1　計算の複雑さと ＮＰ-完全問題

1.1　問題のクラス ＮＰ

「問題」の意味　ここの問題という言葉の意味をはじめに説明する。ここでグラフの平面性判定の問題を考えると、これは「ある特定のひとつのグラフ (例えば、$K_{3;3}$) についてそれが平面グラフであるか」を判定する問題と、「グラフが任意に与えられたとき、そのグラフについて平面グラフであるか」を判定する問題のふた通りの意味にとれる。この二つは意味は異なるのではっきり区別する必要がある。以下では、後者の意味で「問題」という言葉を使う。このとき、上で具体的なひとつのグラフについて平面性を判定する問題を、問題の例問 (instance) と呼ぶ。つまり、「問題」とは、例問の（可付番無限個の）集まりである。

以下では、問題を Yes, No の答えを求める判定問題の形で扱う。見かけ上判定問題ではないような最大化あるいは最小化問題も判定問題に帰着できる。例えば、「与えられたグラフに含まれる最大マッチングの大きさを求めよ。」という問題も、「与えられたグラフに大きさ k 以上のマッチングが存在するか。」という判定問題が解ければ、２分探索でそれを繰り返し利用して最大値問題が解ける。ゆえにこれから取り扱う問題をすべて判定問題と限っても一般性は失われない。

形式を整えて言うと、問題の例問を記号集合 § 上で符号化した語の全体を ¦ とする。そして、判定問題の答えが Yes になるような例問の符号化の全体を $¦_Y$ µ $¦_D$ とする。ひとつの判定問題 ¦ とは、任意に与えられた w 2 $¦_D$ に対して「 w 2 $¦_Y$ となるか」を判定する問題になる。ここで任意の語 w 2 §¤ に対して w 2 $¦_D$ であるかどうかは、一般に容易に確かめられるので、任意の語 w 2 §¤ を入力として「w 2 $¦_Y$ となるか」を判定する問題と考えて同じである。この点は以後特に厳密に区別しない。

多項式時間問題とクラス P

Def. 1 ある問題 ¦ について、それを解くチューリング機械 (計算機のアルゴリズム) A があったとする。このときある多項式 p があって、任意の入力 w に対して、p(jwj) 以内のステップで常に A が停止するとき、この問題は多項式時間の手間で解けるといい、アルゴリズム A は多項式時間のアルゴリズムであるという。ここで jwj は語 w の長さを表わす。多項式時間の手間で解ける問題の全体のなすクラスを P と書く。

我々は、多項式時間のアルゴリスムを、実際的で、効率的であるという意味で良いアルゴリズムであるとみなす。「良い (nice)」とか「効率的 (e±cient) である」という言葉自体は、数学の用語ではないので、その概念ををを多項式時間のアルゴリズムという概念と同一視して正しいかどうかは証明できないが、このように考えることがこの分野では一致した理解になっている。指数時間以上の計算時間がかかるアルゴリスムは、計算時間が爆発するので、問題のサイズがすこし大きくなっただけで実用的な時間以内で解くことができなくなる。

例 1 入力サイズ n に対して計算時間の増加が、関数のオーダーによってどのように違うか、モデル化して比べてみる。

同じ問題に対して計算時間の異なるアルゴリズムがあったとしてそれらの計算時間を表にして比較する。ここで f (n) をサイズ n の問題で各アルゴリズムがチェックする場合の数、そして１つの場合は 10^{-6} 秒でチェックできるとする。
計算時間の表

サイズ n	f(n) = $10^6 n^2$	n^6	e^n	n!
10	100 秒	1 秒	6:11 時間	3:63 秒
20	400 秒	1:06 分	923 年	4060 年
40	26:7 分	1:14 時間	4478 億年	2:59 £ 10^{26} 億年
100	2:78 時間	277:8 時間	5:11 £ 10^{23} 億年	2:99 £ 10^{134} 億年

指数関数以上のオーダーの関数は、計算時間がすぐに爆発してしまい、問題の可能解の総調べ的方法は問題のサイズが少しでも大きくなるとすぐに破綻すると分かる。また、頭に大きな定数がかかっていても、問題サイズが大きくなれば関数のオーダーが高いもののほうがすぐに大きくなる。

クラス NP　クラス NP とは、非決定性チューリング機械で受理される言語或いは解かれる問題、の全体である。もう少し平易な形に定義しなおすと、

Def. 2 問題 ¦ µ §¤ がクラス NP に属するとは、ある多項式時間で解ける問題 ¦0 と多項式 p(x) が存在して、任意の入力 w 2 §¤ に対して

$$w \in ¦_Y \quad (\ ) \quad \text{ある語 v があって } (w;v) \in ¦^0_Y \text{ かつ } |v| \leq p(|w|)$$

が成立することである。

ここで語 v は、語 w が ¦ に属するためのいわば証拠になっている。w 2 ¦Y のときは、うまい証拠 v を推測できれば、(w; v) 2 ¦0 が多項式時間で判定できるので、問題の答えが多項式時間で Yes と分かる。逆に、問題の答えが No であるときは、このままではすべての v の取り方について調べなければならないことになり、それが多項式時間で終わることは保証されない。

