



To Mock a Mocking Bird And Other Logic Puzzles : Including an Amazing Adventure to Combinatory Logic

Raymond M. Smullyan 著

Oxford University Press (2000), 256p., ISBN : 978-0192801425

本書は論理パズルの本である。本書は論理パズルの本としても質が高く面白いのだが、この場であえて紹介する理由は、本書が計算モデルの1つ、「結合子論理 (Combinatory Logic)」^{☆1}に基づく論理パズルを扱っているからである。つまり、本書では、パズルを解くことで結合子論理を楽しく学べるのである。本書は、特に以下のような方にオススメできるだろう。

- 論理パズルが好きな方
- 理論計算機科学分野に興味のある方
- かな漢字変換システム SKK ユーザ^{☆2}

お馴染の正直者と嘘吐きのパズル

本書の約3分の1 (1~8章) はいわゆる「論理パズル」から構成されている。この前半部分において、多く扱われているパズルは、正直者 (いつでも本当のことを言う) と嘘吐き (いつでも嘘を言う、もちろん正直者と見分けがつかない) から、求める情報を得るものである。

しかしながら、本書の正直者と嘘吐きのパズルはバリエーションに富んでいる。たとえば、4章では、「accurate」と「inaccurate」という概念が登場する。ここで inaccurate な者は、真であることを偽だと、そして偽であることを真だと認識している。なので、inaccurate な嘘吐きにたとえば「1+1は2ですか?」と聞くと、彼は「1+1は2でない」と認識しており嘘をつくので「はい」と答える。また、5章以降はストーリー仕立てになっており、退屈させない。5~8章も章ごとにバリエーションが異なっており、中でも8章はメタパズルとなっている。この章で

は、たとえば「正直者が嘘吐きか分からない人物のAという主張からBということが確信できた。ではAはどのような主張か」といったパズルが出題される。

述語論理に基づく論理パズル

一般に「論理パズル」というと命題論理に基づくものが主であろう。本書の1~8章のうち3章以外はそうである。3章では述語論理に基づく論理パズルが扱われている。より具体的には、床屋のパラドックス——「ある床屋は、自分で髭をそらないすべての人の髭を剃り、それ以外の人髭は剃らない」ことが矛盾を導くこと——にちなんだパズルが出題されている。読者の中には大学時代に ϵ - δ 論法における「すべての ϵ に対し、ある δ が存在して…」という議論に苦労した方も多いのではないだろうか (少なくとも私はそうだった)。パズルを通して、述語論理に触れてみるのもよいかもかもしれない。

パズルで学ぼう結合子論理

本書のメインとなるのは、後半部分 (9~25章) の結合子論理に基づく論理パズルである。本書は、結合子論理の各結合子を「鳥」^{☆3}として紹介している。

☆1 組合せ回路 (Combinational Logic) ではない。

☆2 SKKの名前は、結合子論理における等式 $SKK = I$ に由来する (<http://openlab.jp/skk/SKK.html>)。なお、本稿はSKKを利用して作成された。

☆3 本書によると、このことは、結合子論理に大きく貢献した論理学者 Haskell Curry がバードウォッチング好きであったことに由来する。余談であるが、プログラミング言語 Haskell は彼の名に由来する。なお、Curry というプログラミング言語も存在し、こちら彼の名に由来する。

いくつかの「鳥」を紹介したいので、本書の言葉を借りて鳥について説明しよう。森には鳥がいて、鳥 A に鳥 B の名前を呼び掛けると A は、ある鳥の名前を返答する。この名前の鳥を AB で表す。ここで、 $(AB)C$ と書いて「『 A の B に対する返答』の C に対する返答」を表し、 $A(BC)$ と書いて「 A の『 B の C に対する返答』に対する返答」を表すこととしよう。簡単のために $(AB)C$ を ABC と書くことにする。「情報処理」の読者にとっては、「鳥」をプログラム（の挙動）だと思ったほうが分かりやすいかもしれない。すなわち、 AB は B を入力とし A を実行した戻り値（それもまたプログラムである）と読む。

本書のパズルは、主に「…という鳥がいる森には～という鳥もいることを示せ」という形式である。たとえば、あるパズルでは、「mockingbird ($Mx = xx$ を満たす鳥 M) と bluebird ($Bxyz = x(yz)$ を満たす鳥 B) がいる森には、任意の鳥 A に対し、 $Ax = x$ を満たす鳥 x がいる」ことを示すことが要求される（11章の第2問）。

本書に登場する鳥の中で我々にとって興味深いのは、sage bird であろう。この sage bird Θ は $\Theta x = x(\Theta x)$ を満たす鳥で、 Y 結合子や不動点結合子と呼ばれ、再帰関数と関連が深い。具体的には Θ を用いると、再帰的定義を用いずとも再帰関数と同等の処理を実現できることが知られている。きっと「 Y Combinator」と好きなプログラミング言語名と一緒に検索するといろいろと例が見つかることだろう。ここで、 Θ の定義が再帰的だと思われるかもしれないが、mockingbird と bluebird そして、cardinal ($Cxyz = xzy$ を満たす鳥 C) のみから sage bird は定義できる（13章の第2問）。

また、starling ($Sxyz = xz(yz)$ を満たす鳥 S) と kestrel ($Kxy = x$ を満たす鳥 K) も興味深い。驚くべきことに、この2つの鳥がいれば、すべての計算が表現できるのである！ すなわち、starling や

kestrel がいる森は Turing 完全なのだ。本書の、23章では実際にブール値、24章では自然数の計算を行う鳥についてのパズルが含まれている。

ここではあまり触れないが、17章や25章のパズルは Gödel の不完全性定理と関係が深い。これらの章は、数理論理学に興味のある人には特に楽しめることだろう。

プログラムとして書いて遊ぼう！

結合子論理は簡単だが強力な（関数型）プログラミング言語でもある。本書の別の楽しみ方は、プログラムとして結合子を実現し、実行して動作を確かめることである。好きなプログラミング言語で S と K を実装し、実際にそれだけでさまざまな計算ができるかどうかを確かめてみるのも面白いだろう。特に関数オブジェクトや高階関数のある型なしの言語なら鳥をそのまま関数として実現できる。ただし、多くの言語では、sage bird の代わりに $\Theta' xy = x(\Theta' x)y$ となる Θ' を用いなければならないだろう。Lazy K というプログラミング言語は実際に S と K を用いてプログラムを書くことができる（というより S と K しか使えない）。こうした言語を触ってみるのもまた一興か。また、普通のプログラミング言語で書いたプログラムから、 S と K のみからなるプログラムを出力するコンパイラを書くのも面白いかもしれない。

パズルとして面白い。プログラムとして動かしてみてもさらに面白い。こうしたさまざまな楽しみ方のできる本書を私はオススメしたい。

(2014年10月29日受付)

松田一孝（正会員） kztok@is.s.u-tokyo.ac.jp

2009年東京大学大学院情報理工学系研究科博士課程修了。博士（情報理工学）。2010年東北大学大学院情報科学研究科助教。2012年より東京大学大学院情報理工学系研究科助教。プログラミング言語、関数プログラミングに興味を持つ。