A-012

ＳＡＴはＮＰ完全か？：Ｐａｒｔ ２

Is SAT NP- complete? : Part 2

山口人生
(Yamaguchi Jinsei)

１．序論

復習を兼ねて、ＮＰの標準定義を掲載しておこう。まず、使用するアルファベットΣを決める。すると、このΣから生成される“語”（Σの要素の有限列）の集合Σ^*が一つ決まる。この時、Σ^*の部分集合ＬをΣ上の“言語”と言う。以上の準備の下で、ＮＰを

$(\forall L \subset \Sigma^*)((L \in NP) \Leftrightarrow (\exists R$：検査関係$)(\exists k$：自然数$)(\forall w$：語$)(w \in L \Leftrightarrow (\exists y$：語$)(|y| \leqq |w|^k$ ａｎｄ $R(w,y))))$ ・・・（１）

と定義する。ここで、 $\{(w \# y) \mid R(w,y)\}$ は多項式時間ＴＭ計算可能。次に、還元の定義は、

「言語Ｌから言語Ｌ’への還元ｒとは、Σ^*を定義域とし、Ｌ’のアルファベットΣ’から生成されるΣ'^*を値域とするＴＭ計算可能関数で、

$(\forall w$：語$)(w \in L \Leftrightarrow r(w) \in L')$

なる条件を満たすものをいう。」

となる。この概念を基準にして、“多項式時間還元”を、“ ｒは多項式時間計算可能 ”という条件を追加することで定義する。以上の準備の下、Cook の定理が主張しているのは、

「ＮＰの任意の要素Ｌは、（ＮＰの特定要素である）ＳＡＴへ多項式時間還元可能」

という結論である。これを証明するために、Cook は（１）の右辺の条件

$\exists y(|y| \leqq |w|^k$ ａｎｄ $R(w,y))$・・・（２）

を利用した。（２）を知識表現する論理式G(w)が多項式時間で構成できることを示したのである。つまり、

「（２） ⇔ G(w)が充足可能」・・・（３）

なる条件が成立する（ように見える）G(w)の存在を提示してみせた。

構成法の大体の骨子を述べておくと、次のようになる。G(w)に対する適切な {1, 0} 代入は、“ ＴＭの計算過程を表現したもの ”だと想定される。その中で、特に適切な代入は、“ 入力wを計算する時点表示の遷移列 ”に対応する。当然、代入によっては、“ 正当な遷移列になりそこなった残骸 ” になる場合もある。しかし、少なくとも、

「w∈Ｌ ⇔ G(w)を 1 にする代入の存在がR(w,y)なるwの検証計算を表現する」

は成立している。G(w)で使用される論理変数の数は、かなり多いが、|w|に対し、多項式オーダで収まるように構成される。その結果、|G(w)|自体も|w|の多項式オーダで形式表現可能になる。かくして、無事、ＬがＳＡＴに多項式時間還元できたというのが彼の主張である。従来は、私以外の全てのプロが、こう思っていた。問題は、この場合のwの定義域である。前回の[4]における私の主張は、

「この“w⇔G(w)”対応は、定義域がΣ^*ではなく、Ｌに限定されている。それゆえ、還元にはなっていない。」

というものであった。しかし、上の対応を見ると、一見、定義域は全体集合Σ^*になっているように見える。（２）だけ見て、（３）で構成したからである。ここに、間違い発見の困難さがあった。そして、これこそが、巷の研究者が、私の主張を怪しく思う理由なのである。更に、若手の場合、

「この定理で、Cook はチューリング賞を獲得した。いわば、その道のプロ全員が認知した結論だ。その結果に、今更、文句を付けても、言い掛かりに決まっている。」

という拒否反応のようなものが働くから、素直に受け入れることはできまい。では、結局、どちらの言い分が正しいのであろうか？勿論、私の主張の方が正しい。今から、その理由を開陳していく。

２．Ｐａｒｔ １ の解説

決定問題を還元するとは、“ wの正否決定をG(w)の決定問題に委ねる ”ということである。wの正否が決まった後の変換を還元とは言わない。よって、還元の時点では、入力wが正解か不正解かは判明してない。それでも、Cook 還元で、行き先のG(w)は構成（の保証が）されなければならない。前回、問題にしたのは、

「そもそも、任意の入力wに対し、G(w)がマトモに構成可能かどうか？」

であった。上述のように、G(w)は独特の形をしたＣＮＦであり、これに対する適切な {1, 0} 代入が、“時点表示の遷移列”を表現すると想定されている。ここで、wが正解入力の場合には、このwが accept される時点表示の遷移列がある。よって、最低限、これを表現する必要がある。つまり、一般に、G(w)を構成する論理変数として、終了状態を表現する論理変数Ｘが必ず必要になる。実際、従来

㈱Ｉ．Ｉ．Ｉ．代表取締役会長

の総ての証明において、終了状態を表示する論理変数はキチンと出現している。例えば、ｐ(ｎ)を多項式計算ステップ数の上限とした時、

「“ ｐ(ｎ)ステップ目が accept 状態になる ”を表現する論理変数Ｘｐ(ｎ) accept を単独で節にする」

というＧの構成法がある。かなり多くの教科書が、この構成法を採用している。実は、Cook のオリジナル論文が、こちら方式であった。よって、以後、暫くは、この方式を基準にして議論していくことにする。

