

Curry's Last Problem: Imitating λ - β -reduction in Combinatory Logic

J. Roger Hindley,

Mathematics Department, Swansea University, Swansea SA2 8PP, U.K.

Email: j.r.hindley@swansea.ac.uk

From talks at Mathematical Logic Group, Shizuoka, 27–11–1998, and
British Logic Colloquium, Gregynog, 23–9–1999.

Notes revised January 2010

Abstract

The last problem on which Curry worked before he died in 1982 was that of defining a reduction in combinatory logic to correspond closely to the usual beta-reduction in lambda-calculus. He did not succeed. Several solutions to this problem have been posed since then, but despite some ingenuity in their formulation, I believe more work still remains to be done. Here some criteria for acceptability of a beta reduction are discussed.

The problem of defining an analogue of $\lambda\beta$ -reduction in combinatory logic has so far only partly been solved, despite the input of some ingenuity, and is probably one of the main outstanding problems in combinatory logic. When first posed it seemed to be reasonably well defined, but the closer people looked at it the vaguer it began to appear. Up to now the best solution is that in [M'84], see also [M'89], but I believe there is still room for further improvement.

Here I shall describe the problem, not try to give answers. The notation is from [H-S 86/08, Chapters 2 and 9]; note that here the atomic combinators are **S**, **K**, **I**, not just **S**, **K**, and abstraction in CL is denoted by “[x]” not “ λ^*x ”.

The natural mapping $(\)_\lambda$ from CL (i. e. combinatory logic) to λ (i. e. λ -calculus) is defined by

$$\mathbf{S}_\lambda \equiv \lambda xyz.xz(yz), \quad \mathbf{K}_\lambda \equiv \lambda xy.x, \quad \mathbf{I}_\lambda \equiv \lambda x.x.$$

This mapping induces two equivalence relations in CL, called $=_{c\beta}$ and $=_{c\beta\eta}$:

$$X =_{c\beta} Y \iff X_\lambda =_{\lambda\beta} Y_\lambda, \tag{1}$$

$$X =_{c\beta\eta} Y \iff X_\lambda =_{\lambda\beta\eta} Y_\lambda. \tag{2}$$

In the 1950's, Curry defined a relation called *strong reducibility* which generated $=_{c\beta\eta}$ [C+'58, §6F], [C+'72, §11E]. The problem now is to do the same for $=_{c\beta}$ and produce a reducibility relation which has acceptable other properties. Here I shall try to give a precise meaning to “acceptable”.

The first task is to define a mapping H from λ to CL. The key step is:

$$(\lambda x.M)_H \equiv [x].(M_H), \quad (3)$$

where $[\]$ is defined by an abstraction algorithm in CL. Many such algorithms are known; some simple ones are:

$$\begin{aligned} [\]^w \text{ or } [\]^{abf} & \text{ [H-S 86/08, Def. 9.20, called there } \lambda^w \text{ or } \lambda^{abf}], \\ [\]^\eta \text{ or } [\]^{abcf} & \text{ [H-S 86/08, Def. 2.14, } \lambda^\eta \text{ or } \lambda^{abcf}], \\ [\]^\beta & \text{ [H-S 86/08, Def. 9.34, } \lambda^\beta], \\ [\]^0 & \text{ [M'84, Def. 4.1].} \end{aligned}$$

Desirable properties for an abstraction algorithm:

- (A0) $[x].Y$ does not contain x , and $([x].Y) x \triangleright_{weak} Y$;
- (A1) $[Z/v]([x].Y) \equiv ([x].[Z/v]Y)$ if $x \notin FV(vZ)$;
- (A2) $X_{\lambda H} \equiv X$ for all CL-terms X ;
- (A3) $M_{H\lambda} \equiv M$ for all λ -terms M ;
- (A4) $\lambda x.Y$ is functional (fnl), i.e. has form **S**, **SU**, **SUV**, **K**, **KU**, or **I**.

No known definition of $[\]$ has all these properties. We can reduce our demands by weakening “ \equiv ” in (A1) – (A3) to “ $=_\beta$ ” or “ $=_{\beta\eta}$ ”. The corresponding properties may be called “(A1 $_\beta$)” – “(A3 $_\beta$)” or “(A1 $_{\beta\eta}$)” – “(A3 $_{\beta\eta}$)”.