NP に属する問題の具体例をあげる。

(1) グラフのハミルトニアン閉路問題

Instance: 無向グラフ G = (V; E)　V = $\{v_1; v_2; \ldots; v_n\}$

Question: グラフの各点をちょうど一度ずつ通って始点に戻る閉路が存在するか判定する。

このとき、点集合 V のひとつの順列 $(v_{i_1}; v_{i_2}; \ldots; v_{i_n})$ をハミルトニアン路の候補として提示する。すると、これが G のハミルトニアン閉路であるかどうかを判定するのは容易で、特に入力の長さの多項式時間以内で判定できる。上の記法にあわせると、最初の問題は

$$¦_1 = \{G \ : \ G \text{ にはハミルトニアン閉路が存在する}\}$$

これに対して

$$¦^0_1 = \{(G;¾) \ : \ G \text{ の点の順列}¾\text{はハミルトニアン閉路をなす}\}$$

とこれが前述の定義を満たして、ハミルトニアン閉路問題がクラス NP に属すると分かる。

このように、解の候補のひとつをうまく推測して提示することができかつそれが求める解になっていることが多項式時間で判定できるような問題は、すぐにクラス NP に属するとわかる。このとき、問題の答えが否定的であるときは、このやり方では全部の可能な答えを尽くさない限り、何も分からない。

(2) 集合分割 問題

Instance: 有限集合 E と E の各元 a 2 E のサイズ s(a) 2 Z^+ が与えられたとする。
Question: ある部分集合 A µ E が存在して

$$\sum_{a \in A} s(a) = \sum_{b \in E \setminus A} s(b)$$

となるか?

この問題を表現する言語は

$$L_2 = \{(E; s) : s \text{ に対する上の等式を満たす } E \text{ の部分集合 } A \text{ がある}\}$$

これに対して

$$L_2' = \{(E; s; A) : E \text{ の部分集合 } A \text{ は上の等式を満たす}\}$$

とすれば、与えられた (E; s; A) が L_2' に含まれるか否かはすぐに判定できる。つまり単に足し算して、比べればよい。また、ひとつの部分集合を解の候補として提示するのは、簡単である。ゆえに、L は NP に属する。

(3) 完全マッチング問題

Instance: 無向単純グラフ G = (V; E)
Question: 辺の部分集合 M µ E で、どの二つの辺も端点を共有せずかつ G のどの点も M の中のひとつの辺の端点になっているようなものは、存在するか判定する。

もとの問題は

$$L_3 = \{G : G \text{ は無向単純グラフで、完全マッチングを持つ}\}$$

である。次の L_3' が、上述の NP の定義を満たす、つまり問題

$$L_3' = \{(G; M) : G \text{ の辺集合 } M \text{ は } G \text{ の完全マッチングである}\}$$

が P に属することは明らかなので L_3 はクラス NP に属する。

P µ NP は明らかである。(実際、上の定義で証拠を提示する部分がなくなったものと考えればよいからである。) これが強い不等号なのか、等号が成立するのかはまだ知られていない。(P = NP 問題)

クラス NP, co{NP と問題の良い特徴付け

Def. 3 その補集合 $\bar{L} = \Sigma^* \setminus L$ が NP に属するような問題 L の全体のなすクラスを co-NP と書く。

簡単に言えば、問題の答えが No であるときには、うまい証拠を提示してやれば、その証拠から多項式時間で答えが No であることを引き出せるということである。上の例のうち完全マッチング問題は co-NP に属する。実際、

$$L_3^* = \{(G; A) : A \text{ µ } V(G); \text{ かつ } G \setminus A \text{ が } |A| \text{ 個より多い奇数連結成分を持つ}\}$$

が P に属するからである。これは Tutte のよく知られた完全マッチング定理から導かれる。

定理 1 (Tutte の完全マッチング定理) グラフ G = (V;E) に完全マッチングが存在するための必要十分条件は、任意の点部分集合 A µ V に対して、誘導部分グラフ G ¡ A の中の奇数サイズの連結成分の数が jAj 個以下であることである。

P µ co-NP だから P µ (NP \ co-NP) である。NP \ co-NP に問題が属するということは、問題の答えが Yes の場合、No の場合ともに答えを簡単に導ける証拠が存在するということである。たとえば、完全マッチング問題では Tutte の完全マッチング定理により、その証拠が実際に示せる。このようなとき、この Tutte の定理は問題の良い特徴付け (good characterization) を与えると言う。これまでに知られている良い特徴付けを持つ問題のほとんどは、多項式時間のアルゴリズムを持つことが分かっている。

良い特徴付けの定理が成立する $ 良いアルゴリズムが存在する

そうでないものの例としては、有名な素数判定問題がある。「与えられた整数が素数であるか？」という問題が、co-NP に属することはすぐに分かる。実際、約数の候補を挙げてわり算を１回すればよい。反対にこれが NP に属することは、Fermat の定理を使うことで示される。しかしながら、素数判定の多項式時間アルゴリズムは未だ知られていない。

(注) Fermat の定理。p が素数で (a;p) = 1 ならば $a^{p-1} \equiv 1 \pmod{p}$。

1.2 NP{完全性

多項式還元 問題 ¦ 2 NP が NP{完全であるとは、NP に属するすべての問題を多項式時間の手間で ¦ に還元できることを言う。これから、NP{完全なある問題が多項式時間で解けると分かると、クラス NP に属するすべての問題が多項式時間で解けると分かる。この意味で NP{完全問題は、クラス NP の中でもっともむずかしい問題のことであるとも言える。

