

安全な One-way Function について

陳 致中 笠井 琢美

電気通信大学 計算機科学科

あらまし 論理式の充足可能性問題に基づいて一つの関数を構成する。この関数の計算は効率的に計算できるが、この関数の逆関数を計算することは少なくとも論理式の充足可能性問題を解くことと同じ困難さを持つ。更に、この関数は定義域のほとんどの所で一対一になる。また、対関数 (pairing function) を利用して、この関数を実際に計算するのにかかる時間とメモリを減らすアルゴリズムを提案する。

1、まえがき

関数 $f(x)$ が One-way Function であるとは $f(x)$ の計算が容易であるが、この f からその逆関数 f^{-1} を求めることが計算量的に無理であるときをいう。暗号学において今まで得られたほとんどの結果は One-way Function が存在するという仮定に基づいている。しかし、One-way Function が本当に存在するかと言う問題はまだ証明されていない。暗号学においてこの問題が一番基本的な問題であろう。

もし、近似不能な述語 (unapproximable predicate) $B(x)$ が存在しかつ関数 $f(x)$ と $B(f(x))$ の計算が容易であれば、 $f(x)$ が One-way Function になるということを A.C.Yao は証明している ([2] に参照)。この結果より、平方剰余問題が難しいと言う仮定 (QRA) あるいは離散対数問題が難しいと言う仮定 (DLA) をすれば、One-way Function が簡単に構成される。実は、この二つの仮定に基づいて多くの結果は獲られている。平方剰余問題と離散対数問題のどちらも明かに NP に属するが、NP 完全であるかどうかということはわかっていない。よって、この二つの問題に基づいて構成した One-way Function の安全性に疑問を持つ可能性が小さくないであろう。また、実用的な点から見て、これらの仮定に基づく One-way Function は非常に長い桁数の整数計算を必要とし時間量が高価なものとなる。

NP 完全問題に基づいて One-way Function を構成するのは非常に自然である。もちろん、NP 完全問題を One-way Function の構成に適用する場合には次のような点に留意する必要がある: NP 完全の議論は、ある問題の計算の複雑さを最悪の場合の計算量で評価している。One-way Function の安全性を議論するには、最悪の場合だけでは不十分で、平均的な場合の計算量だけでなく、ほとんどすべての場合に対して考察しなければならない。よって、NP 完全問題を One-way Function の構成に適用するとき、どの NP 完全問題を使うかを選ばなければならない。ほとんどの NP 完全問題が論理式の充足可能性問題から還元されているのであるので、論理式の充足可能性問題は NP 完全問題の中で一番安全であろう。

定義域 (Domain) を制限せず、一対一でありかつ逆関数の計算が NP 困難であ

るような関数は、著者の知るかぎり、まだ構成されていない。本稿では、論理式の充足可能性問題に基づいて一つの関数を構成する。この関数の逆関数を計算することが少なくとも論理式の充足可能性問題を解くことと同じ難しさを持つということを示す。従って、この関数の逆関数を求めることが難しいであろう。また、この関数が定義域のほとんどの所で一対一になるといういい性質を持つことを示す。その次に、対関数 (pairing function) を利用して、この関数を計算するのにかかる時間とメモリを減らすアルゴリズムを提案する。

2. 関数 $\Phi(X)$ の構成

本節では、 $\Phi(X): \{0,1\}^N \rightarrow \{0,1\}^M$ を構成する。ここで、 $N=2n^2 * \log(30n) + 4n$ 、 $M=2n^2 * (\log(30n) + 2)$ 、 n は任意の整数である。 $\Phi(X)$ を計算するのに、三つのリスト L_1 、 L_2 、 L_s と L_1 上の全順序 \langle_1 、 L_2 上の全順序 \langle_2 、 L_s 上の全順序 \langle を用いる。 L_1 、 L_2 、 L_s 、 \langle_1 、 \langle_2 、 \langle は付録 A に定義されている。

$X = u v b$ を入力とする。ただし、 $u = u_1 u_2 \dots u_m$ 、 $v = v_1 v_2 \dots v_m$ 、 $b = b_1 \dots b_{4n-1}$ 、 $1 \leq i \leq m$ なる各 i に対して、 $|u_i| = |v_i| = \lceil \log(30n) \rceil$ 、 $|b| = |b_1 \dots b_{4n-1}| = 4n-1$ 、 $m=n^2$ とする。関数 $\Phi(X)$ の構成は五つのステップに分かっている。第一ステップ目で、 u でリスト L_1 から m 個のクローズ C_1 、 C_2 、 \dots 、 C_m を取る。第二ステップ目で、 v でリスト L_2 から m 個のクローズ C_{m+1} 、 C_{m+2} 、 \dots 、 C_{2m} を取る。論理式 $F_{uv} = C_1 \wedge C_2 \wedge \dots \wedge C_{2m}$ とおく。その次に、第三ステップ目で、 b で F_{uv} を次のように変換する：もし $b_i = 0$ ならば、 x_i と \bar{x}_i を交換する。第四ステップ目で、全順序 \langle で F_{uv} をソートする。得られた論理式 $C_1' \wedge C_2' \wedge \dots \wedge C_{2m}'$ を F_x とする。そして、第四ステップ目で、リスト L_s を用いて、 F_x を 0 と 1 の文字列に符号化する。最後に得られたビット列 $w = w_1 w_2 \dots w_{2m}$ は出力 Y である。ただし、 $|w_i| = |u_i| + 2$ 。

これから、 L_1 からの C_1 、 C_2 、 \dots 、 C_m の取り方を説明する。 L_2 からの C_{m+1} 、 C_{m+2} 、 \dots 、 C_{2m} の取り方はほとんど同じであるので省略する。 L_1 からの C_1 、 C_2 、 \dots 、 C_m の取り方は次のとおりである： L_1 の先頭の要素を一番の要素とし、 $\text{Rep}(u_i)$ を u_i が表す自然数とする。すると、 C_1 は L_1 の $\text{Rep}(u_1) + 1$ 番目の要素であり、 C_2 は L_1 の $[\text{Rep}(u_1) + 1] + [\text{Rep}(u_2) + 1]$ 番目の要素であり、 \dots 、 C_i は L_1 の $[\text{Rep}(u_1) + 1] + [\text{Rep}(u_2) + 1] + \dots + [\text{Rep}(u_i) + 1]$ 番目の要素である。

