決定不能問題
特にタイル張り問題

メニュー

1. 計算課題と計算可能性
2. チューリングマシン
3. 停止性問題の決定不可能性
4. タイル貼り問題の決定不可能性
5. タイル貼り問題の難しさの度合い
6. しめくくり

メニュー

1. 計算課題と計算可能性
2. チューリングマシン
3. 停止性問題の決定不可能性
4. タイル貼り問題の決定不可能性
5. タイル貼り問題の難しさの度合い
6. しめくくり

- 今回の発表では「決定不可能な計算課題」を紹介します

計算課題

- 計算課題とは想定される入力の型と質問の組である
- 例: 計算課題 IsPrime
  - 入力: 自然数 n
  - 質問: n は素数か

- ただし，入力は自然数に符号化できるものしか扱わないものとする (入力が実数とかはダメ)
- また当然のことだが，質問は数学的に意味をなすものでないとダメ (x は美人である，とかはダメ)

決定可能の定義

- 計算課題 X は次のとき決定可能という
- 次の関数 f_X が計算可能であるとき
  - $f_X(x) = \begin{cases} 1, & \text{計算課題}X\text{の}x\text{に対する答えはyesである} \\ 0, & \text{計算課題}X\text{の}x\text{に対する答えはnoである} \end{cases}$

関数が計算可能とはなんぞや？

計算可能の定義

- 実は自然数上の関数が計算可能であるということの定義はたくさんあって，しかもそのすべてが同値である
  - 帰納的関数
  - チューリングマシン
  - ラムダ計算
  - レジスタマシン
  - etc

- この発表では計算可能性の定義として二つ「C言語のサブセット」「チューリングマシン」を導入する

C言語のサブセット

- 「C言語のサブセット」はC言語に次のような変更を加えたものである
- まず追加点：
  - int型は扱える範囲に制限がなく絶対値がどんなに大きい整数でも扱えるものとする

- 次に削除点
  - 入出力 (printf, scanfなど)
  - グローバル変数
  - 標準ライブラリ (randなど)
  - その他外部ライブラリ (並列処理や機種依存コマンドなど)

- また，ハードウェアのエラーは決して発生せず，メモリは十分に与えられているとする．つまり，いつどんな環境で実行しても常に実行結果は一定になるような理想的状況だけを考える． ([2]より)

計算可能の定義

- 「C言語のサブセット」のあるプログラムにより計算できる関数のことを「計算可能」な関数と呼ぶ

IsPrime の決定可能性

- さて「C言語のサブセット」が定義できたところで計算課題 IsPrime が決定可能なことを見よう
- 以下のプログラムより，IsPrime は決定可能である

```
isPrime(int n) {
        if (n <= 1) return 0;
        for (int i = 2; i < n; i ++) {
                if ((n % i) == 0) return 0;
        }
        return 1;
}
```

メニュー

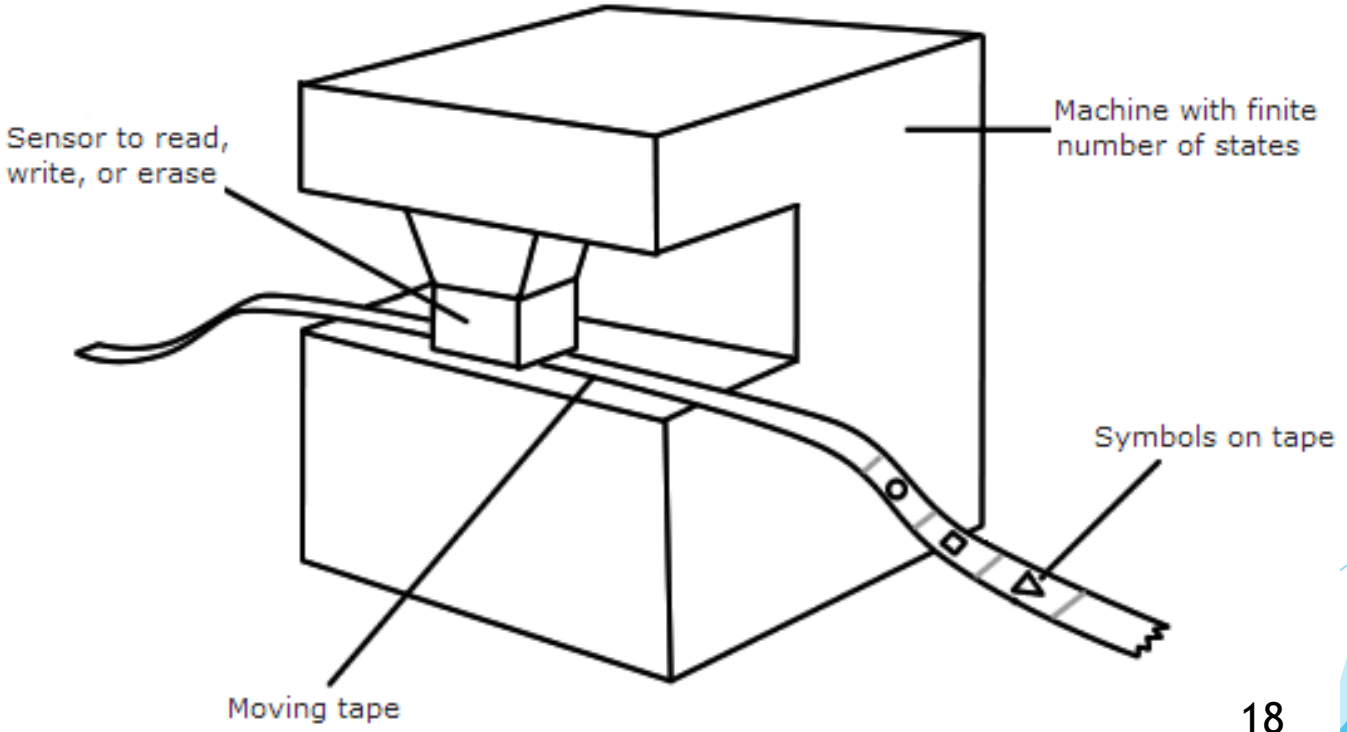
1. 計算課題と計算可能性
2. チューリングマシン
3. 停止性問題の決定不可能性
4. タイル貼り問題の決定不可能性
5. タイル貼り問題の難しさの度合い
6. しめくくり

