproof until an *Unlock*, then it permutes to a mix_2. $dmix_1$ permutes into the middle subproof until *Unlock*, then permutes to an $smix_1$, which is why $dmix_1$ is also a multicut. All permutations over rules that do not affect the locked cut formulas are in the manner already demonstrated. In particular, $dmix^r$ may permute directly over *Lock* rules in the left subproof for these rules do not affect the cut formula A. We demonstrate what occurs at each significant *Unlock*:

$$\frac{\dfrac{[:A], \Gamma; \Delta \vdash A}{[:A], \Gamma; \Delta \vdash e}\ Unlock \qquad [\Theta'_j, A^+ :D_j]_{j=1}^m, A^?, \Gamma'; \Delta', A^* \vdash C}{[\Delta' :C], [\Theta'_j :D_j]_{j=1}^m, \Gamma\Gamma'; \Delta \vdash e}\ dmix^r$$

$$\downarrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{[:A], \Gamma; \Delta \vdash A}{[:A], \Gamma\Delta; \vdash A}\ Dr^* \qquad [\Theta'_j, A^+ :D_j]_{j=1}^m, A^?, \Gamma'; \Delta', A^* \vdash C}{[\Theta'_j :D_j]_{j=1}^m, \Gamma\Gamma'\Delta; \Delta' \vdash C}\ mix_2}{[\Delta' :C], [\Theta'_j :D_j]_{j=1}^m, \Gamma\Gamma'\Delta; \vdash e}\ Unlock}{[\Delta' :C], [\Theta'_j :D_j]_{j=1}^m, \Gamma\Gamma'; \Delta \vdash e}\ Pr^*$$

The inductive measure is reduced by $\mathcal{U}$. This permutation would fail if, instead of $[:A]$ we had $[\Theta :A]$ with a non-empty Θ: Pr is not enough and more powerful structural rules that would allow such a permutation are unsound. Now for $dmix^\ell$:

$$\frac{[:A]^?, \Gamma; \Delta \vdash A \qquad \dfrac{[\Theta'_j, A :D_j]_{j=1}^m, \Gamma'; \Delta'\Theta'_k, A \vdash D_k}{[\Theta'_j, A :D_j]_{j=1}^m, \Gamma'; \Delta' \vdash e}\ Unlock}{[\Delta\Theta'_j :D_j]_{j=1}^m, \Gamma\Gamma'; \Delta' \vdash e}\ dmix^\ell$$

$$\downarrow$$

$$\dfrac{\dfrac{[:A]^?, \Gamma; \Delta \vdash A \qquad [\Theta'_j, A :D_j]_{j=1}^m, \Gamma'; \Delta'\Theta'_k, A \vdash D_k}{[\Delta\Theta'_j :D_j]_{j=1}^m, \Gamma\Gamma'; \Delta\Delta'\Theta'_k \vdash D_k}\ mix_1}{[\Delta\Theta'_j :D_j]_{j=1}^m, \Gamma\Gamma'; \Delta' \vdash e}\ Unlock$$

In the last step, the unlocked element $[\Delta\Theta'_k :D_k]$ already exists in the set. The inductive measure is also reduced by $\mathcal{U}$. The subtle point here is that, unlike $dmix^r$, we cannot permute $dmix^\ell$ into a mix_2: Δ will be in the wrong location in the conclusion. Consequently, $dmix^\ell$ and other cuts that depend

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

on it, including mix_1, can unlock at most one copy of the cut formula at a time (A instead of A^+ in $[\Theta'_j, A :D_j]_{j=1}^m$). Once these cuts are proved admissible, however, we can apply them repeatedly to remove all copies of locked cut formulas. Now for $dmix_1$

$$\dfrac{\dfrac{}{[:A]^?, \Gamma; \Delta \vdash A}\,(*) \quad \left\{ \dfrac{[\Theta_i :A]_{i=1}^n, \Gamma''; \Delta''\Theta_k \vdash A}{[\Theta_i :A]_{i=1}^n, \Gamma''; \Delta'' \vdash e}\, Unlock \right\} \quad \dfrac{}{[\Theta'_j, A :D_j]_{j=1}^m, \Gamma'; \Delta', A \vdash C}\,(*L_A)}{[\Theta_i\Delta' :C]_{i=1}^n [\Delta\Theta'_j :D_j]_{j=1}^m, \Gamma\Gamma'\Gamma''; \Delta'' \vdash e}\, dmix_1$$

$$\downarrow$$

$$\dfrac{\dfrac{\dfrac{}{[:A]^?, \Gamma; \Delta \vdash A}\,(*) \quad \{[\Theta_i :A]_{i=1}^n, \Gamma''; \Delta''\Theta_k \vdash A\} \quad \dfrac{}{[\Theta'_j, A :D_j]_{j=1}^m, \Gamma'; \Delta', A \vdash C}\,(*L_A)}{[\Theta_i\Delta' :C]_{i=1}^n [\Delta\Theta'_j :D_j]_{j=1}^m, \Gamma\Gamma'\Gamma''; \Delta''\Theta_k\Delta' \vdash C}\, smix_1}{[\Theta_i\Delta' :C]_{i=1}^n [\Delta\Theta'_j :D_j]_{j=1}^m, \Gamma\Gamma'\Gamma''; \Delta'' \vdash e}\, Unlock$$

The unlocked $[\Theta_k\Delta' :C]$ already exists in $[\Theta_i\Delta' :C]_{i=1}^n$. The inductive measure in this case is simply reduced by $\mathcal{H}_3$. The other values including $\mathcal{U}$ are unchanged.

Permutations of $\mathbf{mix_2}$: mix_2 permutes into the right subproof until an initial rule or a left-introduction on a copy of the cut formula is reached. These permutations are mostly in the same manner as for mix_1. For parametric introduction rules, the transformations of Figure 4 are not required. A structural rule ending the right subproof changes nothing in terms of what needs to be cut, and are permuted over straightforwardly. When these permutations are exhausted (without reaching an initial rule), we consider the left-subproof, of which the last inference rule can only be a right-introduction, a *Lock*, *Unlock*, *Pr* or *Dr*. A right-introduction rule would result in a key case (see below). *Lock* can only produce a $[:A]$, which changes nothing. The situation for *Unlock* and *Pr* are the same as for mix_1. The additional difficulty is the *Dr* rule:

$$\dfrac{\dfrac{F, \Gamma; F \vdash A}{F, \Gamma; \vdash A}\, Dr \quad \dfrac{}{A^?, \Gamma'; \Delta', A^*, A \vdash C}\,(*L_A)}{F, \Gamma\Gamma'; \Delta' \vdash C}\, mix_2$$