L_s を用いる F_x の符合化の仕方は L_1 からの取り方の逆演算とも言える。 L_s の先頭の要素を一番の要素とする。 C_1' が L_s の N_1 番目のクローズであり、 C_2' が L_s の N_2 番目のクローズであり、 \dots 、 C_{2m}' が L_s の N_{2m} 番目のクローズであるならば、 w_1 は $N_1 - 1$ の二進表現であり、 w_2 は $N_2 - N_1 - 1$ の二進表現であり、 \dots 、 w_{2m} は $N_{2m} - N_{2m-1} - 1$ の二進表現である。

$|w_i| = |u_i| + 2$ としてもいいことを証明しなければならない。この証明は次の Claim 1 と Claim 2 の証明になる。

Claim 1. リスト L_1 において (l_{i1}, l_{j1}, l_{k1}) と (l_{i2}, l_{j2}, l_{k2}) の距離が d であるならば、 L_s において (x_{i1}, x_{j1}, x_{k1}) と $(\bar{x}_{i2}, \bar{x}_{j2}, \bar{x}_{k2})$ の距離が $(8/3)d + 14$ 以下である。ただし、 $i_1 \leq i_2$ 、 $j_1 \leq j_2$ 、 $k_1 \leq k_2$ 。

証明、簡単な数え上げである。 ■

Claim 2. リスト L_2 において (l_{i1}, l_{j1}, l_{k1}) と (l_{i2}, l_{j2}, l_{k2}) の距離が d であるならば、 L_s において (x_{i1}, x_{j1}, x_{k1}) と $(\bar{x}_{i2}, \bar{x}_{j2}, \bar{x}_{k2})$ の距離が $(8/3)d + 14$ 以下である。ただし、 $i_1 \leq i_2$ 、 $j_1 \leq j_2$ 、 $k_1 \leq k_2$ 。

証明、 簡単な数え上げである。 ■

3、 $\Phi(X)$ が一対一になる確率

本節で、 $\Phi(X)$ がほとんどの入力に対して一対一になることを示す。

$X = uvb$ とする。 u と v で取ってきたクローズからなる論理式を F_{uv} とし、 F_{uv} を b で変換された後に得られた論理式を F_{uvb} とし、 F_{uvb} を全順序 \prec でソートした後に得られた論理式を F とする。

$X' = u'v'b'$ に対して、 $\Phi(X') = \Phi(X)$ となったと仮定する。 u' と v' で取ってきたクローズからなる論理式を $F_{u'v'}$ とする。すると、 b' で $F_{u'v'}$ を変換し、さらに全順序 \prec でソートした後に得られた論理式は F となる。また、次の補題が成り立つ。

補題: ある b_1 が存在して、 b_1 で F_{uv} を変換し、さらに全順序 \prec でソートした後に得られた論理式は $F_{u'v'}$ となる。

証明: まず、 F を b で変換して、さらに全順序 \prec でソートした後に得られた論理式が F_{uv} に戻り、 F を b' で変換して、さらに全順序 \prec でソートした後に得られた論理式が $F_{u'v'}$ に戻ることに注意。すると、 $b_1 = b \otimes b'$ で F_{uv} を変換し、さらに全順序 \prec でソートした後に得られた論理式が $F_{u'v'}$ となる。ここで、 \otimes は exclusive nor ($0 \otimes 0 = 1, 0 \otimes 1 = 0, 1 \otimes 0 = 0, 1 \otimes 1 = 1$) である。 ■

上の補題より、もしすべての $b' \in \{0, 1\}^{4n-1} - \{1^{4n-1}\}$ に対して、 b' で F_{uv} を変換した後に得られた論理式 $F_{u'v'}$ にあるクローズの中で一つ以上のクローズが L_1 の元でもないし L_2 の元でもないならば、 $\Phi(X)$ が一対一になる。よって、すべての $b' \in \{0, 1\}^{4n-1} - \{1^{4n-1}\}$ に対して、 b' で F_{uv} を変換した後に得られた論理式 $F_{u'v'}$ にあるクローズの中で一つ以上のクローズが L_1 の元でもないし L_2 の元でもない確率 P を計算すればよい。以下で、 P を求める。

以下で、 $b' = b'_1 b'_2 \dots b'_{4n-1} \in \{0, 1\}^{4n-1} - \{1^{4n-1}\}$ を固定して議論する。

b' にある0の個数を r とする。すると、 $1 \leq r \leq 4n-1$ 。 $b_{i1} = b_{i2} = 1, b_{j1} = b_{j2} = 0, i_1 < j_1 < i_2 < j_2$ と仮定する。すると、もし、 $(x_0, x_{i1}, \bar{x}_{j1}), (x_0, \bar{x}_{j1}, x_{i2}), (x_0, \bar{x}_{j1}, \bar{x}_{j2}), (\bar{x}_0, \bar{x}_{i1}, x_{j1}), (\bar{x}_0, x_{j1}, x_{j2}), (x_{i1}, \bar{x}_{j1}, x_{i2}), (\bar{x}_{j1}, x_{i2}, \bar{x}_{j2})$ の中で一つ以上が F_{uv} に現れれば、 b' で変換した後に得られた論理式は L_1 と L_2 から取れない。 $(x_0, x_{i1}, \bar{x}_{j1}), (x_0, \bar{x}_{j1}, x_{i2}), (x_0, \bar{x}_{j1}, \bar{x}_{j2}), (x_{i1}, \bar{x}_{j1}, x_{i2}), (\bar{x}_{j1}, x_{i2}, \bar{x}_{j2})$ のようなクローズをいいクローズという。もっと一般的に、便宜上、次のようにいいクローズを定義する。

