Sabine Frittella
Giuseppe Greco
Alexander Kurz
Alessandra Palmigiano
Vlasta Sikimić

Multi-type Sequent Calculi

Abstract.

Display calculi are generalized sequent calculi which enjoy a 'canonical' cut elimination strategy. That is, their cut elimination is uniformly obtained by verifying the assumptions of a meta-theorem, and is preserved by adding or removing structural rules. In the present paper, we discuss a proof-theoretic setting, inspired both to Belnap's Display Logic [2] and to Sambin's Basic Logic [6], which generalises these calculi in two directions: by explicitly allowing different types, and by weakening the so-called *display* and *visibility* properties.

The generalisation to a multi-type environment makes it possible to introduce specific tools enhancing expressivity, which have proved useful e.g. for a smooth proof-theoretic treatment of multi-modal and dynamic logics [4, 3]. The generalisation to a setting in which full display property is not required makes it possible to account for logics which admit connectives which are neither adjoints nor residuals, or logics that are not closed under uniform substitution.

In the present paper, we give a general overview of the calculi which we refer to as *multi-type calculi*, and we discuss their canonical cut elimination meta-theorem.

Keywords: cut elimination, display calculi, multi-type sequent calculi, non-classical logics, modal logic, dynamic logics.
Math. Subject Class. 2010: 03F05, 06D50, 06D05, 03G10, 06E15.

1. Introduction

The range of non-classical logics has been rapidly expanding, driven by influences from other fields which have opened up new opportunities for applications. The logical formalisms which have been developed as a result of this interaction have attracted the interest of a wider research community than the logicians, and their theory has been intensively investigated, especially w.r.t. their semantics and computational complexity.

However, most of these logics lack a comparable proof-theoretic development. More often than not, the hurdles preventing a standard proof-theoretic development for these logics are due precisely to some of their defining features which make them suitable for applications, such as e.g. their not being closed under uniform substitution, or the fact that (the semantic interpretations of) key connectives are not adjoints.

These difficulties caused the existing proposals in literature to be often ad hoc, not easily generalisable, and more in general lacking a smooth proof-theoretic behaviour. In particular, the difficulty in smoothly transferring results from one logic to another is a problem in itself, since these logics typically come in large families (consider for instance the family of dynamic logics), and hence proof-theoretic approaches which uniformly apply to each logic in a given family are in high demand (for an expanded discussion of the existing proof systems for dynamic epistemic logics, see [5, section 3]).

The problem of the transfer of results, tools and methodologies has been addressed in the proof-theoretic literature for the families of substructural and modal logics, and has given rise to the development of several generalisations of Gentzen sequent calculi (such as hyper-, higher level-, display- or labelled-sequent calculi).

The present paper focuses on the core technical aspects of a proof-theoretic methodology and set-up closely linked to Display logic [2] and Basic logic [6]. Instances of this set-up have appeared in [5], [4] and [3] to account for (nonclassical versions of) Baltag-Moss-Solecki's dynamic epistemic logic and Propositional Dynamic Logic respectively. In ongoing work, this set-up is being applied to monotone modal logic, game logic, and linear logic.

The present set-up generalizes display calculi in two directions: by allowing multi-type languages, and by dropping the so-called *visibility* (i.e. all active formulas in all rules are in isolation or, equivalently, all the active contexts in all rules are empty), or *segregation* property, a requirement which in principle substantially reduces the proof-power of the calculi which are to satisfy it, unless they also enjoy e.g. the display property.

The additional expressivity provided by the multi-type setting allows for an improved proof-theoretic treatment of multi-modal and dynamic logics [4, 3]. The generalisation to a setting in which display and visibility property are weakened makes it possible to account both for logics—such as monotone modal logic—which admit connectives which are neither adjoints nor residuals, and for logics that are not closed under uniform substitution—such as dynamic epistemic logic.

The present paper is organized as follows: in section 2, we illustrate the main features of the multi-type setting without relying on one specific system in particular. In section 3, we list a set of conditions generalising the (properly) displayable calculi of [9]. In Section 4, we discuss how these conditions guarantee the cut elimination meta-theorem for the multi-type calculi enjoying them.