Def. 4 問題の多項式還元の定義を示す。アルゴリズム A が、多項式時間アルゴリズムであるとは、ある多項式 p があって任意の入力列 w に対して対応する決定性チューリング機械が高々 p(jwj) ステップ以内で停止することである。問題 ¦ から $¦^0$ への多項式還元とは、¦ から $¦^0$ への写像 f : ¦ ! $¦^0$ で以下を満たすもの。

(1) f は多項式アルゴリズムで計算できる。

(2) 任意の I 2 ¦ で I 2 $¦_Y$ ( ) f(I) 2 $¦^0_Y$

上の多項式還元が存在するとき、¦ @ $¦^0$ と書くことにする。

Def. 5 ¦ 2 NP に対して、クラス NP 中の任意の問題 ª から ¦ へ多項式還元が存在するとき、つまり ª @ ¦ であるとき、¦ は NP-完全であるという。

補題 1 $¦_1$ @ $¦_2$ でかつ $¦_2$ 2 P ならば $¦_1$ 2 P.

(Proof) 明らか。 ¤

補題 2 $¦_1$ @ $¦_2$, $¦_2$ @ $¦_3$ ならば $¦_1$ @ $¦_3$

(Proof) 明らか。 ¤

補題 3 ¦1; ¦2 2 NP で ¦1 が NP-完全であるとき、¦1 @ ¦2 ならば ¦2 も NP-完全である。

クラス NP 中の任意の問題 ª に対して ª @ ¦1 @ ¦2 より明らか。 ¤

これから、次の手順にもとづけば与えられた問題 ¦ が NP-完全であることを証明できる。

(1) ¦ がクラス NP に属することを示す。

(2) すでに NP-完全であることが分かっている問題 ¦cmp をひとつ選ぶ。

(3) ¦cmp の例問に対して ¦ の例問を多項式時間で構成する写像 f を作る。

(4) 任意の P 2 ¦cmp で P の判定問題 ¦ での答えが Yes になることと 、f(P) の答えが Yes になることが必要十分になることを示す。

ある NP{完全問題に多項式時間アルゴリズムが見つかるとすべての NP の問題が多項式時間で解けることになり、P=NP=co{NP となる。が、そういうことは起こりそうにもないと思われている。それゆえ、ある問題が NP-完全もしくはそれ以上に手間のかかる問題であることが分かれば、多項式で手間が押さえられるという意味で効率の良いアルゴリズムを見つけだそうとするのは、とりあえずあきらめるのが普通である。ただ、正確には現在のところ、 P = NP か P 6= NP のどちらが正しいのかは知られていない。 (多項式時間で解けない問題が NP の中にひとつでも存在することを示せば、P 6= NP と分かるが、それが知られていない。)

Def. 6 また、少なくとも NP-完全以上の難しさを持つ問題を NP-困難 (NP-hard) と呼ぶ。正確には、P = NP でない限り、多項式時間のアルゴリズムが存在しないような問題で、NP に属するか否か問わないとき、NP-困難な問題であるという。

充足可能性問題 (satis¯ability problem) ひとつ問題が NP-完全であることが示されれば、それをタネに多項式還元の手順を繰り返して多数の問題が芋づる式に NP-完全であることを示すことができる。この最初のひとつとして NP-完全であることが証明された問題が Boole 式の充足可能性問題である。[Cook]

充足可能性問題を説明する。U = fx1; x2; : : : ; xng を変数とするブール式は、次のように帰納的に定義できる。

(1) xi (i = 1; : : : ; n) はブール式である。

(2) s と t がブール式であれば、(s ∧ t); (s _ t); : s はブール式である。

ブール式の例を挙げれば、

$$(x_1 \wedge x_4); \qquad (:(x_2 \wedge x_3) \wedge (:x_2 \_ :x_3) \qquad :(((x_1 \_ x_3) \wedge :x_4) \wedge (x_1 \_ x_2))$$

各変数 xi は 1 (=True) または 0 (=False) の値をとる。与えられたブール式において、その各変数に 0, 1 の値を割り当てる写像 t : U ! f0; 1g を truth assigment と呼ぶ。ひとつの truth assignment のもとでの

ブール式の値は、次から定まる。

$$0 \wedge 0 = 0;\quad 0 \wedge 1 = 1 \wedge 0 = 0;\quad 1 \wedge 1 = 1$$
$$0 \vee 0 = 0;\quad 0 \vee 1 = 1 \vee 0 = 1;\quad 1 \vee 1 = 1$$
$$\neg 0 = 1;\quad \neg 1 = 0$$

充足可能性問題とは、与えられたブール式に対してその値を 1 (True) にする truth assignment が存在するか、判定する問題である。ここでは、ブール式は常に論理積標準形で与えられるとして、充足可能性問題を扱うことにする。変数の集合が $U = \{x_1, x_2, \ldots, x_n\}$ であるとき、$\mathcal{U} = \{x_1, \neg x_1, x_2, \neg x_2, \ldots, x_n, \neg x_n\}$ の要素を項と呼ぶ。またその論理和 $a_1 \vee \cdots \vee a_k$ $(a_j \in \mathcal{U})$ を節と呼ぶことにする。各 $a_{i,j}$ が項であるとき、次のような節の論理積の形のブール式 f を論理積標準形という。

$$f = (a_{1,1} \vee a_{1,2} \vee \cdots \vee a_{1,m_1}) \wedge (a_{2,1} \vee a_{2,2} \vee \cdots \vee a_{2,m_2}) \wedge \cdots \wedge (a_{n,1} \vee a_{n,2} \vee \cdots \vee a_{n,m_n}) \qquad (1)$$

