CS 5860

Tue Sept. 27, 2011

Logic of Quantified Statements
(First-Order Logic)

PLAN

1. Review propositional example
2. Proofs - propositional case
3. Evidence for quantified statements
4. Proofs - quantifier case

Review

What is the evidence for $A \& (B \vee C) \Rightarrow (A \& B \vee A \& C)$?

It is the same as an ML program of type: `A * (`B + `C) → `A * `B + `A * `C

Here is such a program:

\x. <u>let</u> (a, bc) = x <u>in</u> <u>if</u> isl(bc) <u>then</u> inl(a, outl(bc))
<u>else</u> inr(a, outr(bc))

In Nuprl λ(x. spread(x; a, bc. decide(bc; b. inl <a,b>
c. inr <a,c>)))

Note offical ML syntax is (a, bc) = x in. If you use EventML you need to use that syntax exactly.

If you wrote the above ML program, the type inference algorithm would produce the type `A * (`B + `C) → `A * `B + `A * `C

CS 5860

Sept. 27, 2011

2. Proofs – propositional case

Sometimes it is difficult to see how to write a program for these simple types. I put forward the challenges to try writing code for

(a) $(P \vee \sim P) \Rightarrow$ Pierce's Law $(((P \Rightarrow Q) \Rightarrow P) \Rightarrow P)$

(b) $\sim\sim(P \vee \sim P)$

Recall that $\sim P$ is $(P \Rightarrow False)$

The evidence term for (b) is

$$\lambda(h.\ ap(h;\ inr(\lambda(p.\ ap(h;\ inl(p)))))))$$

Here is the proof (note expansion of $\sim P$ to $P \Rightarrow False$ as necessary)

$\vdash \sim\sim(P \vee \sim P)$ by $\lambda(h.\_\_)$

$h: \sim(P \vee \sim P) \vdash False$ by $ap(h;\_\_;\ v.v)$ $[v = ap(h;\_\_)]$

$v: False \vdash False$ by v

$\vdash P \vee \sim P$ by $inr(\_)$

$\vdash P \Rightarrow False$ by $\lambda(p.\_\_)$

$p:P \vdash False$ by $ap(h;\_\_;\ w.w)$

$w: False \vdash False$ by w

$\vdash P \vee \sim P$ by $inl(\_)$

$\vdash P$ by p

note $w = ap(h; inl(p))$

CS5860

Tue Sept 27, 2011 Sample Proof Extract

It is challenging to create the realizing evidence for this computationally valid formula. A proof procedure makes it easy, and we can extract the evidence from the proof as this example shows. Pierce's Law is $((P \Rightarrow Q) \Rightarrow P) \Rightarrow P$. It's a tautology.

$\vdash (P \vee \neg P) \Rightarrow ((P \Rightarrow Q) \Rightarrow P) \Rightarrow P$ by $\lambda(d.\ \_\_\_\ )$

$d{:}(P \vee \neg P) \vdash ((P \Rightarrow Q) \Rightarrow P) \Rightarrow P$ by $decide(d;\ p.\ \_\_\_;\ np.\ \_\_\_)$

$p{:}P \vdash ((P \Rightarrow Q) \Rightarrow P) \Rightarrow P$ by $\lambda(f.\ \_\_\_)$

$\rightarrow$ $p{:}P,\ f{:}((P \Rightarrow Q) \Rightarrow P) \vdash P$ by p

$np{:}\neg P \vdash ((P \Rightarrow Q) \Rightarrow P) \Rightarrow P$ by $\lambda(f.\ \_\_\_)$ see *

$\rightarrow$ $np{:}\neg P,\ f{:}((P \Rightarrow Q) \Rightarrow P) \vdash P$ by $ap(f;\ \_\_\_;\ v.\ \_\_\_)$

$v{:}P \vdash P$ by v

$\vdash (P \Rightarrow Q)$ by $\lambda(x.\ \_\_\_)$

$\lambda(x.any(ap(np;x)))$

$x{:}P \vdash Q$ by $ap(np;\ \_\_\_;\ w.\ \_\_\_)$

$w{:}\ False \vdash Q$ by $any(w)$

$\vdash P$ by x

* $\lambda(f.\ ap(f;\ \lambda(x.any(ap(np;x)))))$

$\lambda(d.\ decide(d;\ p.\ \lambda(f.p);\ np.\ \lambda(f.ap(f;\ \lambda(x.any(ap(np;x)))))))$

This is the final extract or realizer.

CS 5860

Tue Sept. 27

3. Evidence for quantified formulas

Here are interesting examples

(a) $\forall x. (P(x) \Rightarrow Q(x)) \Rightarrow \forall x. P(x) \Rightarrow \forall x. Q(x)$

(b) $\forall x. (P(x) \Rightarrow C) \Rightarrow (\exists x. P(x)) \Rightarrow C$

(c) $\forall x. (P(x) \Rightarrow C) \Leftrightarrow ((\exists x. P(x)) \Rightarrow C)$

(d) $\exists y \forall x. R(x,y) \Rightarrow \forall x \exists y\, R(x,y)$

What is the meaning of (a)?

(a) $(x:D \to (P(x) \to Q(x))) \to (x:D \to P(x)) \to (x:D \to Q(x))$

Nuprl $\lambda(f. \lambda(p. \lambda(x. (f(x))\, p(x))))$

ML \f.\p.\x. (f(x))(p(x)) but ML does not have the dependent type.

Meaning of (b)?

$\forall x\, (P(x) \Rightarrow C) \Rightarrow (\exists x\, P(x) \Rightarrow C)$

$(x:D \to (P(x) \to C)) \to (x:D \times P(x)) \to C$

$\lambda(f. \lambda(e.\ spread(e; x,p.\ (f(x))\, p)))$