Clearly, mix_2 cannot be permuted directly above this *Dr*. The cut is permuted above the left-introduction rule on the cut formula itself. Consider the representative case when $A = P \,\&\, Q$, and with the presence of $[:A]$:

$$\dfrac{\dfrac{[:P \,\&\, Q], F, \Gamma; F \vdash P \,\&\, Q}{[:P \,\&\, Q], F, \Gamma; \vdash P \,\&\, Q}\, Dr \quad \dfrac{[\Theta'_j, (P \,\&\, Q)^+ :D_j]_{j=1}^m, (P \,\&\, Q)^?, \Gamma'; \Delta', (P \,\&\, Q)^*, P \vdash C}{[\Theta'_j, (P \,\&\, Q)^+ :D_j]_{j=1}^m, (P \,\&\, Q)^?, \Gamma'; \Delta', (P \,\&\, Q)^*, P \,\&\, Q \vdash C}\, \&L}{[\Theta'_j :D_j]_{j=1}^m, F, \Gamma\Gamma'; \Delta' \vdash C}\, mix_2$$

Form the following, mix_2, for which the inductive measure is reduced by $\mathcal{H}_2$:

$$\dfrac{\dfrac{[:P \,\&\, Q], F, \Gamma; \vdash P \,\&\, Q \quad [\Theta'_j, (P \,\&\, Q)^+ :D_j]_{j=1}^m, (P \,\&\, Q)^?, \Gamma'; \Delta', (P \,\&\, Q)^*, P \vdash C}{[\Theta'_j :D_j]_{j=1}^m, F, \Gamma\Gamma'; \Delta', P \vdash C}\, mix_2}{[\Theta'_j :D_j]_{j=1}^m, F, \Gamma\Gamma'; \Delta', P \,\&\, Q \vdash C}\, \&L$$

Now, form a mix_1 beneath the mix_2

$$\dfrac{\dfrac{\dfrac{[:P \,\&\, Q], F, \Gamma; F \vdash P \,\&\, Q \quad [\Theta'_j :D_j]_{j=1}^m, F, \Gamma\Gamma'; \Delta', P \,\&\, Q \vdash C}{[\Theta'_j, F :D_j]_{j=1}^m, F, \Gamma\Gamma'; \Delta', F \vdash C}\, mix_1}{[\Theta'_j, F :D_j]_{j=1}^m, F, \Gamma\Gamma'; \Delta' \vdash C}\, Dr}{[\Theta'_j :D_j]_{j=1}^m, F, \Gamma\Gamma'; \Delta' \vdash C}\, (Dr^\bullet)$$

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

This mix_1 (actually just cut_1) is reducible because the inductive measure is reduced by $\mathcal{H}_1$. This is the reason that $\mathcal{H}_1$ has precedence over $\mathcal{H}_2$ in the lexicographical ordering. Since mix_2 removed all copies of the cut formula, the values $\mathcal{U}$ and $\mathcal{P}$ cannot increase (they will decrease if non-zero). Note that only the 'regular" introduction rule $\&L$ was used beneath mix_2. The cut rules were designed so that none of the transformations of Figure 4 will be required, as they would generally increase $\mathcal{U}$.

Key cases: The key-cases for cut_2 are similar to those of mix_1 except that mix_2 replaces $dmix^\ell$. We demonstrate one case

$$\dfrac{\dfrac{[:A\vee B],\Gamma;\vdash A}{[:A\vee B],\Gamma;\vdash A\vee B}\vee R \quad \dfrac{[\Theta'_j,(A\vee B)^+ :D_j]_{j=1}^m, A, A\vee B,\Gamma';\Delta'\vdash C \quad ..,B,A\vee B,\Gamma';\Delta'\vdash C}{[\Theta'_j,(A\vee B)^+ :D_j]_{j=1}^m, A\vee B,\Gamma';\Delta',A\vee B\vdash C}\vee L}{[\Theta'_j :D_j]_{j=1}^m,\Gamma\Gamma';\Delta'\vdash C}\, mix_2$$

Eliminate copies of cut formulas with

$$\dfrac{[:A\vee B],\Gamma;\vdash A\vee B \qquad [\Theta'_j,(A\vee B)^+ :D_j]_{j=1}^m, A, A\vee B,\Gamma';\Delta'\vdash C}{[\Theta'_j :D_j]_{j=1}^m, A,\Gamma\Gamma';\Delta'\vdash C}\, mix_2$$

$$\dfrac{[:A\vee B],\Gamma;\vdash A \qquad [\Theta'_j,(A\vee B)^+ :D_j]_{j=1}^m, A\vee B,\Gamma';\Delta',A\vee B\vdash C}{[\Delta' :C],[\Theta'_j :D_j]_{j=1}^m,\Gamma\Gamma';\vdash A}\, dmix^r$$

The inductive measure for the new mix_2 is reduced by $\mathcal{H}_2$ with $\mathcal{H}_1$ unchanged, while for $dmix^r$ it is reduced by $\mathcal{H}_1$. Now, form the cut on subformulas

$$\dfrac{\dfrac{[\Delta' :C],[\Theta'_j :D_j]_{j=1}^m,\Gamma\Gamma';\vdash A \qquad [\Theta'_j :D_j]_{j=1}^m, A,\Gamma\Gamma';\Delta'\vdash C}{[\Delta' :C],[\Theta'_j :D_j]_{j=1}^m,\Gamma\Gamma';\Delta'\vdash C}\, mix_2}{[\Theta'_j :D_j]_{j=1}^m,\Gamma\Gamma';\Delta'\vdash C}\, Lock$$

The $dmix_r$ rule can be applied even if the left subproof has a non-empty multiset, such as with $[:A \multimap B],\Gamma;A\vdash B$. Thus, the same technique is valid for all cases including $\multimap$. □

Theorem 6.8. *The cut rules of ACL are admissible.*

Proof. By induction on the number of cuts in a proof: repeatedly apply Lemma 6.7 to topmost cuts with only cut-free subproofs. □

7. Consequences of cut elimination; soundness and completeness

A large number of properties are implied by cut elimination. Only the principal cuts, cut_1 and cut_2 are required to establish these properties: we do not require cuts between arbitrary proof fragments. The most significant of these properties are soundness and completeness with respect to the use of *Lock*/*Unlock*, and with respect to the semantics.

Combined with initial elimination (Theorem 4.2), cut elimination allows us to show the logical validity of the *Unlock* rule:

Corollary 7.1. $[\Theta :A],\Gamma;\Delta\vdash B$ *is provable if and only if* $-(\Theta^\otimes \multimap A),\Gamma;\Delta\vdash B$ *is provable.*

Proof. 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) holds by cut

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

elimination and initial elimination as the following shows:

$$\dfrac{\dfrac{\dfrac{\dfrac{;\Theta \vdash \Theta^{\otimes} \qquad ;A \vdash A}{;\Theta^{\otimes} \rightarrowtriangle A, \Theta \vdash A}\ \rightarrowtriangle L}{[\Theta : A]; \Theta^{\otimes} \rightarrowtriangle A \vdash \bot}\ Unlock}{[\Theta : A]; \vdash -(\Theta^{\otimes} \rightarrowtriangle A)}\ \rightarrowtriangle R \qquad -(\Theta^{\otimes} \rightarrowtriangle A), \Gamma; \Delta \vdash B}{[\Theta : A], \Gamma; \Delta \vdash B}\ cut_2$$

□

This corollary is also important for the semantic completeness proof.