Of the above-listed definitions of $[\]$, all satisfy (A0), and

$$\begin{aligned} [\]^w & \text{ satisfies (A1), (A2}_\beta), (A3_\beta), (A4); \\ [\]^\eta & \text{ satisfies (A1), (A2), (A3}_\eta); \\ [\]^\beta & \text{ satisfies (A1}_\beta), (A2), (A3_\beta), (A4); \\ [\]^0 & \text{ satisfies (A1), (A2}_\beta), (A3_\beta), (A4). \end{aligned}$$

Desirable properties for a combinatory β -reduction (called here “ $\triangleright_{c\beta}$ ”).

We state the properties first, and motivate them later.

- (R0) $\triangleright_{c\beta}$ should satisfy all the defining rules of weak reducibility \triangleright_{weak} ;
- (R1) $\triangleright_{c\beta}$ should generate the equivalence $=_{c\beta}$ defined in (1);
- (R2) $X \triangleright_{c\beta} Y \implies [Z/v]X \triangleright_{c\beta} [Z/v]Y$;
- (R3) $\triangleright_{c\beta}$ should be confluent;
- (R4) $X \triangleright_{c\beta} Y \implies [v].X \triangleright_{c\beta} [v].Y$;
- (R5) $[N_H/x](M_H) \triangleright_{c\beta} ([N/x]M)_H$;
- (R6) The irreducible CL-terms should form a decidable set and should correspond closely in some way to the irreducible λ -terms;
- (R7) $\triangleright_{c\beta}$ should be contained in the $\beta\eta$ -strong reducibility relation [C⁺’58, §6F].

(R8) The metatheory of $\triangleright_{c\beta}$ should be reasonably simple.

These properties are unlikely to be all satisfiable together.

(R2) is motivated by wanting a variable v to genuinely denote an *arbitrary* object.

(R3) is motivated by the belief that the main purpose of a reduction is to help prove statements of the form " $X \not\equiv_{c\beta} Y$ ", and that confluence is the most often-used tool for doing this.

(R4) is included just to make the analogy with λ -calculus close.

(R5) is included for a purely technical reason: it is the β -analogue of a key lemma in the proof of the confluence theorem for $\beta\eta$ -strong reduction [C⁺'72, §11E]; in that proof the theorem was deduced from the confluence of $\triangleright_{\beta\eta}$ in λ ; if a direct proof-method for CL was ever discovered, (R5) could be omitted.

(R7) is probably of less interest than the others.

(R8), though vague, is of course extremely important in practice; if it failed, everyone would use λ -calculus and $\triangleright_{\lambda\beta}$ instead of whatever $\triangleright_{c\beta}$ we proposed.

Several attempts to define a suitable relation $\triangleright_{c\beta}$ have been made. (Some background remarks are in [H'77, §3].) The most naive attempt is

$$X \triangleright_{c\beta} Y \iff X_\lambda \triangleright_{\lambda\beta} Y_\lambda. \quad (4)$$

But this relation is not confluent (proved by D. T. van Daalen).

The most sophisticated attempt is in [M'84, Def. 4.2] as modified in [M'89, Def. 1]. It satisfies all of (R0)–(R4), and (R5) if the H-mapping in (R5) is defined using $[]^\beta$ not $[]^0$. So, if an analysis of normal forms is not needed, this $\triangleright_{c\beta}$ is good. But (R6) has not yet been proved for it. (And (R8)?)

Further work still needs to be done, either on Mezghiche's reduction or a completely new one.

References

- [C⁺'58] H. B. Curry, R. Feys, *Combinatory Logic Vol. 1*, North-Holland Co. 1958.
- [C⁺'72] H. B. Curry et al., *Combinatory Logic Vol. 2*, North-Holland Co. 1972.
- [H'77] J. R. Hindley, Combinatory reductions and lambda reductions compared, *Zeitschrift für Mathematische Logik* 23 (1977), 169-180.
- [H-S 86/08] **J. R. Hindley** and **J. P. Seldin**, *Lambda-Calculus and Combinators, an Introduction*, Cambridge University Press, England, 2008. (2nd edn. of *Introduction to Combinators and λ -calculus*, 1986.)
- [M'84] M. Mezghiche, Une nouvelle $c\beta$ -reduction dans la logique combinatoire, *Theoretical Computer Science* 31 (1984), 151–163.
- [M'89] M. Mezghiche, On pseudo- $c\beta$ -normal form in combinatory logic, *Theoretical Computer Science* 66 (1989), 323–331.