A New Machine-checked Proof of Strong Normalisation for Display Logic

Jeremy E. Dawson and Rajeev Goré

Automated Reasoning Group
Computer Sciences Laboratory
Res. Sch. of Inf. Sci. and Eng.
Institute of Advanced Studies

Formal Methods Group
Dept. of Computer Science
Fac. of Eng. and Inf. Tech.
The Faculties

Australian National University

jeremy@csl.anu.edu.au
csl.anu.edu/~jeremy

rpg@csl.anu.edu.au
csl.anu.edu.au/~rpg

Goré supported by an Australian Research Council Queen Elizabeth II Fellowship.

Dawson supported by an Australian Research Council Large Research Grant.

Overview and Motivation

RA: Algebra of binary relations (sets of ordered pairs) over underlying set U

Foundation of relational databases

Display Calculi: Generalised sequent framework due to Nuel Belnap

Single cut-elimination proof if rules satisfy 8 easily checked conditions

dRA: Display Calculus for the logic of relation algebras

Isabelle/HOL: Logical framework that allows us to formalise mathematics

Goal: Use Isabelle/HOL to check the strong normalisation proof of Wansing

Actual outcome: problem with proof, found and checked a new proof

Display Calculus $\delta\mathrm{RA}$ for Relation Algebra

RA: Algebra of binary relations (sets of ordered pairs) over underlying set U
Extends classical propositional logic (algebra of sets), we only show classical connectives and rules.

Atomic constants: $c ::= \bot \mid \top \mid \ldots$

Atomic formulae: $p ::= p_0 \mid p_1 \mid p_2 \mid \cdots$ (relation symbols)

Formulae: $A\ B \quad ::= \quad c \quad \mid \quad p \quad \mid \quad \neg A \quad \mid \quad A \wedge B \quad \mid \quad A \vee B \quad \mid \quad \ldots$

Structure: $X\ Y \quad ::= \quad A \quad \mid \quad I \quad \mid \quad *X \quad \mid \quad X\ ,\ Y \quad \mid \quad \ldots$

Sequent: $X \vdash Y$ where X is the *antecedent* and Y is the *succedent*

Sequent rule: (name) $\dfrac{X_1 \vdash Y_1 \cdots X_n \vdash X_n}{X \vdash Y}$ $\dfrac{\textit{premisses}}{\textit{conclusion}}$

Display Calculus $\delta\mathrm{RA}$: Display Postulates

$$\frac{*X \vdash Y}{\overline{*Y \vdash X}} \qquad \frac{X \vdash *Y}{\overline{Y \vdash *X}} \qquad \frac{**X \vdash Y}{\overline{X \vdash Y}} \qquad \begin{array}{c} X \vdash Z\,,\,*Y \\ \hline\hline X\,,\,Y \vdash Z \\ \hline\hline Y \vdash *X\,,\,Z \end{array} \qquad \begin{array}{c} Z\,,\,*Y \vdash X \\ \hline\hline Z \vdash X\,,\,Y \\ \hline\hline *X\,,\,Z \vdash Y \end{array}$$

Theorem (Belnap82): For every sequent $X \vdash Y$ and every substructure Z of $X \vdash Y$, there is a structurally equivalent sequent $Z \vdash Y'$ or $X' \vdash Z$ that has Z displayed as the whole of its antecedent or succedent.

e.g. $\dfrac{*p_3 \vdash p_0\,,\,(p_0 \land p_1)}{*(p_0\,,\,(p_0 \land p_1)) \vdash p_3} \qquad \dfrac{*p_3\,,\,*(p_0 \land p_1) \vdash p_0}{*p_3 \vdash p_0\,,\,(p_0 \land p_1)}$

Display Calculus $\delta\mathrm{RA}$: Logical Rules

- Left and right introduction rules for every logical connective
- (id) $p \vdash p$ (cut) $\dfrac{X \vdash A \quad A \vdash Y}{X \vdash Y}$
- $(\vee \vdash)$ $\dfrac{A \vdash X \quad B \vdash Y}{A \vee B \vdash X\ ,\ Y}$ decoding/rewrite $(\vdash \vee)$ $\dfrac{Z \vdash A\ ,\ B}{Z \vdash A \vee B}$
- $(\wedge \vdash)$ $\dfrac{A\ ,\ B \vdash Z}{A \wedge B \vdash Z}$ rewrite/decoding $(\vdash \wedge)$ $\dfrac{X \vdash A \quad Y \vdash B}{X\ ,\ Y \vdash A \wedge B}$
- Introduced formula is always displayed
- Gentzen overloading:

Antecedent Part	$\top$	$\neg$	$\wedge$
Connective	I	$*$	$,$
Succedent Part	$\bot$	$\neg$	$\vee$

- Example: $*(p_0\ ,\ p_1) \vdash *(p_1\ ,\ p_3)$ means $\neg(p_0 \vee p_1) \vdash \neg(p_1 \wedge p_3)$

Display Logic $\delta\mathrm{RA}$: Structural Rules

A *structural* rule contains no logical connectives and no formula variables.

$$(I^{+}_{-} \vdash)\ \frac{\dfrac{X \,,\, I \vdash Y}{X \vdash Y}}{I \,,\, X \vdash Y} \qquad (\vdash A)\ \frac{W \vdash X \,,\, (Y \,,\, Z)}{W \vdash (X \,,\, Y) \,,\, Z} \qquad (\vdash P)\ \frac{Z \vdash Y \,,\, X}{Z \vdash X \,,\, Y}$$

$$(M \vdash)\ \frac{X \vdash Z}{X \,,\, Y \vdash Z} \qquad (\vdash C)\ \frac{Z \vdash X \,,\, X}{Z \vdash X}$$

Cut Elimination

Theorem [Belnap 82]: Any display calculus whose rules obey eight conditions given by Belnap enjoys cut-elimination. If $X \vdash Y$ is derivable, then it is derivable without the cut rule.

Formalising δRA in Isabelle/HOL

Formulae: $A\ B \quad ::= \quad c \quad | \quad p \quad | \quad \neg A \quad | \quad A \wedge B \quad | \quad A \vee B \quad | \quad \ldots$

```
datatype formula = Btimes formula formula ("_ && _" [68,68] 67)
                 | Bplus formula formula ("_ v _" [64,64] 63)
                 | Bneg formula ("--_" [70] 70)
                 | Btrue ("T")
                 | Bfalse("F")
                 | ...
                 | PP string
                 | FV string
```

Formalising δRA in Isabelle/HOL

Structure: $X\ Y \quad ::= \quad A \quad | \quad I \quad | \quad *X \quad | \quad X \, , \, Y \quad | \quad \ldots$

```
datatype structr = Comma structr structr                     ,
                 | Star structr                              *
                 | I                                         $I
                 | ...
                 | Structform formula
                 | SV string                                 $''X''

datatype sequent = Sequent structr structr                    |-
```

`$''X'' |- ''A''` via `print-translation` and `parse-translation`

becomes `Sequent (SV ''X'') (Structform (FV ''A''))`

A Deep Embedding of Rules

$$\text{(name)}\ \frac{X_1 \vdash Y_1 \cdots X_n \vdash X_n}{X \vdash Y} \qquad\qquad \frac{X \vdash Y \,,\, Z}{\overline{\overline{\ast Y \,,\, X \vdash Z}}}$$

```
datatype rule = Rule (sequent list) sequent
              | Bidi sequent sequent
              | InvBidi sequent sequent
```

```
cs1    "cs1 == Bidi ($''X'' |- $''Y'',, $''Z'')
                    (*$''Y'',, $''X'' |- $''Z'')"
```

Deep embedding of variables – `''X''`, not `?X` – needed to describe properties of rules (eg Belnap's eight conditions).

This means we must define substitution explicitly.

A shallow(er) embedding might not model rules at all.