チューリングマシン

- チューリングマシン $M = \langle \Sigma, b, Q, q_0, q_h, \delta \rangle$ は次のような6つ組である
  - Σ は有限集合で $b \in \Sigma$
  - Σ の元をテープ記号，b を空白記号という
  - Q は有限集合で $q_0, q_h \in Q$
  - Q の元を状態と呼び，q_0 は初期状態，q_h を停止状態という
  - $\delta: \Sigma \times (Q \setminus \{q_h\}) \to \Sigma \times Q \times \{L, R\}$を遷移関数という

チューリングマシンのイメージ図

- http://64350135.weebly.com/turings-background.htmlより転載



A Turing Machine

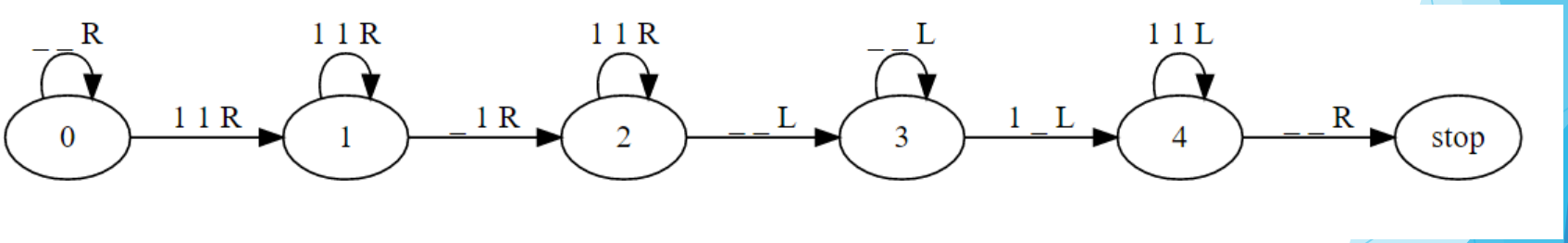
チューリングマシン

- チューリングマシンは状態が格納された本体と左右無限に伸びたテープとテープ上の記号を読み書きするヘッドからなる
- テープはマス目に区切られており，ひとつのマス目にはテープ記号がちょうど一つ書き込まれている．ヘッドは各時点でテープのマスのひとつを見ている
- 本体の状態は各時点で状態集合 Q のいずれかの状態にある

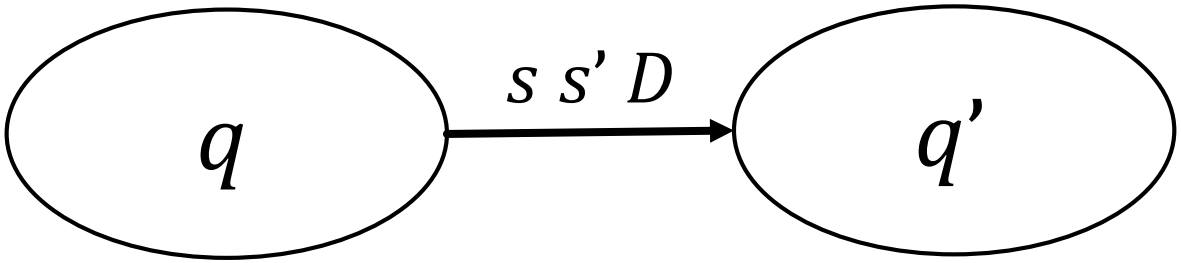
- ある時点でヘッドが見ているテープのマス目に書かれたテープ記号がa, 本体の状態が$q \neq q_h$であるとする
- このとき遷移関数$\delta(a, q) = (a', q', D)$によって，つぎの時点での
  - ヘッドが見ていたマス目上の記号をa'に書き換え
  - 本体の状態をq'にし
  - ヘッドを$D = L$ならひとつ左に，$D = R$ならひとつ右に動かす

チューリングマシンのグラフ表示

- Q の元を頂点として遷移関数 δ によって定まる対応を辺とするグラフでチューリングマシンを表せる
- 例：



- グラフの作り方
- $\delta(s, q) = (s', q', D)$ なら次のように q から q' への辺をはる



- つまりこの辺は「今状態 q にいてテープが s を読んでいるなら，テープの記号を s' に書き換えて D の方向(L なら左，R なら右)へ行き，状態を q' に変える」ということを意味する

