CURRY'S ANTICIPATION OF THE TYPES USED IN PROGRAMMING LANGUAGES

Jonathan P. Seldin
Department of Mathematics and Computer Science
University of Lethbridge
Lethbridge, Alberta, Canada
jonathan.seldin@uleth.ca
http://home.uleth.ca/~jonathan.seldin

Annual Meeting
Canadian Society for History and Philosophy of Mathematics
Toronto, Ontario
24–26 May 2002

September 17, 2002

Computer data stored as strings of 0s and 1s

A given string can be interpreted by a program in more than one way

Example:

10111110011010000000000000000000

Can interpret as:

- An unsigned integer. This is just a binary integer. Value $= 2^{19} + 2^{21} + 2^{22} + 2^{25} + 2^{26} + 2^{27} + 2^{28} + 2^{29} + 2^{31} = 3,194,486,784$

- A signed integer. First bit, 1, is - sign. The value is $-(2^{20} + 2^{22} + 2^{23} + 2^{26} + 2^{27} + 2^{28} + 2^{29} + 2^{30}) = -1,047,003,136$

- A floating point real. First bit, 1, is -. Next 8 bits, 01111100, are binary for the exponent, which is $124 - 128 = -4$. Remaining bits are mantissa, which is

$$\begin{aligned} & 11010000000000000000000 \\ = \ & 0.1101_2 = 2^{-1} + 2^{-2} + 2^{-4} \\ = \ & \frac{13}{16} = 0.8125_{10} \end{aligned}$$

So the value is -0.8125×2^{-4}

Types

Examples: `int`, `real`, `bool`

Variables must often be declared: `num : int`, `radius : real`, `cond : bool`

We may want compound types: `int -> bool` is the type of a function from integers to booleans

These are modelled by *typed* λ*-calculus*

λ-calculus

We write "f is $x \mapsto x^2$" for "$f(x) = x^2$"

$f(3) = 3^2 = 9$

Why not write "$(x \mapsto x^2)(3) = 3^2 = 9$"?

In the 1930s, Alonzo Church wrote

$$(\lambda x \,.\, x^2)3 = 3^2 = 9$$

Currying

Given $f(x, y) = x - y$

$$\begin{aligned} (\mathsf{curry} f)3 &= \lambda y \,.\, 3 - y \\ ((\mathsf{curry} f)x)y &= f(x, y) \end{aligned}$$

Write $MNPQ$ for $(((MN)P)Q)$

Formal λ-calculus (Church, 1932/33, 1941)

Variables: $x, y, z, u, v, w, \ldots$

Perhaps some constants

Terms: variables, constants, (MN), $\lambda x \,.\, M$

Contractions: replacement of

$$\begin{array}{rcl} \lambda x \,.\, M & \text{by} & \lambda y \,.\, [y/x]M \\ (\lambda x \,.\, M)N & \text{by} & [N/x]M \end{array}$$

Reductions: sequence of contractions
Notation: $\triangleright$

Conversions: sequence of contractions and reverse contractions
Notation: $=_*$

Meaningless terms: $(\lambda x \, . \, xx)(\lambda x \, . \, xx)$
Reduces only to itself (infinite loop)

$(\lambda x \, . \, xxx)(\lambda x \, . \, xxx)$
Reduces to $(\lambda x \, . \, xxx)(\lambda x \, . \, xxx)(\lambda x \, . \, xxx)$
(infite expanding loop)

Avoid these terms: assign types (Church, 1940)

Types are atomic types and $\alpha \rightarrow \beta$

Assumptions assign types to variables

Rules are

$$\frac{\begin{array}{c}[x : \alpha] \\ M : \beta\end{array}}{\lambda x \, . \, M : \alpha \rightarrow \beta} \; (\rightarrow \mathrm{i})$$

and

$$\frac{M : \alpha \rightarrow \beta \quad N : \alpha}{MN : \beta} \; (\rightarrow \mathrm{e})$$

Combinatory Logic (Schönfinkel, 1920; Curry, 1929, 1930)

Variables: $x, y, z, u, v, w, \dots$

Constants: $\mathsf{I}, \mathsf{K}, \mathsf{S}$, and perhaps others

Terms: variables, constants, (MN)

Contractions: replacement of

$$\begin{array}{rcl} \mathsf{I}X & \text{by} & X \\ \mathsf{K}XY & \text{by} & X \\ \mathsf{S}XYZ & \text{by} & (XZ)(YZ) \end{array}$$