定義: リテラル l_i が肯定リテラルであるとは $l_i = x_i$ のときをいう。リテラル l_i が否定リテラルであるとは $l_i = \bar{x}_i$ のときをいう。

定義: クローズ (l_i, l_j, l_k) がいいクローズとは b' で次のように (l_i, l_j, l_k) を変換した後に得られたクローズにあるリテラルがすべて肯定リテラルかすべて否定リテラルであるときをいう: $b'_{i1} = 0$ ならば l_i を \bar{l}_i に変え、 $b'_{j1} = 0$ ならば l_j を \bar{l}_j に変え、 $b'_{k1} = 0$ ならば l_k を \bar{l}_k に変える。

明らかに、もし、一つ以上のいいクローズが F_{uv} に現れれば、 b' で変換した後に得られた論理式は L_1 と L_2 から取れない。 F_{uv} にいいクローズがないという事件が起こる確率を P_r とする(r は b' にある0の個数である)。これから、 P_r を計算する。

定理: もし $1 \leq r \leq 2n-1$ ならば、 $P_r < \exp[-(2/25)Jnr]$ 。もし $2n \leq r \leq 4n-1$ ならば、 $P_r < \exp[-(2/25)Jn(4n-r-1)]$ 。

証明: b' にある 0 の個数が r であるとき、リスト L_1 にあるいいクローズの個数を TOT とすると、 $TOT = (4n-r-1)r + \frac{1}{2}r(r-1) + \frac{1}{2}(4n-r-1)r(r-1) + \frac{1}{2}(4n-r-1)(4n-r-2)r$ となる。明らかに、この TOT 個のいいクローズがリスト L_1 の末に近づけば近づくほど、 P_r が大きくなる。いいクローズがリスト L_1 の末に近づく程度は、 b' のどの r ビットが 0 になるかに依存する。 L_2 についても同じ議論が成り立つ。以下の議論は特にリスト L_1 についてのものであるが、 L_2 についても同じ議論が成り立つ。便宜上、 L_2 についての議論を省略する。

もし $1 \leq r \leq 2n-1$ ならば、 b' の $4n-r$ 番目、 $4n-r+1$ 番目、 \dots 、 $4n-1$ 番目のビットを 0 にしたときには、いいクローズはリスト L_1 の末に一番近づく。このとき、リスト L_1 の先頭から $LEN = [\frac{1}{2}(4n-1)(4n-2)] + [\frac{1}{2}(4n-2)(4n-3)] + \dots + [\frac{1}{2}(2n)(2n-1)]$ 番目のクローズまでの区間に注目しよう。この区間の任意の点から $30n$ 先の所までの間に少なくとも $2r$ 個のいいクローズがある。よって、 $1 \leq r \leq 2n-1$ のとき、 $P_r < [1 - (2r/30n)]^{2^q}$ 。ただし、 $q = \min(n^2, \lfloor LEN/(30n) \rfloor) = \lfloor (3/5)n^2 \rfloor$ 。従って、 $P_r < \exp[-2 \lfloor (3/5)n^2 \rfloor (2r/30n)] = \exp(-\lfloor (2/25) \rfloor nr)$ 。

もし、 $2n \leq r \leq 4n-1$ ならば、 b' の 1 番目、2 番目、 \dots 、 r 番目のビットを 0 にしたときには、いいクローズはリスト L_1 の末に一番近づく。このとき、リスト L_1 の先頭から $[(4n-2) + (4n-3) + \dots + (4n-r)]$ 番目のクローズまでの区間に注目しよう。この区間にある任意の連続の三つのクローズにはちょうど一つがいいクローズである。また、リスト L_1 の $\frac{1}{2}(4n-1)(4n-2)$ 番目のクローズから $LENGTH = [\frac{1}{2}(4n-1)(4n-2)] + [\frac{1}{2}(4n-2)(4n-3)] + \dots + [\frac{1}{2}(4n-r-1)(4n-r-2)]$ 番目のクローズまでの区間に注目しよう。この区間にある任意の点から $30n$ 先の所までの間に少なくとも $2(4n-r-1)$ 個のいいクローズがある。よって、 $P_r < \{ [1 - (8nr - r^2 - 3r)/(48n^2 - 36n + 6)]^{q_1} * [1 - 2(4n-r-1)/(30n)]^{q_2} \}^2$ 。ただし、 $q_1 = \lfloor \frac{1}{2}(4n-1)(4n-2)/(30n) \rfloor$ 、 $q_2 = \min(n^2, \lfloor (LENGTH - \frac{1}{2}(4n-1)(4n-2))/(30n) \rfloor)$ 。 $2n \leq r \leq 4n-1$ を思い出すと、 $P_r < \exp[-2(3/5)n^2 * (4n-r-1)/(15n)] < \exp[-\lfloor (2/25) \rfloor n(4n-r-1)]$ となるのは明かであろう。 ■

上の定理より、 r に関する簡単な帰納法で $\binom{4n-1}{r} P_r < \exp[-\lfloor (2/25) \rfloor n]$ を示すことができる。

さて、 $P > 1 - \sum_{r=1}^{4n-1} P_r \binom{4n-1}{r} > 1 - (4n-1) \exp[-\lfloor (2/25) \rfloor n] > 1 - \exp(-cn)$ 。

ただし、 c は定数である。

よって、 $\Phi(X)$ が一対一になる確率 P は $1 - \exp(-cn)$ である。

4. Φ^{-1} の計算の難しさ

3 節で議論したように、 $y \in \text{Range}(\Phi)$ に対し、 $\Phi(x)$ となる x がいくつあるかもしれない。以下で、 $\Phi^{-1}(y)$ で $\Phi(x)=y$ を満たす x の集合を表す。