チューリングマシンの実行例

メニュー

1. 計算課題と計算可能性
2. チューリングマシン
3. 停止性問題の決定不可能性
4. タイル貼り問題の決定不可能性
5. タイル貼り問題の難しさの度合い
6. しめくくり

停止性問題

- 次の計算課題を停止性問題という
  - 入力: プログラム p とそれに対する入力 x
  - 質問: p に入力 x を与えたとき実行は停止するか

- これを計算課題と考えるためには，まずプログラムを自然数に符号化 (コーディング) できないといけない
- しかし，プログラムの自然数への符号化を定義するのは難しくない (実際コンピュータの中でそれが2進法を使って行われている)
- ここではすでにプログラムの自然数への符号化が済んだものとしよう

- p をあるプログラムのコードとしたとき $\{p\}$ でそのプログラムが計算する関数を表すことにする
- また，関数fがxで定義されていることを $f(x)\downarrow$，定義されていないことを$f(x)\uparrow$と書く
- なお，pが表すプログラムにxを入力して停止しない場合は，$\{p\}(x)$は未定義と定める．すなわち$\{p\}(x)\uparrow$

- すると自然数上の2変数関数 halt を
  $\mathrm{halt}(p,x) = \begin{cases} 1, & (\{p\}(x)\downarrow) \\ 0, & (\{p\}(x)\uparrow) \end{cases}$ と定めることで，
  停止性問題の決定可能性は halt の計算可能性と同じである

停止性問題の決定不能性の証明

- $\mathrm{halt}(p, x)$ が計算可能だと仮定する
- d を次のプログラムのコードとする

```
diag(x) {
    if (halt(x, x) == 1) {
        while (0 == 0) {}
    } else {
        return 0;
    }
}
```

- すると$\{d\}(d)\downarrow \Leftrightarrow \mathrm{halt}(d, d) = 0 \Leftrightarrow \{d\}(d)\uparrow$となって矛盾

停止性問題その2

- 停止性問題を少し変更した次の計算課題を考える
  - 入力: プログラム p
  - 質問: p に 0 を入力したとき停止するか
- これの決定可能性は次の関数の計算可能性のことである
- $\mathrm{halt}_0(p) = \begin{cases} 1, & (\{p\}(0)\downarrow) \\ 0, & (\{p\}(0)\uparrow) \end{cases}$

補題

ある計算可能関数Sが存在して任意のpとxに対して

$$\{p\}(x) = \{S(p,x)\}(0).$$

以下のアルゴリズムをC言語のプログラムに落とし込み，そのプログラムで計算される関数をSとすればよい．
$p, x \in \mathbb{N}$が入力されたとする．
pが表すプログラムが

```
f(x) {
    (fの本体)
}
```

という形だとする．そのとき次のプログラムを作る．

```
f'(y) {
    int x = x;
    (fの本体)
}
```

そしてf'のコードを返す． □

halt_0の計算不可能性の証明

- halt_0が計算可能だと仮定する.
- このとき補題のSとhalt_0を計算するプログラムを使って以下のようにhaltを計算するプログラムが書けてしまう．これはhaltの計算不可能性に矛盾.

```
halt(p, x) {
        int p' = S(p, x);
        return halt0(p');
}
```

チューリングマシンの停止性問題

- 次の計算課題を考える
  - 入力: チューリングマシンM
  - 質問: Mに空を入力したとき停止するか

- この計算課題についてもやはりチューリングマシンの符号化を定めないといけないが，やはりそれは難しくないので省略.
- C言語のサブセットのときと同様の議論をすることでこの計算課題も決定不可能といえる

メニュー

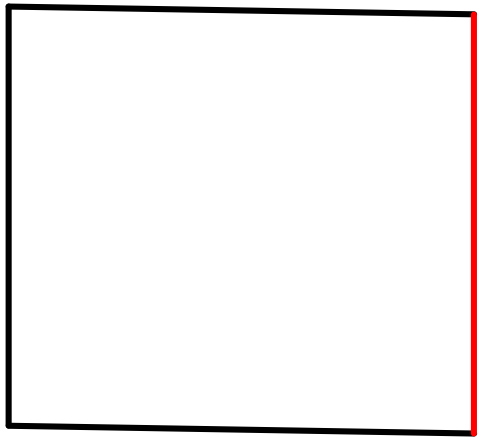
1. 計算課題と計算可能性
2. チューリングマシン
3. 停止性問題の決定不可能性
4. タイル貼り問題の決定不可能性
5. タイル貼り問題の難しさの度合い
6. しめくくり

タイル貼り問題

- 次の計算課題を考える
- 入力：4辺にそれぞれ色のついた単位正方形の有限集合 (タイル集合)
- 質問：平面全体を与えられたタイルのコピーで埋め尽くせるか
- ただし隣り合った辺同士は同じ色でないといけないものとする
- またタイルの回転，裏返しは禁止する