Formalising Derivations

```
datatype dertree = Der sequent rule (dertree list)
                 | Unf sequent
```

Leaves of successful derivation have form `Der (''A'' |- ''A'') id []`

A derived rule will have some `Unf`inished leaf sequents

`wfb :: "dertree => bool"` returns true if the end-sequent of `dertree` and the end-sequents of its immediate children in `dertree list` are a well-formed instance of `rule`

`allDT :: "(dertree => bool) => dertree => bool"` returns true if every node of `dertree` satisfies the given boolean valued function

`allDT wfb` returns true if every rule application in `dertree` is well formed

`allDT (frb rules)` returns true if every rule application in `dertree` is a member of set `rules`

Reducing a Left-and-Right Principal Cut

- A cut on A is *left- [right-] principal* if the left [right] inference immediately above the cut introduces A via a logical rule, written $(\vdash A)$ [$(A \vdash)$]

$$\dfrac{\dfrac{\Pi_L}{X \vdash A}(\vdash A) \quad \dfrac{\Pi_R}{A \vdash Y}(A \vdash)}{X \vdash Y}$$

- C8: All left-and-right principal cuts can be replaced by cuts on strictly smaller subformulae of A

- Base Case: (cut) $\dfrac{p \vdash p \quad p \vdash p}{p \vdash p}$ reduced to just $p \vdash p$

- Similar cases for $\top$, $\bot$

Example of a Principal reduction (C8 condition)

$$\dfrac{\dfrac{\dfrac{\Pi_{ZAB}}{Z \vdash A, B}}{Z \vdash A \lor B}\,(\vdash \lor) \qquad \dfrac{\dfrac{\Pi_{AX}}{A \vdash X} \quad \dfrac{\Pi_{BY}}{B \vdash Y}}{A \lor B \vdash X, Y}\,(\lor \vdash)}{Z \vdash X, Y}\,(cut)$$

Principal cut on formula $A \lor B$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Pi_{ZAB}}{Z \vdash A, B}}{*A, Z \vdash B}\,\text{(dp)} \qquad \dfrac{\Pi_{BY}}{B \vdash Y}}{*A, Z \vdash Y}\,(cut)}{Z \vdash A, Y}\,\text{(dp)}}{Z, *Y \vdash A}\,\text{(dp)} \qquad \dfrac{\Pi_{AX}}{A \vdash X}}{\dfrac{Z, *Y \vdash X}{Z \vdash X, Y}\,\text{(dp)}}\,(cut)$$

Transformed derivation uses cuts only on A and B

Must exhibit and hard-wire such transformations for every logical connective

So if cut is *not* left-and-right principal then we try to make it so ...

A Parametric Reduction — making a cut left-right-principal

- Given a derivation ending in a non left-and-right principal cut on formula A, we trace the ancestor occurrences of this A up Π_L:
- Let $\Pi[A := Y]$ mean "replace all (non-principal) ancestors of A with Y"

$$
\dfrac{\dfrac{\dfrac{\dfrac{\Pi_L^1}{V \vdash A}(\vdash A)}{\Pi_L^2}(\pi)}{X \vdash A}(\rho) \qquad \dfrac{\Pi_R}{A \vdash Y}}{X \vdash Y}(cut)
\qquad\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\Pi_L^1}{V \vdash A}(\vdash A) \qquad \dfrac{\Pi_R}{A \vdash Y}}{V \vdash Y}(cut)}{\Pi_L^2[A := Y]}(\pi)}{X \vdash Y}(\rho)
$$

- New cut is left-principal (similarly make it also right principal)
- If Π_L branches (A introduced at multiple places), more complex (!)
  If Π_L^1 is empty, A introduced by $A \vdash A$, no new cuts needed.

Strong Normalisation

Cut-admissibility proceeds by eliminating the top-most cut

Strong normalisation: define reduction steps, eliminate any cut that can be reduced, and prove that this cut-elimination procedure terminates:

- $\Pi_n < \Pi_{n-1} < \cdots < \Pi_2 < \Pi_0$
- Π_n is cut-free
- n is finite
- weak-normalisation (attacking top most cut) is an instance of this method

Usually define reductions as either a parametric or principal move, with proviso that we must not cross a cut in a parametric move

Why Formalise (in Isabelle) ?

Strong-normalisation proofs are notoriously difficult involving many cases

We tried to formalise Heinrich Wansing's proof of strong-normalisation for display logic, but got stuck at one particular case

Found a problem ... Wansing assumes that Π_L^1 is unchanged!

