A Lost Proof

Christoph Benzmüller and Manfred Kerber
Saarland University (Germany) and University of Birmingham (UK)

Motivation

- Relationship of higher order and first order logic
- Necessity to include orders beyond the problem formulation
  - order of primitive substitutions
  - orders and proof lengths (Gödel)
- Practical and conceptual limitations of current automated reasoning systems
- Difference between creative human reasoning and brute force automated reasoning

Specifics in Higher Order Theorem Proving
(classical simple type theory)

- Comprehension axioms $\exists N_{\overline{\alpha^n} \to \beta}.\, \forall \overline{z^n}.\, N(z^1, \ldots, z^n) = B_\beta$
  Can be avoided: use λ-binding construct to denote N
- Extensionality axioms $\forall M_{\alpha\to\beta}.\, \forall N_{\alpha\to\beta}.\, M = N \leftrightarrow \forall x.\, M(x) = N(x)$
  $\forall P_o.\, \forall Q_o.\, P = Q \leftrightarrow (P \leftrightarrow Q)$
  Search space problem induced by blind forward search.
  Axioms avoidable in extensional higher order resolution [Benzmüller&Kohlhase98]
- Primitive Substitution $\dfrac{[Q_\gamma\ \overline{u^k}]^\alpha \vee \mathbf{C} \quad \mathbf{P} \in \mathcal{GB}_\gamma^{\{\neg,\vee\}\cup\{\Pi^\beta \mid \beta\in\mathcal{T}\}}}{\{[Q_\gamma\ \overline{u^k}]^\alpha \vee \mathbf{C}\}_{\{Q\leftarrow\mathbf{P}\}}}\,Prim$
  Infinitely branching (no order restriction); not goal directed.

Boolos' Proof

The Example

1. $\forall n.\, f(n,1) = s(1)$
2. $\forall x.\, f(1, s(x)) = s(s(f(1,x)))$
3. $\forall n.\, \forall x.$
   $f(s(n), s(x)) = f(n, f(s(n), x))$
4. $D(1)$
5. $\forall x.\, (D(x) \to D(s(x)))$

   $\therefore$
6. $D(f(s(s(s(s(1)))), s(s(s(s(1))))))$

- Induction proof: from (4) and (5), we get $\forall x.\ D(x)$, hence $D(f(s(s(s(s(1)))), s(s(s(s(1))))))$ by $\forall$-elimination.

- But induction is not given, hence the first order proof consists of brute force modus ponens applications: infeasible number of single steps ($2^{(2^{\cdots 2})}$ with 64K '2s')

Boolos' Second Order Proof

Instances of comprehension axioms:

$$\exists N.\, \forall z.\, N(z) \leftrightarrow (\forall X.\, X(1) \wedge \forall y.\, (X(y) \to X(s(y))) \to X(z))$$
$$\exists E.\, \forall z.\, E(z) \leftrightarrow (N(z) \wedge D(z))$$

Central idea: "assume the induction principle holds for number z – corresponding to $N(z)$ – then we can show for any predicate X a property $X(z)$ by induction."

The proof employs the following lemmata:

Lemma 1: $N(1)$, $\forall y.\ (N(y) \to N(s(y)))$, $N(s(s(s(s(1)))))$, $E(1)$, $\forall y.\, (E(y) \to E(s(y)))$, $E(s(1))$
Lemma 2: $\forall n.\, N(n) \to \forall x.\, (N(x) \to E(f(n,x)))$

Define $M(n) \leftrightarrow (\forall x.\ N(x) \to E(f(n,x)))$. We want $\forall n.\ (N(n) \to M(n))$. Enough to show $M(1)$ and $\forall n.\ (M(n) \to M(s(n)))$, since then from $N(n)$ follows $M(n)$ by definition of $N(n)$ as $N(z) \leftrightarrow (\forall X.\ X(1) \wedge \forall y.\ (X(y) \to X(s(y))) \to X(z))$ We can instantiate X by M, in particular, the definition of N does not refer to M and is a proper definition. The rest of the proof of the lemma is mainly a further reduction of the problem in a similar way.

The theorem itself is an easy application of the two lemmata.

Subgoal to prove	comprehension axiom applied
$\forall n.\, N(n) \to (\forall x.\, N(x) \to E(f(n,x)))$	$\exists M.\, \forall n.\, M(n) \leftrightarrow (\forall x.\, N(x) \to E(f(n,x)))$
$\forall x.\, N(x) \to E(f(1,x))$	$\exists Q.\, \forall x.\, Q(x) \leftrightarrow E(f(1,x))$
$\forall x.\, N(x) \to E(f(s(n),x))$ from $\forall x.\, N(x) \to E(f(n,x))$	$\exists P.\, \forall x.\, P(x) \leftrightarrow E(f(s(n),x))$

Automation in First Order?

- Definition principle required
- But even then the proof fails: we may try to define $N(n)$ as $M(1) \wedge \forall y.\ (M(y) \to M(s(y))) \to M(n)$, but this is no longer a proper definition, since now N is defined in terms of M and M in terms of N
- The original definition of N heavily depends on the universal second-order quantifier $\forall X$, in which X can be later instantiated by predicates which are defined in terms of N itself

Automation in Higher Order?

- Initial problem formulation does not contain any HO variable: comprehension axioms have to be added;
  possible form: $\forall B_o.\, \exists N_{\overline{\alpha^n}\to o}.\, \forall \overline{z^n}.\, N(\overline{z^n}) = B$
  $\xrightarrow{systems}$ need to introduce additional axioms
- Required instances of comprehension principles cannot be synthesised by HO unification: 'blind' primitive substitution is only way out
  $\xrightarrow{systems}$ need to guess the 'right' instances
- Are there possible alternatives to Boolos' trick with other axioms: extensionality axioms, tertium non datur, ...
  $\xrightarrow{systems}$ need to decide which additional axioms are useful
- Current (automated) systems do completely avoid additional axioms; they are not designed to support a proof like Boolos'

Ways Out: A Speculation

- High-level reasoning, e.g. proof planning [Bundy88]
- Knowledge intensive reasoning based on structured KB's
- Reflection on the proof construction process at object level
- Agent-based integration of different reasoning techniques; possibly even on different abstraction layers
- Problem re-representation
  [Polya62]: "*Of course you want to restate the problem (transform it into an equivalent problem) so that it becomes more familiar, more attractive, more accessible, more promising.*"
  [McCarthy]: mutilated checkerboard problem
- Selecting useful comprehension axioms probably related to concept formation [Colton00]
- Semantic guidance; model-based techniques [Kerber94]

Conclusion

Neither first order nor higher order theorem provers currently provide mechanisms to automatically support proofs like the one of Boolos. This is not just a technical but a conceptual problem (which is probably not very well known):

The expressiveness and power of higher order logic is not employed to its full extend in recent (automated) higher order theorem provers. $\longrightarrow$ sufficient for automating mathematics?

Related Work

- Goal directed treatment of Primitive Substitution; Chad Brown (CMU) is currently investigating a constraint based approach
- Lemma speculation in first order theorem proving (with induction)