Induction and Coinduction in Sequent Calculus

Alwen Tiu
École Polytechnique and Penn State University

Joint work with
Dale Miller (*INRIA/École Polytechnique*)
Alberto Momigliano (*University of Leicester*)

July 3, 2003

Outline

1. Motivations
2. Deductive systems as logical specifications
3. Reasoning in proof search
4. A design of logic
5. A notion of definition
6. A fixed point interpretation of definitions
7. Induction and coinduction
8. Consistency
9. Examples
10. Related work
11. Conclusion and future work

Motivations

- Using logic to *specify* and to *reason about* deductive systems, e.g., sequent calculus, structured or natural operational semantics, etc.

- We are interested in formalizing structural induction and reasoning methods for non-finite behaviors (e.g., bisimulation). The latter typically involves coinduction.

Deductive systems as logical speficiations

- The static structures of a deductive system, i.e., its syntactic expressions, are encoded as terms in logic. The dynamic structures, i.e., its inference rules, can be encoded as logical theories, which typically involves a simple class of formula, e.g. Horn clauses.

- Consider a fragment of an operational semantics for imperative languages

$$\frac{B \Downarrow \mathit{true} \qquad M \Downarrow V}{(\mathit{if}\ B\ \ M\ \ N) \Downarrow V} \qquad\qquad \frac{B \Downarrow \mathit{false} \qquad N \Downarrow V}{(\mathit{if}\ B\ \ M\ \ N) \Downarrow V} \quad .$$

  These inference rules can be specified as the following Horn clauses:

$$\begin{array}{ll} \forall B \forall M \forall N \forall V [B \Downarrow \mathit{true} \wedge M \Downarrow V & \supset (\mathit{if}\ B\ \ M\ \ N) \Downarrow V] \\ \forall B \forall M \forall N \forall V [B \Downarrow \mathit{false} \wedge N \Downarrow V & \supset (\mathit{if}\ B\ \ M\ \ N) \Downarrow V] \end{array}$$

Reasoning in proof search

- Properties of a logical specification are expressed as logical formulas, e.g.,
$$\forall B \forall M \forall V. M \Downarrow V \supset (\textit{if } B\ M\ M) \Downarrow V$$
and *proof search* is used to verify if the properties hold.

- Advantages: formal proofs, (partial) proof automation, proof generalization, better syntax.

- The properties we can prove depend on the strength of the (meta) logic. Typical interesting properties involves the use of *structural induction* (e.g., subject reduction) or *coinduction* (e.g., bisimulation) as proof methods. We consider making these proof methods explicit in a proof system, as inference rules.

A design of logic

- We currently focus on developing the proof theory part, no formal semantics yet.
- Guidelines for the design: *cut-elimination*, and examples and applications. The latter is mostly drawn from previous works by Miller and McDowell on encoding abstract transition systems in sequent calculus.
- The core logic is intuitionistic logic where formulas are of type o (following Church) and we allow quantification on higher-order type, as long as it does not contain the type o.

The core inference rules

$$\frac{}{A, \Gamma \longrightarrow A}\, init \qquad \frac{\Delta \longrightarrow B \quad B, \Gamma \longrightarrow C}{\Delta, \Gamma \longrightarrow C}\, cut$$

$$\frac{}{\bot, \Gamma \longrightarrow B}\, \bot\mathcal{L} \qquad \frac{}{\Gamma \longrightarrow \top}\, \top\mathcal{R} \qquad \frac{B, B, \Gamma \longrightarrow C}{B, \Gamma \longrightarrow C}\, c\mathcal{L}$$

$$\frac{B, \Gamma \longrightarrow D}{B \wedge C, \Gamma \longrightarrow D}\, \wedge\mathcal{L} \qquad \frac{C, \Gamma \longrightarrow D}{B \wedge C, \Gamma \longrightarrow D}\, \wedge\mathcal{L} \qquad \frac{\Gamma \longrightarrow B \quad \Gamma \longrightarrow C}{\Gamma \longrightarrow B \wedge C}\, \wedge\mathcal{R}$$