Φ^{-1} の計算の難しさを証明するのに集合 $\text{Pref}(\Phi^{-1})$ を導入する。

$\text{Pref}(\Phi^{-1}) = \{ \langle y, w \rangle \mid y \in \text{Dom}(\Phi^{-1}) \text{ かつ } w \text{ が } \Phi^{-1}(y) \text{ のある元の接頭語である} \}$

$\Phi(X)$ の長さ X の長さの差が小さいので、次の命題が成り立つ。

命題: $\text{Pref}(\Phi^{-1}) \in P \iff \Phi^{-1}$ を多項式時間で計算する決定性 Turing transducer が存在する。

証明: [3] に参照。 ■

上の命題より、 Φ^{-1} の計算の難しさを証明するのに、 $\text{Pref}(\Phi^{-1})$ の難しさを証明すればよい。次に $\text{Pref}(\Phi^{-1})$ の難しさを示す。そのために、問題 Prange を導入する。

問題 Prange: 入力 $Y \in \{0, 1\}^M$ が与えられたら、 $Y = \Phi(X)$ を満たす X が存在するかどうかを判定する問題。

補題: 問題 Prange が NP 困難 (under many-one reduction) である。

証明: まず、次の問題 3SAT' が NP 困難であることを示す。

3SAT' = { F | F は乗法標準形であり、かつ F にある各クローズがちょうど三つの互いに異なるリテラルを持ち、かつどのクローズにもあるリテラルとそのリテラルの否定が同時に現れなく、かつ F にあるクローズの総数が変数の総数の二倍であり、かつ F が充足可能である。 }

M を非決定的に多項式時間で止まるチューリング機械とする。 x を M への入力とする。 Cook は、 M が x を受理するときは論理式 F_0 が充足可能であるときまたそのときに限るということを証明した。 Cook が作った論理式 F_0 にクローズの総数は変数の総数の 1.9 倍ぐらいとなっている。そこで、 F_0 より、パディング手法を用いて、 M が x を受理する必要十分条件が論理式 $F_0' \in 3SAT'$ を満たす論理式 F_0' を構成することができる。よって、問題 3SAT' が NP-困難である。

次に問題 P が 3SAT' に還元可能であることを示す。

F を論理式とする。一般性を失わずに、 F が、充足可能かどうかだけを除き 3SAT' の他の条件を満たすと仮定してもいい。

F にある各クローズ (ℓ_i, ℓ_j, ℓ_k) を $(\ell_i, \ell_j, h_1) \wedge (\bar{t}, \ell_k, \bar{h}_1) \wedge (\bar{\ell}_i, \bar{\ell}_j, h_2) \wedge (\bar{\ell}_k, t, \bar{h}_2)$ で置き換える。ただし、 t は 1 を表し、 h_1 と h_2 は新しい変数であり、二重否定が肯定になる。こうした後に得られた論理式を F_1 とする。また、 F_1 にある変数 (t を含む) を n 個とする。そうすると、 F_1 にあるクローズの総数が $\lceil 8(n-1)/5 \rceil$ となる。

t に名前 x_0 を付け、 F_1 にある他の変数に名前 $x_2, x_4, \dots, x_{2n-2}$ を付ける。更に、 $3n$ 個の新しい変数を導入し、 $x_1, x_3, \dots, x_{2n-1}, x_{2n}, x_{2n+1}, \dots, x_{4n-1}$ と名前付ける。

これから、乗法標準形論理式 F_2 を構成する。便宜上、 F_2 をクローズの集合と見なす。次に、 F_2 を構成するアルゴリズムを示す。このアルゴリズムの中に、 Total が F_2 に入れるクローズの個数であり、 L_1 が付録 A に定義されたリストであり、 $L_1[i]$ が L_1 の i 番目のクローズである。

F_2 の構成:

```

F2 ← {}; Total ← n2 - ⌈4(n-1)/5⌉; Span ← 30n; Pointer ← 0;
N1 ← { 2i | 0 ≤ i ≤ n-1 }; N2 ← { 2i-1 | 1 ≤ i ≤ n } ∪ { i | 2n ≤ i ≤ 4n-1 };
Dummy ← { (xi,  $\bar{x}_j$ , xk) | i ∈ N1 かつ j, k ∈ N2 かつ i < j < k }
           ∪ { (xi, xj,  $\bar{x}_k$ ) | i ∈ N1 かつ j, k ∈ N2 かつ i < j < k };
While Total > 0 do
  Pointer ← Pointer + Span;
  while L1[Pointer] ∉ Dummy do Pointer ← Pointer - 1 od;
  F2 ← F2 ∪ { L1[Pointer] }; Total ← Total - 1;
od;

```

出力: F_2

ここでまず、上のアルゴリズムの正当性を示す。明らかに、このアルゴリズムが必ず止まる。というのは、リスト L_1 のどこからも幅 $30n$ 先までの間に必ず Dummy の元があるからである。また、Pointer の値がリスト L_1 のサイズ $2n(4n-1)(4n-2)$ を越えることはない。さらに、Pointer の最後に指している所からリスト L_1 の末までの間にあるクローズはすべて Dummy の元である。上のアルゴリズムが n の多項式時間で止まるのも明白であろう。

以下で便宜上、 F_1 もクローズの集合と見なす。

$F' = F_1 \cup F_2 \cup F_3$ と定める。ここで $F_3 = \{(\bar{r}_i, r_j, r_k) \mid (r_i, r_j, r_k) \in F_2\}$ 。
(F_2 の元 (r_i, r_j, r_k) がリスト L_1 の d 番目のクローズであれば、 (\bar{r}_i, r_j, r_k) もリスト L_2 の d 番目のクローズであることに注意)。さらに、付録 A で定義された半順序 \prec で F' をソートし、リスト L_5 を使って F' を符号化する。 F' の符号化が存在することは F_2 と F_3 の構成方と 3 節の claim 1 と claim 2 によって保証されている。以下で便宜上、 F' と F' の符号化を区別せずに議論する。

