Cut Elimination in a Multi-type Display Calculus

Vlasta Sikimić
joint work with: S. Frittella, G. Greco, A. Kurz, A. Palmigiano

March 10, 2014

Desiderata for Good Proof Systems for DLs

- An **independent** account of dynamic logics;
- Intuitive, **user-friendly** rules;
- **Good performances**, i.e. a system with the following properties:
  - soundness,
  - completeness,
  - cut elimination,
  - subformula property,
  - decidability.
- A **modular** account of dynamic logics:
  - more logics can be obtained by adding/subtracting rules,
  - transfer of results with minimal changes.

- Line of research between philosophy and logic.
- Inferential theory of meaning.
- Meaning as **correct** use!
- ... but you need a good proof system supporting all this...
- Main question: Which criteria are needed for a good proof system?
  - ...
  - Belnap's answer:
    Those which guarantee a **canonical** cut-elimination procedure.
- These criteria have **never before** been tested on dynamic logics.

Our Criteria

Two main ingredients

- Display calculus;
- Multi-type environment.

Provide:

- Modular reconstruction of the whole space of dynamic logics;
- Canonical cut elimination.

- are a special type of sequent calculi;
- contain more structural connectives than standard Gentzen calculi;
- are characterised by structural rules called *display postulates*:

$$\frac{X\,;Y \vdash Z}{Y \vdash X > Z}\,(;\,,>)$$

Display Property

Display property

Each structure in a display-sequent can be isolated ('displayed') in a precedent or, exclusively, succedent position.

precedent / succedent position:

$$\begin{array}{c} Y \vdash X > Z \\ \hline X \,;\, Y \vdash Z \\ \hline Y \,;\, X \vdash Z \\ \hline X \vdash Y > Z \end{array}$$

- No side conditions;
- Modular cut elimination;
- Došen's principle brought to its extremes in the presence of types.

Concrete results:

- **Dynamic calculus** (multi-type display calculus) for BMS;
- Belnap-style **cut elimination** for the dynamic calculus.

- Multi-type environment: Ag Act Fnc Fm;
  - no parameters in the language, but all types are **first-class citizens**;
- Additional expressivity:
  - operational connectives **merging different types**:

$$
\begin{array}{lcll}
\vartriangle_1, \blacktriangle_1 & : & \mathsf{Act} \times \mathsf{Fm} \to \mathsf{Fm} & \langle\alpha\rangle A \rightsquigarrow \alpha \vartriangle_1 A \\
\vartriangle_2, \blacktriangle_2 & : & \mathsf{Ag} \times \mathsf{Fm} \to \mathsf{Fm} & \langle\mathsf{a}\rangle A \rightsquigarrow \mathsf{a} \vartriangle_2 A \\
\vartriangle_3, \blacktriangle_3 & : & \mathsf{Ag} \times \mathsf{Fnc} \to \mathsf{Act} &
\end{array}
$$

- Modular perspective: by adding or subtracting types one can reach the whole space of dynamic logics;
- No labels.

The syntactic need for enforcing the distinction between functional and general actions in the specific situation arises because of the presence of the rule *balance*:

$$\frac{X \vdash Y}{F \vartriangle_0 X \vdash F \,{-}\!\!\rhd_0 Y}\,.$$

D.EAK: formulas as side conditions (and rules with labels);

$$\textit{swap-in}_L \; \frac{\mathit{Pre}(\alpha)\,;\{\alpha\}\{\mathtt{a}\}X \vdash Y}{\mathit{Pre}(\alpha)\,;\{\mathtt{a}\}\{\beta\}_{\alpha\mathtt{a}\beta}\, X \vdash Y}$$

Dynamic Calculus: no side conditions and no labels.

$$\text{swap-in}_L^D \; \frac{\mathtt{a} \blacktriangle_2 (\alpha \blacktriangle_1 X) \vdash Y}{(\mathtt{a} \blacktriangle_3 \alpha) \blacktriangle_1 (\mathtt{a} \blacktriangle_2 (\alpha \vartriangle_1 \mathrm{I}\,; X)) \vdash Y}$$

A Requirement for a Multi-type Calculus

- We introduce the notion of type-uniformity:

A sequent $x \vdash y$ is type-uniform when the structures x and y are of the same type.

Type-uniformity

Each sequent in the Dynamic Calculus for EAK is type-uniform.

- Now, Dynamic Calculus enjoys Belnap-style cut elimination.

A Multi-type Environment and Cut Elimination

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

- A cut rule is *strongly type-uniform* if its premises and conclusion are of the same type.

Belnap's Conditions in a Nutshell

Principal step

The elimination of a cut rule on two principal formulae is checked in a standard (Gentzen-like) manner.

Parametric step

- *Parametric* formulas are constituents that are held constant in the application of the rule;
- Conditions C2-C4 track the history of a formula;
- Conditions C6-C7 guarantee that the right substitution is possible.

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

A case in which *a* is parametric, but introduced by at least one premise rule previously.

- Condition C_5' about the shape of the axioms $\rightsquigarrow$ quasi/properly displayable calculus;
- The remaining conditions guarantee preservation of types including:
  - type-uniformity;
  - strong type-uniformity of the cut rule.

Summary

- $\checkmark$ Introduction of a multi-type calculus
  upshot $\rightsquigarrow$ modularity;
- $\checkmark$ Revision of Belnap's cut elimination for a multi-type environment
  upshot $\rightsquigarrow$ new requirements for proof-theoretic semantics;
- $\checkmark$ Dynamic Calculus as a case study.

Conclusions and Further Directions

Conclusions

- Added value both for proof-theoretic semantics and for DEL:
  - **enlarged range of applications** of proof-theoretic semantics;
  - **an independent perspective** on dynamic logics.

Further directions

- A uniform proof-theoretic account for **more dynamic logics** (already applied to game logic).
- Types and Moore sentences.