

8. クラスNPと多項式時間帰着

1

クラスPは決定性TMによって定義された。
この計算機モデルを非決定性に変更すると、
問題のクラスがどのように変化するかを考察する。

実は、厳密な意味では、この違いは
明らかになっていない。
(つまり、決定性TMと非決定性TMの能力
の差が、多項式倍しかないのかどうかは、
未解決である。)

2

8-1. 非決定性時間限定TM

まず、非決定性TMの計算時間を定める。

非決定的計算過程は、初期様相を根とする木として表現可能である。
この計算の木において、最も浅い受理様相の深さをNTMの計算時間と定義する。
すなわち、NTMが非決定的選択を繰り返したとき、最も速く受理状態に達するときの総ステップ数が、NTMの計算時間である。

なお、非決定性計算において、ある様相への初期様相からの遷移系列を、**計算パス**と呼ぶこともある。

NTMの計算時間とは、受理様相までの最も短い計算パスの長さともみなせる。

3

計算の木と非決定性計算時間

4

時間限定非決定性TM

入力サイズが n のとき、
 $T(n)$ 時間限定非決定性チューリングマシン
($T(n)$ -NTM)とは、
計算時間が $T(n)$ 以下であるような非決定性チューリングマシン (NTM) のことである。

5

8-2. クラスNPの定義

$NTIME(T(n)) = \{L \mid L \text{は } O(T(n)) \text{時間限定NTM}$
で判定される言語}

と定義する。

このとき、**クラスNP**を、

$$NP = \bigcup_k NTIME(n^k)$$

と定義する。

つまり、非決定性多項式時間で
計算可能な問題の集合がNPである。

6

クラスPとクラスNPの名前の由来

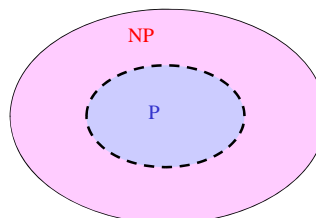
クラスP : 多項式時間TM
(Polynomial Time Turing Machine)
で解ける問題の集合

クラスNP : 非決定性多項式時間TM
(Non-deterministic Polynomial Time TM)
で解ける問題の集合

7

クラスPとクラスNPの関係

定義から明らかであるが、
NPはPを包含する。



8

8-3. クラスNPの問題

名称: ハミルトン閉路問題HC
インスタンス: グラフ $G = (V, E)$

問:
Gにハミルトン閉路が存在するか?
すなわち、Vのすべての点を通るような単純な閉路が存在するか?

また、ハミルトン閉路問題のYESのインスタンスからなる言語を

$L_{HC} = \{w \mid w \text{はハミルトン閉路をもつグラフ}\}$
と表す。

9

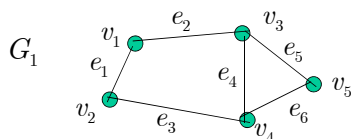
非決定性計算例

L_{HC} を判定する次のような、
非決定性多項式時間アルゴリズム存在する。
なお、
 $E = \{e_1, e_2, \dots, e_m\}$
とする。

1. 辺 e_1 から e_m まで、
ハミルトン閉路で辺を用いるかどうかを、
非決定的に定める。
2. 1. で定めた辺集合が、ハミルトン閉路になっている
かどうかをチェックする。
3. 2. において、ハミルトン閉路になっていければ
YES、なってなければNO