これ以降、論理積標準形のブール式 f は、各節を項の集合、式の全体を節の集合とみなせば、集合族として $f = \{\{a_{1,1}, \ldots, a_{1,m_1}\}, \ldots, \{a_{n,1}, \ldots, a_{n,m_n}\}\}$ のように表現したとして扱う。

まとめると、充足可能性問題 (SAT) は、

(1) Instance : 変数の集合 $U = \{x_1, x_2, \ldots, x_n\}$ と論理積標準形のブール式 f

(2) Question : f の値を 1 (True) にする truth assignment $t : U \to \{0,1\}$ が存在するか。

定理 2 ブール式の充足可能性問題は、NP-完全である。

基本的な NP-完全問題　基本的な NP-完全問題として、次のようなものがある。

1. 3-充足可能性問題 (3-Satisfiability, 3SAT)

   a) Instance: すべての節のサイズが3 であるとなっているブール式 $f = \{\{a_{1,1}, a_{1,2}, a_{1,3}\}, \ldots, \{a_{n,1}, a_{n,2}, a_{n,3}\}\}$

   b) Question: f の値を 1 にする truth assignment が存在するか？

   c) NP-完全性

      充足可能性問題 (SAT) が 3-SAT に多項式還元出来ることを示して、その NP-完全性を示そう。

      SAT の Instance f が任意に与えられたとする。$U = \{x_1, \ldots, x_n\}$ をその変数の集合、$W = \{x_1, \neg x_1, \ldots, x_n, \neg x_n\}$ を項の集合とする。

      $$f = (c_1) \wedge (c_2) \wedge \cdots \wedge (c_m); \qquad c_i = z_{i,1} \vee z_{i,2} \vee \cdots \vee z_{i,k_i}$$

      f の各節 $c_i = z_1 \vee \cdots \vee z_k$ を次に置き換える。

      i. $k = 1$ の場合。$D_i = (z_1 \vee y_1^i \vee y_2^i) \wedge (z_1 \vee y_1^i \vee \neg y_2^i) \wedge (z_1 \vee \neg y_1^i \vee y_2^i) \wedge (z_1 \vee \neg y_1^i \vee \neg y_2^i)$,

      ii. $k = 2$ の場合。$D_i = (z_1 \vee z_2 \vee y_1^i) \wedge (z_1 \vee z_2 \vee \neg y_1^i)$

      iii. $k \geq 3$ の場合。$D_i = (z_1 \vee z_2 \vee y_1^i) \wedge \left(\bigvee_{j=1}^{k_i - 4} z_{j+2} \vee \neg y_j^i \vee y_{j+1}^i\right) \wedge (\neg y_{k_i - 3}^i \vee z_{k_i - 1} \vee z_k)$

ここで各 y_j^i は新しく導入された変数である。これらから定義されるブール式

$$f' = D_1 \wedge D_2 \wedge \cdots \wedge D_k$$

は、節の中の項の数がすべて３で、3-SAT の instance になっている。このとき、$f = 1$ にする真偽値の割り当て t が存在すれば、$f' = 1$ をみたす真偽値の割り当て t' が存在することと、その逆も成立することを示せばよい。(以下略。)
ここで、f' が多項式時間の手間で計算できることもほぼ明らか。ゆえに、3-SAT は NP-完全と分かった。

2. 3-次元マッチング (3-Dimensional Matching, 3DM)

   a) Instance: 互いに素な集合 X, Y, Z で $|X| = |Y| = |Z| = k$ であるものと、3 次元辺集合 $A \subseteq X \times Y \times Z$

   b) Question: 辺集合の部分集合 $M \subseteq A$ で $|M| = k$ となりかつ、M 中のどの 2 元 $e = (e_1, e_2, e_3) \in M$, $f = (f_1, f_2, f_3) \in M$ を取っても各座標の中で重ならない、つまり $e_i \neq f_i$ $(i = 1, 2, 3)$ となるものが存在するか？

   c) NP-完全性の証明。 3-次元マッチング問題は、3-SAT を用いて NP-完全であることが示せる。

3. グラフの点被覆 (Vertex Cover)

   a) Instance: グラフ $G = (V, E)$ と正の整数 $K \leq |V|$

   b) Question: サイズが K 以下の点集合による辺の被覆が存在するか？

   c) NP-完全性の証明

   グラフの点被覆問題は、3-SAT の問題を点被覆問題に多項式還元できることを示すことで、NP-完全であることが次のように示せる。
   3-SAT で変数の集合を $U = \{x_1, x_2, \ldots, x_n\}$、与えられたブール式を $f = (c_1) \wedge (c_2) \wedge \cdots \wedge (c_m)$ とする。これから、グラフ G と正の整数 K を次のように定める。
   (a) 各変数 x_i に対して、点集合 $V_i = \{a_i, \bar{a}_i\}$ を考え、辺 $\{a_i, \bar{a}_i\}$ を挿入する。
   (b) 各節 $c_j = u_j \vee v_j \vee w_j$ に対して、点集合 $V_j' = \{u_j, v_j, w_j\}$、辺集合 $E_j' = \{(u_j, v_j), (v_j, w_j), (u_j, w_j)\}$ を与える。
   (c) (a) の単独辺と (b) の三角形をつなぐ辺を次のように加える。各節 $c_j = u_j \vee v_j \vee w_j$ の中の各項 u_j, v_j, w_j で、$u_i = x_j$ のとき辺 (u_i, a_j) を、$u_i = \neg x_j$ のとき辺 $(u_i, \bar{a}_j)$ を加える。同様に他の v_j, w_j からも対応する項を表わす単独辺中の点への辺を加える。
   このようにしてできた点集合 $V = (\bigcup_{i=1}^{n} V_i) \cup (\bigcup_{j=1}^{m} V_j')$ 上のグラフを G 、上限値を $K = n + 2m$ としたときの点被覆問題を考える。
   まず、$C \subseteq V$ が条件 $|C| \leq K = n + 2m$ を満たす G の点被覆であるとする。これは、各 i で単独辺の a_i または $\bar{a}_i$ の点を含まなければならない。同様に、V_j' 上の三角形のの中の少なくともふたつの点を含まなければならない。サイズの条件から C は各単独辺