例

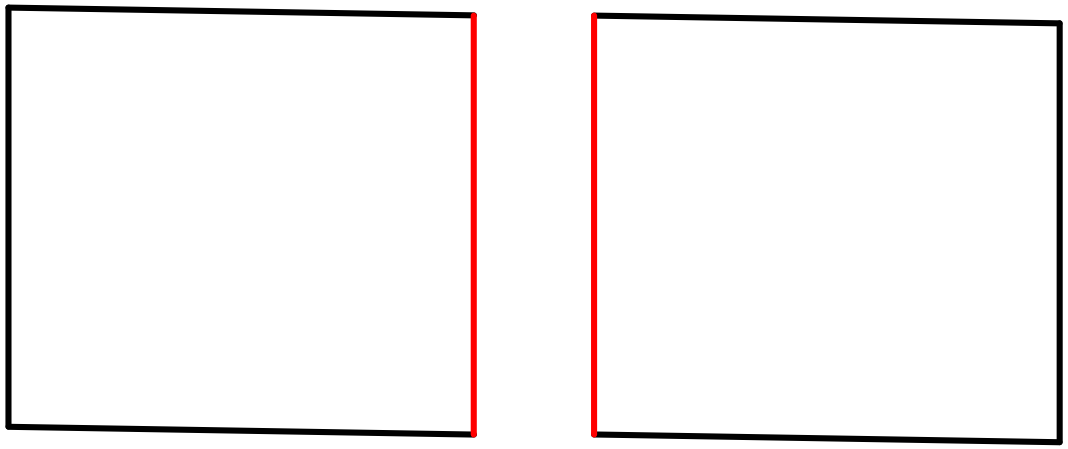
- タイル集合が次の1個のタイルからなる集合のとき



- これはタイル貼り不可能

例

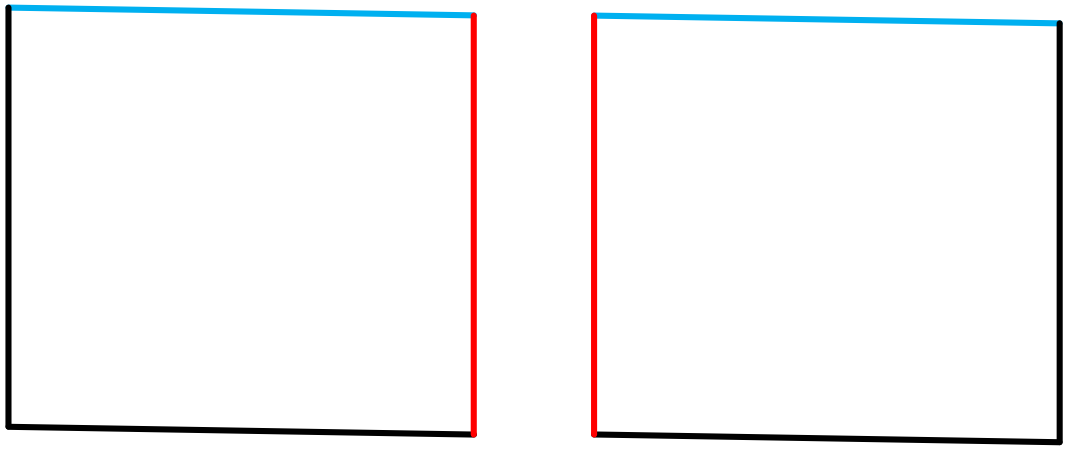
- タイル集合が次の2個のタイルからなる集合のとき



- これはタイル貼り可能

例

- タイル集合が次の2個のタイルからなる集合のとき



- これはタイル貼り不可能

- 定理

タイル貼り問題は決定不能.

- この定理の代わりにもう少し証明のやさしい次の定理を示す

- 定理

原点制約付きタイル貼り問題は決定不能

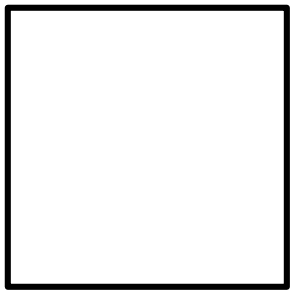
- 原点制約付きタイル貼り問題
- 入力：タイル集合 T と T の要素 t
- 質問：t を原点に配置した T の平面のタイル貼りができるか

- 証明の方針
- 与えられたチューリングマシンに対して適当にタイル集合 T を作れば，タイル貼りによってチューリングマシンの動作をシミュレートできることを言う

- 与えられたチューリングマシンを $M = \langle \Sigma, b, Q, q_0, q_h, \delta \rangle$ とする
- 次のような5種類のタイル (空白タイル，記号タイル，合流タイル，遷移タイル，初期タイル)を用意する
- タイルの辺には色が塗られていると問題を設定したが，これはタイルに記号が書かれていると変更しても本質的に違いはないのでこれからは色ではなく記号がついているものとする