$$\frac{B, \Gamma \longrightarrow D \quad C, \Gamma \longrightarrow D}{B \vee C, \Gamma \longrightarrow D}\, \vee\mathcal{L} \qquad \frac{\Gamma \longrightarrow B}{\Gamma \longrightarrow B \vee C}\, \vee\mathcal{R} \qquad \frac{\Gamma \longrightarrow C}{\Gamma \longrightarrow B \vee C}\, \vee\mathcal{R}$$

$$\frac{\Gamma \longrightarrow B \quad C, \Gamma \longrightarrow D}{B \supset C, \Gamma \longrightarrow D}\, \supset \mathcal{L} \qquad \frac{B, \Gamma \longrightarrow C}{\Gamma \longrightarrow B \supset C}\, \supset \mathcal{R}$$

$$\frac{B[y/x], \Gamma \longrightarrow C}{\exists x.B, \Gamma \longrightarrow C}\, \exists\mathcal{L} \qquad \frac{\Gamma \longrightarrow B[t/x]}{\Gamma \longrightarrow \exists x.B}\, \exists\mathcal{R}$$

$$\frac{B[t/x], \Gamma \longrightarrow C}{\forall x.B, \Gamma \longrightarrow C}\, \forall\mathcal{L} \qquad \frac{\Gamma \longrightarrow B[y/x]}{\Gamma \longrightarrow \forall x.B}\, \forall\mathcal{R}$$

A notion of definition

We extend the core logic by allowing non-logical constants to be introduced. To each predicate p, we associate a *definition clause*

$$\forall \bar{x}. p\, \bar{x} \triangleq B\, \bar{x}$$

where $B\, \bar{x}$ is some formula. We call $p\, \bar{x}$ the head of the definition and $B\, \bar{x}$ the body. A *definition* is a collection of definition clauses. The notion of definitions has been previously studied by Schroeder-Heister, Eriksson, Girard, Miller and McDowell. Given some stratifications on definitions (e.g., the head of a definition cannot occur negatively in the body), we can prove cut-elimination.

Definition and equality

Notice that in the notion of definition shown before there are no pattern matching on the head of the definition; they are encoded in the body, e.g., to encode a predicate nat to express natural numbers we write

$$\mathit{nat}\ x \stackrel{\triangle}{=} [x = 0] \vee \exists y.[x = (sy)] \wedge \mathit{nat}\ y,$$

instead of the more familiar definition

$$\begin{array}{lcl} \mathit{nat}\ 0 & \stackrel{\triangle}{=} & \top \\ \mathit{nat}\ (sx) & \stackrel{\triangle}{=} & \mathit{nat}\ x. \end{array}$$

This requires us to take equality predicate as primitive. Both presentations are operationally equivalent. However, the former presentation allows for a simpler formulation of the (co)induction rules to be introduced later.

Introduction rules for definitions and equality

- Given a definition $p\,\bar{x} \stackrel{\triangle}{=} B\,\bar{x}$, the introduction rules for p are

$$\frac{B\,\bar{t}, \Gamma \longrightarrow C}{p\,\bar{t}, \Gamma \longrightarrow C}\ \mathit{def}\mathcal{L} \qquad \frac{\Gamma \longrightarrow B\,\bar{t}}{\Gamma \longrightarrow p\,\bar{t}}\ \mathit{def}\mathcal{R}$$

- The rules for equality

$$\frac{\{\Gamma\theta \longrightarrow C\theta \mid \ s\theta =_{\beta\eta} t\theta\}}{[s = t], \Gamma \longrightarrow C}\ \mathrm{eq}\mathcal{L} \qquad\qquad \frac{}{\Gamma \longrightarrow [t = t]}\ \mathrm{eq}\mathcal{R}$$

That is, on the right, pattern matching is used; on the left, we use unification. Note that *eigenvariables can be instantiated* in $\mathrm{eq}\mathcal{L}$.

