Generic Methods for Formalising Sequent Calculi Applied to Provability Logic

Jeremy Dawson and Rajeev Goré

Logic and Computation Group
School of Computer Science
College of Engineering and Computer Science
Australian National University
Canberra ACT 0200, Australia
http://users.rsise.anu.edu.au/~jeremy/
http://users.rsise.anu.edu.au/~rpg/

September 21, 2010

Outline

Introduction

Formalisation of cut-admissibility for the GLS sequent system

- cut-admissibility applies for many sequent systems
- proofs can be tedious — details omitted ("other cases are similar")
- we try to get common elements of the proofs for re-use
- provability logic has unusual features (*GL* rule has formula on both sides of $\vdash$), proof more complex
- previous proofs wrong, or allegedly so but actually OK
- formalised proof in Isabelle/HOL confirms the result, omits no details, and uses many lemmas applicable for other logics

Sequents and Multisets, Sets and Provability Logic

- sequents $\Gamma \vdash \Delta$ where Γ and Δ are "collections" of formulae
- Our "collections" are multisets (unordered, but repetitions counted)
- Tree-shaped derivations, conclusion at the bottom
- Tree branches where rule has > 1 premise, leaf where rule has no premises

Provability Logic

- explicit weakening and contraction rules
- usual (additive) rules for $\neg, \wedge, \vee, \rightarrow$
- additional rule *GLR* which characterises **GL**:

$$\frac{\Box X, X, \Box B \vdash B}{\Box X \vdash \Box B}\ GLR \text{ or } GLR(B) \text{ or } GLR(X, B)$$

- in our formalisation, cut or multicut rules not part of GLS

$$(\text{cut})\ \frac{\Gamma \vdash A, \Delta \qquad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}$$

$$(\text{multicut})\ \frac{\Gamma' \vdash A^n, \Delta' \qquad \Gamma'', A^m \vdash \Delta''}{\Gamma', \Gamma'' \vdash \Delta', \Delta''}$$

Deep and Shallow Embeddings — Derivations

- Deep or shallow embeddings of *derivations*, *rules* and *variables*.
- *shallow* means that a feature in the logic is identified with the same feature of Isabelle/HOL

Derivations:

- Deep: the actual derivation tree is a data structure in HOL

```
datatype 'a dertree = Der 'a ('a dertree list)
    | Unf 'a (* unfinished leaf not proved *)
```

  there is a predicate which tests whether each node of an derivation tree is an instance of a rule
- Shallow: no derivation tree data structure, but an inductive definition in HOL saying what formulae are derivable; (the course of a proof, in HOL, of a formula, could be described by a derivation tree)

Deep and Shallow Embeddings — Rules and Variables

Rules:

- Deep: each rule is a data structure in HOL, and the definition of derivability refers to the set of rules as a parameter
- Shallow: the set of rules is encoded in the definition of derivability

Variables (only for deep embedding of rules):

- Deep: each rule contains references to names variable(s), and HOL functions instantiate each variable as required
- Shallow: each "rule" is in fact the set of all possible instantiations of the "rule", achieved using Isabelle variables

Shallow embedding of rules seems to necessarily imply shallow embedding of variables and the process of instantiating them

Generic Derivability Predicates

```
types 'a psc = "'a list * 'a" (* single step inference *)
consts
  derl, adm :: "'a psc set => 'a psc set"
  derrec    :: "'a psc set => 'a set => 'a set"
```

An inference rule of type `'a psc` is a list of premises and a conclusion. Then

- `derl rls` is the set of rules derivable from the rule set `rls`,
- `adm rls` is the set of admissible rules of the rule set `rls`, and
- `derrec rls prems` is the set of sequents derivable using rules `rls` from the set `prems` of premises.