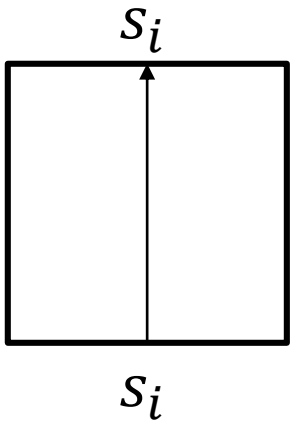
空白タイル

- 4辺とも空白記号が書かれたタイル



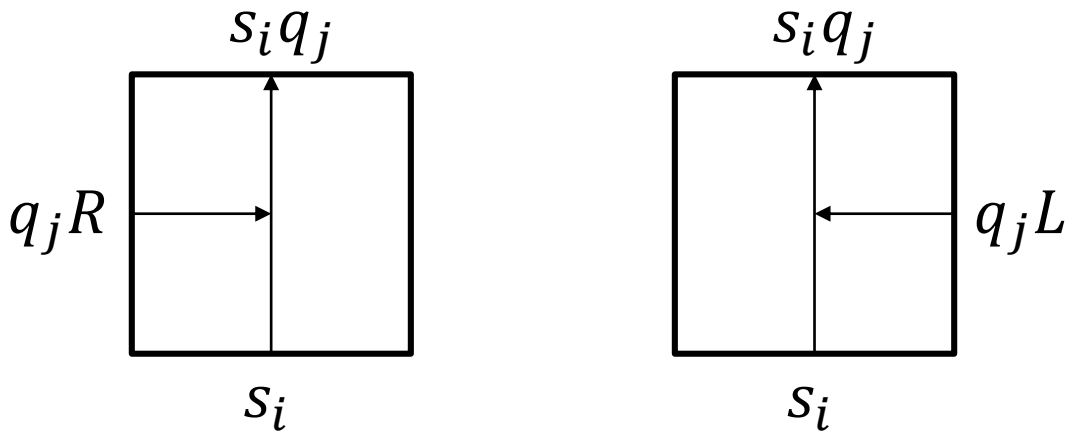
記号タイル

- 各 $s_i \in \Sigma$ について次のタイルを用意する



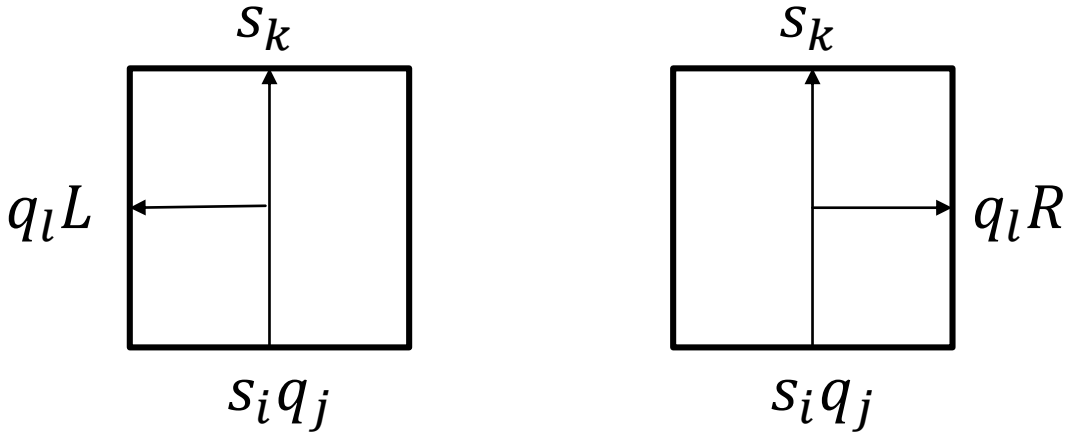
合流タイル

- 記号 $s_i \in \Sigma$ と状態 $q_j \in Q$ の組ごとに二種類ずつ用意する



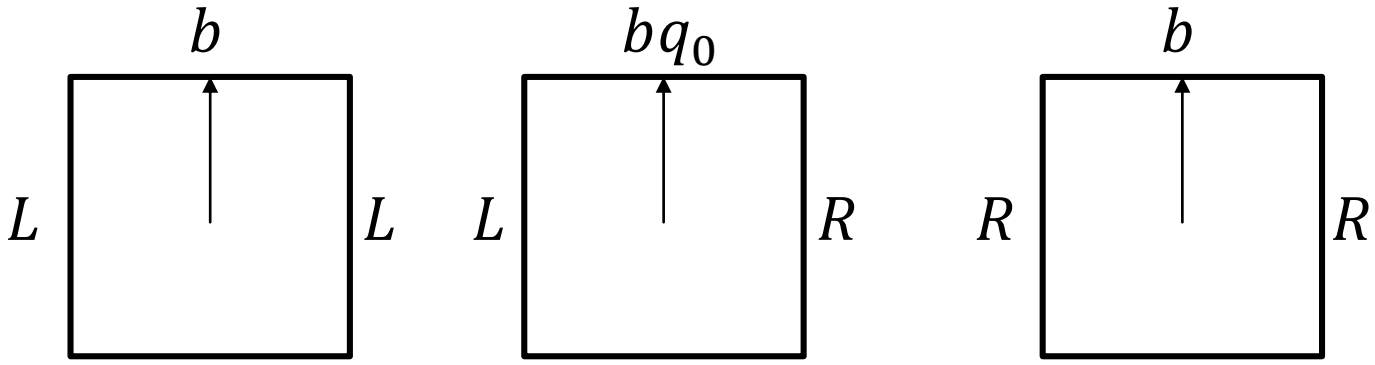
遷移タイル

- 次のタイルは $\delta(s_i, q_j) = (s_k, q_l, D)$ であるような組 (s_i, q_j, s_k, q_l, D) に対してのみ用意する．ここで $D = L$ なら左のタイルを $D = R$ なら右のタイルを作る．