Encoding logical specifications as definitions

- Example: consider a fragment of the operational semantics for eval

$$
\begin{array}{rcl}
M \Downarrow V & \triangleq & \ldots \\
& & (\exists B, M', N.[M = (\textit{if}\ B\ \ M'\ \ N)] \wedge B \Downarrow \textit{true} \wedge M' \Downarrow V) \vee \\
& & (\exists B, M', N.[M = (\textit{if}\ B\ \ M'\ \ N)] \wedge B \Downarrow \textit{false} \wedge N \Downarrow V) \vee \\
& & \ldots
\end{array}
$$

- Prove the statement:

$$\forall B \forall M \forall V. M \Downarrow V \supset (\textit{if}\ B\ \ M\ \ M) \Downarrow V$$

A fixed point interpreation of definitions

A definition clause can be seen as expressing a fixed point equation. That is, a definition $p\,\bar{x} \stackrel{\triangle}{=} B\,\bar{x}$ can be read as [Girard]

"$p\,\bar{x}$ if and only if $\bar{x}$ is some terms $\bar{t}$ such that $B\,\bar{x}$ holds".

In other words, provability of a judgment

$$\longrightarrow p\,\bar{t}$$

expresses the fact that $p\,\bar{t}$ is in a solution (not necessarily the least one) of the corresponding fixed point equation of p. Stratification of definitions ensures that each definition is monotone. Hence, we can generalize the rules for definition to capture least fixed points (induction) and greatest fixed points (coinduction).

Induction and Coinduction

- Based on fixed point interpretation, the induction rules make use of the notion of *pre-fixed point*, or invariants. Given a definition clause $p\,\bar{x} \stackrel{\triangle}{=} B\,\bar{x}$ the induction rules for p are

$$\frac{B_I\bar{x} \longrightarrow I\,\bar{x} \quad I\,\bar{t}, \Gamma \longrightarrow C}{p\,\bar{t}, \Gamma \longrightarrow C}\,\mathcal{IL} \qquad\qquad \frac{\Gamma \longrightarrow B\,\bar{t}}{\Gamma \longrightarrow p\,\bar{t}}\,\mathcal{IR}$$

  where $I\,\bar{x}$ is a formula denoting an invariant of the induction and $B_I\,\bar{x}$ is $B\,\bar{x}$ where every occurrence of p is replaced by I.

- The coinduction rules are defined dually.

$$\frac{B\,\bar{t}, \Gamma \longrightarrow C}{p\,\bar{t}, \Gamma \longrightarrow C}\,\mathcal{IL} \qquad\qquad \frac{I\,\bar{x} \longrightarrow B_I\bar{x} \quad \Gamma \longrightarrow I\,\bar{t}}{\Gamma \longrightarrow p\,\bar{t}}\,\mathcal{IR}$$

Consistency

Consider the definiton $p \stackrel{\triangle}{=} p$. The least fixed point is $\emptyset$ while the greatest fixed point is $\{p\}$ (Herbrand universe). Therefore one would expect to have the following proofs:

$$\dfrac{\dfrac{}{\bot \longrightarrow \bot}\,\textit{init} \qquad \dfrac{}{\bot \longrightarrow \bot}\,\textit{init}}{p \longrightarrow \bot}\,\mathcal{IL} \quad \text{and} \quad \dfrac{\dfrac{}{\top \longrightarrow \top}\,\textit{init} \qquad \dfrac{}{\longrightarrow \top}\,\top\mathcal{R}}{\longrightarrow p}.$$

These two proofs are not composable, otherwise the logic would be inconsistent!

