Sequent Calculus: Classical Arithmetic (Lecture 5)

Dale Miller

INRIA Saclay & Ecole Polytechnique, France

Pisa, 15 April 2014

Presenting and applying a focused proof system for classical logic.

Arithmetic via equality and fixed points

We shall add

- first-order *term equality*
  following Girard [1992] and Schroeder-Heister [1993], and
- *fixed points* (for recursive definitions)
  following Baelde, McDowell, M, Tiu [1996-2008].

They will both be *logical connectives:* that is, they are defined by introduction rules.

Equality as logical connective

Introductions in an unfocused setting.

$$\frac{}{\vdash \Theta, t = t} \qquad \frac{}{\vdash \Theta, s \neq t}\ \ddagger \qquad \frac{\vdash \Theta\sigma}{\vdash \Theta, s \neq t}\ \dagger$$

$\ddagger$ s and t are not unifiable.

$\dagger$ s and t to be unifiable and σ to be their mgu

Equality as logical connective

Introductions in an unfocused setting.

$$\frac{}{\vdash \Theta, t = t} \qquad \frac{}{\vdash \Theta, s \neq t}\ \ddagger \qquad \frac{\vdash \Theta\sigma}{\vdash \Theta, s \neq t}\ \dagger$$

$\ddagger$ s and t are not unifiable.

$\dagger$ s and t to be unifiable and σ to be their mgu

Introductions in a focused setting.

$$\frac{}{\vdash \Theta \Downarrow t = t} \qquad \frac{}{\vdash \Theta \Uparrow \Gamma, s \neq t}\ \ddagger \qquad \frac{\vdash \Theta\sigma \Uparrow \Gamma\sigma}{\vdash \Theta \Uparrow \Gamma, s \neq t}\ \dagger$$

N.B. Unification was used before to *implement* inference rules: here, unification is in the *definition* of the rule.

Some theorems about equality

Equality is an equivalence relation...

- $\forall x\ [x = x]$
- $\forall x, y\ [x = y \supset y = x]$
- $\forall x, y, z\ [x = y \wedge y = z \supset x = z]$

and a congruence.

- $\forall x, y\ [x = y \supset (f\ x) = (f\ y)]$
- $\forall x, y\ [x = y \supset (p\ x) \supset (p\ y)]$

Let 0 denote zero and s denote successor.

- $\forall x\ [0 \neq (s\ x)]$
- $\forall x, y\ [(s\ x) = (s\ y) \supset x = y]$

A hint of model checking

Encode a non-empty set of first order terms $S = \{s_1, \ldots, s_n\}$ $(n \geq 1)$ as the one-place predicate

$$\hat{S} = [\lambda x.\ x = s_1 \vee^+ \cdots \vee^+ x = s_n]$$

If S is empty, then define $\hat{S}$ to be $[\lambda x.\ f^+]$. Notice that

$$s \in S \quad \text{if and only if} \quad \vdash \ \vdash \cdot \Uparrow \hat{S}\ s.$$

A hint of model checking

Encode a non-empty set of first order terms $S = \{s_1, \ldots, s_n\}$ $(n \geq 1)$ as the one-place predicate

$$\hat{S} = [\lambda x.\ x = s_1 \vee^+ \cdots \vee^+ x = s_n]$$

If S is empty, then define $\hat{S}$ to be $[\lambda x.\ f^+]$. Notice that

$$s \in S \quad \text{if and only if} \quad \vdash \ \ \vdash \cdot \Uparrow \hat{S}\ s.$$

The statement

$$\forall x \in \{s_1, \ldots, s_n\}.P(x) \quad \text{becomes} \quad \vdash \cdot \Uparrow \forall x.[\hat{S}x \supset Px].$$

$$\dfrac{\dfrac{\vdash P(s_1) \Uparrow \cdot}{\vdash P(x) \Uparrow x \neq s_1} \qquad \cdots \qquad \dfrac{\vdash P(s_n) \Uparrow \cdot}{\vdash P(x) \Uparrow x \neq s_n}}{\vdash \cdot \Uparrow \forall x.[x \neq s_1 \wedge^- \cdots \wedge^- x \neq s_n] \vee^- P(x)}$$

Fixed Points as connectives

The *fixed points* operators μ and ν are De Morgan duals and simply unfold.

$$\frac{\vdash \Theta \Uparrow \Gamma, B(\nu B)\bar{t}}{\vdash \Theta \Uparrow \Gamma, \nu B\bar{t}} \qquad \frac{\vdash \Theta \Downarrow B(\mu B)\bar{t}}{\vdash \Theta \Downarrow \mu B\bar{t}}$$

B is a formula with $n \geq 0$ variables abstracted; $\bar{t}$ is a list of n terms.

Here, μ denotes neither the least nor the greatest fixed point. That distinction arises if we add induction and co-induction.