Although the formula $--A \rightarrowtriangle A$ is not valid (unless A is green), cut elimination allows a syntactic proof of the following admissible rule:

Lemma 7.2. If $--A$ is provable then A is also provable

Proof. Using cut elimination, we can easily show that the $\rightarrowtriangle R$ introduction rule is *invertible:* $;\vdash --A$ is provable iff $;-A \vdash \bot$ is provable. The last inference rule of such a cut-free proof can be either a $\rightarrowtriangle L$ rule, in which case $;\vdash A$ is provable, a *Lock* rule with premise $[:\bot];-A \vdash \bot$ or $[-A : \bot];-A \vdash \bot$ or a *Pr* rule with premise $-A;\vdash \bot$. The second possibility can be disregarded since by Corollary 7.1, this sequent is provable iff $\bot \rightarrowtriangle \bot;-A \vdash \bot$ is provable. By cut elimination again, this sequent is equivalent to $;-A \vdash \bot$. If the premise of *Lock* is $[-A : \bot];-A \vdash \bot$, by Corollary 7.1 this is equivalent to $---A;-A \vdash \bot$. But $---A$ is provably equivalent to $-A$ (even in minimal logic), and so by eliminating a cut_2 the sequent is equivalent to $-A;-A \vdash \bot$. But the provability of this sequent is equivalent to the provability of $-A;\vdash \bot$ (by weakening and *Dr*). By Corollary 7.1, this sequent is in turn equivalent to $[:A];\vdash \bot$. The last rule above this sequent can only be another *Lock* on $[:\bot]$, which is again a null operation, or an *Unlock*, which means that $;\vdash A$ is provable. The final possibility (Pr) also has premise $-A;\vdash \bot$. □

This admissible rule is easy to show semantically, but the syntactic argument is also an important step in the semantic completeness proof.

7.1. *Semantic completeness*

The soundness of the sequent calculus with respect to its semantics is argued as usual by induction on the structure of proofs. The completeness proof differs from other phase space completeness proofs principally in the following ways. First, we prove the existence of a counter-model from the assumption of an unprovable formula. Second, the unit/root of the monoid that we build is not an empty set of formulas. Instead, it is a *maximally consistent set with respect to* $\bot$, which has the characteristics of a Hintikka set. Our proof also uses cut elimination, which we want to prove procedurally in any case.

The most important argument of the proof is that, since the root is maximally consistent and since $B \oplus -B$ is provable, by cut elimination exactly one of B or $-B$ must be in the root for every formula B. Thus, *any proper addition to the root will cause it to derive* $\bot$, i.e., $\bot$ is represented by all worlds above the root. Thus, a counter-model with the proper interpretation of $\bot$ can be constructed.

The completeness proof applies Lemma 4.3 and Corollary 7.1 and requires the following:

Proposition 7.3. *If* $;\vdash A$ *is not provable, then* $;\vdash A \oplus \bot$ *is also not provable.*

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

Proof. This holds by the contrapositive of Lemma 7.2, and by cut elimination because $A \oplus \bot$ is provably equivalent to $--A$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{;A \vdash A}{;A \vdash A\oplus\bot}\,\oplus R}{[:A\oplus\bot];A\vdash\bot}\,\mathit{Unlock}}{[:A\oplus\bot];\vdash -A}\,\mathrel{-\!\!\rhd} R \qquad ;\bot\vdash\bot}{[:A\oplus\bot];--A\vdash\bot}\,\mathrel{-\!\!\rhd} L}{[:A\oplus\bot];--A\vdash A\oplus\bot}\,\oplus R}{;--A\vdash A\oplus\bot}\,\mathit{Lock}$$

The other direction is straightforward. □

This lemma allows us to build a set of formulas that are maximally consistent with respect to both A, the formula assumed to be unprovable, as well as $\bot$. This set forms the root world of the model that does not 'force" $\bot$ ($r \notin \bot^p$).

Theorem 7.4. *A formula is provable if and only if it is valid.*

The remaining details of this proof, save for Lemma A1, mostly emulate Girard (1995) and Okada (2002). The proof is found in Appendix A.

8. Natural deduction and computation

This section defines an alternative, natural deduction system with proof terms. We also define a call-by-value based reduction system for these terms. Of course, there are other ways to reduce terms and this section is intended as a demonstration of the computational significance of ACL. The new system restricts to the two arrows $\rightarrow$ and $\mathrel{-\!\!\rhd}$, and the two forms of *false*. We prefer to associate a proof term with an entire subproof, as in Parigot (1992), and not just the stoup formula. The system is found in Figure 5. There is much to explain, in particular the special elimination rules $\mathrel{-\!\!\rhd}\sharp E$ and $\rightarrow\sharp E$, which distinguishes types of the form $E \mathrel{-\!\!\rhd} R$ and $E \rightarrow R$, where E is green but R is red (hence the syntactic restrictions in the regular $\mathrel{-\!\!\rhd} E$ and $\rightarrow E$ rules). These forms *delimit* the scope of control operators. First, however, we explain how this system relates to our sequent calculus, linear λ-calculus, and $\lambda\mu$-calculus.

All formulas on the left-hand side of sequents are indexed by variables (e.g., A^x). We assume that variables are always distinguishable and are renamed to avoid clash when necessary. In particular, appropriate new variables are used to index copies of formulas in the Dr, Pr and $Unlock_e$ rules (notation $\{x/y\}$ represents substitution). In the Pr rule, we assume that duplicate copies of the formula are merged into the set context by selecting the appropriate $y_1 \ldots y_n$. In proof examples, however, we will assume that these variable substitutions occur implicitly to avoid notational clutter.

We have generalized the Pr and $\bot E$ rules to arbitrary green formulas, which is valid by Lemma 4.3. We have modified the *Lock* rule based on the more general Peirce's formula

$$((P \mathrel{-\!\!\rhd} e) \rightarrow P) \mathrel{-\!\!\rhd} P,$$

so that e can be any green literal. We chose not to allow arbitrary green formulas here in order to simplify term reduction. The notation $[\Theta : A]^d_e$ has the logical meaning of the formula $(\Theta^{\otimes} \mathrel{-\!\!\rhd} A) \mathrel{-\!\!\rhd} e$, and is indexed by d. 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. The new lock/unlock rules are clearly

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

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

$$\frac{t : \Gamma; \Delta, A^y \vdash B}{t\{x/y\} : A^x, \Gamma; \Delta; \vdash B}\, Dr \qquad \frac{t\{y_1/x_1 \ldots y_n/x_n\} : \Gamma, B_1^{y_1} \ldots B_n^{y_n}; \vdash E}{!t : B, \Gamma; B_1^{x_1} \ldots B_n^{x_n} \vdash E}\, Pr$$

$$\frac{s : \Gamma; \Delta, A^x \vdash B}{\lambda x.s : \Gamma; \Delta \vdash A \multimap B} \multimap I \qquad \frac{s : A^x, \Gamma; \Delta \vdash B}{\lambda^! x.s : \Gamma; \Delta \vdash A \to B} \to I$$

$$\frac{s : \Gamma; \Delta \vdash C \multimap B \quad t : \Gamma; \Delta' \vdash C}{(s\ t) : \Gamma; \Delta\Delta' \vdash B} \multimap E \qquad \frac{s : \Gamma; \Delta \vdash C \to B \quad t : \Gamma; \vdash C}{(s\_t) : \Gamma; \Delta \vdash B} \to E$$

$$\frac{u : \Gamma; \Delta \vdash E \multimap R \quad v : \Gamma; \Delta' \vdash E}{(u\ \sharp v) :\ \Gamma; \Delta\Delta' \vdash R} \multimap \sharp E \qquad \frac{u : \Gamma; \Delta \vdash E \to R \quad v : \Gamma; \vdash E}{(u\_\sharp v) :\ \Gamma; \Delta \vdash R} \to \sharp E$$

