Logical- and Meta-Logical Frameworks
Lecture 3

Carsten Schürmann

August 9, 2006

Recall from last time

Conclusion LF, the dependently typed logical framework

One corner of the λ-cube.

No imprepdicativity, no induction principles thus adequate emcondings possible.

Canonical forms inductively defined.

All implemented in the Twelf system.

Homework Complete one case of the adequacy theorem proof for negE in one direction, and negE D_1 $D_2 \Uparrow$ true B in the other.

Example

Judgments: N even, N odd, $N + M = K$
Evidence:

$$\frac{}{z \text{ even}}\ \text{ev0} \qquad \frac{N \text{ odd}}{s\ N \text{ even}}\ \text{evN} \qquad \frac{N \text{ even}}{s\ N \text{ odd}}\ \text{odN}$$

$$\frac{}{0 + M = M}\ \text{pl0} \qquad \frac{N + M = K}{(s\ N) + M = s\ K}\ \text{plN}$$

Example in Twelf

Let's look at even.elf

Theorem

Theorem: If $\mathcal{O}_1 :: N$ odd and $\mathcal{O}_2 :: M$ odd then $\mathcal{P} :: N + M = K$ and $\mathcal{E} :: K$ even.

Proof: by induction on $\mathcal{O}_1$.

Case: $\mathcal{O}_1 = \dfrac{\dfrac{}{z \text{ even}}\,\text{ev0}}{s\ z \text{ odd}}\,\text{odN}$

$\mathcal{P}_1 :: \text{z} + M = M$ by pl0

$\mathcal{P}_2 :: (\text{s } z) + M = (\text{s } M)$ by plN on $\mathcal{P}_1$

$\mathcal{E} :: (\text{s } M)$ even by evN $\mathcal{O}_2$

Theorem

Case: $\mathcal{O}_1 = \cfrac{\cfrac{\mathcal{O}'_1 :: N \text{ odd}}{\text{s } N \text{ even}}\text{evN}}{\text{s (s } N) \text{ odd}}\text{odN}$

$\mathcal{P}_1 :: N + M = K$ and $\mathcal{E}_1 :: K$ even
by ind. hyp. on $\mathcal{O}'_1$, $\mathcal{O}_2$

$\mathcal{P}_2 :: (\text{s } N) + M = (\text{s } K)$ by plN on $\mathcal{P}_1$

$\mathcal{P}_3 :: (\text{s (s } N)) + M = (\text{s (s } K))$ by plN on $\mathcal{P}_2$

$\mathcal{O}_3 :: (\text{s } K)$ odd by odN on $\mathcal{E}_1$

$\mathcal{E} :: (\text{s (s } K))$ even by evN on $\mathcal{O}_3$

Functional encoding

- Elimination forms [Coq]
- Case analysis and recursion [Delphin]
  - Termination,
  - Coverage,
  - Hypothetical judgments.

$$\begin{aligned} \text{thm} \quad : \quad & \forall M : \text{nat} \forall N : \text{nat} \forall O_1 : \text{odd } M \forall O_2 : \text{odd } N \\ & \exists K : \text{nat} \exists P : \text{plus } M\ N\ K \exists E : \text{even } K \end{aligned}$$

$$\begin{array}{l} \textbf{fun } \text{thm (odN ev0) } O_2 = \text{(plN pl0) (evN } O_2) \\ \quad \mid \text{thm (odN (evN } O_1)) \ O_2 = \\ \quad\quad \textbf{let} \\ \quad\quad\quad (P, E) = \text{thm } O_1 \ O_2 \\ \quad\quad \textbf{in} \\ \quad\quad\quad (\text{plN (plN } P), \text{evN (odN } E)) \\ \quad\quad \textbf{end} \end{array}$$

LF functions vs. meta-functions

LF functions

$$\begin{aligned} \mathsf{thm} \quad : \quad & \Pi M : \mathsf{nat}.\, \Pi N : \mathsf{nat}.\, \Pi O_1 : \mathsf{odd}\ M.\, \Pi O_2 : \mathsf{odd}\ N. \\ & \Pi K : \mathsf{nat}.\, \Pi P : \mathsf{plus}\ M\ N\ K.\, \Pi E : \mathsf{even}\ K. \end{aligned}$$

Meta-functions

$$\begin{aligned} \mathsf{thm} \quad : \quad & \forall M : \mathsf{nat} \forall N : \mathsf{nat} \forall O_1 : \mathsf{odd}\ M \forall O_2 : \mathsf{odd}\ N \\ & \exists K : \mathsf{nat} \exists P : \mathsf{plus}\ M\ N\ K \exists E : \mathsf{even}\ K \end{aligned}$$