2. Multi-type calculi

The present section is aimed at illustrating the environment of multi-type calculi. Our starting point is a propositional language, the terms of which form n pairwise disjoint types $\mathsf{T_1} \ldots \mathsf{T_n}$. As in display calculi, we allow both terms at the operational level—like for instance a propositional formula—and more terms at the structural level than the usual ones builded by means of Gentzen's comma—like for instance by using Belnap's star. We will use a, b, c and x, y, z to respectively denote operational and structural terms of unspecified (possibly different) type. Further, we assume that operational and structural connectives are given *both* within each type *and* also between different types.

As an example, consider the calculus given in [4] for the intuitionistic version of Baltag-Moss-Solecki's dynamic epistemic logic (IEAK) [8]. This calculus admits four types: formulas (Fm), actions (Act), functional actions (Fnc) and agents (Ag). Typically, the languages of dynamic logics are constructed using e.g. agents or actions as parameters for the modal operators. The basic intuition is that the multi-type environment makes it possible to regard these parameters as first-class citizens, on a par with formulas. Hence, IEAK-formulas such as $\langle\alpha\rangle\langle\mathsf{a}\rangle A$—where $\langle\alpha\rangle$ and $\langle\mathsf{a}\rangle$ are the modal operators associated with the (functional) action α and with the agent a, respectively—are translated in the multi-type setting as $\alpha \vartriangle_0 (\mathsf{a} \vartriangle_2 A)$ using binary connectives which take different types as arguments, as illustrated below:

$$\vartriangle_0 : \mathsf{Fnc} \times \mathsf{Fm} \to \mathsf{Fm} \qquad\qquad \vartriangle_2 : \mathsf{Ag} \times \mathsf{Fm} \to \mathsf{Fm}.$$

Each type has its own natural domain of interpretation. For instance, the domain of interpretation for formulas can be a powerset algebra, or the algebra of downward-closed subsets of some poset (cf. [7]); the domain of interpretation of agents can be a set; finally, following [1], the domains of interpretation for actions and functional actions can be a quantale and a monoid, respectively (note that, unlike general actions, functional actions are typically not closed under non-deterministic choice).

Depending on the features of each domain, different properties can be required of the connectives. For instance, in the case of the connectives above, we stipulate that they be completely join-preserving in their second coordinate, and hence, by general order-theoretic facts, each of them has a right adjoint in its second coordinate. Therefore, the following additional connectives have a natural interpretation:

$$\mathrel{-\!\!\blacktriangleright}_0 : \mathsf{Fnc} \times \mathsf{Fm} \to \mathsf{Fm} \qquad\qquad \mathrel{-\!\!\blacktriangleright}_2 : \mathsf{Ag} \times \mathsf{Fm} \to \mathsf{Fm}.$$

However, the present setting does not permit us to assume that the right adjoints in the first coordinate exist for either $\vartriangle_0$ or $\vartriangle_2$. The lack of certain adjoints occurs in many situations, and is due to diverse reasons. For instance, by definition, the modal operators in monotone modal logic do not have adjoints. We will come back to this point later on.

Just like in standard display calculi, each operational connective corresponds to some structural connective (or proxy), that is interpreted in a context-sensitive way. To illustrate some introduction rules for operational connectives in the case of our running example, the translation mapping IEAK-formulas into multi-type terms (of which the example above $\langle\alpha\rangle\langle\mathtt{a}\rangle A \mapsto \alpha \vartriangle_0 (\mathtt{a} \vartriangle_2 A)$ is an instance) motivates the following introduction rules for the binary connective $\vartriangle_i$, for $i = 0, 2$:

$$\frac{a \bigcap_i A \vdash X}{a \vartriangle_i A \vdash X} \qquad \frac{x \vdash a \qquad X \vdash A}{x \bigcap_i X \vdash a \vartriangle_i A}$$

as the natural counterparts of the following standard introduction rules for diamond-like modal operators:

$$\frac{\{\mathtt{a}\}A \vdash X}{\langle\mathtt{a}\rangle A \vdash X} \qquad \frac{\{\alpha\}A \vdash X}{\langle\alpha\rangle A \vdash X} \qquad \frac{X \vdash A}{\{\mathtt{a}\}X \vdash \langle\mathtt{a}\rangle A} \qquad \frac{X \vdash A}{\{\alpha\}X \vdash \langle\alpha\rangle A}$$

Notice that, in the multi-type setting, each sequent is always interpreted in one domain. However, when connectives take arguments of different types, premises of binary rules are of course interpreted in different domains.