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

Fig. 5. Natural deduction for a fragment of ACL. E is green, R is red, e is a green literal. $[\Theta : A]$ may only appear to the left of ; If B is red, then C must also be red;

sound. Furthermore, the original rules can be recovered by $Lock_\bot$ and by $Unlock_\bot$ combined with *break*:

$$\cfrac{\cfrac{t : \Gamma; \Delta\Theta \vdash A}{[d]t : [\Theta : A]^d_\bot, \Gamma; \Delta \vdash \bot}\, Unlock_\bot}{\mathcal{B}([d]t) : [\Theta : A]^d_\bot, \Gamma; \Delta \vdash e}\, \bot E$$

The soundness and completeness of the modified rules guarantee that cuts are still admissible. The generalized rules are more useful in that they allow us to use the green formulas more meaningfully as types. 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. However, we have modified the *Pr* rule so that it affects all formulas in the affine linear context: this modification is conservative with respect to the original *Pr* because of the *Dr* rule:

$$\cfrac{\cfrac{\Gamma, B; \Delta \vdash e}{\Gamma, B, \Delta; \vdash e}\, Dr^*}{\Gamma; B, \Delta \vdash e}\, Pr$$

The generalized *Id* rule holds by Lemma 4.2. Concerning rules for the connectives and constants, the natural deduction style introduction and elimination rules are shown to be equivalent to sequent calculus in the usual way, using cut elimination:

Lemma 8.1. A formula containing only $\to$ and $\multimap$ as connectives is provable in natural deduction if and only if it is provable in sequent calculus.

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

Proof. These arguments are standard and appear elsewhere, but perhaps less so for the affine-linear $\multimap$ and for $\bot$:

$$\dfrac{\Gamma;\Delta' \vdash B \quad \dfrac{\Gamma;\Delta \vdash B \multimap C \quad ;B, B\multimap C \vdash C}{\Gamma;\Delta, B \vdash C}\, cut_1}{\Gamma;\Delta\Delta' \vdash C}\, cut_1 \qquad \dfrac{\Gamma;\Delta \vdash \bot \quad ;\bot \vdash E}{\Gamma;\Delta \vdash E}\, cut_1$$

$$\dfrac{\dfrac{\Gamma;\Delta \vdash B \quad ;B\multimap C \vdash B \multimap C}{\Gamma;\Delta, B\multimap C \vdash C}\multimap E \quad \dfrac{\Gamma;\Delta', C \vdash D}{\Gamma;\Delta' \vdash C \multimap D}\multimap I}{\Gamma;\Delta\Delta', B\multimap C \vdash D}\multimap E \qquad \dfrac{\Gamma;\Delta,\bot \vdash \bot}{\Gamma;\Delta,\bot \vdash E}\,\bot E$$

By Lemma 4.3, $;\bot \vdash E$ is provable for any green formula E. The rules $\rightarrow \sharp E$ and $\multimap \sharp E$ change the proof term, not provability. □

There are two types of lambda abstraction: λ and $\lambda^!$, that correspond to $\multimap$ and $\rightarrow$, respectively. There are also two types of application: $(s\ t)$ and (s_t): these correspond, respectively, to $A \multimap (A \multimap B) \multimap B$ and $A \rightarrow (A \rightarrow B) \multimap B$, two forms of *Modus Ponens* that are possible. 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. The solution to this problem in our system is 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 $\rightarrow$ elimination. The term $\lambda x.((\lambda^! f.\lambda y.f\ (f\ y))\ x)$ is not typable.

The rule *Dr* carries no computational meaning except for variable renaming. The rule *Pr*, however, is more significant. 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 $!(y\ \lambda x.t)$, 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 $\lambda x \lambda y.!xyy$. Alternatively, 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 $!x.t$*. We considered the possibility of including such a binder to simulate dynamic scoping in our previous paper on PCL (Liang and Miller 2013b). However, here the approach is problematic, because the variables that *Pr* affects are not distinguishable from λ-bound variables (unlike γ-bound variables that are uniquely associated with locked formulas), which means that $!x$ should be subject to substitution during β-reduction. Thus, we chose the simpler representation.

For background on the following, a clear explanation of $\lambda\mu$-calculus and control operators can be found in de Groote (1994).

Terms $\gamma d.s$ represent contraction and are equivalent to $\mu d.[d]s$ in classical $\lambda\mu$ calculus. Whereas γ can be seen as a logical constant of type $((P \multimap e) \rightarrow P) \multimap P$, μ has type $\neg\neg A \rightarrow A$ in classical logic. This does not mean that we cannot derive the more general $\mathcal{C}$ control operator (Felleisen et al. 1987), compared to *call/cc*. We can prove the purely classical $\neg\neg E \rightarrow E$, or the hybrid $((A \multimap \bot) \rightarrow 0) \multimap A$. Given that there are two implications, two constants for false, and two colours, there are 64 versions of the double negation axiom that can be considered in the unified logic.

Figure 6 displays two sample proofs. The first is for our version of Peirce's formula. The proof term is equivalent to the version found in the $\lambda\mu$-calculus, $\lambda x.\mu d.[d](x\ \lambda y.\mu f.[d]y)$: the only difference is the explicit contraction on d and the vacuous $\mu f.$, which represents a weakening on the right-hand side in Parigot's multiple-conclusion system. Had we chosen to store locked formulas on the right-hand side, we would also need such a mechanism (which we did in Liang and Miller (2013b)).

The second proof is new: $\frown e \multimap (\frown\frown A \multimap A)$. Here, A is any formula, red or green. The assumption $\frown e = e \multimap 0$ 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'

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

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

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

Fig. 6. Sample proofs: Peirce's formula and use-once control operator.

of type $\frown e$ (or $\sim e$) needs to be present. A similar proof derives a *call-once/cc* operator, this time with no colour restriction on Q: $\lambda x \lambda z.\gamma d.x(\lambda y.\mathcal{A}(z\ [d]y)) : ((P \to Q) \rightarrowtriangle P) \rightarrowtriangle \frown e \rightarrowtriangle P$.

8.1. *Structural rules and delimited control*

The transitions between different modes of proof, in the form of structural rules, also mark boundaries that determine how cuts can be permuted. This behaviour is similar to those of delimited control operators. This correspondence is consistent with the recent work of Ilik (2012), which shows that delimited control behaviour can be seen as resulting from the transition between non-classical and classical modes of proof. In this section, we explore this phenomenon and define a call-by-value based reduction system for our terms that includes the evaluation of control operators.