Examples : Generic Derivability Predicates

Shallow Embedding of Derivations, Deep Embedding of Rules:

$(\{\Gamma \vdash P,\ \ \Gamma \vdash Q\},\ \ \Gamma \vdash P \wedge Q) \in$ `rules` (etc for other rules)

$c \in$ `prems` $\Longrightarrow c \in$ `derrec rules prems`

$[\![(ps, c) \in$ `rules` ; $ps \subseteq$ `derrec rules prems` $]\!] \Longrightarrow$
$c \in$ `derrec rules prems`

Shallow Embedding of Derivations and of Rules:

$c \in$ `prems` $\Longrightarrow c \in$ `ders prems`

$[\![\Gamma \vdash P \in$ `ders prems` ; $\Gamma \vdash Q \in$ `ders prems` $]\!] \Longrightarrow$
$\Gamma \vdash P \wedge Q \in$ `ders prems`

Theorems about the Generic Derivability Predicates

- `derl_deriv_eq` states that derivability using derived rules implies derivability using the original rules
- `derrec_trans_eq` states that derivability from derivable sequents implies derivability from the original premises.

```
derl_deriv_eq : "derl (derl ?rls) = derl ?rls"
derrec_trans_eq : "derrec ?rls (derrec ?rls ?prems)
                  = derrec ?rls ?prems"
```

The induction principle (simplified) from the definition of `derrec` :

$$\frac{x \in derrec\ rls\ prems \qquad \forall c \in prems.\ P\ c \qquad \forall (ps, c) \in rls.\ (\forall p\ in\ ps.\ P\ p) \Rightarrow P\ c}{P\ x}$$

Induction on two derivations

Induction for a property of two derivations (eg cut-admissibility!)

$$\frac{cl \in derrec\ rlsl\ \{\} \qquad cr \in derrec\ rlsr\ \{\} \qquad \forall (lps, lc) \in rlsl.\ \forall (rps, rc) \in rlsr.\ (\forall lp \in lps.\ P\ lp\ rc) \wedge (\forall rp \in rps.\ P\ lc\ rp) \Rightarrow P\ lc\ rc}{P\ cl\ cr}$$

to prove $P(\mathcal{C}_l, \mathcal{C}_r)$, the induction hypothesis is that $P(\mathcal{P}_{li}, \mathcal{C}_r)$ and $P(\mathcal{C}_l, \mathcal{P}_{rj})$ hold for all i and j:

$$\dfrac{\dfrac{\mathcal{P}_{l1} \dots \mathcal{P}_{ln}}{\mathcal{C}_l}\rho_l \qquad \dfrac{\mathcal{P}_{r1} \dots \mathcal{P}_{rm}}{\mathcal{C}_r}\rho_r}{?}\ (cut\ ?)$$

Sequents, Formulae and Rules

formula language: connectives, variables and primitive propositions:

```
datatype formula = FC string (formula list) (* connective *
                 | FV string                 (* variable *)
                 | PP string      (* primitive proposition *
```

A sequent is a pair of multisets of formulae, written $\Gamma \vdash \Delta$.
Given a rule such as $(\vdash \wedge)$ in the two forms below,

$$\mathcal{C}_s = \frac{\vdash A \quad \vdash B}{\vdash A \wedge B} \qquad \mathcal{C}_e = \frac{X \vdash Y, A \quad X \vdash Y, B}{X \vdash Y, A \wedge B}$$

we call $\mathcal{C}_e$ an *extension* of $\mathcal{C}_s$: $X \vdash Y =$ `extend` $(X \vdash Y)\ (\vdash A)$
`pscmap` f applies f to premises and conclusion,
so, using $+$ for multiset union,

$$\texttt{extend}\ (X \vdash Y)\ (U \vdash V) = (X + U) \vdash (Y + V)$$
$$\mathcal{C}_e = \texttt{pscmap}\ (\texttt{extend}\ (X \vdash Y))\ \mathcal{C}_s$$

The GLS Rules

Then we define `glss`, the set of rules of GLS by defining:

- `glil` and `glir`: the unextended left and right introduction rules, like $\mathcal{C}_s$ above;
- `wkrls` and `ctrrls` A: the unextended weakening and contraction (on A) rules;
- `glne`: all of the above;
- `glr` B: the GLR(B) rule;
- `glss`: the axiom $A \vdash A$ (not requiring A to be atomic), the GLR(B) rule for all B, and all extensions of all rules in `glne`.