V_A is $(*A \,,\, X)$ $\qquad$ V_Y is $(*Y \,,\, X)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Pi_L^1}{V_A \vdash A}(\vdash A)}{X \vdash A, A}\text{(dp)}}{X \vdash A}\text{(ctr)} \quad \dfrac{\Pi_R}{A \vdash Y}}{X \vdash Y}(cut)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Pi_L'^1}{V_Y \vdash A}(\vdash A) \quad \dfrac{\Pi_R}{A \vdash Y}}{V_Y \vdash Y}(cut)}{X \vdash Y, Y}\text{(dp)}}{X \vdash Y}\text{(ctr)}$$

Branch in left subtree — before and after

$$\dfrac{\dfrac{\dfrac{\Pi'}{X' \vdash A}\,(\text{intro-}A) \quad \dfrac{\Pi''}{X'' \vdash A}\,(\text{intro-}A)}{\dfrac{\Pi_L}{X \vdash A}\,(\rho)}\,(\pi) \qquad \dfrac{\Pi_R}{A \vdash Y}}{X \vdash Y}\,(cut)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Pi'}{X' \vdash A} \quad \dfrac{\Pi_R}{A \vdash Y}}{X' \vdash Y}\,(cut) \qquad \dfrac{\dfrac{\Pi''}{X'' \vdash A} \quad \dfrac{\Pi_R}{A \vdash Y}}{X'' \vdash Y}\,(cut)}{\Pi_L[A \mathrel{?\!=} Y]}\,(\pi)}{X \vdash Y}\,(\rho)$$

Two successive introductions – before

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Pi}{X, B \vdash *B}\,(\rho)}{X, B \vdash \neg B}\,(\vdash \neg)}{*\neg B, X \vdash *B}\,(\mathrm{dp})}{*\neg B, X \vdash \neg B}\,(\vdash \neg)}{X \vdash \neg B, \neg B}\,(\mathrm{dp})}{X \vdash \neg B}\,(\vdash \mathrm{cont}) \qquad \dfrac{\Pi_R}{\neg B \vdash Y}}{X \vdash Y}\,(cut)
$$

Two successive introductions – after

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Pi}{X, B \vdash *B}\,(\rho)}{X, B \vdash \neg B}\,(\vdash \neg) \qquad \dfrac{\Pi_R}{\neg B \vdash Y}}{X, B \vdash Y}\,(cut)}{*Y, X \vdash *B}\,(\mathrm{dp})}{*Y, X \vdash \neg B}\,(\vdash \neg) \qquad \dfrac{\Pi_R}{\neg B \vdash Y}}{\dfrac{\dfrac{*Y, X \vdash Y}{X \vdash Y, Y}\,(\mathrm{dp})}{X \vdash Y}\,(\vdash \mathrm{cont})}\,(cut)
$$

Various Binary Orderings

1. $\Pi_1 <_{\texttt{LRP}} \Pi_0$ if the bottom inferences of derivations Π_1 and Π_0 are both cut, and either

   a. the cut in Π_1 is left-principal or right-principal, and the cut in Π_0 is neither, or

   b. the cut in Π_1 is (left- and right-)principal, and the cut in Π_0 is not (left- and right-)principal.

2. $\Pi_1 <_{\texttt{cut}} \Pi_0$ if the bottom inferences of derivations Π_1 and Π_0 are both cut, and if either

   a. the size of the cut-formula of Π_1 is smaller than that of Π_0, or

   b. each derivation has the same cut-formula, and $\Pi_1 <_{\texttt{LRP}} \Pi_0$.

Various Binary Orderings

1. $\Pi_1 <_{\mathtt{sn1}} \Pi_0$ if Π_0 and Π_1 are the same except that one of the immediate subtrees of Π_0 is strongly normalisable and reduces to the corresponding immediate subtree of Π_1.

2. $\Pi_1 <_{\mathtt{dt}} \Pi_0$ iff any of the following hold:
   a. the bottom inference of Π_1 is not cut, but that of Π_0 is
   b. $\Pi_1 <_{\mathtt{cut}} \Pi_0$
   c. $\Pi_1 <_{\mathtt{sn1}} \Pi_0$.

Theorem 1 *The relations* `LRPorder`, `cutorder`, `sn1order`, *and* `dtorder` *are well-founded*

Despite notation, these relations are *not* reflexive, and some are *not* transitive.