以下で、 $F \in 3SAT' \iff \exists X (F' = \Phi(X))$ ということを示す。

$F \in 3SAT'$ とする。そうすると、 F を真にする割当 ass がある。 F にある変数を y_1, y_2, \dots, y_l とする。 F を次のように変換する：もし、 y_i が ass によって偽と割り当てられたら、 F にある y_i とその否定とを交換して、得られた論理式を F_a とする。明らかに、 F_a にある各クローズは少なくとも一つの肯定リテラルを持つ（ここで、肯定リテラルとは y_i のこと）。

求めたい X を $u_1 u_2 \dots u_m v_1 v_2 \dots v_m$ とする。ここで $1 \leq i \leq m$ を満たす i に対して、 $|u_i| = |v_i| = \lceil \log(30n) \rceil$ 、 $|b| = |b_0 b_1 \dots b_{4n-1}| = 4n$ 、 $m = n^2$ 。また、一般性を失わずに、 F_1 にある変数 $x_0, x_2, \dots, x_{2n-2}$ の中で、 x_0, x_2, \dots, x_{2l} が F にある変数を表し、 $x_{2l+2}, x_{2l+4}, \dots, x_{2n-2}$ が F_1 にある他の変数を表すとしてもいい。 b_0, b_2, \dots, b_{2l} の値を ass に従って与え、 $b_{2l+2}, b_{2l+4}, \dots, b_{2n-2}$ の値を付録 B で示されるように与え、 b の他のビットをすべて 1 にする。それから、 b で F' を次のように変換する： $0 \leq i \leq 4n-1$ に対し、 $b_i = 0$ なら F' にある x_i とその否定とを交換し、得られた論理式を F_b' とする。そうすると、 F_b' にちょうど二つの肯定リテラル（ここで、肯定リテラルとは x_i のこと）を持つクローズの総数はちょうど一つの肯定リテラルを持つクローズの総数と等しい。また、 F_b' に三つの肯定リテラルを持つクローズと一つの肯定リテラルも持たないクローズは存在しない。これは、 F_2 と F_3 の構成と b の定め方から明かであろう。ここで、特に注意すべきなのは F_1 にある変数の割当がどう変わっても $F_2 \cup F_3$ が変わらないということである。この事実より、 $u_1, u_2, \dots, u_m, v_1, v_2, \dots, v_m$ が存在して、 $u_1, u_2, \dots, u_m, v_1, v_2, \dots, v_m$ で F_b' にあるすべてのクローズが取れる。よって、 $F' = \Phi(X)$ を満たす X が存在する。

一方、 $\exists X (F' = \Phi(X))$ とする。また、 $X = u_1 u_2 \dots u_m v_1 v_2 \dots v_m$ とする。そうすると、 b で F' を次のように変換する： $b_i = 0$ ならば、 F' にある変数 x_i とその否定とを交換する。変換した後に得られた F' にあるどのクローズも少なくとも一つの肯定リテラルを持つ（ここで、肯定リテラルとは x_i のこと）。よって、 b が F' を真にする割当である。 F' の構成仕方より、 $b_f = b_0 b_2 \dots b_{2n-2}$ は F を真にする割当である。従って、 $F \in 3SAT'$ 。 ■

定理： $\text{Pref}(\Phi^{-1})$ は NP 完全である。

証明：明らかに、 $\text{Pref}(\Phi^{-1})$ が非決定性チューリング機械によって多項式時間で受理される。また、Prange が $\text{Pref}(\Phi^{-1})$ に多項式時間還元可能 (Turing reducible) であることも明かであろう。 ■

5、 $\Phi(X)$ の計算

$\Phi(X)$ の構成のしかたは2節でおおまかに説明した。2節で説明したように $\Phi(X)$ を計算するのに一番時間とメモリがかかる所はリスト L_1 とリスト L_2 からクローズを取ることと最後の符号化である。次のアルゴリズム Pickout でリスト L_1 からメモリと時間があまりかからずにクローズの取り方を示す。リスト L_2 からクローズの取り方がリスト L_1 からクローズの取り方と似ているので、ここで省略する。

理解しやすいため、精密なアルゴリズム Pickout を示すまえに、まず、アルゴリズム Pickout を説明する。ビット列 $u = u_1 u_2 \dots u_m$ と自然数 n を入力とする。2節で、説明したように、 u で L_1 から m 個のクローズを取る。 $1 \leq r \leq m$ に対し、 r 番目のクローズ $(l_{i_1}, l_{i_2}, l_{i_3})$ はリスト L_1 の $N_r = \sum_{j=1}^r [\text{Rep}(u_j) + 1]$ 番目の要素である。一番簡単な方法はリスト L_1 を予めメモリの中に計算しておいて、 N_r が分かってから L_1 の N_r 番目のクローズを取ってくればよい。しかし、明らかにこの単純な方法はメモリをたくさん要求する。下のアルゴリズム Pickout では、 L_1 をメモリの中に蓄えずに、 N_r で直接に $(l_{i_1}, l_{i_2}, l_{i_3})$ を計算できることを示す。明らかに、 i_1, i_2, i_3 が計算でき、かつ $l_{i_1}, l_{i_2}, l_{i_3}$ の中のどれが否定リテラルとなっているかを計算できれば、 $(l_{i_1}, l_{i_2}, l_{i_3})$ を計算できる。ここで、まず、リスト L_1 の形をよく見よう。 L_1 の形が図1に示されるとおりである。

図1の左端は L_1 の先頭であり、右端は L_1 の末端である。また、 $0 \leq i \leq 4n-3$ をみたす i に対し、block i にあるすべてのクローズは (x_i, \dots, \cdot) と $(\bar{x}_i, \dots, \cdot)$ の形をしている。block i のサイズを size i とすると、 $\text{size } i = 3 \binom{4n-1-i}{2}$ 。