Whenever adjoints are available, display postulates such as the following ones can be added to the system. For $i = 0, 2$, the structural connectives $\bigcap_i$ and $\sqsupset_i$ are associated with $\vartriangle_i$ and $-\!\!\blacktriangleright_i$, respectively. Assuming that $\vartriangle_i$ are left adjoints guarantees the soundness of the rules below whether or not the operational connectives $-\!\!\blacktriangleright_i$ belong to the language.

$$\frac{x \bigcap_i y \vdash z}{\frac{}{y \vdash x \sqsupset_i z}}$$

Most types in the example above naturally arise from the specification of the logic IEAK in its original presentation. However, the example below shows that additional types might be added for technical reasons. Consider for instance the following rule *balance* introduced in [4, section 5]:

$$\frac{X \vdash Y}{F \bigcap_0 X \vdash F \supset_0 Y}$$

This rule is sound for functional actions but not for general actions. Introducing functional and general actions as different types makes it possible for

balance to be closed under uniform substitution of arbitrary structures for congruent operational terms *within each type*, and thus for the whole system to satisfy conditions C'_6 and C'_7 below (see section 3).
Our syntactic stipulations require that each term be assigned to a unique type *unambiguously.* This is a crucial requirement for the cut elimination meta-theorem of the next section, and will be explicitly stated in condition C'_2 below (see section 3). The notion of type-uniformity will be critical to the multi-type setting.

DEFINITION 2.1. A sequent $x \vdash y$ is *type-uniform* if x and y are of the same type.

The proposition below is shown by a straightforward induction on the minimal depth of derivation trees.

PROPOSITION 2.2. *If a multi-type proof system is such that axioms are type-uniform and rules preserve type-uniformity, then every derivable sequent is type-uniform.*

In a display calculus, the cut rule is typically of the following form:

$$\frac{X \vdash A \qquad A \vdash Y}{X \vdash Y}\ Cut$$

where X, Y are structures and A is a formula. This translates straightforwardly to the multi-type environment, by the stipulation that cut rules of the form

$$\frac{x \vdash a \qquad a \vdash y}{x \vdash y}\ Cut$$

are allowed in the given multi-type system for each type. Systems not enjoying the display property will be endowed with the following more general *surgical cut* rules:

$$\frac{x \vdash a \qquad (z \vdash y)[a]^{pre}}{(z \vdash y)[x/a]^{pre}}\ LCut \qquad \frac{(x \vdash z)[a]^{suc} \qquad a \vdash y}{(x \vdash z)[y/a]^{suc}}\ RCut$$

and the cut rules, like all other rules, will be required to preserve type-uniformity.
This finishes the discussion about the generalisation of display calculi along the multi-type dimension. The second dimension of our generalisation concerns weakening the *visibility* property. The introduction rules for connectives will be still required to satisfy the visibility condition, but principal operational terms introduced by means of axioms will not be required anymore

to occur in isolation (see condition C_5', section 3). However, a companion condition will be still required (see condition C_5'', section 3), making sure that it is always possible to equivalently replace any such axiom with a suitable one in which the given principal operational term occurs in isolation.

3. Proper multi-type calculi

A multi-type calculus is *proper* if it satisfies the following list of conditions (see [5] for a discussion on C_5' and C_5''):

C_1: Preservation of operational terms. Each operational term occurring in a premise of an inference rule *inf* is a subterm of some operational term in the conclusion of *inf*.

C_2: Shape-alikeness of parameters. Congruent parameters (i.e. non-active terms in the application of a rule) are occurrences of the same structure.

C_2': Type-alikeness of parameters. Congruent parameters have exactly the same type. This condition bans the possibility that a parameter changes type along its history.

C_3: Non-proliferation of parameters. Each parameter in an inference rule *inf* is congruent to at most one constituent in the conclusion of *inf*.

C_4: Position-alikeness of parameters. Congruent parameters are either all precedent or all succedent parts of their respective sequents. In the case of calculi enjoying the display property, precedent and succedent parts are defined in the usual way (see [2]). Otherwise, these notions can still be defined by induction on the shape of the structures, by relying on the polarity of each coordinate of the structural connectives.

