Naive Skolemization Makes Lambda Logic Inconsistent

Chad E. Brown

March 30, 2006

Abstract

We show Skolemization is inconsistent with lambda logic. We further show basic extensions of lambda logic are inconsistent.

We argue that Skolemized versions of theorems of lambda logic are inconsistent with lambda logic (as given in [2] and reported on in [3, 4]). We use the existence of a fixed point combinator Y for the λ-calculus such that for all W we have $Ap(W, Ap(Y, W)) = Ap(Y, W)$ (see [1]).

Theorem 1 *There is a formula ϕ of lambda logic such that ϕ is provable, but the Skolemized version of ϕ implies a contradiction.*

Proof: Let ϕ be $\forall x \exists y (x \neq y)$. The non-triviality axiom of [2] states $\mathsf{T} \neq \mathsf{F}$, where T is $(\lambda xy.x)$ and F is $(\lambda xy.y)$. One can easily prove ϕ using $\mathsf{T} \neq \mathsf{F}$. However, the Skolemized form of ϕ is $\forall x (x \neq f(x))$. Applying a fixed-point operator Y to the λ-term $(\lambda x.f(x))$, we obtain a fixed point X. That is, $f(X) = Ap((\lambda x.f(x)), X) = X$ is provable. Hence $X \neq f(X) = X$. $\Box$

In the proof above, we made use of a term $(\lambda x.f(x))$. One way to avoid the contradiction is to assign an "arity" of 1 to the Skolem function f and disallow λ-binding of variables occurring in the first argument of f. This form of Skolemization is described for Church's Type Theory in [6, 7].

The theorem above implies Skolemization (even of valid formulas) can introduce inconsistencies. Note that Theorem 6.1 of [2] proves a conservation result for Skolemization with respect to lambda logic with the axiom of choice (AC):

$$(\forall x \exists y P(x, y)) \supset (\exists f \forall x.P(x, Ap(f, x)))$$

The theorem only holds vacuously, as lambda logic with the axiom of choice is inconsistent.

Theorem 2 *Lambda logic + AC is inconsistent.*

Proof: Consider the particular instance

$$(\forall x \exists y (x \neq y)) \supset (\exists f \forall x (x \neq Ap(f, x)))$$

of AC. As above, one can easily deduce $(\forall x \exists y (x \neq y))$. Using the instance of AC, we conclude the existence of f such that $\forall x (x \neq Ap(f, x))$. This contradicts the fixed-point theorem. □

One may consider the weaker description axiom (D):

$$(\forall x \exists ! y P(x, y)) \supset (\exists f \forall x . P(x, Ap(f, x)))$$

Theorem 3 *Lambda logic + D is inconsistent.*

Proof: One can easily prove

$$\forall x \exists ! y (((y = \mathsf{T}) \wedge (x \neq \mathsf{T})) \vee ((y = \mathsf{F}) \wedge (x = \mathsf{T}))).$$

Using (D), there is an f such that $Ap(f, \mathsf{T}) = \mathsf{F}$ and $\forall x ((x \neq \mathsf{T}) \supset (Ap(f, x) = \mathsf{T}))$. Applying the fixed-point theorem, we obtain a contradiction. □

The use of the fixed-point theorem above allows us to find very easy contradictions above. The fixed-point construction is not typable and so the contradictions would not be found by Otter-lambda when searching in type-safe mode (as described in [3]).

However, a formulation of lambda logic with implicit typing may also be inconsistent with Skolemization. Such a system would seem to be vulnerable to Girard's paradox (a form of the Burali-Forte paradox), especially in the formulation relative to ML-style polymorphism given in [5].

References

[1] H. P. Barendregt. *The λ-Calculus*. Studies in logic and the foundations of mathematics, North-Holland, 1984.

[2] Michael Beeson. Lambda logic. In David A. Basin and Michaël Rusinowitch, editors, *IJCAR*, volume 3097 of *Lecture Notes in Computer Science*, pages 460–474. Springer, 2004.

[3] Michael Beeson. Implicit typing in lambda logic. In *LPAR-05 Workshop: Empirically Successful Automated Reasoning in Higher-Order Logic (ES-HOL)*, pages 5–24, 2006.

[4] Michael Beeson. Mathematical induction in otter-lambda. 2006. to appear, available online http://www.mathcs.sjsu.edu/faculty/beeson/Papers/pubs.html.

[5] Thierry Coquand. An analysis of girard's paradox. In *LICS*, pages 227–236. IEEE Computer Society, 1986.

[6] Dale A. Miller. *Proofs in Higher-Order Logic.* PhD thesis, Carnegie Mellon University, 1983. 81 pp.

[7] Dale A. Miller. A compact representation of proofs. *Studia Logica*, 46(4):347–370, 1987.