10

ハミルトン閉路問題のインスタンス1



$\{(v_1, v_2), (v_1, v_3), (v_2, v_4), (v_3, v_4), (v_3, v_5), (v_4, v_5)\}$

G_1 には、

$e_1 e_3 e_6 e_5 e_2$

というハミルトン閉路が存在する。

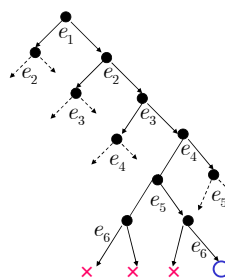
よって、

$G_1 \in L_{HC}$

11

非決定性計算

$G_1 \in L_{HC}$ であるが、その受理までの計算をみってみる。



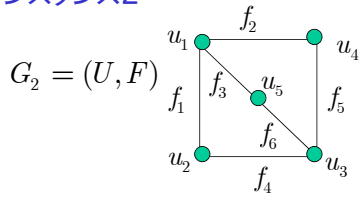
この木において、
右ならその辺を用いて、
左なら用いないとする。

この受理される場合の
計算パスの長さは
明らかに $O(m)$ である。

このように、ハミルトン閉路問題は、
明らかにNPに属する。

12

インスタンス2



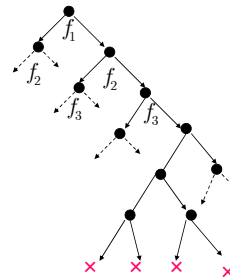
G_2 には、ハミルトン閉路が存在しない。
よって、

$$G_2 \notin L_{HC}$$

13

非決定性計算

$G_2 \notin L_{HC}$ であるので、すべての計算パスが非受理の様相に遷移する。



この受理される場合の計算パスの長さは明らかに $O(m)$ である。

このように、ハミルトン閉路問題は、明らかにNPに属する。

14

以上より、次の命題が成り立つ。

ハミルトン閉路問題はクラスNPに属する。
つまり、

$$L_{HC} \in NP$$

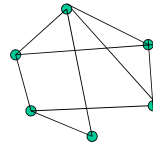
である。

15

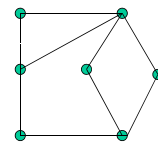
練習

次のグラフにハミルトン閉路があるかどうかを決定せよ。

(1)



(2)



16

クラスNPの問題2

名称: SAT(充足可能性問題、SATisfiability problem)
インスタンス: 和積形の論理式 $f(x_1, x_2, \dots, x_n)$

問: $f = 1$ となる x_1, \dots, x_n への0, 1の割り当てが存在するか?

また、SATのYESのインスタンスからなる言語を次のように表す。

$$L_{SAT} = \{f \mid f = 1 \text{ となる } x_1, \dots, x_n \text{ への0, 1の割り当てが存在する。}\}$$

17

充足可能なインスタンス

$$f_1(x_1, x_2, x_3, x_4) = (\overline{x_1} + x_3 + \overline{x_4})(x_2)(x_1 + x_4)(x_2 + x_3 + \overline{x_4})$$

この例では、

例えば、

$$x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 1$$

と割り当てればブール関数は充足される。

よって、

$$f_1 \in L_{SAT}$$

である。

18

充足不可能なインスタンス

$$f_2(x_1, x_2) = (x_1 + x_2)(x_1 + \overline{x_2})(\overline{x_1})$$

このインスタンスは、恒偽であり、充足不可能である。

よって、

$$f_2 \notin L_{SAT}$$

19

SATの非決定性アルゴリズム

SATはNPに属する。
すなわち、

$$L_{SAT} \in NP$$

である。

証明

SATを解く、非決定性多項式時間アルゴリズムを示せばよい。
次に、そのアルゴリズムを示す。

20

1. 変数に対して x_1 から x_n まで、0か1を非決定的に割り当てる。 $f(x_1, \dots, x_n)$
2. 1.での割り当てで、ブール関数が1かどうかをチェックする。
3. 2.において、1になっていければYES、なってなければNO

このアルゴリズムが、非決定性多項式時間であることは明らかである。
以上より、

$$L_{SAT} \in NP$$

QED 21

練習

L_{SAT} に属する3変数以上のブール関数と、
 L_{SAT} に属さない3変数以上のブール関数を示せ。

22

8-4 TMとNTMの時間の関係

前に、DTMによってNTMをシミュレートできることを示した。
ここでは、時間まで考慮に入れて考察する。

$T(n)$ を $T(n) > n$ であるような関数とする。
このとき、すべての $T(n)$ 時間限定NTMに対して、
それと等価な $2^{O(T(n))}$ 時間限定DTMが存在する。