Break versus abort

The manner in which colouring information determines how cut can be permuted corresponds to an interesting computational effect. Consider

$$
\cfrac{s : \Gamma; \Delta \vdash E_1 \rightarrowtriangle E_2 \qquad \cfrac{t : \Gamma; \Delta' \vdash \bot}{\mathcal{B}(t) :\ \Gamma; \Delta' \vdash E_1}\ \bot E}{(s\ \mathcal{B}(t)) :\ \Gamma; \Delta\Delta' \vdash E_2}\ \rightarrowtriangle E
$$

With E_1 and E_2 both green, there are two ways to permute this cut. The first is by usual β-reduction, once s has been reduced to a lambda-term. A second possibility aborts β-reduction and reduces to the following:

$$
\cfrac{t : \Gamma; \Delta\Delta' \vdash \bot}{\mathcal{B}(t) :\ \Gamma; \Delta\Delta' \vdash E_2}\ \bot E
$$

With weakening, the same t still proves the premise (Δ-variables do not appear free in t). The term s is discarded. The same choice exists for $\to$. However, if E_2 was not green, then the only choice is β-reduction. In contrast to an *abort*: $\mathcal{A}(t)$, which uses 0-elimination, the *break* generated by a $\mathcal{B}(t)$

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

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 also say that it *catches* the break.

This is the function of the special-case $\mathrel{-\!\triangleright} \sharp E$ (and $\rightarrow \sharp E$) rules. The 'delimiter' $\sharp$ is a type annotation that indicates a transition from green to red; it has no meaning independently of such a context. β-reduction can be cancelled by $\mathcal{B}$ but resumes on $\sharp$: that is, $(\lambda x.s)\ \mathcal{B}(t)$ reduces to $\mathcal{B}(t)$, but $(\lambda x.s)\ \sharp\mathcal{B}(t)$ reduces to $s\{\mathcal{B}(t)/x\}$. The delimiter $\sharp$ has no meaning on its own, and must be dropped after substitution. This gives the marker a dynamic behaviour. Instances of subterms $(s\ t)$, where s is of type $E \mathrel{-\!\triangleright} R$ are not well-typed: they must be in the form $(s\ \sharp t)$. For example, $(\lambda x.g\ \sharp(f_2\ x))\ \sharp(f_1\ \mathcal{B}(u))$ should reduce to $(g\ \sharp\mathcal{B}(u))$. Both f_1 and f_2 are aborted. Here, f_2 must be of some type $e \mathrel{-\!\triangleright} e'$ and g of type $e' \mathrel{-\!\triangleright} R$. Reducing to a term that contains $(f_2\ \sharp\mathcal{B}(u))$ is not type-sound. This behaviour is dynamic because one cannot determine which $\sharp$ will stop the break without reducing the term.

Capturing delimited continuations

Besides *Unlock* and $\bot E$, Pr is also sensitive to colouring information. Pr also marks a boundary between classical and non-classical behaviour. Consider the following scenario:

$$\dfrac{\lambda x.u : \Gamma;\vdash E_2 \mathrel{-\!\triangleright} R \qquad \dfrac{\lambda y.v : \Gamma;\vdash E_1 \mathrel{-\!\triangleright} E_2 \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{t : \Gamma_1;\Delta_1 \vdash E_1}{[d]t :\ [:E_1]^d_{e'},\Gamma_1;\Delta_1 \vdash e'}\,Unlock_{e'}}{\vdots}}{s :\ [:E_1]^d_{e'},\Gamma\Delta;\vdash E_1}}{\gamma d.s : \Gamma\Delta;\vdash E_1}\,Lock_{e'}}{!\gamma d.s : \Gamma;\Delta \vdash E_1}\,Pr}{(\lambda y.v)!\gamma d.s : \Gamma;\Delta \vdash E_2}\,\mathrel{-\!\triangleright} E}{(\lambda x.u)\ \sharp((\lambda y.v)!\gamma d.s) : \Gamma;\Delta \vdash R}\,\mathrel{-\!\triangleright} \sharp E$$

With R red but E_1, E_2 and e' green, it is possible to permute the cut with proof $\lambda y.v$ above the Pr, and then above the *Unlock*. 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:

$$\dfrac{\Gamma;\vdash E_2 \mathrel{-\!\triangleright} R \qquad \dfrac{\dfrac{\dfrac{\Gamma;\vdash E_1 \mathrel{-\!\triangleright} E_2 \qquad \dfrac{\dfrac{\dfrac{\dfrac{\Gamma;\vdash E_1 \mathrel{-\!\triangleright} E_2 \quad \Gamma_1;\Delta_1 \vdash E_1}{\Gamma_1\Gamma;\Delta_1 \vdash E_2}\,\mathrel{-\!\triangleright} E}{[:E_2]^d_{e'},\Gamma_1\Gamma;\Delta_1 \vdash e'}\,Unlock_{e'}}{\vdots}}{[:E_2]^d_{e'},\Gamma\Delta;\vdash E_1}}{[:E_2]^d_{e'},\Gamma\Delta;\vdash E_2}}{\Gamma\Delta;\vdash E_2}\,Lock_{e'}}{\Gamma;\Delta \vdash E_2}\,Pr}{\Gamma;\Delta \vdash R}\,\mathrel{-\!\triangleright} \sharp E$$

Terms of the form $!\gamma d.s$ are different from $\gamma d.!s$ in terms of how cuts can be permuted: in the latter case Pr is applied above *Lock*, i.e., the continuation is saved *before* a switch to a classical, 'stateless' context. Terms of the latter form do not affect the permutability of cuts above *Lock* (a cut with $E_2 \mathrel{-\!\triangleright} R$ can take place beneath Pr, but still above *Lock*). However, if the continuation is saved *after* the ! switch, as shown here, then terms of transitional type $e \mathrel{-\!\triangleright} R$ cannot be captured above the *Lock* (and corresponding *Unlock*s) because Pr cannot be replicated beneath.

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 Ong and Stewart (1997)). Terms $\gamma d.t$ are not

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

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 define a reduction strategy to a variant of weak head-normal form, we first identify several subcategories of terms, P, H, V and V'. Terms are first reduced to *pre-values* P before the choice of further reduction is made. These terms will include 'rigid' terms H of the form $(x\ T_1 \ldots T_n)$, where x is a variable. V represents those pre-values that can be treated as values that do not come after $\sharp$, such as those appearing in the redex $(\lambda x.T)V$, while V' represents those pre-values that can be treated as values after $\sharp$. For simplicity, we shall only consider λ instead of $\lambda^!$ terms as additional rules for $\lambda^!$ would be repetitive. The rules for the conventional *abort*, $0E$ are left out for the same reason. Not all of the following terms are intended to be typable.

Definition 8.2.