(Co)Inductive definitions

We require that a definition to be used either as an inductive definition or as a coinductive one, but not both, in a proof. We therefore distinguish inductive from coinductive definitions. An inductive definition is written as $p\,\bar{x} \stackrel{\mu}{=} B\,\bar{x}$, the coinductive one is $p\,\bar{x} \stackrel{\nu}{=} B\,\bar{x}$.

We have cut-elimination (and hence consistency), with some restrictions on the coinduction rules.

Example: append

- Consider the familiar append clause that concatenate two lists.

$$\begin{array}{rcl} \text{append } l_1 \ l_2 \ l_3 & \stackrel{\mu}{=} & (l_1 = nil \wedge l_2 = l_3) \vee \\ & & \exists l_1' \exists l_3' \exists x . l_1 = (x :: l_1') \wedge l_3 = (x :: l_3') \wedge \text{append } l_1' \ l_2 \ l_3'. \end{array}$$

- We would like to show that whenever $\text{append } l \ l_2 \ l$, then it must be the case that l_2 is the empty list (nil). Formally,

$$\forall l \forall l_2 . \text{append } l \ l_2 \ l \supset l_2 = nil.$$

- We use the invariant $I = \lambda l_1 \lambda l_2 \lambda l_3 . l_1 = l_3 \supset l_2 = nil$.

The induction is on the first and third argument. The inductive step is formally proved as follows.

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{}{\longrightarrow l_1' = l_1'}\ \mathrm{eq}\mathcal{R}
      }{(x :: l_1') = (x :: l_3') \longrightarrow l_1' = l_3'}\ \mathrm{eq}\mathcal{L}
    }{l_1 = (x :: l_1'), l_3 = (x :: l_3'), l_1 = l_3 \longrightarrow l_1' = l_3'}\ \mathrm{eq}\mathcal{L}; \mathrm{eq}\mathcal{L}
    \qquad
    \dfrac{}{\ldots, l_2' = nil \longrightarrow l_2' = nil}\ \mathit{init}
  }{l_1 = (x :: l_1'), l_3 = (x :: l_3'), (l_1' = l_3' \supset l_2' = nil), l_1 = l_3 \longrightarrow l_2' = nil}\ \supset \mathcal{L}
}{
  \dfrac{l_1 = (x :: l_1'), l_3 = (x :: l_3'), (l_1' = l_3' \supset l_2' = nil) \longrightarrow l_1 = l_3 \supset l_2' = nil}{l_1 = (x :: l_1') \wedge l_3 = (x :: l_3') \wedge \boxed{(l_1' = l_3' \supset l_2' = nil)} \longrightarrow \boxed{l_1 = l_3 \supset l_2' = nil}}\ \wedge\mathcal{L}; \wedge\mathcal{L}
}\ \supset \mathcal{R}
$$

Example: CCS one-step transitions

$$\frac{}{A.P \xrightarrow{A} P} \qquad \frac{P \xrightarrow{A} P'}{P \mid Q \xrightarrow{A} P' \mid Q} \qquad \frac{Q \xrightarrow{A} Q'}{P \mid Q \xrightarrow{A} P \mid Q'} \qquad \frac{P(\mu x.P\ x) \xrightarrow{A} Q}{\mu x.P\ x \xrightarrow{A} Q}$$

$$\frac{P \xrightarrow{A} R}{P + Q \xrightarrow{A} R} \qquad \frac{Q \xrightarrow{A} R}{P + Q \xrightarrow{A} R} \qquad \frac{P \xrightarrow{\downarrow A} R \quad Q \xrightarrow{\uparrow A} R}{P \mid Q \xrightarrow{\tau} R \mid S} \qquad \frac{P \xrightarrow{\uparrow A} R \quad Q \xrightarrow{\downarrow A} R}{P \mid Q \xrightarrow{\tau} R \mid S}$$