証明

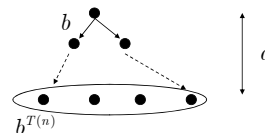
Nを $T(n)$ 時間で動作するNTMとする。
前にみてきたように、NTMの計算の木を幅優先で探索しながらシミュレートする3テープ決定性DTM D_1 が存在する。
まず、この D_1 の計算時間が $2^{O(T(n))}$ であることを示す。

23

Nの分岐の最大値を b とする。(つまり、計算木において、どの節点に対しても、高々 b の子供しかいない。)

Nの計算木において、最も浅い受理様相の深さを d とする。

このとき、深さ $d = T(n)$ であり、
このには深さには高々 $b^{T(n)}$ の頂点しかない。



また、根から深さ d までの計算木に現れる頂点の総数は、 $2b^{T(n)}$ 以下、すなわち $O(b^{T(n)})$ である。

24

計算木の各頂点に対して、シミュレーションは $O(T(n))$ 時間で行える。

よって、 D_1 の計算時間は、

$$O(T(n)) \times O(b^{T(n)}) = 2^{O(T(n))}$$

である。

D_1 は3テープTMであったので、これを1テープTM D_2 でシミュレートする。このシミュレートは、2乗時間で行える。よって、 D_2 は、

$$(2^{O(T(n))})^2 = 2^{2 \times O(T(n))} = 2^{O(T(n))}$$

時間で N をシミュレートすることができる。 *QED* 25

8-5. 検証可能性

ここでは、クラスNPのもう一つの特徴づけを与える。クラスNPは、直感的には、答えがの正当性が多項式時間で検証できる問題の集合ともみなせる。

あるアルゴリズム V に対して、言語 A を次のように定義できるとき、 V を A の**検証装置 (Verifier)**という。

$$A = \{w \mid V \text{ は文字列 } w \text{ に対して } \langle w, c \rangle \text{ を受理}\}$$

このとき、 c を**証拠 (witness)**という。(なお、証拠としては、答えそのものであることが多い。ただし、答え以外の証拠もあるので、注意が必要。)

26

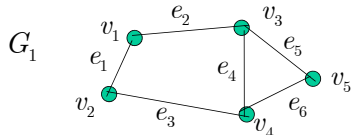
多項式時間検証可能性

検証装置の時間は、 w の長さに対してのみ計られる。したがって、**多項式時間検証装置**とは、 w の長さに対して、多項式時間で c の検証を行うアルゴリズムである。言語 A が決定的多項式時間検証装置をもつとき、 A を**多項式時間検証可能**という。

27

多項式時間検証の例1

L_{HC} は多項式時間検証可能である。



$\langle w, c \rangle = \langle G, e_1 e_2 e_3 e_5 e_6 \rangle$
 ここで、 $c = e_1 e_2 e_3 e_5 e_6$

c は順序が異なるので、ハミルトン閉路ではないが、ハミルトン閉路で用いる辺集合を与えている。これより、多項式時間検証可能である。

QED 28

多項式時間検証の例2

L_{SAT} は多項式時間検証可能である。

$$f_1(x_1, x_2, x_3, x_4)$$

$$= (\overline{x_1} + x_3 + \overline{x_4})(x_2)(x_1 + x_4)(x_2 + x_3 + \overline{x_4})$$

$$x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 1$$

$\langle w, c \rangle = \langle f_1(x_1, x_2, x_3, x_4), 0111 \rangle$

この証拠のチェックは明らかに多項式時間で行える。

QED 29

クラスNPと検証可能性

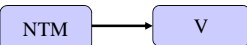
ここでは、クラスNPが多項式時間検証可能な言語と等価であることを示す。

言語 L がクラスNPに属するための、必要十分条件は L が多項式時間検証可能であることである。

証明

$w \in L$ を判定するNTMから $\langle w, c \rangle$ を検証する検証装置 V を構成し、
 $\langle w, c \rangle$ を検証する V から
 $w \in L$ を判定するNTMを構成する。