An Axiomatic Type Class for Multisets and Sequents

the class pm0

ordering $\leq$ on multisets analogous to $\subseteq$ for sets: $N \leq M$ if, for all x, N contains no more occurrences of x than does M.

We define a type class pm0:

For any type in class pm0, the operations $+$ and 0 form a commutative monoid and the following two properties hold.

$$A + B - A = B \qquad A - B - C = A - (B + C)$$

```
axclass pm0 < comm_monoid_add, minus
  pm0_plus_minus  : "A + B - A = B"
  pm0_minus_minus : "A - B - C = A - (B + C)"
```


An Axiomatic Type Class for Multisets and Sequents

the class pm_ge0

class pm_ge0: it also has $\leq$ and 0, axioms of pm0 and these:

$$0 \leq A \qquad B \leq A \Rightarrow B + (A - B) = A$$
$$m \leq n \Leftrightarrow m - n = 0 \qquad x < y \Leftrightarrow x \leq y \wedge x \neq y \qquad a \sqsubseteq b \Leftrightarrow a \leq b$$

Lemma

Multisets are in pm0 and pm_ge0 using our definition of $\leq$, and, if Γ and Δ are of any type in the classes pm0 or pm_ge0, then so is sequent $\Gamma \vdash \Delta$.

This class in fact gives us a lattice

Lemma

Any type of class pm_ge0 forms a lattice, using the definitions

$$c \wedge d = c - (c - d) \qquad c \vee d = c + (d - c)$$

Simplification Procedures for Multisets and Sequents

Isabelle has "simplification procedures":

- $a - b + c + b$ to $a + c$ (integers)
- $a + b + c - b$ to $a + c$ (integers *or* naturals)

We applied most of the simplification procedures for naturals to types of the classes pm0 and pm_ge0

The Induction Pattern in Cut-Admissibility Proofs

Definition of gen_step2ssr

In the diagram below, to prove $P(\mathcal{C}_l, \mathcal{C}_r)$, the induction hypothesis is that $P(\mathcal{P}_{li}, \mathcal{C}_r)$ and $P(\mathcal{C}_l, \mathcal{P}_{rj})$ hold for all i and j:

$$\dfrac{\dfrac{\mathcal{P}_{l1} \ldots \mathcal{P}_{ln}}{\mathcal{C}_l} \mathcal{R}_l \qquad \dfrac{\mathcal{P}_{r1} \ldots \mathcal{P}_{rm}}{\mathcal{C}_r} \mathcal{R}_r}{?} \ (cut\ ?)$$

gen_step2ssr expresses that property P holds, given appropriate inductive hypotheses, for last rules on each side $\mathcal{R}_l$ and $\mathcal{R}_r$.
P might be that cut-admissibility holds for cut-formula A, rule set `rls`, assuming it holds for smaller (subformula relation `sub`)

The Induction pattern in Cut-Admissibility Proofs

Definition of gen_step2ssr

Definition (gen_step2ssr)

For a formula A, a property P, a subformula relation `sub`, a set of rules `rls`, inference rule instances $\mathcal{R}_l = (\mathcal{P}_{l1} \ldots \mathcal{P}_{ln}, \mathcal{C}_l)$ and $\mathcal{R}_r = (\mathcal{P}_{r1} \ldots \mathcal{P}_{rm}, \mathcal{C}_r)$, `gen_step2ssr P A sub rls` $(\mathcal{R}_l, \mathcal{R}_r)$ means:

if forall A' such that $(A', A) \in$ `sub` and all `rls`-derivable sequents $\mathcal{D}_l$ and $\mathcal{D}_r$, $P\ A'\ (\mathcal{D}_l, \mathcal{D}_r)$ holds
and for each $\mathcal{P}_{li}$ in $\mathcal{P}_{l1} \ldots \mathcal{P}_{ln}$, $P\ A\ (\mathcal{P}_{li}, \mathcal{C}_r)$ holds
and for each $\mathcal{P}_{rj}$ in $\mathcal{P}_{r1} \ldots \mathcal{P}_{rm}$, $P\ A\ (\mathcal{C}_l, \mathcal{P}_{rj})$ holds
then $P\ A\ (\mathcal{C}_l, \mathcal{C}_r)$ holds.