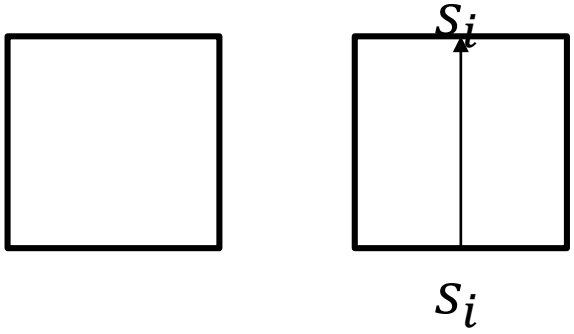


初期タイル

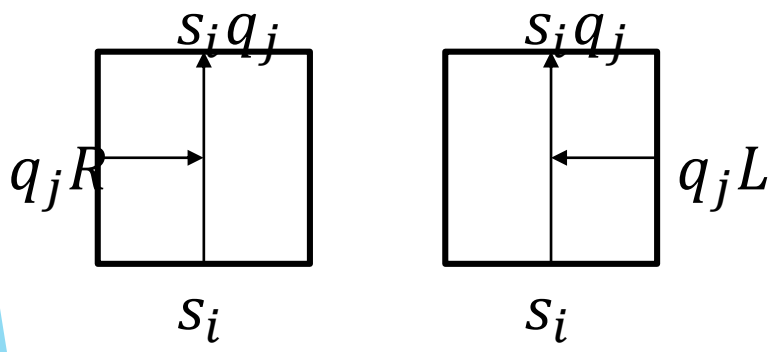
- 次の3つのタイルを初期タイルという



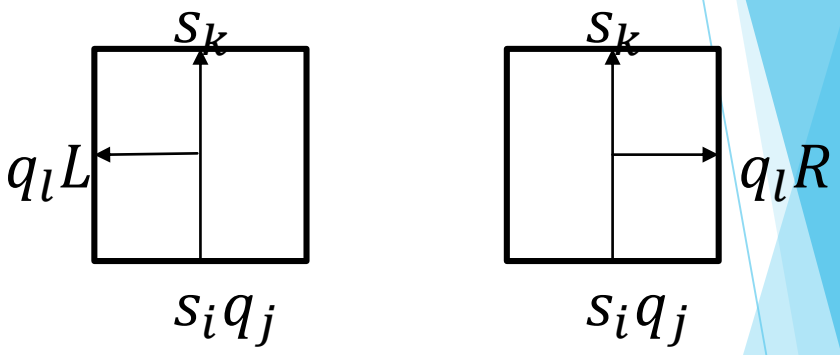
空白タイル　記号タイル



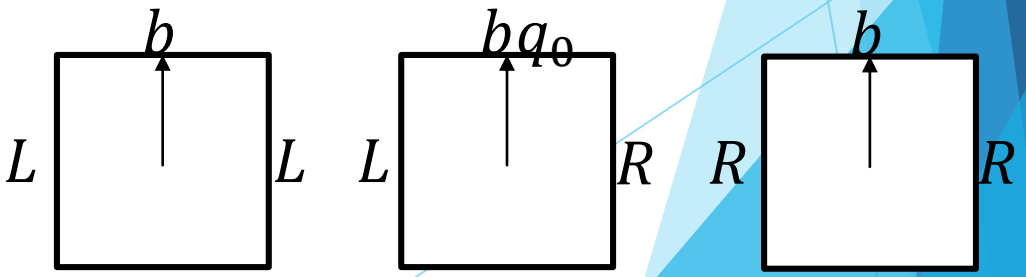
合流タイル



遷移タイル (δによる対応の分だけ用意)



初期タイル



このタイルを原点におく

- 補題

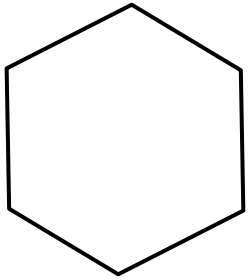
初期タイルの中央のものを原点に置くタイル貼りが可能である必要十分条件はチューリングマシン M に空列を与えた計算が停止しないことである

- 定理：「原点制約付きタイル貼り問題は決定不能」の証明
- 決定可能だと仮定
- すると次のようにしてチューリングマシンの停止性判定が決定可能となって矛盾
- チューリングマシン M が与えられたとき，上に書いた方法でタイル集合を作る
- このときタイル貼りできるか判定する
- できたら M は停止しないと答え，できなければ停止すると答えればよい □

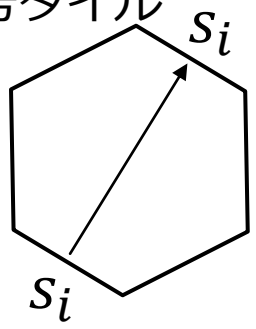
- 正方形で平面を充填することを扱ったが，正三角形，正六角形の場合もタイル貼り問題は決定不能である