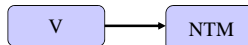
30



1. NTMの非決定的に選択される記号をすべて集めて証拠 c とする。
2. c の表す枝での計算をVはシミュレートする。
3. 2. においてNが受理するならVも受理し、Nが拒否するならVも拒否する。

このシミュレーションは明らかに正しく動作する。

31



Vは n^k 時間で動作するDTMと仮定する。

1. 長さ n^k の文字列 c を非決定的に選択する。
2. 入力 $\langle w, c \rangle$ に対して、VをNTMでシミュレートする。
(VはDTMなので、NTMで容易にシミュレートできる。)
3. 2. において、Vが受理するなら受理し、Vが拒否するなら拒否する。

以上より、クラスNPは多項式時間検証可能な問題の集合でもあることが示された。

QED 32

8-6. 多項式時間帰着

ここでは、問題間の難しさを調べるために、多項式時間帰着について述べる。直感的には、多項式時間帰着とは問題の変換のことである。

ある問題を解く際に、他の問題が利用可能な場合がよくある。この際に、もし利用される側の問題に効率的なアルゴリズムが存在していたならば、利用する側の問題も効率よく解ける可能性がある。

帰着の考え方は問題が難しいことを示すときにも利用される。難しいことがわかっている問題Aのすべてのインスタンスが、別の問題Bに変換可能ならば変換された問題Bを利用して、元の問題Aに対するアルゴリズムが得られる。このことから問題Bは、問題Aより易しくはないことを示している。つまり、問題Bも難しいといえる。

33

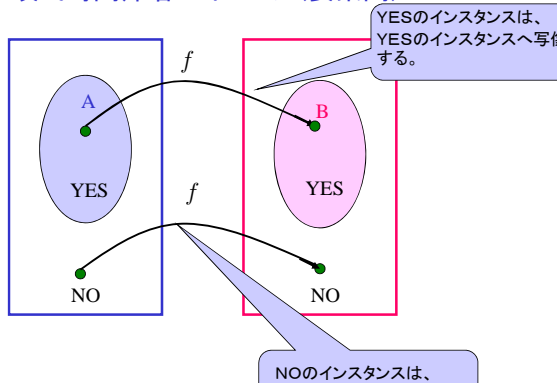
多項式時間帰着の定義

言語Aと言語Bに対して、すべての $w \in A$ に対して、
 $w \in A \Leftrightarrow f(w) \in B$

であるような多項式時間帰着関数 $f: \Sigma^* \rightarrow \Sigma^*$ が存在するとき言語Aは言語Bに多項式時間(多対一)帰着 (Polynomial time many to one reduction) 可能という。ここで、多項式時間帰着関数とは、関数の計算(変換)が多項式時間で行えるもののことである。言語Aから言語Bへ多項式時間帰着可能であることを、
 $A \leq_m^p B$ あるいは $A \leq B$ と書く。

34

多項式時間帰着のイメージ(要素間)