The Induction pattern in Cut-Admissibility Proofs

Theorem using gen_step2ssr

The theorem gen_step2ssr_lem for P states that if the step of the inductive proof holds for all cases of final rules $\mathcal{R}_l$ and $\mathcal{R}_r$ on each side, then P holds in all cases.

Theorem (gen_step2ssr_lem)

If

- *A is in the well-founded part of the subformula relation `sub`,*
- *sequents $\mathcal{S}_l$ and $\mathcal{S}_r$ are `rls`-derivable, and*
- *for all formulae A', and all rules $\mathcal{R}_l$ and $\mathcal{R}_r$, our induction step condition `gen_step2ssr P A' sub rls` $(\mathcal{R}_l, \mathcal{R}_r)$ holds*

then $P\ A\ (\mathcal{S}_l, \mathcal{S}_r)$ holds.

The Induction pattern in Cut-Admissibility Proofs

Lemma for the left parametric case

Inductive step where the cut-formula A is parametric on the left. (`prop2 mar erls` A $(\mathcal{C}_l, \mathcal{C}_r)$ means that the conclusion of a multicut on `A` with premises $\mathcal{C}_l$ and $\mathcal{C}_r$ is derivable using rules `erls`)

Theorem (lmg_gen_steps)

For any relation `sub` and any rule set `rls`, given an instance of multicut with left and right subtrees ending with rules $\mathcal{R}_l$ and $\mathcal{R}_r$:

if *weakening is admissible for the rule set `erls`,*
and all extensions of some rule $(\mathcal{P}, X \vdash Y)$ are in the rule set `erls`,
and $\mathcal{R}_l$ is an extension of $(\mathcal{P}, X \vdash Y)$,
and the cut-formula A is not in Y (meaning that A is parametric on the left)
then *`gen_step2ssr (prop2 mar erls) A sub rls` $(\mathcal{R}_l, \mathcal{R}_r)$ holds.*

The proof of Goré & Ramanayake, and our proof

The proof of Goré & Ramanayake

- Proves admissibility of (*cut*) (we prove admissibility of (*multicut*))
- Induction on height of derivation and on "width"
- Induction on size of cut-formula.

In contrast, in our proof

- we prove admissibility of (*multicut*)
- Induction on "fact of" derivation and on `del0` (approximates to ∂^0, related to width)
- Well-founded induction on immediate subformula relation

Using a deep embedding — explicit derivation trees

To define `del0` on a derivation we need an explicit derivation tree

A *valid* tree is one whose inferences are in the set of rules and which as a whole has no premises.

Lemma

Sequent $X \vdash Y$ is derivable, shallowly, from the empty set of premises using rules `rls` (ie, is in `derrec rls {}`) iff some explicit derivation tree `dt` is valid wrt. `rls` and has a conclusion $X \vdash Y$.

```
"(?a : derrec ?rls {}) =
   (EX dt. valid ?rls dt & conclDT dt = ?a)"
```

"mix and match" a deep embedding (derivation trees) with a shallow embedding (inductively defined sets of derivable sequents)

Defining del0

Definition (del0)

For derivation tree `dt` and formula B, define `del0` B `dt`:

- if the bottom rule of `dt` is $GLR(Y, A)$ (for *any* Y, A), then `del0` B `dt` is 1 (0) if $\Box B$ is (is not) in the antecedent of the conclusion of `dt`
- if the bottom rule of `dt` is not GLR, then `del0` B `dt` is obtained by summing `del0` B `dt'` over all premise subtrees `dt'` of `dt`.

ie, you go up each branch of an explicit derivation tree until you find an instance of the GLR rule, and count 1 where B is in Y

$$\frac{\Box Y, Y, \Box A \vdash A}{\Box Y \vdash \Box A}$$

The Proof

Lemma

If μ is a valid derivation tree with conclusion $\Box X, X, \Box B \vdash B$, and `del0` $B\ \mu = 0$, then $\Box X, X \vdash B$ is derivable.

