Equality and fixed points as logical connectives

Dale Miller

INRIA-Saclay & LIX, École Polytechnique
Palaiseau, France

Bertinoro, 11-15 April 2011

Lecture 6: Equality and fixed points in proof systems.

Principle references: Baelde, "On the proof theory of regular fixed points", Tableaux 2009, LNAI 5607, pp. 93–107. Baelde & M, "Least and greatest fixed points in linear logic" LPAR 2007, LNCS 4790, pp. 92-106.

Classical logic and one-sided sequents

We shall now restrict our attention to classical logic.

We follow two conventions when dealing with classical logic.

- We shall assume that formulas are always placed into *negation normal form*: that is, negation will have only atomic scopes.
- Sequents will be one-sided. In particular, the two sided sequent

$$\Sigma : B_1, \ldots, B_n \vdash C_1, \ldots, C_m$$

will be converted to

$$\Sigma : \quad \vdash \neg B_1, \ldots, \neg B_n, C_1, \ldots, C_m.$$

Equality as logical connective

"It's a logical connective when it has introduction rules and satisfies cut-elimination."

Introductions in an unfocused setting.

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

Introductions in a focused setting.

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

$\ddagger$ s and t are not unifiable.

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

N.B. Unification was used before to *implement* inference rules. Here, it is in the *definition* of the inference 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 the empty set, the set $\hat{S}$ to be $[\lambda x.\ \bot]$. Notice that

$$s \in S \quad \text{if and only if} \quad \vdash_C \ \hat{S}\ s.$$

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 the empty set, the set $\hat{S}$ to be $[\lambda x.\ \perp]$. Notice that

$$s \in S \quad \text{if and only if} \quad \vdash_C \ \hat{S}\ s.$$

Now let $T = \{t_1, \ldots, t_m\}$. $S \subseteq T$ if and only if $\vdash_c \forall x.[\hat{S}x \supset \hat{T}x]$.

- What does this proof look like in a focused setting (suggestion: polarize the disjunctions positively).
- Can you encode n-ary relations instead just sets? Labelled transition systems?

Fixed Points as connectives

Of course, the sets and relations we can encode in this style are finite. Let us introduce the *fixed point* operator. It simply does unfolding.

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

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.

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

Above, as a logic program and below, as fixed points.

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

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

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.

Some references

[1] J.-M. Andreoli. Logic programming with focusing proofs in linear logic. *J. of Logic and Computation*, 1992.

[2] C. Liang and D. Miller. Focusing and polarization in linear, intuitionistic, and classical logics. *Theoretical Computer Science*, 2009.

[3] D. Miller and V. Nigam. Incorporating tables into proofs. *CSL 2007: Computer Science Logic*, LNCS 4646.

[4] V. Nigam and D. Miller. A framework for proof systems, *J. of Automated Reasoning*, 2010.

[5] D. Miller, Unity in Computational Logic, ACM-BCS-Visions Conference, 2010.

[6] D. Miller, Communicating and trusting proofs: Towards a broad spectrum proof certificate, draft available on my web page.