- 正六角形の場合は次のようなタイル集合を考えればよい

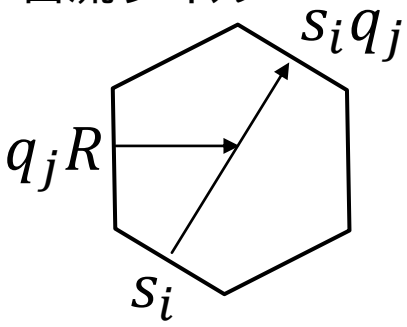
空白タイル

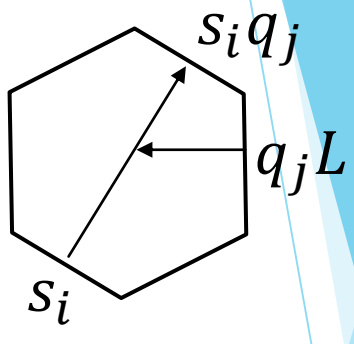


記号タイル

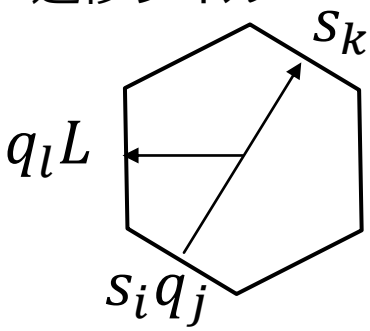


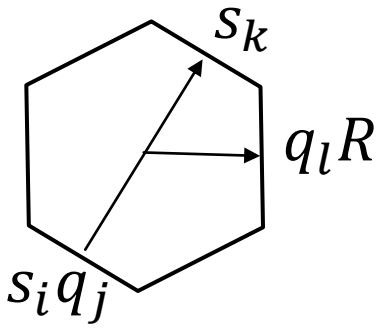
合流タイル



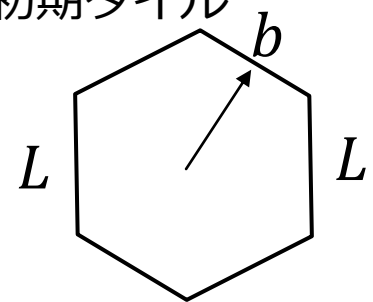


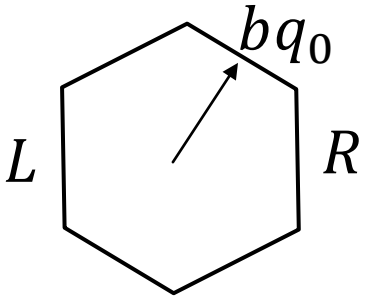
遷移タイル

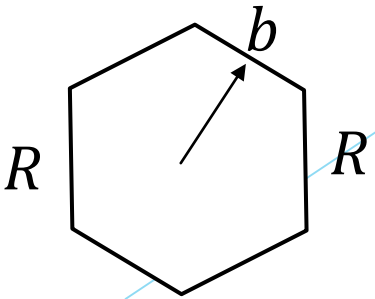




初期タイル







メニュー

1. 計算課題と計算可能性
2. チューリングマシン
3. 停止性問題の決定不可能性
4. タイル貼り問題の決定不可能性
5. タイル貼り問題の難しさの度合い
6. しめくくり

- これから次の定理を証明する.

タイル集合が与えられたとする.
このとき平面全体をタイル貼りできることと任意の大きさの正方形がタイル貼りできることは同値.

- 平面全体をタイル貼り可能⇒任意の大きさの正方形がタイル貼り可能は自明
- そこで，任意の大きさの正方形がタイル貼り可能だと仮定して平面全体をタイル貼り可能であることを導く．
- 今与えられたタイル集合を $T = \{t_1, \ldots, t_n\}$ とおく．

- 平面全体にタイルを貼るということは次の写像 f を定めることである

$$f: \mathbb{Z}^2 \to T$$

- この f がタイル貼り問題の条件を満たすことは次のように書かれる

$$\forall (x, y) \in \mathbb{Z}^2, \begin{pmatrix} R\big(f(x,y)\big) = L\big(f(x+1,y)\big) \text{ かつ} \\ T\big(f(x,y)\big) = B\big(f(x,y+1)\big) \end{pmatrix}$$

- ただし $R(t), L(t), T(t), B(t)$ はそれぞれタイル t の右，左，上，下の辺の色

- ここで写像 $\mathbb{Z}^2 \to T$ の全体の集合 $T^{\mathbb{Z}^2} = \{f : \mathbb{Z}^2 \to T\}$ に位相を入れる
- T: 離散位相空間
- $T^{\mathbb{Z}^2}$: T を直積因子とする直積位相空間

位相空間論の復習

- ご覧のとおり，この証明には位相空間論を使う．そこで位相空間論の復習を挟む

コンパクトの定義

- 位相空間 X がコンパクトであるとは次のことを言う：

開集合 O_λ $(\lambda \in \Lambda)$ の族 $\{O_\lambda\}_{\lambda\in\Lambda}$ が条件

$$X = \bigcup_{\lambda\in\Lambda} O_\lambda$$

を満たすならば，Λの有限部分集合 $\{\lambda_1, \ldots, \lambda_n\}$ が存在し

$$X = \bigcup_{i=1}^{n} O_{\lambda_i}$$

となる

Prop.