Assessment

- Σ not part of LF (otherwise typing not unique)
- No cases in LF (otherwise no adequacy)

Relational encoding

Judgment:

$$\text{thm}\ (\overset{\mathcal{O}_1}{N\ \text{odd}})\ (\overset{\mathcal{O}_2}{M\ \text{odd}}) = (\overset{\mathcal{P}}{N + M = K})\ (\overset{\mathcal{E}}{K\ \text{even}})$$

Philosophical We use exactly the same technology as before

Problem Meta theoretical justification that this is a proof lies outside the formal system.

In other words How do we know that we define the evidence for this judgment correctly?

Relational encoding

Evidence: (for the base case)

$$\text{thm}\ \left(\frac{\overline{z\ \text{even}}}{s\ z\ \text{odd}}\right)\ \left(\begin{array}{c}\mathcal{O}_2\\ M\ \text{odd}\end{array}\right) = \left(\frac{\overline{z + M = M}}{(s\ z) + M = (s\ M)}\right)\ \left(\frac{\begin{array}{c}\mathcal{O}_2\\ M\ \text{odd}\end{array}}{((s\ M)\ \text{even})}\right)$$

Relational encoding

Evidence: (for the inductive case)

$$\dfrac{\text{thm}\ (\overset{\mathcal{O}_1}{N\ \text{odd}})\ (\overset{\mathcal{O}_2}{M\ \text{odd}}) = (\overset{\mathcal{P}}{N + M = K})\ (\overset{\mathcal{E}}{K\ \text{even}})}{\text{thm}\ \left(\dfrac{\dfrac{\overset{\mathcal{O}_1}{N\ \text{odd}}}{(s\ N)\ \text{even}}}{(s\ (s\ N))\ \text{odd}}\right)\ (\overset{\mathcal{O}_2}{M\ \text{odd}}) = \left(\dfrac{\dfrac{\overset{\mathcal{P}}{N + M = K}}{(s\ N) + M = (s\ K)}}{(s\ (s\ N)) + M = (s\ (s\ K))}\right)\ \left(\dfrac{\dfrac{\overset{\mathcal{E}}{K\ \text{even}}}{(s\ K)\ \text{odd}}}{(s\ (s\ K))\ \text{even}}\right)}$$

Example in Twelf

Representation of judgment

```
thm : odd N -> odd M -> plus N M K -> even K -> type.
```

Representation of base case

```
b: thm (odN ev0) O2 (plN pl0) (evN O2).
```

Representation of inductive case

```
i: thm O1 O2 P E
-> thm (odN (evN O1)) O2 (plN (plN P)) (evN (odN E)).
```

Discussion

- Use ideas of judgment and evidence to express meta theorems.
- Recall example: Find evidence $\mathcal{D} :: A \supset \neg\neg A \text{ true}$.
- Proving a meta theorem: Find evidence

$$\mathcal{D} :: \text{thm}\ (\overset{\mathcal{O}_1}{N \text{ odd}})\ (\overset{\mathcal{O}_2}{M \text{ odd}}) = (\overset{\mathcal{P}}{N + M = K})\ (\overset{\mathcal{E}}{K \text{ even}})$$

- Thus search for derivations is important.

Overview search techniques

Recall from Lecture 1:

- **Bottom-Up** (backward-chaining) Consider rules that *match* the conclusion.
- **Top-Down** (forward-chaining) Consider rules that *matches* premisses
- **Mixed** A little bottom-up, a little top-down.
- **Remark** The search techniques are independent from the logic. They depend on how to *match judgments*.
- **Remark** In LF: Given Γ, given A, find M, s.t. $\Gamma \vdash M \Uparrow A$.
- **Technique** Logic Programming: The sublanguage is called Elf.

Elf's search semantics

Propositions

$$P ::= a\ M_1 \dots M_n$$

Goal formulas

$$G ::= P \mid \Pi x : A.\, G \mid D \to G$$

- $x : A$ are *universally* quantified parameters
- D are dynamic extension of the signature
- Note relation to λProlog

Definite clauses

$$D ::= P \mid \Pi x : A.\, D \mid G \to D$$

- $x : A$ are *existentially* quantified parameters
- D are subgoals
- Note relation to λProlog

Elf's search semantics (cont'd)

Logic Variables Notation: $\hat{X}$, $\hat{D}$, $\hat{E}$

Unification $\Gamma \vdash M = N : A$ and $\Gamma \vdash A = B : K$

- Make M and N equal.
- Higher-order unification undecidable.
- Pattern unification + constraints.

