An Implicational Logic for Conjecturing and Distributed Proof Attempts

Lucas Dixon

1 Nov 2007

School of informatics

The Issue

- **Asynchronous** and **distributed** contribution to a formalisation.

- A *common* situation:
  - **Proving a conjecture in parallel with using it**:
    e.g. Fermat's Last theorem involves...
    * Lemma: "Elliptic Curves = Modular" can be converted to Galois Representation.
    * Theorem: Galois representation of "Elliptic Curves = Modular" proved by Iwasawa theory.
  - **Adding to existing theory libraries**,
    e.g. missing lemmas, new theorems...

- Problem: **lots of re-execution** of proof scripts.

The Meta-Logic of Theories

- A **theory** holds a set **theorems** (theorems are derivations of sequents: $\Gamma \vdash A$).
- There is a **meta-logic** to working with theorems, it says:
  - Theorems are given names so they can be referred to.
  - New theorems are derived using only the system's axioms applied to old theorems.
- **How do we make a conjecture?**
  - Add a new theorem of the form: $A \vdash A$?
  - Add it as an (temporary) axiom? (Isabelle's sorry)
  - Application of the cut rule ?

Conjectures as cuts... ?

When you realise you need a conjecture A, use the cut rule:

$$\frac{A, \Delta \vdash B \qquad \Delta \vdash A}{\Delta \vdash B}\,\text{cut}$$

- Conjecture never becomes a theorem in the theory.
- Can only use the conjecture on this branch of the proof.

Conjecture by dangling assumptions... ?

- Leave the conjecture as dangling subgoals/assumptions wherever you plan to use it.

- To make these subgoals go away: prove the conjecture first *and then* apply it to *every* appropriate subgoals.

- Still prove the lemmas before using them:

Parallel Development: conjecture can be proved in parallel with other proofs intend to use it (trail of `FIXME` comments in the file)

Script re-execution: proving the conjecture requires re-checking all proofs after (and modifying them to use the conjecture appropriately).

Conjectures as axioms I promise to remove... ?

- What I actually do: conjectures are added as new axioms, an identical theorem can start to be proved in parallel with the use of the axiom.

- **Parallel Development:** but must remember to remove the axiom and replace it with the proved lemma.

- **Script re-execution:** once an conjecture is proved, need to re-execute everything afterwords.

- **Ugly** to have both axiom and proof attempt of conjecture, not to mention annoying to keep terms in sync.

A Logic of Conjecturing: Idea

Rephrase the rules for implication to support conjectures.

Theory: a set of results (theorems, assumptions, and conjectures) where each result as a unique name.

Result: ${}^{x}[A \vdash p : s]$

- x = the unique name of the result.
- A = the set of result names of assumptions.
- p = the proof of this result; $?$ for unproved, $\circ$ for assumed, and $x\{g_0, ..., g_n\}$ for proved by x with subgoals g_0 to g_n.
- s = the statement that this result makes, in some object language.

A Logic of Conjectures: Making a Theory

$$\frac{}{\{\}}\ \texttt{empty} \qquad \frac{\Delta}{\Delta \cup \{^{x}[A \vdash \circ : s]\}}\ \texttt{assume} \qquad \frac{\Delta}{\Delta \cup \{^{x}[A \vdash ? : s]\}}\ \texttt{conjecture}$$

- where:
  - x is a unique name (fresh) in Δ, and
  - A is a set of assumption names that already exist in Δ.
- Uniqueness of names is an invariant of theories: no freshness conditions.

Example, part 1

ND:

$$\dfrac{\vdots}{A \to B, B \to C \vdash A \to C} \to\mathtt{I}$$

ILC:

$$\dfrac{\dfrac{}{\Delta \equiv \{^{\mathbf{a}}[\vdash \circ : \mathbf{A}],^{a_2}[\vdash \circ : A],^{\mathbf{ab}}[\mathbf{a_2} \vdash \circ : \mathbf{B}],^{b}[\vdash \circ : B],^{\mathbf{bc}}[\mathbf{b} \vdash \circ : \mathbf{C}]\}} \mathtt{assume*}}{\Delta \cup \{^{g_1}[\mathbf{a}, \mathbf{ab}, \mathbf{bc} \vdash ? : C]\}} \mathtt{conjecture}$$