メモリに size 0、size 1、 \dots 、size $4n-3$ を蓄えれば、 i_1 が簡単に計算されることは明かであろう。 i_1 が計算されたとし、 $N_r' = (\sum_{j=0}^{i_1} \text{size } j) - N_r$ とおく。すると、 $(l_{i_1}, l_{i_2}, l_{i_3})$ は、block i_1 の右端から数え始めるときの N_r' 番目のクローズである。これから、 N_r' より対関数 (pairing function) を使って i_2 と i_3 を計算する。

定義: 関数 $G(i, j) = \frac{1}{2} [(i+j)^2 + 3i+j]$ のことを対関数と呼ぶ。

命題^[1]: 関数 $Q_1(k) = \lfloor \frac{1}{2} (\lfloor (8k+1)^{1/2} \rfloor + 1) \rfloor - 1$ 、 $Q_2 = 2k - [Q_1(k)]^2$ と定義すると、 $k = G(i, j)$ のとき $i = \lfloor \frac{1}{2} [Q_2(k) - Q_1(k)] \rfloor$ 、 $j = Q_1(k) - \lfloor \frac{1}{2} [Q_2(k) - Q_1(k)] \rfloor$ 。

無限リスト $L_G = \langle (0, 0), (0, 1), (1, 0), (0, 2), (1, 1), (2, 0), \dots, (i, j), \dots \rangle$ と定義する。ただし、 $0 \leq i, 0 \leq j$ に対して、 (i, j) がリスト L_G の第 $G(i, j) + 1$ 番目の元である。また、集合 $S_R = \{(k, l) \mid i_1 + 1 \leq k < l \leq 4n-1\}$ と定義する。更に、 S_R を次のようにソートする： (k_1, l_1) と (k_2, l_2) を S_R の元とすると、 $k_1 > k_2$ のときあるいは $k_1 = k_2$ 並びに $l_1 > l_2$ のとき、 (k_1, l_1) を (k_2, l_2) の前に置く。こうソートした後に得られたリストを L_R とする。明らかに、リスト L_1 の block i_1 とリスト L_R の間にいい対応がある。

補題: (i, j) と (k, l) をそれぞれリスト L_G と L_R の q 番目の元とすると、 $k = 4n - 2 - (i + j)$ 、 $l = 4n - 1 - i$ 。

証明: 自明である。 ■

上の補題より、 $q_1 = \lfloor \frac{1}{3} N_r' \rfloor$ 、 $q_2 = N_r' - 3q_1$ 、 $q = q_1 + \lfloor \frac{1}{3} (3 - q_2) \rfloor$ とし、 $q = G(i, j)$ となる i, j を命題のように計算すれば、 $i_2 = 4n - 2 - (i + j)$ 、 $i_3 = 4n - 1 - i$ となる。 $(\ell_{i_1}, \ell_{i_2}, \ell_{i_3})$ の三つのリテラルの中でちょうど一つが否定リテラルになる。明らかに、 $q_2 = 0$ ならば一番目のリテラルが否定であり、 $q_2 = 2$ ならば二番目のリテラルが否定であり、 $q_2 = 3$ ならば三番目のリテラルが否定である。

上の説明をまとめると、次のアルゴリズムになる。

アルゴリズム Pickout:

入力: ビット列 $u = u_1 u_2 \dots u_m$ と n 。ここで、 $1 \leq i \leq m$ を満たす i に対して $|u_i| = \lceil \log(30n) \rceil$ 、 $m = n^2$ 。

```

Block[0] ← 0;   Block[1] ←  $\frac{3}{2}(4n-1)(4n-2)$ ;   /Block がサイズ  $4n-1$ 
N1 ← 0;   Q ← {};   /のリストである。/
for j ← 2 to  $4n-2$  do
  Block[j] ← Block[j-1] +  $\frac{3}{2}(4n-j)(4n-j-1)$  od;
for j ← 1 to m do
  if Rep(uj) ≥ 30n halt fi;   /Rep(ui) が ui が表す自
  N1 ← N1 + Rep(uj) + 1;   /然数である。/
  if Block[k] < N1 ≤ Block[k+1] i1 ← k fi;   /k を binary
  N2 ←  $\lfloor (Block[i_1] - N_1) / 3 \rfloor$ ; N3 ←  $(Block[i_1] - N_1) - 3 * N_2$ ; search で見
  if N3 ≠ 0 N2 ← N2 + 1 fi;   /つける/
  r1 ←  $\lfloor \frac{1}{2} [ \lfloor (8N_2 + 1)^{1/2} \rfloor + 1 ] \rfloor$ ; r2 ←  $2N_2 - r_1^2$ ;
  px ←  $\lfloor \frac{1}{2} (r_2 + r_1) \rfloor$ ; py ←  $r_1 - \lfloor \frac{1}{2} (r_2 + r_1) \rfloor$ ;
  qx ← px - py; qy ← px + py; i2 ←  $4n - 2 - q_y$ ; i3 ←  $4n - 1 - \lfloor \frac{1}{2} (q_x + q_y) \rfloor$ ;
  if N3 = 0 ℓi1 ←  $\bar{x}_{i_1}$ ; ℓi2 ←  $x_{i_2}$ ; ℓi3 ←  $x_{i_3}$ ; fi;
  if N3 = 1 ℓi1 ←  $x_{i_1}$ ; ℓi2 ←  $x_{i_2}$ ; ℓi3 ←  $\bar{x}_{i_3}$ ; fi;
  if N3 = 2 ℓi1 ←  $x_{i_1}$ ; ℓi2 ←  $\bar{x}_{i_2}$ ; ℓi3 ←  $x_{i_3}$ ; fi;
  Q ← Q ∪ { (ℓi1, ℓi2, ℓi3) };
od;