C_5': Quasi-display of principal constituents. If an operational term a is principal in the conclusion sequent s of a derivation π, then a is in display, unless π consists only of its conclusion sequent s (i.e. s is an axiom).

C_5'': Display-invariance of axioms. If a is principal in an axiom s, then a can be isolated by applying Display Postulates and the new sequent is still an axiom.

C_5''': Closure of axioms under surgical cut. If $(x \vdash y)([a]^{pre}, [a]^{suc})$, $a \vdash z[a]^{suc}$ and $v[a]^{pre} \vdash a$ are axioms, then $(x \vdash y)([a]^{pre}, [z/a]^{suc})$ and $(x \vdash y)([v/a]^{pre}, [a]^{suc})$ are again axioms.

C_6': Closure under substitution for succedent parts within each type. Each rule is closed under simultaneous substitution of arbitrary structures for congruent operational terms occurring in succedent position, *within each type.*

C$'_7$: Closure under substitution for precedent parts within each type. Each rule is closed under simultaneous substitution of arbitrary structures for congruent operational terms occurring in precedent position, *within each type*.

Condition C$'_6$ (and likewise C$'_7$) ensures, for instance, that if the following inference is an application of the rule R:

$$\frac{(x \vdash y)\big([a]^{suc}_i \mid i \in I\big)}{(x' \vdash y')[a]^{suc}}\ R$$

and $\big([a]^{suc}_i \mid i \in I\big)$ represents all and only the occurrences of a in the premiss which are congruent to the occurrence of a in the conclusion (if $I = \varnothing$, then the occurrence of a in the conclusion is congruent to itself), then also the following inference is an application of the same rule R:

$$\frac{(x \vdash y)\big([z/a]^{suc}_i \mid i \in I\big)}{(x' \vdash y')[z/a]^{suc}}\ R$$

where the structure z is substituted for a.

This condition caters for the step in the cut elimination procedure in which the cut needs to be "pushed up" over rules in which the cut-formula in succedent position is parametric (cf. Section 4).

C$'_8$: Eliminability of matching principal constituents. This condition requests a standard Gentzen-style checking, which is now limited to the case in which both cut formulas are *principal*, i.e. each of them has been introduced with the last rule application of each corresponding subdeduction. In this case, analogously to the proof Gentzen-style, condition C$'_8$ requires being able to transform the given deduction into a deduction with the same conclusion in which either the cut is eliminated altogether, or is transformed in one or more applications of the cut rule, involving proper subterms of the original operational cut-term. In addition to this, specific to the multi-type setting is the requirement that the new application(s) of the cut rule be also *strongly type-uniform* (cf. condition C$_{10}$ below).

C$_9$: Type-uniformity of derivable sequents. Each derivable sequent is type-uniform.

C$_{10}$: Preservation of type-uniformity of cut rules. All cut rules preserve type-uniformity (cf. Definition 2.1).

4. Cut elimination meta-theorem

THEOREM 4.1. *Any multi-type sequent calculus satisfying C_2-C_{10} is cut-admissible. If also C_1 is satisfied, then the calculus enjoys the subformula* *property.*

Proof. We follow the proof in [9, Section 3.3, Appendix A]. For the sake of conciseness, we will expand only on the parts of the proof which depart from the treatment in [9]. In particular, we consider elimination of surgical cuts (cf. page 5). As to the principal move, the only difference concerns the case of a surgical cut application both premises of which are axioms. Condition C_5''' guarantees that this cut application can be eliminated. The remaining principal moves go exactly as in [9], thanks to C_8'. As to the parametric moves, we are in the following situation:

$$\begin{array}{ccc} \vdots\, \pi_{2.1} & & \vdots\, \pi_{2.n} \\ (x_1 \vdash y_1)[a_{u_1}]^{pre} & \cdots & (x_n \vdash y_n)[a_{u_n}]^{pre} \end{array}$$

$$\dfrac{\begin{array}{c} \vdots\, \pi_1 \\ z \vdash a \end{array} \qquad\qquad \begin{array}{c} \ddots\vdots\iddots\ \pi_2 \\ (x \vdash y)[a]^{pre} \end{array}}{(x \vdash y)[z]^{pre}}$$

where $(x \vdash y)[z]^{pre}[w]^{suc}$ means that z and w respectively occur in precedent and succedent position in $x \vdash y$, and the cut term a is parametric in the conclusion of π_2 (the other case is symmetric).