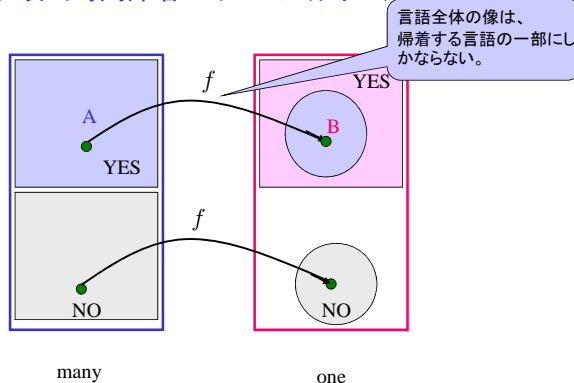


YESのインスタンスは、YESのインスタンスへ写像する。

NOのインスタンスは、NOのインスタンスへ写像する。

35

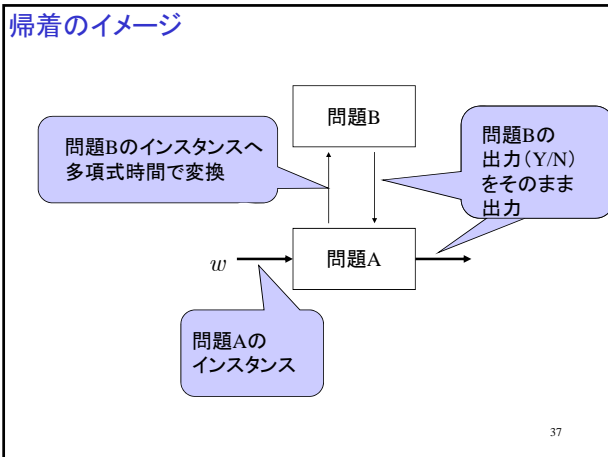
多項式時間帰着のイメージ(クラス)



言語全体の像は、帰着する言語の一部にしかならない。

many one

36



37

帰着とクラスP

帰着の性質から次の命題が成り立つ。

$A \leq_m^P B$ かつ $B \in P$ ならば $A \in P$ である。

証明 問題Bを判定する多項式時間TM(アルゴリズム)をTとし、 f をAからBへの多項式時間帰着とする。

このとき、TM Tを利用して、問題Aを判定するTM T' (アルゴリズム)が構成できる。

Aへのインスタンス w に対して、

- $f(w)$ を計算する。
- 入力 $f(w)$ に対してTを動作させ、その出力をT'の出力とする。

このTM T' は明らかに多項式時間で動作する。 *QED* 38

帰着の例

ここではブール関数の問題からグラフの問題への帰着を示す。まず、これらの問題を示す。

39

3SAT

ここでは、SATを特殊化した問題を考える。そのためにSATの問題を再考する。

ブール変数に対してその否定と肯定を **リテラル(literal)**という。例えば、 x_1, \bar{x}_1 等がリテラル。また、リテラルを論理和で結んだ式を **節(clause)**という。例えば、 $(x_1 + x_2 + \bar{x}_3)$ 等が節。節を論理積で結んだ式が **和積標準形(Conjunctive Normal Form)**であり、**CNF論理式**と約される。

すべての節が3つのリテラルからなるようなCNF論理式(3CNF)に対して、充足可能なものすべてからなる言語を **3SAT**と呼ぶ。

40

変数の個数自体には制限が無いことに注意

名称: 3SAT(3充足可能性問題、3SATisfiability problem)
 インスタンス: 3CNF論理式 $f(x_1, x_2, \dots, x_n)$

問: $f = 1$ となる x_1, \dots, x_n への0, 1の割り当てが存在するか?

また、この問題に対応する言語を、
 $L_{3SAT} = \{f \mid f \text{は充足可能な3CNF}\}$
 と定める。

41

k クリーク問題

無向グラフ $G = (V, E)$ に対して、完全部分グラフを **クリーク(clique)**という。 k 点の完全部分グラフを **k クリーク**という。

インスタンス

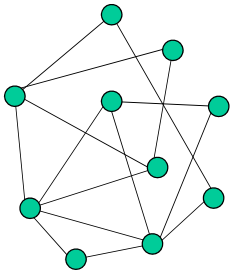
5クリークの例

42

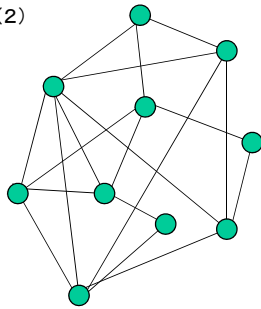
練習

次のグラフから最大のクリークを見つけよ。

(1)



(2)



43

名称: kクリーク、
 インスタンス: $\langle G, k \rangle$
 問: G中に、kクリークが存在するか?