Proof.

Applying the GLR rule to the $\Box X, X, \Box B \vdash B$ gives $\Box X \vdash \Box B$.
Tracing upwards, change each $\Box B$ to $\Box X$ in the usual way.
Contraction is not problematic since we use, as the inductive hypothesis, that *all* occurrences of $\Box B$ can be replaced by $\Box X$. □

Defining muxbn

$$\dfrac{\mu\left\{\dfrac{\Pi_l}{\Box X, X, \Box B \vdash B}\right.}{\Box X \vdash \Box B}\,GLR(B) \qquad \dfrac{\Pi_r}{\Box B^k, Y \vdash Z}\,\rho$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$\Box X, Y \vdash Z \quad (multicut\ ?)$$

Figure: A multicut on cut formula $\Box B$ where $\Box B$ is left-principal via GLR

Definition (muxbn)

`muxbn` B n holds iff: for all instances of Figure 1 (for fixed B) such that `del0` B $\mu \leq n$, the multicut in Figure 1 is admissible.

Proofs of muxbn

Lemma

If μ is a valid derivation tree with conclusion $\Box X, X, \Box B \vdash B$, and `del0` B $\mu = 0$, and multicut on B is admissible, and $\Box B^k, Y \vdash Z$ is derivable, then $\Box X, Y \vdash Z$ is derivable.
That is, if multicut on B is admissible, then `muxbn` B 0 holds.

Proof.

$\Box X \vdash \Box B$ is derivable from $\Box X, X, \Box B \vdash B$ via $GLR(X, B)$. By Lemma 8, $\Box X, X \vdash B$ is derivable. The rest of the proof is by induction on the derivation of $\Box B^k, Y \vdash Z$, in effect, by tracing relevant occurrences of $\Box B$ up that derivation.
Suppose an inference $GLR(Y, C)$ is encountered, with B in Y.
(see next slide) □

$$\dfrac{\Box B^k, B^k, \Box Z, Z, \Box C \vdash C}{\Box B^k, \Box Z \vdash \Box C}\,GLR(Y, C)$$

Z is Y with B deleted.
By induction, $\Box X, B^k, \Box Z, Z, \Box C \vdash C$ is derivable.
From there we have the derivation shown below.

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{Lemma 8}}{\Box X, X \vdash B} \qquad \Box X, B^k, \Box Z, Z, \Box C \vdash C}{\Box X, \Box X, X, \Box Z, Z, \Box C \vdash C}\,mcut(B)}{\Box X, X, \Box Z, Z, \Box C \vdash C}\,ctr}{\Box X, \Box Z \vdash \Box C}\,GLR(C)$$

Additional weakening steps necessary if $\Box B$ in Z or if B in $\Box Z$ (shown by machine-checking!)

From muxbn B n to muxbn B $(n+1)$

$$\dfrac{\mu\left\{\dfrac{\Pi_l}{\Box X, X, \Box B \vdash B}\right.}{\Box X \vdash \Box B}\,GLR(B)$$

Suppose `del0` B $\mu = n + 1$.
Since `del0` B $\mu > 0$, the tree $\mu/\Box X \vdash \Box B$ contains one or more branches with a GLR rule, with $\Box B$ in the antecedent. (one such branch shown).

$$\dfrac{\Box G, G, \Box B^k, B^k, \Box A \vdash A}{\Box G, \Box B^k \vdash \Box A}\,GLR(A)$$
$$\vdots$$
$$\dfrac{\Box X, X, \Box B \vdash B}{\Box X \vdash \Box B}\,GLR(X, B)$$