Conditions C_2-C_4 make it possible to follow the history of that occurrence of a, since these conditions enforce that the history takes the shape of a tree, of which we consider each leaf. Let a_{u_i} (abbreviated to a_u from now on) be one such uppermost-occurrence in the history-tree of the parametric cut term a occurring in π_2, and let $\pi_{2.i}$ be the subderivation ending in the sequent $x_i \vdash y_i$, in which a_u is introduced.

Wansing's case (1) splits into two subcases: (1a) a_u is introduced in display; (1b) a_u is not introduced in display. Condition C_5' guarantees that (1b) can only be the case when a_u has been introduced via an axiom.

If (1a), then we can perform the following transformation:

$$\dfrac{\begin{array}{c} \vdots\, \pi_1 \\ z \vdash a \end{array} \qquad \begin{array}{c} \vdots\, \pi_{2.i} \\ a_u \vdash y_i \\ \vdots\, \pi_2 \\ (x \vdash y)[a]^{pre} \end{array}}{(x \vdash y)[z]^{pre}} \quad \rightsquigarrow \quad \begin{array}{c} \dfrac{\begin{array}{c} \vdots\, \pi_1 \\ z \vdash a \end{array} \qquad \begin{array}{c} \vdots\, \pi_{2.i} \\ a_u \vdash y_i \end{array}}{z \vdash y_i} \\ \vdots\, \pi_2[z/a]^{pre} \\ (x \vdash y)[z]^{pre} \end{array}$$

where $\pi_2[z/a]^{pre}$ is the derivation obtained by substituting z for every occurrence in the history of a. Notice that the assumption that a is parametric in the conclusion of π_2 and that a_u is principal in *inf* imply that π_2 has more than one node, and hence the transformation above results in a cut application of strictly lower height. Moreover, the assumptions that the original cut preserves type-uniformity (C_{10}), that every derivable sequent is type-uniform (C_9), and the type-alikeness of parameters (C'_2) imply that the sequent $a_u \vdash y_i$ is of the same type as the sequents $z \vdash a$. Hence, in particular, the new cut preserves type-uniformity. Finally, condition C'_7 implies that the substitution of z for a in π_2 gives rise to an admissible derivation $\pi_2[z/a]^{pre}$ in the calculus (use C'_6 for the symmetric case). If (1b), i.e. if a_u is the principal formula of an axiom, the situation is illustrated below in the derivation on the left-hand side:

$$
\dfrac{\begin{array}{c}\vdots\,\pi_1\\ z \vdash a\end{array} \qquad \begin{array}{c}(x_i \vdash y_i)[a_u]^{pre}[a]^{suc}\\ \vdots\,\pi_2\\ (x \vdash y)[a]^{pre}\end{array}}{(x \vdash y)[z]^{pre}}
\quad\rightsquigarrow\quad
\begin{array}{c}
\dfrac{\begin{array}{c}\vdots\,\pi_1\\ z \vdash a\end{array} \qquad a_u \vdash y'[a]^{suc}}{z \vdash y'[a]^{suc}}\\
\vdots\,\pi'\\
(x_i \vdash y_i)[z/a_u]^{pre}[a]^{suc}\\
\vdots\,\pi_2[z/a]^{pre}\\
(x \vdash y)[z]^{pre}
\end{array}
$$

where $(x_i \vdash y_i)[a_u]^{pre}[a]^{suc}$ is an axiom. Then, condition C''_5 implies that some sequent $a_u \vdash y'[a]^{suc}$ exists, which is display-equivalent to the first axiom, and in which a_u occurs in display. This new sequent can be either identical to $(x_i \vdash y_i)[a_u]^{pre}[a]^{suc}$, in which case we proceed as in case (1a), or it can be different, in which case, condition C''_5 guarantees that it is an axiom as well. Further, if π is the derivation consisting of applications of display postulates which transform the latter axiom into the former, then let $\pi' = \pi[z/a]^{pre}$. As discussed when treating (1a), the assumptions imply that π_2 has more than one node, so the transformation described above results in a cut application of strictly lower height. Moreover, the assumptions that the original cut preserves type-uniformity, that every derivable sequent is type-uniform, and the type-alikeness of parameters imply that the sequent $a_u \vdash y'[a]^{suc}$ is of the same type as the sequent $z \vdash a$. Hence, the new cut is strongly type-uniform. Finally, condition C'_7 implies that substituting z for a in π_2 and in π gives rise to admissible derivations $\pi_2[z/a]$ and π' in the calculus (use C'_6 for the symmetric case).