また、この問題に対応する言語を、

$$L_{CQ} = \{ \langle G, k \rangle \mid G \text{は} k \text{クリークを持つ。} \}$$

と定める。

44

多項式時間帰着

L_{3SAT} は L_{CQ} に多項式時間帰着可能である。

証明

f を p 個の節を持つ次のような n 変数 3CNF とする。

$$f(x_1, \dots, x_n) = (a_1 + b_1 + c_1)(a_2 + b_2 + c_2) \dots (a_p + b_p + c_p)$$

ここで、各 a_i, b_i, c_i はリテラルを意味し、
 $a_i, b_i, c_i \in \{x_1, x_2, \dots, x_n, \bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}$ である。

45

この f から k クリーク問題のインスタンスである $\langle G, k \rangle$ を生成する。
 すなわち、グラフ G と整数 k を生成する。

まず、整数 k は節数 p に設定する。すなわち、
 $k = p$
 とする。

次にグラフ G の構成法を示す。

まず、各 a_i, b_i, c_i に対応する点集合を作成し、
 各節ごとの k グループに分ける。

辺集合は次の規則で定める。

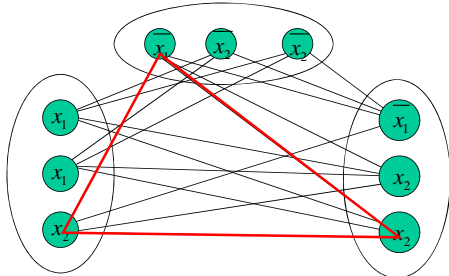
1. 同じグループ内の点間には辺を引かない。
2. 異なるグループ間には、矛盾がない限り
 全ての点間に辺を引く。

(ここで、矛盾とは、 $a_i = x_s, a_j = \bar{x}_s$ のように
 互いに否定の関係にあるものを指す。)

46

$$f(x_1, x_2) = (x_1 + x_1 + x_2)(\bar{x}_1 + \bar{x}_2 + \bar{x}_2)(\bar{x}_1 + x_2 + x_2)$$

に対するグラフ G の構成例を示す。



47

ここで、この帰着が正しく動作することを示す。すなわち、
 「 f が充足可能であるための必要十分条件が、
 G に k クリークが存在することである。」ことを示す。

必要性:

f に充足可能な割り当てが存在すると仮定する。
 この割り当てでは、全ての節で少なくとも一つのリテラルが
 真である。 G においてその真である点を選ぶ。
 そのとき、 G の構成法から選ばれた点間にはすべて
 辺があることがわかる。したがって、 k クリークを持つ。

48

十分性:

G に k クリークがあると仮定する。
 同じグループの点どうしには辺がないので、クリークの
 どの頂点も異なるグループに属する。よって、
 すべてのグループ中の一つの点がクリークに属する。

このとき、クリークに属する点が真となるように、
 割り当てを設定すること(リテラルに真偽の値を設定すること)
 ができる。
 (矛盾するリテラル間には辺がないので、この割り当ては
 可能である。)

以上より、 $f \in L_{3SAT} \Leftrightarrow \langle G, k \rangle \in L_{CQ}$ であり、
 命題が証明された。

*QED*₄₉

練習

次の3CNFのインスタンスに対して、
 帰着で得られるグラフ G を構成せよ。

$$f(x_1, x_2, x_3) = (x_1 + x_2 + x_3)(\overline{x_2} + \overline{x_3} + \overline{x_3})(\overline{x_1} + \overline{x_1} + x_2)(\overline{x_2} + \overline{x_2} + \overline{x_3})$$

また、この f に充足可能な割り当てを見つけ、
 G に対応する4クリークを見つけよ。

50