Reductions: sequences of contractions
Notation: $\triangleright$

Conversions: sequences of contractions and reverse contractions
Notation $=_*$

Definition of abstraction:

$$
\begin{aligned}
[x]x &\equiv \mathsf{I} \\
[x]c &\equiv \mathsf{K}c \\
[x](MN) &\equiv \mathsf{S}([x]M)([x]N)
\end{aligned}
$$

Other combinators:

$$
\begin{aligned}
\mathsf{B}XYZ &\rhd X(YZ) \\
\mathsf{C}XYZ &\rhd XZY \\
\mathsf{W}XY &\rhd XYY
\end{aligned}
$$

Church's original system:
$\lambda x \,.\, M$ defined only if x free in M

Curry's original system:
$[x]M$ always defined

Originally, exact connection between combinatory logic and λ-calculus not clear

Details worked out by Rosser in 1930s

Type assignment: same types, rule ($\rightarrow$ e), and axiom schemes:

($\rightarrow$ I)	$\mathsf{I} : \alpha \rightarrow \alpha$
($\rightarrow$ K)	$\mathsf{K} : \alpha \rightarrow (\beta \rightarrow \alpha)$
($\rightarrow$ S)	$\mathsf{S} : (\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))$

Derived rule:

$$\frac{\begin{array}{c}[x : \alpha] \\ M : \beta\end{array}}{[x]M : \alpha \rightarrow \beta}$$

Proof similar to proof of deduction theorem in propositional calculus

Curry's approach to types (Curry, 1934, 1936)

For Curry, $M : \alpha$ was statement αM of logic

$f : \alpha \to \beta$ stood for $(\forall x)(\alpha x \supset \beta(fx))$

Axioms and rules for types follow by axioms and rules for logic primitives

Curry used logic primitive Ξ, where ΞXY stood for $(\forall x)(Xx \supset Yx)$

Curry thus defined

$$
\begin{aligned}
\mathsf{F} \quad &\equiv \quad \lambda xyz \,.\, (\forall u)(xu \supset y(zu)) \\
&=_* \quad \lambda xyz \,.\, (\forall u)(xu \supset \mathsf{B}yzu) \\
&=_* \quad \lambda xyz \,.\, \Xi x(\mathsf{B}yz)
\end{aligned}
$$

Here $\mathsf{F}\alpha\beta f$ stood for modern $f : \alpha \to \beta$

Idea for theory of predication.

The general theory of this subject arises under the following

Consider a relation between ϕ, α, β; then

$$(x)\ \alpha x .\supset.\ \beta(\phi x)$$

d.h. $(x).\ \alpha x .\supset.\ B\beta\phi x$

$$(\Pi \cdot W)(P, \alpha(B\beta\phi))$$

$$= (\Pi \cdot W)[B_2(P,\alpha)B\beta\phi]$$

$$= B(\Pi \cdot W)(B_2(P,\alpha)B\beta)\phi$$

etc.

i.e. ϕ is a function from α to β. Let us write it $F_{\alpha\to\beta}(\phi)$.

Then we have such questions as

$$F_{\alpha\to\beta}\ \phi \cdot F_{\beta\to\gamma}\ \psi .\supset.\ F_{\alpha\to\gamma}(B\psi\phi)$$

Suppose ϕ is a function of two variables such that

$$\alpha x \cdot \beta y .\supset.\ \gamma[\phi(xy)]$$

then we must have $\alpha x .\supset \lambda(\phi x)$

where $\lambda u = \beta y .\supset. \gamma(uy) : F_{\beta\to\gamma}\ u$

d.h. $F_{\beta\to\gamma} = \lambda$.

So above is $F_{\alpha\to F_{\beta\to\gamma}}\ \chi$

The general problem of combinatory logic is this:

Let $\xi_1, \xi_2, \xi_3, \xi \ldots \xi_n$ be such that

$$F_{\alpha_1\to\beta_1}\xi_1,\ F_{\alpha_2\to\beta_2}\xi_2, \ldots F_{\alpha_n\to\beta_n}\xi_n \text{ etc.}$$

& let Y be a proper combinatory operator.