Examples of fixed points

Natural numbers: terms over 0 for zero and s for successor. Two ways to define predicates over numbers.

$$\begin{array}{rcl} nat\ 0 & \texttt{:-} & true. \\ nat\ (s\ X) & \texttt{:-} & nat\ X. \\ leq\ 0\ Y & \texttt{:-} & true. \\ leq\ (s\ X)\ (s\ Y) & \texttt{:-} & leq\ X\ Y. \end{array}$$

These logic programs can be given as fixed point expressions.

$$nat = \mu(\lambda p \lambda x.(x = 0) \vee^{+} \exists y.(s\ y) = x \wedge^{+} p\ y)$$

$$leq = \mu(\lambda q \lambda x \lambda y.(x = 0) \vee^{+} \exists u \exists v.(s\ u) = x \wedge^{+} (s\ v) = y \wedge^{+} q\ u\ v).$$

Horn clauses can be made into fixed point specifications (mutual recursions requires standard encoding techniques).

Putting computation into an inference rule

Consider proving the positive focused sequent

$$\vdash \Theta \Downarrow (leq\ \ m\ \ n \wedge^{+} N_1) \vee^{+} (leq\ \ n\ \ m \wedge^{+} N_2),$$

where m, n are natural numbers and N_1, N_2 are negative formulas. There are exactly two possible macro rules:

$$\frac{\vdash \Theta \Downarrow N_1}{\vdash \Theta \Downarrow (leq\ \ m\ \ n \wedge^{+} N_1) \vee^{+} (leq\ \ n\ \ m \wedge^{+} N_2)} \text{ for } m \leq n$$

$$\frac{\vdash \Theta \Downarrow N_2}{\vdash \Theta \Downarrow (leq\ \ m\ \ n \wedge^{+} N_1) \vee^{+} (leq\ \ n\ \ m \wedge^{+} N_2)} \text{ for } n \leq m$$

A macro inference rule can contain an entire Prolog-style computation.

One step transitions in CCS

As inference rules in SOS (structured operational semantics):

$$\frac{}{A.P \xrightarrow{A} P} \qquad \frac{P \xrightarrow{A} R}{P + Q \xrightarrow{A} R} \qquad \frac{Q \xrightarrow{A} R}{P + Q \xrightarrow{A} R}$$

$$\frac{P \xrightarrow{A} P'}{P|Q \xrightarrow{A} P'|Q} \qquad \frac{Q \xrightarrow{A} Q'}{P|Q \xrightarrow{A} P|Q'}$$

These can easily be written as Prolog clauses and as a fixed point definition.

The engineering of proof systems (cont)

Consider proofs involving simulation.

$$sim\ P\ Q \ \equiv\ \forall P' \forall A[\ P \xrightarrow{A} P' \supset \exists Q'\ [Q \xrightarrow{A} Q' \wedge sim\ P'\ Q']].$$

Typically, $P \xrightarrow{A} P'$ is given as a table or as a recursion on syntax (*e.g.*, CCS): hence, as a fixed point.

The body of this expression is exactly two "macro connectives".

- $\forall P' \forall A[P \xrightarrow{A} P' \supset \cdot\ ]$ is a negative "macro connective". There are no choices in expanding this macro rule.
- $\exists Q'[Q \xrightarrow{A} Q' \wedge^{+} \cdot\ ]$ is a positive "macro connective". There can be choices for continuation Q'.

These macro-rules now match exactly the sense of simulation.

Future work: Broad spectrum proof certificates

Sequent calculus and focusing proof systems provide:

- The *atoms* of inference (the introduction rules)
- The structure of focusing provides us with the *rules of chemistry*: which atoms stick together and which do not.
- Engineered proofs system made form the *molecules* of inference.

Future work: Broad spectrum proof certificates

Sequent calculus and focusing proof systems provide:

- The *atoms* of inference (the introduction rules)
- The structure of focusing provides us with the *rules of chemistry*: which atoms stick together and which do not.
- Engineered proofs system made form the *molecules* of inference.

An approach to a general notion of *proof certificate*:

- The world's provers print their proof evidence using appropriately engineered molecules of inference.
- A universal proof checker implements only the atoms of inference and the rules of chemistry.

Future work: Broad spectrum proof certificates

Sequent calculus and focusing proof systems provide:

- The *atoms* of inference (the introduction rules)
- The structure of focusing provides us with the *rules of chemistry*: which atoms stick together and which do not.
- Engineered proofs system made form the *molecules* of inference.

An approach to a general notion of *proof certificate*:

- The world's provers print their proof evidence using appropriately engineered molecules of inference.
- A universal proof checker implements only the atoms of inference and the rules of chemistry.

See the two recent draft submissions:

- "Communicating and trusting proofs: The case for broad spectrum proof certificates"
- "A proposal for broad spectrum proof certificates"