```

半順序 \prec で Q をソートして、得られたリストを L_q とする。

出力: L_q

対関数を利用して、リスト L_s を記憶せずに (L_s のサイズが割合大きい)、論理式を符号化するアルゴリズムがわりあい簡単であるので、ここで省略する。

付録 A

変数 $x_0, x_1, \dots, x_{4n-1}$ を用意する。

クローズの集合 S_1, S_2, S_3, S を次のように定義する。

$$S_1 = \bigcup_{0 \leq i < j < k \leq 4n-1} \{ (x_i, x_j, \bar{x}_k), (x_i, \bar{x}_j, x_k), (\bar{x}_i, x_j, x_k) \}$$

$$S_2 = \bigcup_{0 \leq i < j < k \leq 4n-1} \{ (\bar{x}_i, x_j, \bar{x}_k), (\bar{x}_i, \bar{x}_j, x_k), (x_i, \bar{x}_j, \bar{x}_k) \}$$

$$S_3 = \bigcup_{0 \leq i < j < k \leq 4n-1} \{ (x_i, x_j, x_k), (\bar{x}_i, \bar{x}_j, \bar{x}_k) \}$$

$$S = S_1 \cup S_2 \cup S_3$$

S_1 の上に一つの半順序 \prec_1 を導入する:

定義: $C_1 = (\ell_{i_1}, \ell_{j_1}, \ell_{k_1})$ と $C_2 = (\ell_{i_2}, \ell_{j_2}, \ell_{k_2})$ を S の元とする。ここで、

l_i が x_i か \bar{x}_i を表す。 C_1 と C_2 が半順序 \langle_1 を満たす ($C_1 \langle_1 C_2$) とは次のどちらかを満たすときをいう。(1)、 $i_1 < i_2$; (2)、 $i_1 = i_2$ かつ $j_1 < j_2$; (3)、 $i_1 = i_2$ かつ $j_1 = j_2$ かつ $k_1 < k_2$; (4)、 $i_1 = i_2$ かつ $j_1 = j_2$ かつ $k_1 = k_2$ かつ 次の (a) か (b) が成り立つ: (a)、 $C_1 = (x_{i_1}, x_{j_1}, \bar{x}_{k_1})$ かつ $C_2 \in \{(x_{i_1}, \bar{x}_{j_1}, x_{k_1}), (\bar{x}_{i_1}, x_{j_1}, x_{k_1})\}$; (b)、 $C_1 = (x_{i_1}, \bar{x}_{j_1}, x_{k_1})$ かつ $C_2 = (\bar{x}_{i_1}, x_{j_1}, x_{k_1})$ 。

L_1 は半順序 \langle_1 に従って S_1 をソートして得たリストである。

S_2 の上に一つの半順序 \langle_2 を導入する:

定義: $C_1 = (l_{i_1}, l_{j_1}, l_{k_1})$ と $C_2 = (l_{i_2}, l_{j_2}, l_{k_2})$ を S の元とする。ここで、 l_i が x_i か \bar{x}_i を表す。 C_1 と C_2 が半順序 \langle_2 を満たす ($C_1 \langle_2 C_2$) とは次のどちらかを満たすときをいう。(1)、 $i_1 < i_2$; (2)、 $i_1 = i_2$ かつ $j_1 < j_2$; (3)、 $i_1 = i_2$ かつ $j_1 = j_2$ かつ $k_1 < k_2$; (4)、 $i_1 = i_2$ かつ $j_1 = j_2$ かつ $k_1 = k_2$ かつ 次の (a) か (b) が成り立つ: (a)、 $C_1 = (\bar{x}_{i_1}, x_{j_1}, \bar{x}_{k_1})$ かつ $C_2 \in \{(\bar{x}_{i_1}, \bar{x}_{j_1}, x_{k_1}), (x_{i_1}, \bar{x}_{j_1}, \bar{x}_{k_1})\}$; (b)、 $C_1 = (\bar{x}_{i_1}, \bar{x}_{j_1}, x_{k_1})$ かつ $C_2 = (x_{i_1}, \bar{x}_{j_1}, \bar{x}_{k_1})$ 。

L_2 は半順序 \langle_2 に従って S_2 をソートして得たリストである。

S の上に一つの半順序 \langle を導入する:

定義 $C_1 = (l_{i_1}, l_{j_1}, l_{k_1})$ と $C_2 = (l_{i_2}, l_{j_2}, l_{k_2})$ を S の元とする。ここで、 l_i が x_i か \bar{x}_i を表す。 C_1 と C_2 が半順序 \langle を満たす ($C_1 \langle C_2$) とは次のどちらかを満たすときをいう。(1)、 $i_1 < i_2$; (2)、 $i_1 = i_2$ かつ $j_1 < j_2$; (3)、 $i_1 = i_2$ かつ $j_1 = j_2$ かつ $k_1 < k_2$; (4)、 $i_1 = i_2$ かつ $j_1 = j_2$ かつ $k_1 = k_2$ かつ 次の (a)、(b)、(c)、(d)、(e)、(f)、(g) の中でちょうど一つが成り立つ: (a)、 $C_1 = (x_{i_1}, x_{j_1}, x_{k_1})$ かつ $C_2 \in \{(x_{i_1}, x_{j_1}, \bar{x}_{k_1}), (x_{i_1}, \bar{x}_{j_1}, x_{k_1}), (\bar{x}_{i_1}, x_{j_1}, x_{k_1}), (\bar{x}_{i_1}, x_{j_1}, \bar{x}_{k_1}), (\bar{x}_{i_1}, \bar{x}_{j_1}, x_{k_1}), (x_{i_1}, \bar{x}_{j_1}, \bar{x}_{k_1}), (\bar{x}_{i_1}, \bar{x}_{j_1}, \bar{x}_{k_1})\}$; (b)、 $C_1 = (x_{i_1}, x_{j_1}, \bar{x}_{k_1})$ かつ $C_2 \in \{(x_{i_1}, \bar{x}_{j_1}, x_{k_1}), (\bar{x}_{i_1}, x_{j_1}, x_{k_1}), (\bar{x}_{i_1}, x_{j_1}, \bar{x}_{k_1}), (\bar{x}_{i_1}, \bar{x}_{j_1}, x_{k_1}), (x_{i_1}, \bar{x}_{j_1}, \bar{x}_{k_1}), (\bar{x}_{i_1}, \bar{x}_{j_1}, \bar{x}_{k_1})\}$; (c)、 $C_1 = (x_{i_1}, \bar{x}_{j_1}, x_{k_1})$ かつ $C_2 \in \{(\bar{x}_{i_1}, x_{j_1}, x_{k_1}), (\bar{x}_{i_1}, x_{j_1}, \bar{x}_{k_1}), (\bar{x}_{i_1}, \bar{x}_{j_1}, x_{k_1}), (x_{i_1}, \bar{x}_{j_1}, \bar{x}_{k_1}), (\bar{x}_{i_1}, \bar{x}_{j_1}, \bar{x}_{k_1})\}$; (d)、 $C_1 = (\bar{x}_{i_1}, x_{j_1}, x_{k_1})$ かつ $C_2 \in \{(\bar{x}_{i_1}, x_{j_1}, \bar{x}_{k_1}), (\bar{x}_{i_1}, \bar{x}_{j_1}, x_{k_1}), (x_{i_1}, \bar{x}_{j_1}, \bar{x}_{k_1}), (\bar{x}_{i_1}, \bar{x}_{j_1}, \bar{x}_{k_1})\}$; (e)、 $C_1 = (\bar{x}_{i_1}, x_{j_1}, \bar{x}_{k_1})$ かつ $C_2 \in \{(\bar{x}_{i_1}, \bar{x}_{j_1}, x_{k_1}), (x_{i_1}, \bar{x}_{j_1}, \bar{x}_{k_1}), (\bar{x}_{i_1}, \bar{x}_{j_1}, \bar{x}_{k_1})\}$; (f)、 $C_1 = (\bar{x}_{i_1}, \bar{x}_{j_1}, x_{k_1})$ かつ $C_2 \in \{(x_{i_1}, \bar{x}_{j_1}, \bar{x}_{k_1}), (\bar{x}_{i_1}, \bar{x}_{j_1}, \bar{x}_{k_1})\}$; (g)、 $C_1 = (x_{i_1}, \bar{x}_{j_1}, \bar{x}_{k_1})$ かつ $C_2 = (\bar{x}_{i_1}, \bar{x}_{j_1}, \bar{x}_{k_1})$ 。

L_s は半順序 \langle に従って S をソートして得たリストである。

付録 B

$(l_i, l_j, l_k) \in F_a$ とする。すると l_i, l_j, l_k の中で、少なくとも一つが肯定リテラルである (ここで、肯定リテラルとは y_i のこと)。図 2 には、変数 h_1 が x_{h_1} と名前付けられ、 h_2 が x_{h_2} と名前付けられたと仮定している。

参考文献

- [1] Davies, M.: Computability and unsolvability, McGraw-Hill, N. Y. (1958).
- [2] Kranakis, E.: Primality and Cryptography, John Wiley & Sons (1986).
- [3] Watanabe, O.: On One-way Functions (1988).

$\text{block } 0$	$\text{block } 1$	\dots	$\text{block } \frac{4n-2}{4}$	$\text{block } \frac{4n-3}{4}$
-------------------	-------------------	---------	--------------------------------	--------------------------------

図1. リスト L_1 の形

(ℓ_i, ℓ_j, ℓ_k)	$(\ell_i, \ell_j, h_1) \wedge (\bar{\ell}, \ell_k, \bar{h}_1) \wedge (\bar{\ell}_1, \bar{\ell}_j, h_2) \wedge (\bar{\ell}_k, t, \bar{h}_2)$	b_{h_1}	b_{h_2}
(y_i, y_j, y_k)	$(y_i, y_j, h_1) \wedge (\bar{y}, y_k, \bar{h}_1) \wedge (\bar{y}_1, \bar{y}_j, h_2) \wedge (\bar{y}_k, t, \bar{h}_2)$	0	1
(y_i, y_j, \bar{y}_k)	$(y_i, y_j, h_1) \wedge (\bar{y}, \bar{y}_k, \bar{h}_1) \wedge (\bar{y}_1, \bar{y}_j, h_2) \wedge (y_k, t, \bar{h}_2)$	0	1
(y_i, \bar{y}_j, y_k)	$(y_i, \bar{y}_j, h_1) \wedge (\bar{y}, y_k, \bar{h}_1) \wedge (\bar{y}_1, y_j, h_2) \wedge (\bar{y}_k, t, \bar{h}_2)$	1	1
$(y_i, \bar{y}_j, \bar{y}_k)$	$(y_i, \bar{y}_j, h_1) \wedge (\bar{y}, \bar{y}_k, \bar{h}_1) \wedge (\bar{y}_1, y_j, h_2) \wedge (y_k, t, \bar{h}_2)$	0	1
(\bar{y}_i, y_j, y_k)	$(\bar{y}_i, y_j, h_1) \wedge (\bar{y}, y_k, \bar{h}_1) \wedge (y_1, \bar{y}_j, h_2) \wedge (\bar{y}_k, t, \bar{h}_2)$	1	1
$(\bar{y}_i, y_j, \bar{y}_k)$	$(\bar{y}_i, y_j, h_1) \wedge (\bar{y}, \bar{y}_k, \bar{h}_1) \wedge (y_1, \bar{y}_j, h_2) \wedge (y_k, t, \bar{h}_2)$	0	1
$(\bar{y}_i, \bar{y}_j, y_k)$	$(\bar{y}_i, \bar{y}_j, h_1) \wedge (\bar{y}, y_k, \bar{h}_1) \wedge (y_1, y_j, h_2) \wedge (\bar{y}_k, t, \bar{h}_2)$	1	0

図2. $b_{2i+2}, b_{2i+4}, \dots, b_{2n-2}$ の値