さて、仮に、G(w)が構成できたとする。そこで保証しているのは、正解が多項式時間で accept 終了する時点表示遷移列の存在であった。ここで、任意のＮＰ問題Ｌに対し、２通りの“ ＮＰアルゴリズム ”を考えることができる。つまり、

＜１＞　不正解の検証も多項式時間で枝別 reject 終了することを保証するアルゴリズム

＜２＞　不正解の検証については、多項式時間で枝別 reject 終了する保証のないアルゴリズム

である。ＮＰの定義は正解終了条件で規定されているため、どちらでも採用できる。ちなみに、ＳＡＴは＜１＞タイプである。実は、両者は、同値になる。つまり、正解検査時間ｐ(ｎ)を過ぎた時点で、不正解検査も強制終了させればいいからである。よって、＜１＞をＮＰの正式の定義に採用しても文句は言えない。この場合、

「不正解入力の枝別 reject 終了計算も、総て、多項式時間で検査可能」・・・（４）

という知識を表現する必要がある。つまり、“ wが不正解の場合、正当な時点表示遷移列なら、ｐ(|w|)時点で枝別 reject 状態になる ”ことを陽に示す必要がある。しかし、G(w)構成法の場合、原理上、これが出来ないから、“ 定義域がＬに限定されている ”と言ったのである。（“accept でない”は“ reject ”ではない。＜２＞が表現できているからＯＫだという言い逃れはできない。何故、＜１＞が表現できないのかと問うているのだ。「“accept でない”を“ reject ”と解釈する」なんていうのは最悪。）

３．修正Ｇ

これに対し、別の教科書では、G(w)の節として、

「Ｘｐ(ｎ)accept∨Ｘｐ(ｎ)reject・・・」

を採用している。これにより、“ 不正解入力に対しても、ｐ(ｎ)ステップ内で枝別 reject 終了 ”という知識を表現することが可能になると思ったのであろう。しかし、この構成法は自滅する。以下、その理由を述べていこう。

この場合、正解入力には{Ｘｐ(ｎ)accept＝1, Ｘｐ(ｎ)reject＝0} 代入を、不正解入力には、 {Ｘｐ(ｎ)accept＝0, Ｘｐ(ｎ)reject＝1} 代入を対応させるつもりであろう。けれども、これなら、“ 正解入力wに対し、適切な代入σで、G(w)が充足可能になる ”のと同程度の確からしさで、

「不正解入力wに対し、適切な代入γで、G(w)が充足可能になる」・・・（５）

のだ。つまり、

「wが不正解入力の場合、枝別 reject 終了する計算過程を表現できる代入γが存在する」・・・（６）

のである。一見すると、（６）により、上の論点であった多項式時間枝別 reject 終了表現が可能になるように見える。（実は、（６）は、必ずしも、（４）を保証しない。）ところが、（６）が成立すれば、（５）により、Gは還元でなくなるのだ。これで、この修正法は使えないことが判る。

その他の方式は、どうか？例えば、｛Ｘｐ(ｎ)accept＝1, Ｘｐ(ｎ)reject＝1｝という新解釈を採用してみよう。この場合には、見事、（４）の保証になっていると看做せる。しかし、こちらの解釈でも、（５）＋（６）の問題は、形を変えて、そのまま発生する。

ここまで来ると、不正解検査の長さは無視したくなるであろう。しかし、これで、やっと、前回の[4]の主張に辿り着くことができるのである。＜１＞では、wが不正解の場合、G(w)が構成できたことにはならない。そして、＜１＞と＜２＞は同値だと想定されているのだ。よって、以後、これを“ 不正解表現問題 ”と呼ぶことにしよう。以上の探究により、

定理　（ＺＦＣ）

従来の「ＳＡＴはＮＰ完全」の証明は不完全で“ 不正解表現問題 ”をクリヤできなければ、Cook 流の証明は破綻する。　┤

が証明できたことになる。

４．まとめ

実は、前回発表した[4]の層化還元の節は、今回の内容を超えた議論を誘発させるための呼び水の役割を果たしている。なるほど、“ Ｌ∈ＮＰ ”はキチンと定義されている。しかし、ある種のＬ’に対し、Ｌ’∈ＮＰが計算量理論において証明できるかどうかの保証は別儀である。最後には、この方面の議論が、「Ｐ＝ＮＰ？」問題の解決に繋がっていく。

参考文献

[1]S. Cook, The complexity of theorem-proving procedures, *Conference Record of Third Annual ACM Symposium on Theory of Computing*, 151-158, 1971.
[2]山口人生、「Ｐ＝ＮＰ？」問題の解決、*Proceeding of IPSJ 64,* Vol.1, 183-184, 2002.
[3]山口人生、計算量理論の存亡（１）:「Ｐ＝ＮＰ？」問題の解決、 *I. I. I.*, 2002.
[4]山口人生、ＳＡＴはＮＰ完全か？：Cook の定理は間違っていた！、*Proceeding of FIT 2004,* 2004.