Intuitively, $(\Pi_1, \Pi_0) \in \mathtt{dtorder}$ means that Π_1 is closer to being cut-free (in some sense) than is Π_0.

Inheriting Strong Normalisation

We next define `snHered`, a property of derivation trees which indicates that a tree inherits strong normalisation from its immediate subtrees.

Definition 1 *Π satisfies `snHered` iff: if all the immediate subtrees of Π are strongly normalisable then Π is strongly normalisable.*

```
snHered_def = "snHered ?dt ==
  set (nextUp ?dt) <= sn_set --> ?dt : sn_set"
```

Lemma 1 *A derivation tree Π is strongly normalisable iff every subtree of Π has the property `snHered`.*

Key Properties of Reductions

When we apply a reduction to a cut at the bottom of a tree Π_0, obtaining Π_1, then every subtree Π_1^s of Π_1 which has cut at the bottom satisfies one of

- Π_1^s is a proper subtree of Π_0, or
- $\Pi_1^s <_{\mathtt{cut}} \Pi_0$, because
  - (principal reduction) the cut-formula of Π_1^s is smaller than that of Π_0
  - (parametric reduction) Π_1^s and Π_0 have the same cut-formula and $\Pi_1^s <_{\mathtt{LRP}} \Pi_0$.

Strong-Normalisation

Lemma 2 *For a given derivation Π_0, if all derivation trees $\Pi' <_{dt} \Pi_0$ have the property `snHered`, then so does Π_0.*

```
dth = "ALL dt'. (dt', ?dt) : dtorder --> snHered dt'
       ==> snHered ?dt" : thm
```

Proof: The machine-proof is quite complicated, and a careful examination of it highlights why the definition of *dtorder* needs to be so complex.

Theorem 2 *Every derivation tree is strongly normalisable.*

```
all_snH = "snHered ?dt"    : thm
all_sn  = "strongNorm ?dt" : thm
```

Proof: By well-founded induction, it follows from Lemma 2 that every derivation tree satisfies `snHered`; the result follows from Lemma 1.

Where are we?

- We have actually shown using Isabelle that a class of reductions which we have defined is (small enough to be) strongly normalising.
- We would like to show that this class of reductions is correctly defined, ie, it (is large enough that it) contains the reductions used by Wansing.
- We can show that it is large enough to ensure cut-elimination. This guards against the worst definition errors.
- We do this by showing that any tree containing a cut admits at least one reduction in this class. Strong normalization then gives cut-elimination.

Belnap’s Conditions and New C8

Appendix: Belnap’s Conditions.

For every sequent rule Belnap82, page 388 first defines the following notions: in an application *Inf* of a sequent rule (ρ), “constituents occurring as part of occurrences of structures assigned to structure-variables are defined to be **parameters** of *Inf* ; all other constituents are defined as **nonparametric**, including those assigned to formula-variables. Constituents occupying similar positions in occurrences of structures assigned to the same structure-variable are defined as **congruent** in *Inf* ”. The eight (actually seven) conditions shown below are from [Kracht96]:

(C1) Each formula which is a constituent of some premiss of a rule ρ is a subformula of some formula in the conclusion of ρ.

(C2) Congruent parameters are occurrences of the same structure.

(C3) Each parameter is congruent to at most one constituent in the conclusion. Equivalently, no two constituents of the conclusion are congruent.

(C4) Congruent parameters are either all antecedent parts or all succedent parts of their respective sequent.

(C5) If a formula is non-parametric in the conclusion of a rule ρ, it is either the entire antecedent, or the entire succedent (called a **principal** formula).

(C6/7) Each rule is closed under simultaneous substitution of arbitrary structures for congruent parameters.

(C8) If there are inference rules ρ_1 and ρ_2 with respective conclusions $X \vdash \varphi$ and $\varphi \vdash Y$ with φ principal in both inferences (in the sense of C5), and if (cut) is applied to yield $X \vdash Y$ then, either $X \vdash Y$ is identical to $X \vdash \varphi$ or to $\varphi \vdash Y$; or it is possible to pass from the premisses of ρ_1 and ρ_2 to $X \vdash Y$ by means of inferences falling under (cut) where the cut-formula is always a proper subformula of φ.