Then to show the category of $Y\xi_1\xi_2\ldots\xi_n$ is uniquely determin

Instead of $F_{\alpha\to\beta}$ a more convenient notation is

$[\alpha \to \beta]$

and for $F_{\alpha \to F_{\beta\to\gamma}}$ have $[\alpha,\beta \to \gamma]$

etc.

in general if $\phi(x_1 \cdots x_n)$ is such that

$\alpha x_1 x_1 . \alpha_2 x_2 . \cdots . \alpha_n x_n . \supset . \beta(\phi x_1 x_2 \ldots x_n)$

then $[\alpha_1, \alpha_2, \cdots \alpha_n, \to \beta] \ \phi$

then we have $[\alpha, \beta \to \gamma] \equiv [\alpha \to [\beta \to \gamma]]$

and in general $[\alpha_1, \alpha_2, \cdots \alpha_n, \to \beta] \equiv [\alpha_1 \to [\alpha_2 \to [\alpha_3 \to \cdots [\alpha_n \to \beta]]$

postulates are such as

$[\alpha \to \beta]\phi . [\beta \to \gamma]\psi . \supset . [\alpha \to \gamma]\phi(B\psi\phi)$

still better notation, for $\alpha \to \beta$ use $F\alpha\beta$

then $[\alpha,\beta \to \gamma] = F\alpha(F\beta\gamma)$ etc

we have $F\alpha\beta\phi . F\beta\gamma\psi . \supset . F\alpha\gamma(B\psi\phi)$

definition of F is then

$$\begin{aligned}
F\alpha\beta x &= (y)\ \alpha y . \supset . \beta(xy) \\
&= (y)\ \alpha y . \supset . B\beta x y \\
&= (\Pi \cdot W)(P_1 \alpha (B\beta x)) \\
&= (\Pi \cdot W)(B B_2 P_1 \alpha B \beta x) \\
&= (\Pi \cdot W)(C_3 B B_2 P_1 B \alpha \beta x) \\
&= B_3 (\Pi \cdot W)(C_3 B B_2 P_1 B)\alpha\beta x
\end{aligned}$$

$\therefore F = B_3 (\Pi \cdot W)(C_3 B B_2 P_1 B)$

From this the various postulates may perhaps be proved other postulates are such as

$$F\alpha(F\beta y)x \ .\supset.\ F\beta(F\alpha y)(Cx)$$

etc. $F\alpha(F\alpha y)x \ .\supset.\ F\alpha y\,(Wx)$

The postulate for B may be put in the form

$$F\alpha\beta x \,.\, F\beta y y' \ .\supset.\ F\alpha y\,(Bxy)$$

d.h. $F(F\alpha\beta)(F(F\beta y)(F\alpha y))B$

ie $F(F\alpha\beta)(F(F\beta y)(F\alpha y))B$

this is not a constant proposition about B since it involves the variables α and β.

if $\alpha = \beta = \gamma$ we have

~~F(F\alpha\alpha~~ $F(F\alpha\alpha)(F(F\alpha\alpha)(F\alpha\alpha))B$

d.h. $F(F\alpha\alpha)(F\alpha\alpha)(WB)$

let α be the proposition.

Then $F\alpha\alpha N$ when N is negation

whence $F\alpha\alpha(WBN)$

but let p be a proposition $WBNp = NNp = p$.

which is no trouble

The contradiction arises in $W(BN)$

If we have the postulate for B

$F(F\alpha\beta)(F(F\beta\gamma)(F\alpha\gamma))B$

can we deduce contradiction in $W(BN)$ where $W(BN)x = BNxx = N(xx)$

we have $F\pi\pi N$. set $\alpha = \beta = \pi$

$F(F\pi\pi)(F(F\pi\gamma)(F\pi\gamma))B$.

we have only $F\pi\pi x$.⊃. $F\pi\gamma(BNx)$

d.h. $F(F\pi\gamma)(F\pi\gamma)(BN)$

from which we can conclude nothing)

there is of form $F\beta\alpha(F\beta\gamma)BN$

where $\alpha = F\pi\gamma$

$\beta = \pi$

$\gamma = \gamma$

Curry's version of rule ($\rightarrow$ e):

$$\frac{\mathsf{F}XYZ \quad XU}{Y(ZU)}$$

Curry called this the *theory of functionality*

As early as July 1930, Curry was naming implication formulas for combinators:

(PI) $A \supset A$

(PK) $A \supset (B \supset A)$

(PS) $(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$

This is probably the beginning of *propositions-as-types*

7/15/30

3
T300715B

<u>Axioms for Calculus of Propositions for Notational Purposes</u>

P	P_0	$x \to x$
PB	P_b	$(y \to z) \to (x \to y) \to (x \to z)$
	~	$(x \to y) \to ((z \to x) \to (z \to y))$.
PC	P_c	$(x \to (y \to z)) \to (y \to (x \to z))$.
PW	P_w	$(x \to (x \to y)) \to (x \to y)$.
PK	P_k	$(x \to (y \to x))$.
ΛK	Λ_k	$x \wedge y \to x$
ΛW	Λ_w	$x \to x \wedge x$
ΛC	Λ_c	$x \wedge y \to y \wedge x$.
ΛB	Λ_b	$(x \to y) \to (z \wedge x \to z \wedge y)$.
VK	V_k	$x \to x \vee y$
VW	V_w	$x \vee x \to y$
VC	V_c	$x \vee y \to y \vee x$
VB	V_b	$(x \to y) \to ((z \vee x) \to (z \vee y))$.
PΛ	PΛ	$x \wedge (x \wedge y) \to y$
N_0	N_0	$x \vee \bar{x}$
$(NN)_1$	N_1	$x \to \bar{\bar{x}}$
$(NN)_2$	N_2	$\bar{\bar{x}} \to x$
(NB)	N_b	$(x \to y) \to (\bar{y} \to \bar{x})$.
~ (PN).		

Pfm

In his logic, Curry postulated rule (Eq):

$$\frac{X \quad X =_* Y}{Y}$$

In the theory of functionality, this rule also held

In 1950s, Curry prooved that if any term is a type, the system is inconsistent. He proved this (Curry, 1958, p. 349) by proving

$$\beta(\mathsf{WWW})$$

where β is any term. He then lets β be $\mathsf{K}X$ for an arbitrary term X, thus getting

$$\mathsf{K}X(\mathsf{WWW})$$

from which, by Rule (Eq), he gets

$$X$$

But earlier in (Curry, 1958, p. 279), he had *basic functionality*, in which types were all terms in normal form and could not be converted to other terms. This led to a restricted version of Rule (Eq), namely Rule (Eq$'$):

$$\frac{\alpha X \quad X =_* Y}{\alpha Y}$$

About 1966, he separated Rule (Eq) into two rules for functionality:

$$\frac{\alpha X \quad X =_* Y}{\alpha Y}\,\text{(Eqs)} \qquad \frac{\alpha X \quad \alpha =_* \beta}{\beta X}\,\text{(Eqp)}$$

(Curry, 1968, Chapter 14)

Relation to logic

In 1935, Curry's original system (along with that of Church) was proved inconsistent by Kleene and Rosser

Curry's response: examine different kinds of systems for consistency

His original idea (late 1930s, published 1941): systems based on logical primitives

Three kinds of systems:

- $\mathcal{F}_1$: primitive is F

  $\Xi \equiv \lambda xy \,.\, \mathsf{F}xy\mathsf{I}$ or $\Xi \equiv \lambda xy \,.\, \mathsf{F}x\mathsf{I}y$

- $\mathcal{F}_2$: primitive is Ξ

F defined as above, $\mathsf{P} \equiv \lambda xy \,.\, \Xi(\mathsf{K}x)(\mathsf{K}y)$, and $\Pi \equiv \lambda x \,.\, \Xi\mathsf{E}x$, where $\vdash \mathsf{E}X$ for all terms X

- $\mathcal{F}_3$: primitives are P and Π

  $\Xi \equiv \lambda xy \,.\, \Pi(\lambda u \,.\, \mathsf{P}(xu)(yu))$

Curry originally thought that $\mathcal{F}_3$ was stronger than $\mathcal{F}_2$ was stronger than $\mathcal{F}_1$ (on reasonable additional assumptions). However, it has turned out that $\mathcal{F}_2$ and $\mathcal{F}_3$ are of essentially the same strength on any reasonable additional postulates, and that if the equality rules are not separated $\mathcal{F}_1$ is of essentially the same strength.

An Idea in Regard to G.

Take following as rule,

Rule G: $GXYz, XU \vdash YU(zU)$

Then $G = [x, y, z]\ \Xi x.(Syz)$.

Thus G differs from Ξ in putting S for B.

With Curry's idea of 1956 for G, we have

$$\begin{aligned}\mathsf{G} &\equiv \lambda x, y, z \,.\, \Xi x(\mathsf{S}yz) \\ &=_{*} \lambda x, y, z \,.\, (\forall u)(xu \supset \mathsf{S}yzu) \\ &=_{*} \lambda x, y, z \,.\, (\forall u)(xu \supset yu(zu))\end{aligned}$$

This gives us the *dependent function type*:

$$(\Pi x : A)B \equiv (\forall x : A)B \equiv \mathsf{G}A(\lambda x \,.\, B)$$