Motivating The Proof of the Kleene Recursion Theorem

By John Case

Kleene's (Strong) Recursion Theorem (KRT) can be stated as follows, where the notation is from [Rog67]. N is the set of non-negative integers. φ_p is the 1-ary partial computable function computed by program (number) p in some fixed acceptable system for computing all the (1-argument) partial computable functions. $\langle \cdot, \cdot \rangle$ is a fixed, computable pairing function: $N \times N \to N$. We will write $\varphi_p(x, y)$ as an abbreviation of $\varphi_p(\langle x, y \rangle)$.

Theorem $(\forall p)(\exists e_p)(\forall x)[\varphi_{e_p}(x) = \varphi_p(x, e_p)]$. *Furthermore,* $\lambda p \boldsymbol{.} e_p$ *can be taken to be computable.*

Consider an application of KRT where a particular p is chosen so that, $(\forall x, w)[\varphi_p(x, w) = w]$. Then, by KRT, for $e = e_p$,

$$(\exists e)(\forall x)[\varphi_e(x) = e]. \tag{1}$$

Such a program e, independently of its external information x, outputs a self-copy (e). This case is similar to self-replication in single-celled organisms. Actually, the mechanism in *Kleene's proof* of KRT [Rog67, Page 214] is combinatorially and logically quite similar to the mechanism of self-replication in single-celled organisms! This can be made especially clear if we present the special case of that proof required to prove only (1) above *accompanied by an indication of biological analogs.*[1]

Motivated Proof of (1) above.

We'll use $\odot$ to represent a single-celled organism, where the inner circle contains the *DNA*, the genetic material or blueprint, and the space between the outer and inner circle contains the *cytoplasm*, the active and controlling part of the cell. We'll use $\odot_b^p$ to represent the single-celled organism whose cytoplasm (program) is p and whose DNA (blueprint datum) is b. N.B. We are thinking of the cytoplasm as a program and the DNA as a datum! It is more typical to see the DNA thought of a program, but that is not the more useful conceptualization in the present context! In fact it is useful to think of $\odot_b^p$ as cytoplasm p with DNA blueprint datum b stored inside.

Kleene's S-m-n Theorem can be stated as

Theorem *There is a computable function s, such that*

$$(\forall b, p, x)[\varphi_{s(b,p)}(x) = \varphi_p(x, b)]. \tag{2}$$

My interpretation of Kleene's S-m-n Theorem [Cas74, Ric80, Ric81, Roy87, RC94] is that $s(b, p)$ is program p with datum b stored inside (where the computability of s merely means that the transformation of storing b in p is, as it should be, computable). Hence,

$$\odot_b^p \textit{ is a biological analog of } s(b, p). \tag{3}$$

Single-celled organisms which create a daughter cell contain cytoplasmic programs d which have the property that, in any (reasonable) environment x, for any (reasonable) DNA b, the cytoplasm of $\odot_b^d$ first uses b as a blueprint for building more cytoplasm. Then the cytoplasm of $\odot_b^d$ next makes an extra copy of the blueprint b. For convenience, we'll abuse notation a bit and denote the cytoplasm blueprinted by DNA b as b also.[2] Hence, $\odot_b^d$ would then have created a copy of cytoplasm b and an extra copy of the blueprint b. Lastly, $\odot_b^d$ forms from these copies $\odot_b^b$ and breaks it off as a separate, new daughter cell.

[1]This was also discussed in less detail in [Cas74].

[2]N.B. Biochemically and physically, of course, any particular cytoplasm and a blueprint for *that* cytoplasm are very different.

Typically, in the world, the blueprint b that single-celled organisms contain is the blueprint for building their own cytoplasm d. Abusing notation consistently with the above abuse, we'll denote the blueprint datum for cytoplasm d also as d.[3]

We'll use $\rightarrow$ to mean *builds/begets*. Hence, typically in the world, in any (reasonable) environment x, we see that

$$\odot^d_d \rightarrow \odot^d_d, \tag{4}$$

self-replication! However, in the laboratory, the genetic material from a cell $\odot^d_d$ can be replaced by that of a different organism $\odot^{q_0}_{q_0}$ to form an in-the-lab-only single-celled organism $\odot^d_{q_0}$. Then we can get, in any (reasonable) environment x,

$$\odot^d_{q_0} \rightarrow \odot^{q_0}_{q_0}. \tag{5}$$

In the laboratory it is possible to control which DNA datum is stored in a given cell. In effect, the DNA datum functions *in the laboratory setting* as an storable/inputtable parameter!

N.B. We are presenting what amounts, in some cases, to a *simplification* of the biological mechanisms. For example:

- real cells may have genetic material in more than one place and operating in different spots, in the same cell, differently;
- in many cases, it may be too difficult to completely replace the genetic material from one cell of one species by that of another; and,
- in many cases, if one replaces the genetic material from a cell of one species by that of another, the resultant cell may not reproduce at all or its daughter cells may have cytoplasm which is a mixture of that from the two species whence it came.

However, our simplifcations make the analogy with the (non-simplified) proof of KRT much clearer.

Lets expand and explore now the analog in (3) above. One expansion is to treat the *environment* x of a single-celled organism $\odot^p_b$ as analogous to the external *input* x to program $s(b,p)$.