All terms: $T\ ::=\ x\ \mid\ \lambda x.T\ \mid\ \gamma d.T\ \mid\ !T\ \mid\ [d]T\ \mid\ (T\ T)\ \mid\ (T\ \sharp T)\ \mid\ \mathcal{B}(T)$
Rigid-head terms: $H\ ::=\ x\ \mid\ (H\ T)\ \mid\ (H\ \sharp T)$
Pre-values: $P\ ::=\ H\ \mid\ \lambda x.T\ \mid\ \gamma d.P\ \mid\ [d]P\ \mid\ !P\ \mid\ \mathcal{B}(P)$
Values: $V\ ::=\ H\ \mid\ \lambda x.T\ \mid\ [d]P$
$\sharp$-Values: $V'\ ::= V\ \mid\ !P\ \mid\ \mathcal{B}(P)$

Furthermore, we shall use the symbol Q to represent pre-values that are not of the form $!P$, and the symbol R to represent Q-terms that are not of the form $\mathcal{B}(P)$.

In order to reduce cuts inside a γ binder (Lock), we will need to evaluate inside terms $\gamma d.T$ and $!T$. We choose not to unravel the focused meaning of Unlock, and also treat terms $[d]P$ as a value.

Whether a term $!P$ is treated as a value depends on the presence of $\sharp$. In $(u\ !v)$, if u is not of type $e \multimap R$, then no delimitation is required despite the !. It is only when $\sharp$ and ! appear 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.

Definition 8.3. Evaluation contexts E are of the following forms
$E\ ::=\ [\,]\ \mid\ E\ T\ \mid\ P\ E\ \mid\ E\ \sharp T\ \mid\ P\ \sharp E\ \mid\ !E\ \mid\ \gamma d.E\ \mid\ [d]E\ \mid\ \mathcal{B}(E)$

Without the symbols γ, $[d]$, !, $\sharp$ and $\mathcal{B}$, this definition of context E becomes the standard one for call-by-value reduction.

1. $E[Q\ !P] \longrightarrow E[!(Q\ P)]$ (permutation of cut above Pr).
2. $E[!P\ Q] \longrightarrow E[!(P\ Q)]$ (permutation of cut above Pr).
3. $E[!P_1\ !P_2] \longrightarrow E[!(P_1\ P_2)]$ (permutation of cut above Pr).
4. $E[Q\ \mathcal{B}(P)] \longrightarrow E[\mathcal{B}(P)]$ (break).
5. $E[\mathcal{B}(P)\ R] \longrightarrow E[\mathcal{B}(P)]$ (break).
6. $E[Q\ \gamma d.P] \longrightarrow E[\gamma d.Q\ P\{[d]Qu/[d]u\}]$ (capture of evaluation context).
7. $E[P_1\ \sharp\gamma d.P_2] \longrightarrow E[\gamma d.P_1\ \sharp P_2\{[d]P_1\sharp u/[d]u\}]$ (capture of evaluation context).
8. $E[(\lambda x.T)\ V] \longrightarrow E[T\{V/x\}]$ (beta-reduction).
9. $E[(\lambda x.T)\ \sharp V'] \longrightarrow E[T\{V'/x\}]$ (beta-reduction).
10. $E[(\gamma d.P)\ V] \longrightarrow E[\gamma d.P\{[d](u\ V)/[d]u\}\ V]$ ($\lambda\mu$-style reduction).
11. $E[(\gamma d.P)\ \sharp V'] \longrightarrow E[\gamma d.P\{[d](u\ \sharp V')/[d]u\}\ \sharp V']$ ($\lambda\mu$-style reduction).

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

The first three rules permute Pr beneath cuts where possible. A term in the form $(\lambda x.s)\ \gamma d.t$ will be reduced by rule 6, whereas $(\lambda x.s)\ \sharp!\gamma d.t$ must be reduced by beta-reduction (rule 9).

Define a *redex* to be any term of one of the forms r found in an evaluation rule $E[r] \longrightarrow s$. A term t is *well-typed* if there is a proof of $t : \Gamma;\Delta \vdash A$ for some sequent $\Gamma;\Delta \vdash A$. We have the following principle result concerning evaluation.

Lemma 8.4. Every well-typed term T is either a pre-value P or of the form $E[r]$, where r is a redex and where E and r are unique.

Proof. By induction on the structure of terms. We detail the most important and representative cases. What distinguishes a pre-value from a non-reduced term is the presence of application terms $(T_1\ T_2)$ or $(T_1\ \sharp T_2)$ that are not rigid H-terms. Assume that T is not a pre-value and is of the form $(T_1\ T_2)$. We have the following mutually exclusive cases:

1. If T_1 is not a P-term, 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. If T_1 is a P term but T_2 is not, then T is of the form $(P\ T_2)$. Again by inductive hypothesis $T_2 = E'[r]$ so let $E = P\ E'$.
3. If T is of the form $(P_1\ P_2)$, then let $E = []$; If either P_1 or P_2 is prefixed by !, then it is reduced by one of the first three rules, which permutes ! beneath cuts. Since terms $(H\ P)$ are pre-values, and terms $[d]P$ cannot be of arrow type, the only other possibilities are as follows:
   a. $P_1 = \lambda x.T$: the redex r is either $(\lambda x.T)\ V$, $(\lambda x.T)\ \gamma d.P$, or $(\lambda x.T)\ \mathcal{B}(P)$, which are reduced by rules 8, 6, 4, respectively.
   b. $P_1 = \gamma d.P$. The redex is either $(\gamma d.P)\ V$, $(\gamma d.P)\ \gamma f.P'$ or $(\gamma d.P)\ \mathcal{B}(P')$, which are reduced by rules 10, 6, 4, respectively.
   c. $P_1 = \mathcal{B}(P)$. The redex depends on whether P_2 is also of form the form $\mathcal{B}(P')$: $(\mathcal{B}(P)\ \mathcal{B}(P'))$ reduces by rule 4 uniquely and $(\mathcal{B}(P)\ R)$ reduces by rule 5, where R represents P-values not of the form $\mathcal{B}(P')$ or $!P'$.

Now consider the case, when T is a not a P-term and is of the form $(T_1\ \sharp T_2)$. The case when T_1 and T_2 are not both P-terms result in either $E = E'\ \sharp T_2$ or $E = P\ \sharp E'$ by inductive hypotheses, just as in the case of $(T_1\ T_2)$. We observe that a term $(!P_1\ \sharp P_2)$ is not well typed because $E \multimap R$ is *red* and thus Pr cannot be applied if P_1 is of such a type. Thus, let $E = []$ and $T = (Q\ \sharp P)$ (Q cannot be $!P_1$). Q also cannot be of the form $\mathcal{B}(T)$ because it must be of type $E \multimap R$, which is red and thus cannot be the conclusion of *break*. Nor can Q be $[d]T$ because *Unlock* cannot be applied to an arrow type (the restriction to green literals is retained for *Unlock*), nor can it be an H-term since T is not a P-term. Thus, the only possibilities for Q:

1. $Q = \lambda x.T$: the redex is either $(\lambda x.T)\ \sharp V'$ (rule 9) or $(\lambda x.T)\ \sharp\gamma d.P_2$ (rule 7), depending on the form of P.
2. $Q = \gamma d.P_1$: the redex is either $(\gamma d.P_1)\ \sharp V'$ (rule 11) or $(\gamma d.P_1)\ \sharp\gamma f.P_2$ (rule 7).