の点のうちちょうどひとつの点を、各三角形の中のちょうどふたつの点を含むと分かる。この被覆 C から f の truth assignment $t : U \to \{0,1\}$ を以下で定める。

$$t(x_i) = \begin{cases} 1 & a_i \in C \text{ のとき} \\ 0 & \bar{a}_i \in C \text{ のとき} \end{cases}$$

これはもとの f の値を 1 にする truth assignment になっている。逆に、f を充足する truth assignment があれば、そこからサイズ $n + 2m$ の G の点被覆が作れる。(以下略)

4. グラフの Vertex Cover 問題、最大クリーク問題、最大安定集合問題は、互いに同値である。実際、グラフ $G = (V, E)$ と $X \subseteq V$ に対して、

   X が G の Vertex cover $\Leftrightarrow$ $V - X$ が G の安定集合 $\Leftrightarrow$ $V - X$ が $\bar{G}$ のクリーク

   特に、 G がサイズ K 以下の Vertex Cover を持つことと、$\bar{G}$ がサイズ $|V| - K$ 以上のクリークを持つことは自明に同値になる。

   グラフの最大クリーク問題 (Max Clique) (= 最大安定集合問題)

   a) Instance: グラフ $G = (V, E)$, 正の整数 $K \leq |V|$.
   b) Question: G にサイズが K 以上のクリークが存在するか。クリークとは点の部分集合 $W \subseteq V$ でその中のどの 2 点も辺で結ばれているもののこと。
   c) NP-完全性の証明。点被覆問題と最大クリーク問題は同値だから。

5. ハミルトニアン閉路問題 (Hamiltoninan Circuit)

   a) Instance: グラフ $G = (V, E)$
   b) Question: G はハミルトニアン閉路を持つか。つまり、頂点全部の順列 $(u_1, u_2, \ldots, u_n)$ で $(u_i, u_{i+1}) \in E$; $(v_{n+1} = v_1)$ となるものがあるか。
   c) NP-完全性の証明。点被覆問題を使う。 (煩雑なので省略。)

6. 集合分割問題 (Partition)

   a) Instance: 有限集合 S とその各元 $x \in S$ の重み $w(x) \in \mathbb{Z}^+$
   b) Question: $\sum_{x \in A} w(x) = \sum_{x \in S \setminus A} w(x)$ となる部分集合 $A \subseteq S$ が存在するか。
   c) NP-完全性の証明。3DM を使う。

ここまでの NP{完全問題の証明の導出関係をまとめてみると、

SAT	$\Rightarrow$	3SAT	$\Rightarrow$	3DM	$\Rightarrow$	集合分割問題
		#				#
Hamiltonian 閉路	$\Leftarrow$	Vertex Cover	$\Rightarrow$	Dominating Set		ナップザック問題
		#				
部分グラフ同型問題	$\Leftarrow$	最大クリーク	$=$	最大安定集合	$\Rightarrow$	集合パッキング

多項式還元で NP-完全性を示す方法をおおざっぱに3つに分類してみる。

A. 部分問題。ある問題 $\Pi \in NP$ が NP-完全であることを示すのに、すでに NP-完全が既知である問題 Π' のすべての Instance がが Π の中に含まれることを示せばよい。

1. 最小 Blocker
   a) INSTANCE: 集合 S の部分集合族 $\{A_i : i = 1, \ldots, n\}$ と正の整数 L
   b) QUESTION: 部分集合 $B \subseteq S$ で任意の i で $B \cap A_i \neq \emptyset$; かつ $|B| \leq L$
   c) PROOF: グラフの点被覆問題を使う。グラフの辺を2点集合とみなした集合族での最小 Blocker 問題は、もとの点被覆問題に同じになっている。
2. 部分グラフ同型問題
   a) INSTANCE: 2つのグラフ $G = (V_1, E_1)$; $H = (V_2, E_2)$
   b) QUESTION: G は H に同型な部分グラフを含むか。
   c) PROOF: 部分グラフ H として完全グラフをとれば、最大クリーク問題になっている。 最大クリーク問題が NP-完全と分かっていてそれを部分として含んでいるから。
3. ナップザック問題
   a) INSTANCE: 'もの' の集合 S と各元 $x \in S$ の重さ $w(x) \in \mathbb{Z}^+$、価値 $v(x) \in \mathbb{Z}^+$ と重さの上限 $B \in \mathbb{Z}^+$ と利得値の下限 $L \in \mathbb{Z}^+$
   b) QUESTUON: ものの部分集合 A で、重さの和が $\sum_{x \in A} w(x) \leq B$ で、利得の和が $\sum_{x \in A} v(x)$, L となるものがあるか。
   c) PROOF: 集合分割問題を含んでいる。実際、$w(x) = v(x) = s(x)$ $(x \in S)$ かつ $B = L = 1/2 \sum_{x \in S} s(x)$ とおいたときのナップザック問題はもとの集合分割問題に等しい。
4. 二乗の最小和
   a) INSTANCE: 有限集合 A、各元 $a \in A$ のサイズ $s(a) \in \mathbb{Z}^+$ と正の整数 $K; D \in \mathbb{Z}^+$
   b) QUESTION: A の D 個の部分集合の分割 $A_1, A_2, \ldots, A_D$ で $\sum_{i=1}^{D} (\sum_{x \in A_i} s(x))^2 \leq K$ を満たすものがあるか。
   c) PROOF: 集合 A とサイズ $s(x)$ $(x \in A)$ の分割問題は、$D = 2$; $K = 1/2 (\sum_{x \in A} s(x))^2$ とした場合の二乗最小和問題に等しい。