From muxbn B n to muxbn B $(n+1)$

$$\dfrac{\Box G, G, \Box B^k, B^k, \Box A \vdash A}{\Box G, \Box B^k \vdash \Box A}\,GLR(A)\ \text{(delete this)}$$
$$\vdots$$
$$\dfrac{\Box X, X, \Box B \vdash B}{\Box X \vdash \Box B}\,GLR(X, B)$$

Delete top step, adjoin $\Box A$ on the left, extra weakening step:

$$\dfrac{}{\Box A, \Box G, \Box B^k \vdash \Box A}$$
$$\vdots$$
$$\dfrac{\dfrac{\Box A, \Box X, X, \Box B \vdash B}{\Box A, A, \Box X, X, \Box B \vdash B}\,(weakening)\ \text{(extra step)}}{\Box A, \Box X \vdash \Box B}\,GLR(B)$$

Call this $\mu^A/\Box A, \Box X \vdash \Box B$, then `del0` B $\mu >$ `del0` B μ^A, so $\mu^A/\Box A, \Box X \vdash \Box B$ can be left branch of an admissible multicut.

Multicutting with $\Box A, \Box X \vdash \Box B$

$$\dfrac{\Box A, \Box X \vdash \Box B \qquad \dfrac{}{\Box X, X, \Box B \vdash B}}{\Box A, \Box X, X \vdash B}\,(multicut + ctr)$$

$$\dfrac{\Box A, \Box X \vdash \Box B \qquad \dfrac{}{\Box G, G, \Box B^k, B^k, \Box A \vdash A}}{\Box G, G, \Box X, B^k, \Box A \vdash A}\,(multicut + ctr)$$

Now, multicut on B (smaller cut-formula), and contraction, gives

$$\dfrac{\dfrac{\Box G, G, \Box A, \Box X, X \vdash A}{\Box G, \Box X \vdash \Box A}\,GLR}{\Box G, \Box X, \Box B^k \vdash \Box A}\,(weakening)$$

From `del0` B $\mu = n + 1$ to `del0` B $\mu' = n$

$$\dfrac{}{\Box G, \Box B^k \vdash \Box A}$$
$$\vdots$$
$$\dfrac{}{\Box X, X, \Box B \vdash B}$$

We use this proof again, now adjoin $\Box X$ on the left, to get

$$\dfrac{\text{previous slide}}{\Box X, \Box G, \Box B^k \vdash \Box A}$$
$$\vdots$$
$$\dfrac{\Box X, \Box X, X, \Box B \vdash B}{\Box X, X, \Box B \vdash B}\,(contraction)$$

That is, given a derivation μ of $\Box X, X, \Box B \vdash B$ with `del0` B $\mu = n + 1$, we have a derivation μ' with `del0` B $\mu' = n$.

Wrapping it up

Lemma

Assume that multicut-admissibility holds for cut-formula B, and that `muxbn` B n holds. Then `muxbn` B $(n+1)$ holds.

Proof.

See the Figure: given μ, where `del0` B $\mu = n + 1$, we can replace it by by μ', where `del0` B $\mu' = n$. Since `muxbn` B n holds, the multicut in the Figure is admissible, as required. □

Now, since `muxbn` B 0 holds, repeated use of this Lemma gives that `muxbn` B n for all n.

The cut-admissibility theorem

Theorem

Multicut is admissible in GLS.

Proof.

Most of the proof is as usual for cut-elimination proofs, using induction on the size (or structure) of the cut-formula. The difficult case is with a multicut as in the Figure, which is handled by the previous lemma. □

Introduction Sequents, Multisets, Sets and Provability Logic Reasoning About Derivations and Derivability An Axiomatic Type Cla

Conclusion : value of the formalisation

- proofs usually tedious, with many details varying only slightly
- many cases or details usually omitted in paper proofs
- this may lead to erroneous proofs
- formal proof avoids this risk

Our formalisation includes:

- formalisation includes general treatment of derivation trees
- general theorem expressing the appropriate inductive principle
- general lemmas for many cases in this and other proofs