A Logic of Conjecturing: Proving Things

To prove a conjecture x using a result y:

$$\frac{\Delta \cup \{{}^{x}[A \vdash ? : s]\} \qquad {}^{y}[B \vdash p : s] \in \Delta \qquad \mathit{applicable}(y, x)}{\Delta \cup \{{}^{x}[A \vdash y\{i' | i \in B - A\} : s]\} \cup \{{}^{i'}[A \cup \mathit{asms}(i) \vdash ? : \mathit{trm}(i)] \mid i \in B - A\}}\ \texttt{prove}$$

- where...
  - $\mathit{asms}(i)$ = the assumptions of result i w.r.t. Δ.
  - $\mathit{trm}(i)$ = conclusion term of result i w.r.t. Δ.
  - i' = a new name, w.r.t. Δ, generated from i.
  - $\mathit{applicable}(y, x)$ stops circular proofs; done efficiently by caching names.

Remark: tracking dependencies supports minimal rechecking when lemmas are modified/removed.

Example, part 2

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\Delta \equiv \{^{\mathbf{a}}[\vdash \circ : \mathbf{A}],^{a_2}[\vdash \circ : A],^{\mathbf{ab}}[\mathbf{a_2} \vdash \circ : \mathbf{B}],^{b}[\vdash \circ : B],^{\mathbf{bc}}[\mathbf{b} \vdash \circ : \mathbf{C}]\}}\ \texttt{assume*}}{\Delta \cup \{^{g_1}[\mathbf{a}, \mathbf{ab}, \mathbf{bc} \vdash ? : C]\}}\ \texttt{conjecture}}{\Delta \cup \{^{g_1}[a, ab, bc \vdash bc\{g_2\} : C],^{g_2}[\mathbf{a}, \mathbf{ab}, \mathbf{bc} \vdash ? : B]\}}\ \texttt{prove}\ g_1\ \texttt{by}\ bc}{\Delta \cup \{^{g_1}[a, ab, bc \vdash bc\{g_2\} : C],^{g_2}[a, ab, bc \vdash ab\{g_3\} : B],^{g_3}[\mathbf{a}, \mathbf{ab}, \mathbf{bc} \vdash ? : A]\}}\ \texttt{prove}\ g_2\ \texttt{by}\ ab
$$

$$
\dfrac{\Delta \cup \{^{g_1}[a, ab, bc \vdash bc\{g_2\} : C],^{g_2}[a, ab, bc \vdash ab\{g_3\} : B],^{g_3}[\mathbf{a}, \mathbf{ab}, \mathbf{bc} \vdash ? : A]\}}{\Delta \cup \{^{g_1}[a, ab, bc \vdash bc\{g_2\} : C],^{g_2}[a, ab, bc \vdash ab\{g_3\} : B],^{g_3}[a, ab, bc \vdash a : A]\}}\ \texttt{prove}\ g_3\ \texttt{by}\ a
$$

Example, part 3

Mizar/Isar stylish:

$$
\begin{array}{l}
\{\ a_2\texttt{:}\ A \vdash ab\texttt{:}\ B\,\texttt{,} \\
\quad b\texttt{:}\ B \vdash bc\texttt{:}\ C\,\texttt{,} \\
\quad a\texttt{:}\ A\ \} \\
\vdash \\
\quad g_1\texttt{:}\ C\ \texttt{by}\ bc\ \texttt{to}\ g_2 \\
\quad g_2\texttt{:}\ B\ \texttt{by}\ ab\ \texttt{to}\ g_3 \\
\quad g_3\texttt{:}\ A\ \texttt{by}\ a
\end{array}
$$

Remarks

- **ILC supports the process of conjecturing**: it does not describe the nature of conjecturing.
- **Parallel proof attempts:** conjectures can be used and proved in parallel.
- **no re-execution** is needed after proving a conjecture.
- Admissible rules can be useful: assumption $\leftrightarrow$ subgoal, theory merging.
- Implemented: ILC for propositions as 400 lines of SML.
  as 6000 lines in IsaPlanner for Isabelle's intuitionistic meta-HOL.
- **Soundness/Completeness** working on proofs by translation to and from ND calculus.