B. Local Replacement 前の場合ほど自明ではないが、部分的な構造の置き換えで NP-完全な問題を、与えられた問題の特別な場合に帰着できる場合。

1. Feedback Vertex Set
   a) INSTANCE: 有向グラフ $G = (V, E)$, 正整数 $K \leq |V|$
   b) QUESTION: 点部分集合 $V' \subseteq V$ で $|V'| \leq K$ かつ G のすべての有向閉路が V' の点を少なくともひとつ含むようなものがあるか。
   c) PROOF: 点被覆問題を Feedback Vertex Set に帰着させる。

2. グラフの支配集合 (Dominating Set) 問題
   a) INSTANCE: グラフ $G = (V; E)$, 正整数 $K \leq |V|$
   b) QUESTION: サイズ K 以下の点集合 V' で、その補集合 $V - V'$ の任意の点が V' 中の少なくともひとつの点に隣接しているようなものがあるか。
   c) PROOF: 点被覆問題を Dominating Set に帰着させる。
3. 集合パッキング問題
   a) INSTANCE: 有限集合の集まり F, 正整数 $K \leq |F|$
   b) QUESTION: F の中から選んだ K 個の集合の組で、どの2つも互いに共通部分が空になるようなものが存在するか。
   c) PROOF: グラフの安定集合問題を含む。(グラフの点 v に接続する枝の全体の集合を X_v とすれば、このグラフの安定集合はパッキングに同値になるから。)

C. Component Design.

1.3 計算の複雑さ：時間計算量、領域計算量

上では、チューリング機械もしくはその他の万能計算機で問題を解く、あるいは同じことだが対応する言語を認識するのに必要となる計算時間が、入力のサイズの多項式以下で押さえられるかどうかを扱った。このように、計算にかかる時間、計算に要するメモリ領域の大きさ、その他必要な計算資源の量をもとにして問題を分類する分野を、一般に計算の複雑さの理論という。NP-完全の理論では、もっぱら計算時間量を問題にしたが、計算に必要なメモリ量である領域計算量も考えられる。

チューリング機械 M で長さ n の任意の入力列に対し、高々$T(n)$ 回の動作の後停止するとき、M の時間計算量は高々 $T(n)$ であるという。この M で実現されているアルゴリズムも、時間計算量が高々 $T(n)$ であるといい、$O(T(n))$ と書く。例えば、 n 点上の二部グラフの最大マッチングを求める通常よく知られたアルゴリズムは、$O(n^2)$ の手間がかかる。非決定性チューリング機械の動作の列をどのように選択したとしても停止するまでの時間が $T(n)$ 回を越えないとき、この非決定性チューリング機械は時間計算量が高々 $T(n)$ であると言う。

決定性チューリング機械において時間計算量が高々$T(n)$ である言語（問題）の全体のなす族を $DTIME(T(n))$、非決定性チューリング機械での時間計算量が高々 $T(n)$ である言語全体のなす族を $NTIME((T(n))$ と書く。これを使うと、クラス P と NP は

$$P = \bigcup_{k \geq 1} DTIME(n^k); \qquad NP = \bigcup_{k \geq 1} NTIME(n^k) \tag{2}$$

同様に、決定性チューリング機械で領域計算量が高々 $T(n)$ である言語の全体のなす族を $DSPACE(T(n))$、非決定性チューリング機械での領域計算量が高々 $T(n)$ である言語 (問題) 全体のなす族を $NSPACE((T(n))$ と書く。領域計算量が入力サイズの多項式以下で押さえられる問題のクラスを

$$PSPACE = \bigcup_{k \geq 1} DSPACE(n^k); \quad NSPACE = \bigcup_{k \geq 1} NSPACE(n^k) \tag{3}$$

定理 3 これらの計算量の間に次の自明な関係が成立している。

(1) 言語 L が DTIME(f(n)) に属するならば DSPACE(f(n)) にも属する。

(2) L が DSPACE(f(n)) に属しかつ f(n) 6 $\log_2 n$ であるならば、L によって定まる定数 c があって L は DTIME($c^{f(n)}$) に属する。