We need a program analog of *cytoplasm* d above which, for any (reasonable) DNA blueprint b and in any (reasonable) environment x, we have $\odot^d_b \rightarrow \odot^b_b$ (as in (5) above). Hence, we want a program d so that, for any datum b and any input x,

$$\text{program } s(b,d) \text{ outputs program } s(b,b), \tag{6}$$

i.e., we want a program d such that

$$(\forall x, b)[\varphi_{s(b,d)}(x) = s(b,b)]. \tag{7}$$

It would suffice to have a program d, so that,

$$(\forall x, b)[\varphi_d(x,b) = s(b,b)], \tag{8}$$

for, then, by S-m-n, we would have,

$$(\forall x, b)[\varphi_d(x,b) = \varphi_{s(b,d)}(x)], \tag{9}$$

[3]N.B. The blueprint is *not* the blueprint of the whole cell $\odot^d_d$, but only the blueprint of its cytoplasm d. This avoids infinite regress paradoxes, and was first noticed by von Neumann, but for the case of *machine* self-replication. He sought to show that machine self-replication was possible/free-of-paradox [Neu66, Bur70, Cas74]. In fact the *top* level design/refinement of his self-replicating machines [Neu66, Bur70, Cas74] closely matches the description above of how single-celled organisms self-replicate! Of course, the detailed refinement *below* the top level in the biological case involves biochemistry, but not for von Neumann's construction [Neu66, Bur70].

Kleene mentioned to me sometime in the 80's that early on he had used his KRT to better understand Von Neumann's construction; whereas, I had used Von Neumann's construction to better understand Kleene's!

Interestingly too, the proof of KRT is quite analogous to Gödel's construction [Göd86] of particular formal sentences which refer to themselves (and of course similarly analogous to Feferman's scheme for constructing more general self-referential formal sentences [Fef60]). In this analogy, Kleene's S-m-n function plays exactly the role of Gödel's substitution function: Kleene's involves substitution of data into programs and Gödel's, substitution of numerals into sentences.

and, hence, by (9) and (8), we would obtain (7). Of course we *do* have a program d such that (8) — since by S-m-n, $\lambda x, b\boldsymbol{.} s(b, b)$ is (partially) computable. Therefore, we obtain a program d satisfying (7) above.[4]

As we noted above, in the typically occurring biological case, the DNA in a cell is the blueprint for the cytoplasm in *that* cell. Our analog of the blueprint of a program is just its code number. Hence, without abusing notation at all, we can take the blueprint of any program to be that program itself. Hence, if we store the blueprint for program d inside program d, we get the program $s(d, d)$ (which, from (3) above, is analogous to the cell $\odot_d^d$). Then, we have, from (7) above, for $e = s(d, d)$, (1) above (which is our analog of (4) above).

$\square$ FOR (1) ABOVE

To prove KRT more generally, it would clearly suffice, then, to have instead of d satisfying (8) in the proof above a computable *function* d satisfying

$$(\forall p, x, b)[\varphi_{d(p)}(x, b) = \varphi_p(x, s(b, b))]. \tag{10}$$

Such a computable d exists by S-m-n: φ-program $d(p)$ has p stored inside and, on external input $\langle x, b\rangle$, computes $\varphi_p(x, s(b, b))$.

A biological analog of (10) above does not seem to occur in the world, but the self-knowing program $s(d(p), d(p))$ (corresponding to $d(p)$ satisfying (10) just above) would be analogous to a curious single-celled organism $\odot_{d(p)}^{d(p)}$ which creates a daughter cell self-copy *and then uses it to compute/build* $\varphi_p(x, \text{that daughter cell})$.[5]

References

[Bur70] A. W. Burks, editor. *Essays on Cellular Automata.* Univ. of Illinois Press, 1970.

[Cas71] J. Case. A note on the degrees of self-describing turing machines. *Journal of the ACM*, 18(3):329–338, 1971.

[Cas74] J. Case. Periodicity in generations of automata. *Mathematical Systems Theory*, 8:15–32, 1974.

[Fef60] S. Feferman, *Arithmetization of metamathematics in a general setting*, Fundamenta Mathematicae **49** (1960), 35–92.

[Göd86] K. Gödel, *On formally undecidable prositions of principia mathematica and related systems I*, Kurt Gödel. Collected Works. Vol. I (S. Fererman, ed.), Oxford Univ. Press, 1986, pp. 145–195.

[Neu66] J. Von Neumann. *Theory of Self–Reproducing Automata.* Univ. Illinois Press, 1966. Edited and completed by A. W. Burks.

[RC94] J. Royer and J. Case. *Subrecursive Programming Systems: Complexity and Succinctness.* Research monograph in Progress in Theoretical Computer Science. Birkhäuser Boston, 1994.

[Ric80] G. Riccardi. *The Independence of Control Structures in Abstract Programming Systems.* PhD thesis, SUNY Buffalo, 1980.

[Ric81] G. Riccardi. The independence of control structures in abstract programming systems. *Journal of Computer and System Sciences*, 22:107–143, 1981.

[Rog67] H. Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw Hill, New York, 1967. Reprinted, MIT Press, 1987.

[Roy87] J. Royer. *A Connotational Theory of Program Structure.* Lecture Notes in Computer Science 273. Springer-Verlag, 1987.

[4]In (7) above the left-hand side instance of the s transformation is analogous to the transformation of *forming* the mother cell out of her constituents.

In (7) above the right-hand side instance of the s transformation is analogous to the mother cell's transformation of the new cytoplasm she creates and the copy of the DNA datum she makes into her daughter cell.

[5]In terminology from [Cas71]: such a mother cell begets the partially computable distortion of herself wrought by φ_p and x. $(\ddot{\smile})$