One-step transitions can be encoded straightforwardly as inductive definitions, e.g.,

$$\begin{aligned} P \mid Q \xrightarrow{\tau} R \mid S &\stackrel{\mu}{=} \exists A.P \xrightarrow{\downarrow A} R \wedge Q \xrightarrow{\uparrow A} R \\ \mu x.P\ x \xrightarrow{A} Q &\stackrel{\mu}{=} P\,(\mu x.P\ x) \xrightarrow{A} Q \end{aligned}$$

Example: CCS simulation

- More interesting is the encoding of the (strong) *simulation* relation between two processes, i.e., transitions by one process can be imitated by the other, as the definition

$$\textit{sim}\ P\ \ Q \stackrel{\nu}{=} \forall A \forall P'. P \stackrel{A}{\longrightarrow} P' \supset \exists Q'. Q \stackrel{A}{\longrightarrow} Q' \wedge \textit{sim}\ P\ '\ Q'$$

- Consider two processes $P = \mu x.(a.x)$ and $Q = \mu x.((a.x \mid a.x))$. Their transition patterns are

$$P \stackrel{a}{\longrightarrow} P \stackrel{a}{\longrightarrow} P \stackrel{a}{\longrightarrow} \ldots$$

$$Q \stackrel{a}{\longrightarrow} (Q \mid a.Q) \stackrel{a}{\longrightarrow} (Q \mid Q) \stackrel{a}{\longrightarrow} ((Q \mid a.Q) \mid Q) \stackrel{a}{\longrightarrow} \ldots$$

Clearly they are similar, since the only observable action is a.

- This can be proved formally using coinduction rules. The invariant is

$$S := \lambda P \lambda Q.(P = \mu x.a.x) \wedge \exists Q'.Q \xrightarrow{a} Q \mid Q'.$$

- An interesting subcase of the proof is to show that S is indeed a post fixed point, i.e, proving the sequent $S\ R\ T \longrightarrow B_S\ R\ T$ where $(B_S R\ T)$ is the formula

$$\forall A \forall R'.R \xrightarrow{A} R' \supset \exists T_1.T \xrightarrow{A} T_1 \wedge [R' = \mu x.a.x \wedge \exists T_2.T_1 \xrightarrow{a} T_1 \mid T_2]$$

- Intuitively, what we have to show is that the pattern of T in the invariant repeats itself during the transition steps.

$$\cfrac{\cfrac{\cfrac{}{(T \xrightarrow{a} T \mid T_1) \longrightarrow (T \xrightarrow{a} (T \mid T_1))}\ \textit{init}}{(T \xrightarrow{a} T \mid T_1) \longrightarrow ((T \mid T_1) \xrightarrow{a} (T \mid T_1) \mid T_1)}\ \textit{def}\mathcal{R}}{(T \xrightarrow{a} T \mid T_1) \longrightarrow \exists T_2.((T \mid T_1) \xrightarrow{a} (T \mid T_1) \mid T_2)}\ \exists\mathcal{R}$$

Example: soundness of the encoding of simulation

Lemma 1. *For all P and Q, if $\longrightarrow$ sim P Q is provable then Q simulates P.*

Proof By using cuts, cut-elimination and permutability of inference rules.

Related Work

- Calculus of partial inductive definitions [Eriksson], but no cut-elimination.

- Craciunescu has a form of coinduction rule in a constraint logic programming language. But again, no cut-elimination.

- Circular proofs [Santocanale, Cockett]. Cut-elimination is non-terminating in general, but cut can always be pushed up in a proof indefinitely.

Conclusion and Future Work

- We currently have a proof system with both induction and coinduction. We proved cut-elimination and hence consistency of the logic. A prototype of the logic has been implemented by Alberto Momigliano on top of HOL/Isabelle.

- Future work:
  - Extend the logic with the ∇ quantifier (Miller and Tiu) to capture reasoning with *names*.
  - Study the connection to circular proofs, e.g., how to recover the invariants from a circular proof object.
  - Semantics, type systems.
  - Proof search properties, e.g., permutability of rules, structures of invariants.