(3) L が NTIMT(f(n)) に属するならば L によって定まる定数 c があって、L は DTIME($c^{f(n)}$) に属する。

演習 1 上を示せ。

また、明らかに

$$DSPACE(\log n) \mu P \mu NP \mu PSPACE = NSPACE$$

が成り立つ。この包含関係が真の包含関係になるかならないかは知られていない。ただし、DSPACE(log n) ½ PSPACE が真の包含関係として成立することは知られている。Savitch の定理より PSPACE = NSPACE が成り立つ。

1.4 Worst Case Analysis and Mean Time Analysis

これまで問題にしてきた問題の計算時間は、問題のサイズが n の場合のすべてで計算時間が f(n) で押さえられるような関数によって表してきた。これは言いかえれば、計算時間の意味で最悪の (最長の) 問題例を基準にして考えているとも言える。これを Worst Case Analysis という。しかし、現実の場合には、ごく少数の特別な場合を除きたいてい短い計算時間ですむようなたぐいのアルゴリズムもある。理論的には、とても長い時間がかかる可能性があることが示されていても、実世界に現れる問題ではうまく効率的に解けることが経験的に知られているような場合である。線形計画問題を解くための単体法はそのような例のひとつである。このような場合は、問題全体での計算時間の平均値をアルゴリズムの計算時間を表わす尺度としたほうが良いように思える。このような評価法を Mean Time Analysis という。ここで平均値を計算するためには、問題全体という集合の上に適当な測度を導入する必要があるが、妥当と思われるものを定義するのがむずかしい。

1.5 補遺：様々な NP-完全問題

I Graph Theory, Network

(1) 点による辺の被覆

Instance: グラフ $G = (V; E)$, K $(0 < K \leq |V|)$ 正の整数。

Question: サイズ K 以下の点集合で、すべての辺がその中の少なくともひとつの点に接しているようなものがあるか。

(2) 支配集合問題 (Dominating Set)

Instance: グラフ $G = (V; E)$, 正の整数 $K \leq |V|$

Question: サイズ K 以下の点集合 W で、すべての点 $v \in V - W$ が W の中の少なくとも 1 点と隣接するものがあるか。

Note: Vertex Cover から証明できる。

(3) グラフの彩色数

Instance: グラフ $G = (V; E)$, K $(0 < K \leq |V|)$ 正の整数。

Question: K 色以下の色で点の彩色ができるか。Note: 3SAT を使う。

(4) Feedback Vertex Set

Instance: 有向グラフ $G = (V; A)$, 正の整数 $K \geq |V|$

Suestion: サイズ K 以下の点部分集合 V' で、G の任意の有向閉路がその中の少なくとも 1 点含むようなものがあるか。

Note: 無向グラフであれば、補木が答えになる。

(5) Feedback Arc Set

Instance: 有向グラフ $G = (V; A)$, 正の整数 $K \geq |V|$

Qustion: G の任意の有向閉路に対し、その中の少なくとも 1 辺を含むようなサイズが K 以下の辺部分集合 $A' \subseteq A$ が存在するか。

(6) 三角形への分割

Instance: グラフ $G = (V; E)$, $|V| = 3a$

Question: V のサイズが 3 の部分集合への分割 $V_1; \cdots; V_a$ で各 V_i の部分グラフが三角形になっているようなものがあるか。

(7) クリークへの分割

INSTANCE: グラフ $G = (V; E)$, 正整数 $K \leq |V|$

QUESTION: V の分割 $V_1; \cdots; V_m$ で $m \leq K$ となり、各 V_i がクリークになるものがあるか。

(8) 最大クリーク、最大独立集合

Instance: グラフ $G = (V; E)$, 正の整数 $K \geq |V|$

Question: サイズ K 以上のクリークが存在するか。

Note: 独立集合は、補グラフのクリークに等しい。

(9) ハミルトニアン閉路、有向ハミルトニアン閉路

Question: 与えられたグラフにすべての点をちょうど一度ずつ通るような閉路があるか。

(10) ハミルトニアンパス

Question: 与えられたグラフの始点と終点を結ぶ単純パスで、他のすべての点をちょうど一度ずつ通るものを見つける。

(11) 部分グラフ同型問題

(12) Graph Contractability

(13) グラフ準同型

グラフ G で非隣接の 2点を同一視する操作を繰り返して、与えられたグラフ H に同型になるか。

(14) 核 (kernel)

INSTABNCE: 有向グラフ G = (V; A)

QUESTION: 点部分集合 W µ V で、W は安定集合でかつ任意の u 2 V ¡ W に対してある w 2 W があって (w; u) 2 A となるものがあるか。PROOF: 3SAT から。

(15) 極大木の同型 (Isomorphic Spanning Tree)

グラフ G と木 T が与えられて、G が T に同型な極大木を含むか判定する。

(16) グラフでの Steiner Tree 問題

グラフ G とその中の点の部分集合 R が与えられたとき、R の点をすべて含む G の部分木で、枝の重み和が与えられた制限値 B 以下であるものがあるか。

(17) (ふつうの幾何学的な意味での) Steiner Tree 問題

(18) 通信路ネットワークの信頼性

INSTANCE:

(19) 最長路問題

Instance: グラフ G = (V; E), 始点、終点 s; t 2 V, 各辺の長さ l(e) 2 Z_+ (e 2 E), 正の整数 K

Question: s, t を結ぶ単純パスで長さ K 以上のものがあるか。

Note: ハミルトニアン路問題に含まれる。各辺の長さがすべて 1 のときでも NP {完全になる。

(20) 最大カット

Instance: グラフ G = (V; E) と辺の重み w : E ! Z_+

Question: カットセットの中で辺の重み和が最大になるものを求める。

Note: 最小カット問題は、多項式時間で解ける。

II Sets and Partitions

(1) 3-次元 マッチング

Note: 2次元マッチングは多項式時間で解ける。

(2) Exact Cover by 3-Sets

INSTANCE: jEj = 3p である集合 E と、サイズが 3の部分集合の集まり F

QUESTION: 部分族 F^0 µ F で、E の分割になっているものがあるか。

(3) Set Packing

INSTANCE: 有限集合の族 F, 整数 K 6 jFj,

QUESTION: F 中に互いに素な K 個の集合が存在するか。

(4) Minimum Covering of Sets

INSTANCE: 有限集合 S、その部分集合の族 F、正整数 K 6 jFj

QUESTION: F の中の K 個以下の集合の組で、その合併が S をおおうものがあるか。

Proof: X3C を使う。

Comment: jAj = 2 for A 2 F としても NP-完全。

(5) 3-マトロイド INTERSECTION

(6) 集合分割問題

$\sum_{x \in A} w_x = \sum_{x \in E \setminus X} w_x$ となる部分集合 A があるか。

III Commpression and Representation

(1) Shortest Common Superstring

INSTANCE: §: 文字集合、 R µ §¤: 語の有限集合、 K: 正整数

QUESTION: 長さ K 以下の語 w で R の中の任意の列 x 2 R が w の部分列になるようなものがあるか。

PROOF: Vertex Cover を使う。

(2) Consecutive Ones Submatrix

(3) String-To-String Correction

INSTANCE: §: 文字集合, 2つの語 x;y 2 §¤, 正整数 K

Question: 文字を除去するまたは隣り合う文字を交換するという操作を K 回以下施すとして、列 x から列 y が得られるか。

IV スケジューリング

(1) Multiprocessor Scheduling

INSTANCE: processor の数 m 、仕事の集合 T、各仕事の所要時間 l(t) t 2 T0, 全体の締め切り時間 D 2 Z^+

QUESTION: m processors のスケジューリングで、締め切り D に間に合うものが存在するか。

V 数理計画法

(1) 整数計画法

Question: 線形計画法で解を整数値に限ったときの最適解を求める問題。制約式、目的関数は整数値であるとしておく。

(2) ナップザック問題

INSTANCE: item の集合 S, item t 2 S のサイズ s(t) 2 Z^+, 価値 v(t) 2 Z^+, 正整数 B; K,

QUESTION: item の部分集合 T µ S で、$\sum_{t \in T} s(t) \leq B$ かつ $\sum_{t \in T} v(t) \geq K$ となるものがあるか。

VI 整数論、代数

(1) Quadratic Congruence

INSTANCE: 正整数 a; b; c,

QUESTION: 正整数 $x < c$ で $x^2 \acute{} a \pmod b$ となるものがあるか。

(2) Quadratic Diophantine Equations

INSTANCE: 正整数 a; b; c,

QUESTION: 式 $ax^2 + by = c$ は正の整数解 x; y をもつか。

VII パズル

(1) Generalized Hex

INSTANCE: グラフ G = (V; E) とその中の 2点 s;t 2 V,

QUESTION: player 1, 2 がいて s;t 以外の点を交代に選び、player 1 は点を「青」にぬり、player 2 は点を「赤」にぬる。すべての点をぬり終わったあとで点 s から t への途中がすべて青の点ばかりからなるパスがあれば、player 1 の勝ちとする。与えられたグラフのゲームで player 1 は必ず勝てるか。

Comment: 上で点を選ぶところを辺を選ぶにおきかえると、Shannon switching game になる。こちら多項式で解ける。

(2) N £ N 盤の チェッカー

(3) N £ N 盤の碁

VIII 論理学、オートマトン、数理言語

(1) 充足可能性問題

(2) 有限オートマトンが非同値

同一の入力記号集合を持つ 2 つのオートマトンについてそれらが異なる言語を受理するかを判定する。

(3) 文脈自由言語を生成する文法が与えられて、任意の語がその言語に属するか判定する。

IX 雑多なもの

(1) 有限集合上の関数の生成

有限集合 S とその上の関数の集まり F と関数 h : A ! A が与えられたとき、 h が F の関数から合成を繰り返して得られるか。

(2) クラスタリング:

INSTANCE: 有限集合 S, S 中の 2 点間の距離 d(s;t) 2 Z^+, 正整数 B;K,

QUESTION: S を K 個以下の部分集合 $X_1; : : : ; X_k$ に分割してかつ d(x;y) 6 B x;y 2 X_i となるようにできるか。

X 未解決な問題：NP-完全とも P とも未判明なもの

(1) グラフ同型問題:

与えられた 2つのグラフが同型であるか判定する。

(2) 対象のグラフ H を固定したとき、グラフ G が H に位相同型な部分グラフを含むか。

(3) グラフの種数。
与えられたグラフを埋め込むことのできる曲面の種数の最小値を求める。

1.6 数の複雑さ | チェイティンの定義

Def. 7 実数 x の複雑さを、それをプリントアウトするチューリング機械の最短のプログラムの長さ (内部状態の数) によって定義することにする。

定理 4 (チェイティンの定理) 複雑性 n のどんなプログラムも n より大きな複雑性をもつ数を産み出すことはできない。どんなプログラムもそれ自身の複雑さより大きい複雑さの数を産み出すことは出来ない。