- X がコンパクトであり，$A_n\ (n \in \mathbb{N})$は X の空でない閉集合であり，

$$A_1 \supseteq A_2 \supseteq A_3 \supseteq \cdots$$

をみたすとする

- このとき $\bigcap_{n\in\mathbb{N}} A_n \neq \emptyset$ である

チコノフの定理

- 次の定理は今回の証明で本質的な役割を果たす

コンパクト空間の直積空間もまたコンパクトである.
つまり各$\lambda \in \Lambda$についてX_λがコンパクトなら直積空間$\prod_{\lambda\in\Lambda} X_\lambda$ もコンパクトである

Prop.

- $X = \prod_{\lambda \in \Lambda} X_\lambda$ を直積空間，$x \in X$ とする．
- 有限個の $\lambda_1, \ldots, \lambda_n \in \Lambda$ と $x(\lambda_i)$ の近傍V_{λ_i} を使って

$$V = \{y \in X \mid y(\lambda_i) \in V_{\lambda_i}\ (\mathrm{i} = 1, \ldots, n)\}$$

と書ける X の部分集合 V は x の近傍である

Prop.

- Xを位相空間とし，Mを部分集合とする
- このとき任意の $x \in M$ についてある x の近傍 V が存在して $V \subseteq M$ が成立するならば M は開集合である

- 以上，位相空間論の復習でした

タイル貼りの定理の再開

- 写像 $\mathbb{Z}^2 \to T$ の全体の集合 $T^{\mathbb{Z}^2} = \{f : \mathbb{Z}^2 \to T\}$ に位相を入れる
- T: 離散位相空間
- $T^{\mathbb{Z}^2}$: T を直積因子とする直積位相空間

- T は有限集合なのでコンパクトである
- よってチコノフの定理より$T^{\mathbb{Z}^2}$もコンパクト

Lemma

- $F_k =$

$$\left\{ f : T \to \mathbb{Z}^2 \;\middle|\; \begin{array}{c} \forall (x,y) \in \{-k, \ldots, k-1\}^2, \\ \left(\begin{array}{c} R\big(f(x,y)\big) = L\big(f(x+1,y)\big) \text{ かつ} \\ T\big(f(x,y)\big) = B\big(f(x,y+1)\big) \end{array} \right) \end{array} \right\}$$

とおく

F_k は平面のタイル配置であって，原点を中心とする一辺の長さ $2k$ の正方形だけ見たときタイル貼りの条件を満たしているようなもの全体である

このとき

- $F_1 \supseteq F_2 \supseteq F_3 \supseteq \cdots$
- F_k は空でない
- F_k は閉集合

- よって，$T^{\mathbb{Z}^2}$がコンパクトであったことと合わせると，$\bigcap_{k\in\mathbb{N}} F_k \neq \emptyset$である
- $\bigcap_{k\in\mathbb{N}} F_k$は平面のタイル貼りの条件を満たすタイル配置全体の集合であるのでこれで平面のタイル貼りが可能であることが導かれた □

この定理で分かったこと

- タイル集合 T について平面のタイル貼りができるという関係を $\mathrm{CanTilePlane}(T)$
- タイル集合 T と自然数 k について1辺の長さ k の正方形のタイル貼りができるという関係を $\mathrm{CanTileSquare}(T, k)$
- とおくとき,

$$\mathrm{CanTilePlane}(T) \Leftrightarrow (\forall k \in \mathbb{N})\mathrm{CanTileSquare}(T, k)$$

- 両辺の否定をとれば
  $\neg \mathrm{CanTilePlane}(T) \Leftrightarrow (\exists k \in \mathbb{N}) \neg \mathrm{CanTileSquare}(T, k)$
- ここに$\mathrm{CanTileSquare}(T, k)$は計算可能な述語である
- よって右辺は計算可能述語に存在記号をつけた形をしている
- これをもって，集合 $\{T \mid \neg \mathrm{CanTilePlane}(T)\}$ は枚挙可能であるという

- 実際，「枚挙可能」の名のとおり集合 $\{T \mid \neg \mathrm{CanTilePlane}(T)\}$のすべての元をすべてもれなく出力するプログラムが次のように書ける

```
enumerateTilesWhichCannotTilePlane() {
    for ((T, k) in ℕ²) {
        if (!canTileSquare(T, k)) {
            print(T);
        }
    }
}
```

(ただし for ((T, k) in $\mathbb{N}^2$)の部分はすべての$\mathbb{N}^2$の元が出てくるような順番で繰り返しを行うものとする)

- 結論：「タイル貼り問題」は決定不能だが「タイル貼りできないタイル集合」は枚挙することができる

メニュー

1. 計算課題と計算可能性
2. チューリングマシン
3. 停止性問題の決定不可能性
4. タイル貼り問題の決定不可能性
5. タイル貼り問題の難しさの度合い
6. しめくくり

分からなかったこと

- 回転や裏返しを許したらどうか？
- 原点制約付きは Π_1 完全だとわかったが制約なしではどうか？

参考文献

1. 新井敏康 (2011)『数学基礎論』岩波書店
2. 照井一成 (2015)『コンピュータは数学者になれるのか？』 青土社
3. 鹿島亮 (2008)『C言語による計算の理論』サイエンス社
4. Robinson, Raphael M. (1971), “Undecidability and nonperiodicity for tilings of the plane”
5. Siamak Taati (2006) “Wang Tiles”

Any Question?