Goal search $\Gamma \vdash G \Rightarrow M$

- Construct $M : G$ from G.

Immediate entailment $\Gamma \vdash D >> G \Rightarrow M$

- Constrct $M : G$ from G, by focusing on D.

Elf's search semantics (cont'd)

How to search for evidence.

$$\frac{x : D \in \Gamma, \Sigma \qquad \Gamma \vdash x : D >> M : G}{\Gamma \vdash M : P}$$

$$\frac{\Gamma \vdash P = Q : \text{type} \qquad \Gamma \vdash M = N : P}{\Gamma \vdash M : P >> N : Q}$$

$$\frac{\Gamma, x : A \vdash M : G}{\Gamma \vdash \lambda x : A.\, M : \Pi x : AG} \qquad \frac{\Gamma \vdash M\ \hat{X} : [\hat{X}/x]G >> N : Q}{\Gamma \vdash M : \Pi x : A.\, G >> N : Q}$$

$$\frac{\Gamma, D \vdash M : G}{\Gamma \vdash \lambda x : D.\, M : D \rightarrow G} \qquad \frac{\Gamma \vdash D >> M : Q \qquad \Gamma \vdash N : G}{\Gamma \vdash G \rightarrow D >> M\ N : Q}$$

Elf's search semantics (cont'd)

Example Find evidence $\mathcal{D}$ of $A \supset \neg\neg A$ true.

Example Find evidence that since 5 is odd and 7 is odd, 12 is even, using our logic program.

Problem How to control the non-determinism?

Twelf Let's let it run in Twelf. (see even-el.elf)

Elf's search semantics (cont'd)

- Existential variables.
- Back-tracking.
- Embedded implications.
  - + Works with higher-order encodings.
  - + Same syntax as LF signatures.
  - − No user control on search.
  - ∗ No extra logical constants.

When is a Elf program a proof?

If it is a realizer (total function).

Curry-Howard correspondance.

Difficult:

- There are so many logical programs.
- Instantiation of logic variables is not local.
- The hypotheses.

Solution:

1. Mode correctness
2. [World correctness]
3. Termination correctness
4. Coverage correctness

Mode correctness

Definition [*Mode criterion*] During execution, ground inputs are being mapped onto output ground outputs.

[Rohwedder, Pfenning]

Twelf syntax `%mode thm +O`$_1$ `+O`$_2$ `-P -E.`

Algorithm Treverse the constructor type.

- Show that the overall outout is ground assuming the overall input and the output of or subgoals are ground.
- Show that all inputs to the subgoals are ground assuming the overall input to be ground.

Demonstration even-meta.elf.

World correctness

Definition [*World criterion*] During execution the local context is always regular formed.

[Schürmann]

Twelf syntax `%worlds () thm +O`$_1$ `+O`$_2$ `-P -E.`

Algorithm Traverse the constructor type.

- Show that each collection of negative occurences fall within the world schema defined beforehand.

Demonstration even-meta.elf.

Termination correctness

Definition [*Termination criterion*] The execution will eventually terminate.

[Rohwedder, Pfenning, Pientka]

Twelf syntax `%terminates O1 (thm O1 O2 P E).`

Algorithm Traverse the constructor type

- Check if the argument in each recursive call gets smaller.

Properties

- In general undecidable.
- Well-founded subterm ordering.
- Lexicographic and simultaneous extensions.

Demonstration even-meta.elf.

Meta Theory

Definition: [*Coverage criterion*] The execution will always make progress.

[Schürmann, Pfenning]

Twelf syntax `%covers (thm +O1 +O2 -P -E).` Input coverage.

`%total O1 (thm O1 O2 P E).` Includes output coverage.

Algorithms Traverse relevant parts of the signature

- Compute set of coverage candidates.
- Try to cover
- Interpret failure
- Refine set of coverage candidate

Properties

- In general undecidable. [Coquand]
- Algorithms always terminates.
- Open for 10 years.

Conclusion

Conclusion

- Twelf is meta logical framework.
- Logic Programming Semantics give raise to new function arrow.
- Totality = Modes + World + Termination + Coverage.

Homework

- Prove that if n is odd and m is even, then $n + m$ is odd.
- Extra: Implement the double negation theorem.