As to Wansing's case (2), assume that a_u has been introduced as a parameter in the conclusion of $\pi_{2.i}$ by an application *inf* of the rule *Ru*. For instance, in the calculus of [4, Section 5], this situation can arise if a is of type formula Fm and it is introduced by *weakening*, or if a is of type functional actions Fnc and it is introduced by the rule *balance* or *atom*. Since a_u is a leaf in the history-tree of a, this implies that a_u is congruent only to itself in $\pi_{2.i}$. Hence, conditions C'_7, the assumption that the original cut is quasi strongly type-uniform, and the type-alikeness of parameters (C'_2) imply that the sequent $(x_i \vdash y_i)[a_u]^{pre}$ can be replaced in the conclusion of $\pi_{2.i}$ by the sequent $(x_i \vdash y_i)[z/a_u]^{pre}$ by means of an application of the same rule *Ru*. Let $\pi'_{2.i}$ be the resulting derivation.

Therefore, the transformation below yields a derivation where π_1 is not used at all and the cut disappears.

$$\dfrac{\begin{array}{c}\vdots\ \pi_1\\ z \vdash a\end{array}\qquad \begin{array}{c}\vdots\ \pi_{2.i}\\ (x_i \vdash y_i)[a_u]^{pre}\\ \vdots\ \pi_2\\ (x \vdash y)[a]^{pre}\end{array}}{(x \vdash y)[z]^{pre}} \qquad \rightsquigarrow \qquad \begin{array}{c}\vdots\ \pi'_{2.i}\\ (x_i \vdash y_i)[z/a_u]^{pre}\\ \vdots\ \pi_2[z/a]^{pre}\\ (x \vdash y)[z]^{pre}\end{array}$$

From this point on, the proof proceeds like in [9]. It is useful to emphasise that the need to combine principal and parametric moves arises in multi-type settings such as [4] not only because of contraction or additive rules, but also due to the presence of structural rules such as

$$\dfrac{(x \sqcap Y)\,;(x \sqcap Z) \vdash W}{x \sqcap (Y\,; Z) \vdash W} \qquad \dfrac{W \vdash (x \bigcap Y) > (x \sqsupset Z)}{W \vdash x \sqsupset (Y > Z)}$$

■

References

[1] A. Baltag, B. Coecke, and M. Sadrzadeh. Epistemic actions as resources. *Journal of Logic and Computation*, 17(3):555–585, 2007.

[2] Nuel Belnap. Display logic. *Journal of Philosophical Logic*, 11:375–417, 1982.

[3] S. Frittella, G. Greco, A. Kurz, and A. Palmigiano. Multi-type display calculus for propositional dynamic logic. *Journal of Logic and Computation*, 2014 (forthcoming).

[4] S. Frittella, G. Greco, A. Kurz, A. Palmigiano, and V. Sikimić. A multi-type display calculus for dynamic epistemic logic. *J. of Logic and Comp.*, 2014 (forthcoming).

[5] S. Frittella, G. Greco, A. Kurz, A. Palmigiano, and V. Sikimić. A proof-theoretic semantic analysis of dynamic epistemic logic. *JLC*, 2014 (forthcoming).

[6] C. Faggian, G. Sambin, and G. Battilotti. Basic logic: Reflection, symmetry, visibility. *Journal of Symbolic Logic*, 65, 2000.

[7] A. Kurz and A. Palmigiano. Epistemic updates on algebras. *Logical Methods in Computer Science*, 9(4), 2013. DOI: 10.2168/LMCS-9(4:17)2013.

[8] M. Ma, A. Palmigiano, and M. Sadrzadeh. Algebraic semantics and model completeness for intuitionistic public announcement logic. *Annals of Pure and Applied Logic*, 165:963–995, 2014. http://dx.doi.org/10.1016/j.apal.2013.11.004.

[9] H. Wansing. Sequent systems for modal logics. *Hand. of Phi. Logic*, 8:61–145, 2002.