In the other cases when T is not a pre-value, it must be of the forms $\gamma d.T'$, $[d]T'$, $\mathcal{B}(T')$, or $!T'$, where T' is not a pre-value. All of these cases are straightforward. □

This lemma shows both the *progress* of evaluation and that evaluation is deterministic.[§]

Each evaluation rule corresponds to a valid proof transformation (permutation of cut), and thus we can also show that evaluation is type sound (subject reduction):

[§] Technically, a pre-value can also include a redex since T in the definition rigid-head terms H can be prefixed by !: but this does not contradict the statement of the lemma.

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

Lemma 8.5. If $s : \Gamma;\Delta \vdash A$ is provable and $s \longrightarrow t$, then $t : \Gamma;\Delta \vdash A$ is also provable.

Proof. Since any subproof can be replaced by another subproof that ends in the same sequent, a reduction $E[r] \longrightarrow E[r']$ is type sound as long as $r \longrightarrow r'$ is. We show some representative cases of the proof. For rule 1, if $(Q\ !P)$ is well typed then it must be a proof of the following form:

$$\dfrac{Q : \Gamma;\Delta \vdash E_1 \multimap E_2 \quad \dfrac{P : \Gamma'\Delta';\vdash E_1}{!P : \Gamma';\Delta' \vdash E_1}\,Pr}{(Q\ !P) : \Gamma\Gamma';\Delta\Delta' \vdash E_2}\multimap E$$

which is reduced to

$$\dfrac{\dfrac{Q : \Gamma;\Delta \vdash E_1 \multimap E_2 \quad P : \Gamma'\Delta';\vdash E_1}{(Q\ P) : \Gamma\Gamma'\Delta';\Delta \vdash E_2}\multimap E}{!(Q\ P) : \Gamma\Gamma';\Delta\Delta' \vdash E_2}\,Pr$$

Rules 2 and 3 are similar. Consider rule 7 (rules 6, 10, 11 are similar) with no ! before γ, the term P_1 can be captured as part of the continuation despite the presence of $\sharp$.

$$\dfrac{P_1 : \Gamma;\Delta\vdash E\multimap R \quad \dfrac{\begin{array}{c}\dfrac{u : \Gamma_1;\Delta_1\Theta\vdash E}{[d]u : [\Theta:E]^d_{e'},\Gamma_1;\Delta_1\vdash e'}\,Unlock_{e'} \\ \vdots \\ P_2 : [\Theta:E]^d_{e'},\Gamma';\Delta'\Theta\vdash E\end{array}}{\gamma d.P_2 : \Gamma';\Delta'\Theta\vdash E}\,Lock_{e'}}{P_1\ \sharp\gamma d.P_2 : \Gamma\Gamma';\Delta\Delta'\Theta\vdash R}\multimap\sharp E$$

$$\downarrow$$

$$\dfrac{\dfrac{P_1 : \Gamma;\Delta\vdash E\multimap R \qquad \begin{array}{c}\dfrac{\dfrac{P_1 : \Gamma;\Delta\vdash E\multimap R \quad u:\Gamma_1;\Delta_1\Theta\vdash E}{P_1\sharp u : \Gamma\Gamma_1;\Delta\Delta_1\Theta\vdash E}\multimap\sharp E}{[d]P_1\sharp u : [\Delta\Theta:E]^d_{e'},\Gamma\Gamma_1;\Delta_1\vdash e'}\,Unlock_{e'} \\ \vdots \\ P_2\{[d]P_1\sharp u/[d]u\} : [\Delta\Theta:E]^d_{e'},\Gamma\Gamma';\Delta'\Theta\vdash E\end{array}}{P_1\ \sharp P_2\{[d]P_1\sharp u/[d]u\} : [\Delta\Theta:E]^d_{e'},\Gamma\Gamma';\Delta\Delta'\Theta\vdash R}\multimap\sharp E}{\gamma d.P_1\ \sharp P_2\{[d]P_1\sharp u/[d]u\} : \Gamma\Gamma';\Delta\Delta'\Theta\vdash R}\,Lock_{e'}$$

Observe that even if P_2 was of the form $!P$, the new proof would still be valid because the subproof $!P\{[d]P_1\sharp u/[d]u\}$ would still be of type E. The $\sharp$ may affect further reduction of the cuts with P_1 inside the γ binder. This is not the case if ! was outside the γ as the next case we consider shows.

Consider rule 9, the exact forms of which depends on V'. Say $V' = !P$. If the term is well typed then the redex represents a proof of the form

$$\dfrac{\dfrac{T : \Gamma;\Delta,E^x\vdash R}{\lambda x.T : \Gamma;\Delta\vdash E\multimap R}\multimap I \quad \dfrac{P : \Gamma'\Delta';\vdash E}{!P : \Gamma';\Delta'\vdash E}\,Pr}{(\lambda x.T)\sharp !P : \Gamma\Gamma';\Delta\Delta'\vdash R}\multimap\sharp E$$

Here, the $!P$ must be substituted into proof T entirely, resulting in

$$T\{!P/x\} : \Gamma\Gamma';\Delta\Delta'\vdash R.$$

Unlike rule 1, ! cannot be moved to the outside because Pr is not valid with red R in the stoup. Unlike rule 7, the ! here is outside the γ. The case where $V' = \mathcal{B}(P)$ is similar. □

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

The type of delimited control operator that γ implements is dynamic since the captured term $[d]Pu$ or $[d]P\sharp u$ is not itself delimited: it depends on the context that it occurs. We shall not attempt any termination results as it is known that dynamic control/prompt operations can lead to non-terminating behaviour under call-by-value (see Kameyama and Yonezawa (2008)), 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 $\lambda x.T$ are captured as part of the continuation (as opposed to applied immediately with call-by-name). Under such a setting, delimitation is *logically necessitated* in ACL.

9. Conclusion

ACL is the culmination of our attempts to find a unified logic, with a unified proof theory, in which each logic is found easily as a fragment, but which allows proofs in different logics to mix without a collapse into classical logic. The original motivation for this project was Girard's LU system, which is largely based on linear logic and polarization as used in focusing. Our earlier attempts (LKU in particular) were also based largely on focusing and linear logic. However, we found this approach limiting because it does not adequately explain the different treatment of implication ($\rightarrow$ and $\multimap$) in classical and non-classical logics. These connectives are obviously of great importance computationally. We found a semantic explanation of the difference: *$\rightarrow$ and $\multimap$ are always negative, but they can be red or green.* One might mistake green formulas as equivalent to ?-formulas in linear logic, or some form of double negation. However, no such formula can explain structural rules that enable contractions on *other* formulas when a green formula is encountered in a proof. ACL is a new logic with its own semantics and proof theory. The uniqueness of its proof theory is found not just in its structural rules but also in its cut-elimination proof. We have demonstrated the computational relevance of ACL in terms of control operators, although we believe that more can be done in this respect.

Acknowledgments

The author wishes to thank Dale Miller and all members of the *Parsifal* research group at *INRIA/Ecole Polytechnique* for their support during the author's visits. The author also wishes to thank the anonymous reviewers of earlier versions of this article for their valuable comments.

References

Andreoli, J.-M. (1992). Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation* **2** (3) 297–347.

de Groote, P. (1994). On the relation between lambda-mu calculus and the syntactic theory of sequential control. In: *Proceedings of the 5th International Conference Logic Programming and Automated Reasoning LPAR'94*, 31–43.

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

Gentzen, G. (1935). Investigations into logical deduction. In: Szabo, M.E. (ed.) *The Collected Papers of Gerhard Gentzen*, North-Holland (1969) 68–131.

Girard, J.-Y. (1991). A new constructive logic: Classical logic. *Mathematical Structures in Computer Science* **1** 255–296.

Girard, J.-Y. (1993). On the unity of logic. *Annals of Pure and Applied Logic* **59** 201–217.

Girard, J.-Y. (1995). Linear logic: Its syntax and semantics. In Girard, J.-Y., Lafont, Y. and Regnier, L. (eds.) *Advances in Linear Logic*, London Mathematical Society Lecture Note Series, vol. 222. Cambridge University Press.

Hodas, J. and Miller, D. (1994). Logic programming in a fragment of intuitionistic linear logic. *Information and Computation* **110** (2) 327–365.

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

Ilik, D., Lee, G. and H. Herbelin (2010). Kripke models for classical logic. *Annals of Pure and Applied Logic* **161** (11) 1367–1378.

Jagadeesan, R., Nadathur, G. and Saraswat, V. (2005). Testing concurrent systems: An interpretation of intuitionistic logic. In: *Proceedings of the Foundations of Software Technology and Theoretical Computer Science*, Lecture Notes in Computer Science, vol. 2821, Springer, Hyderabad, India, 517–528.

Kameyama, Y. and Yonezawa, T. (2008). Typed dynamic control operators for delimited continuations. In: *Symposium on Functional and Logic Programming*, 239–254.

Kopylov, A.P. (1995). Propositional linear logic with weakening is decidable. In: *Symposium on Logic in Computer Science*, IEEE, 496–504.

Lafont, Y. (1997). The finite model property for various fragments of linear logic. *Journal of Symbolic Logic* **62** 1202–1208.

Liang, C. (2016). Unified semantics and proof system for classical, intuitionistic and affine logics. In: *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16*, New York, NY, USA, July 5–8, 2016, 156–165. URL `http://doi.acm.org/10.1145/2933575.2933581`.

Liang, C. and Miller, D. (2011). A focused approach to combining logics. *Annals of Pure and Applied Logic* **162** (9) 679–697.

Liang, C. and Miller, D. (2013a). Kripke semantics and proof systems for combining intuitionistic logic and classical logic. *Annals of Pure and Applied Logic* **164** (2) 86–111. URL `http://hal.inria.fr/hal-00787601`.

Liang, C. and Miller, D. (2013b). Unifying classical and intuitionistic logics for computational control. In: Kupferman, O. (ed.) *Proceedings of the 28th Symposium on Logic in Computer Science*, 283–292. URL `http://www.lix.polytechnique.fr/Labo/Dale.Miller/papers/lics13.pdf`.

Moschovakis, J. (2015). Intuitionistic logic. In: Zalta, E.N. (eds.) *The Stanford Encyclopedia of Philosophy*, Stanford University, spring edition. URL `plato.stanford.edu/archives/spr2015/entries/logic-intuitionistic`.

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

Luke Ong, C.H. and Stewart, C. (1997). A Curry-Howard foundation for functional computation with control. In: *Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, ACM Press, 215–227.

Parigot, M. (1992). $\lambda\mu$-calculus: An algorithmic interpretation of classical natural deduction. In: *LPAR: Logic Programming and Automated Reasoning, International Conference*, Lecture Notes in Computer Science, vol. 624, Springer, 190–201.

Troelstra, A.S. and Schwichtenberg, H. (2000) *Basic Proof Theory*, Cambridge University Press, 2 edition.

Veldman, W. (1976). An intuitionistic completeness theorem for intuitionistic predicate logic. *Journal of Symbolic Logic* **41** (1) 159–166.

Appendix A. Proof of Completeness

We show that if A is not provable, there exists a counter-model 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 is maximally consistent with respect to A and to $\bot$. We write $\Gamma;\Delta \nvdash A$ to mean $\Gamma;\Delta \vdash A$ *is not provable*.

We know from Proposition 7.3 that if $;\nvdash A$, then $;\nvdash A \oplus \bot$. We also need the following:

Lemma A1. If $B \oplus C, \Gamma; \vdash A \oplus \bot$ is not provable, then either $B, \Gamma; \vdash A \oplus \bot$ or $C, \Gamma; \vdash A \oplus \bot$ is not provable.

Proof. This holds because $A \oplus \bot$ is green, which enables the relaxed Pr rule by Lemma 4.3. If $B \oplus C, \Gamma; \nvdash A \oplus \bot$, then $\Gamma; B \oplus C \nvdash A \oplus \bot$ by the DR rule (arguing the contrapositive). But then by the $\oplus L$ rule either $\Gamma; B \nvdash A \oplus \bot$ or $\Gamma; C \nvdash A \oplus \bot$. Thus, by the (relaxed) Pr rule either $B, \Gamma; \nvdash A \oplus \bot$ or $C, \Gamma; \nvdash A \oplus \bot$. □

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 counter-model 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 7.1, 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 $\Gamma; \nvdash A \oplus \bot$, then $B \oplus -B, \Gamma; \nvdash A \oplus \bot$ because $; \vdash B \oplus -B$ is provable and cut is admissible. By Lemma A1, 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.

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

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. However, we borrow some notation from Girard (1995).

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 $\Gamma_r A\otimes B \in (A\otimes B)^p = cl(A^pB^p)$: by inductive hypothesis $\Gamma_r A \in A^p$, $\Gamma_r B \in B^p$ thus $\Gamma_r AB \in A^pB^p \subseteq cl(A^pB^p) = \bigcap\{F \in D : A^pB^p \subseteq F\}$. But each F in $cl(A^pB^p)$ 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 $cl(A^pB^p)$.

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 $(A\otimes B)^p = cl(A^pB^p) \subseteq Pr(A\otimes B)$: by inductive hypothesis $A^p \subseteq Pr(A)$, $B^p \subseteq Pr(B)$ so $A^pB^p \subseteq Pr(A)Pr(B)$. 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^pB^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

https://doi.org/10.1017/S0960129518000403 Published online by Cambridge University Press

$\Gamma_r B \in B^p = \bigcap Pr(C_i)$, 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 7.4: A formula is provable in sequent calculus if and only if it is valid.

One might expect the completeness proof to shed light on the question of decidability for ACL. Propositional affine linear logic is decidable (Kopylov 1995), and Lafont (1997) 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 generated. This is because there can be infinitely many *proxy* sub-formulas 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 can be finite. It would be somewhat of a surprise, however, if ACL is not decidable. For the time being, we leave the question of propositional decidability to future work.