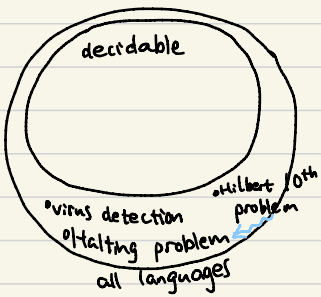


4. Computability

4.1 Universal Turing Machine

Why we need UTM?

1) By building a UTM, we do not need to implement special-purpose hardware for every ? //次要造一个TM就够了

2) Standard-program, i.e. von Neumann architecture (1946)

3) Encode a TM by a string, which enables us to prove incomputability results.

UTM: Input: {程序 / 程序的输入}

eg. 手机: UTM (通用)
下载的应用: TM (专用)

Encoding of a TM

Assume our encoding satisfies the following α: 常数

1) Every string $\alpha \in \{0,1\}^*$ represents some TM, denoted by M_α. (If α is invalid, M_α always rejects).

2) Every TM is represented by infinitely many strings. //可以加一些空状态 or注释

For example, turingmachine.io

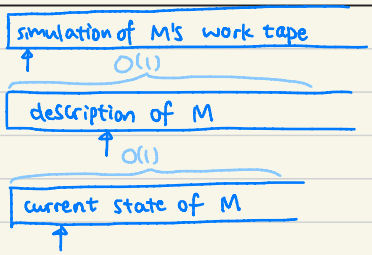
Thm 4.1 A 3-tape universal TM U can be constructed such that for all $\alpha, x \in \{0,1\}^*$, $U(\alpha, x) = M_\alpha(x)$, where $M_\alpha(x)$ is accept, reject, or loop forever. Moreover, if M_α halts within T steps, then U halts in $O_\alpha(T \log T)$ (multitape)

M_α有多少 tape? encode在α里

As a warmup, we construct a UTM that runs in time $O_\alpha(T^2)$.

Proof. By Lemma 3.10, convert M_α to a single-tape TM M', which runs in time $O_\alpha(T^2)$.

Work tapes:



For each step of M, the UTM U:

1) Reads the current symbol on M's work tape.
2) Scan through the description of M to determine the next state and the symbol to be written
3) Overwrites the symbol on the simulated work tape if needed.
4) Moves the simulated head.
5) Transition to the next state.

Overall, all the above steps can be executed in $O_d(1)$. □

Proof of Thm 4.1 The proof has the same overall structure

Let k be the number of work tapes. Let Γ be the tape alphabet.

Assume that the UTM U has alphabet Γ^k.

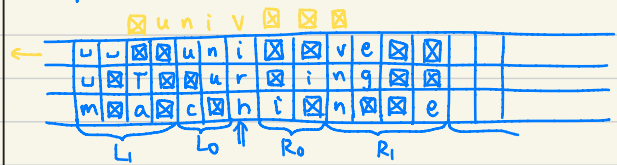
移动指针并修改内容
向左方向移动

$\Gamma \to \Gamma^3$.

Instead of moving the heads, U moves the contents on each tape.

However, the simulation takes $O_M(\Gamma^k)$.

To improve this, we introduce extra buffer space. ⊠



Divide the tape into zones, denoted by $R_0, L_0, R_1, L_1, \cdots, R_{\lfloor n/2 \rfloor}, L_{\lfloor n/2 \rfloor}$.

The center cell does not belong to any zone.

Each zone has size $|R_i| = |L_i| = 2^{i+1}$.

We maintain the following invariants:

1) Each zone is either full, half-full or empty.
   That is, the number of ⊠ cells is L_i (or R_i) is $0, 2^i, 2^{i+1}$
2) The total number of ⊠ in $L_i \cup R_i$ is always 2^{i+1}. That is, one of the following holds:
   a) L_i is full, and R_i is empty.
   b) L_i is empty, and R_i is full.
   c) L_i and R_i are both half-empty.

How to perform a left shift (of the contents)

1) Find the smallest i_0 s.t. R_{i_0} is not empty.
2) Move the leftmost non-⊠ symbol of R_{i_0} to the center. Shift the next $2^{i_0}-1$ non-⊠ symbols from R_{i_0} to $R_0, R_1, \cdots, R_{i_0-1}$, filling each zone exactly half.

$\frac{1}{2}(2 + 2^2 + \cdots + 2^{i_0}) = (2^{i_0+1} - 2)/2 = 2^{i_0} - 1$

R_0, R_1 半 $\Rightarrow L_0, L_1$ full.

R_2 非空 $\Rightarrow L_2$ 非满

center L_0 L_1 L_{i_0-1}

$1 + \frac{1}{2}(2 + 2^2 + \cdots + 2^{i_0}) = 2^{i_0+1} - 1$

$1 + 1 + 2 + 2^2 + \cdots + 2^{i_0-1} = 2^{i_0}$

3) U performs the symmetric operation on the left.
That is, starting from $i_0 - 1$ to 0, it moves half of the non-$\boxtimes$ cells from $L_{i_0-1}, L_{i_0-2}, \cdots, L_0$, plus one additional cell at the center, into L_{i_0}.

4) After that, $R_0, L_0, R_1, L_1, \cdots, R_{i_0-1}, L_{i_0-1}$ are half-full.

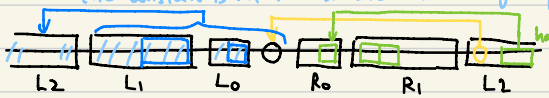
call i_0 the index of this operation.

Initially, all $L_0, R_0, L_1, R_1 \cdots$ are half-full. It takes at least $2^{i_0} - 1$ shifts to make $R_0, R_1, \cdots R_{i_0-1}$ empty.

α: description of another TM
x: Input

Thm 4.1 A 2-tape universal TM can be constructed such that for all $\alpha, x \in \{0,1\}^*$, $U(\alpha, x) = M_\alpha(x)$, where $M_\alpha(x)$ can be accept, reject, or loop forever. Moreover, if M_α halts in T steps, then $U(\alpha, x)$ halts in $O_\alpha(T \log T)$ steps.

the constant is related to the number of tapes in M, encoded in α



left shift (of content) Index is 2

i_0: smallest i_0 s.t. R_{i_0} is not empty
↘ the index of this shift

Observation: once a shift with index i_0 is performed, the next $2^{i_0}-1$ shifts will have indices $< i_0$.

"$\frac{1}{2}(2^1 + \cdots + 2^{i_0}) = 2^{i_0}-1$"

After this shift, $L_{i_0-1}, \cdots, L_0, R_0, \cdots, R_{i_0-1}$ are half-full
The next shift must only affect these blocks.
For each left shift, the num of cells in L +1, R −1.
After $\cdots$ shifts, $L_{i_0-1} \sim L_0$ are full.

amortized analysis

Total work spent in shifting

double-counting

$$k \sum_{i=1}^{T} O(2^{\text{index}(i)}) = k \sum_{j=1}^{O(\log T)} O(2^j) \#\{i : \text{index}(i) = j\}$$

$$\le k \sum_{j=1}^{O(\log T)} O(2^j) \frac{T}{2^j} = k \sum_{j=1}^{O(\log T)} O(T) = O(kT \log T) = O_\alpha(T \log T)$$

□

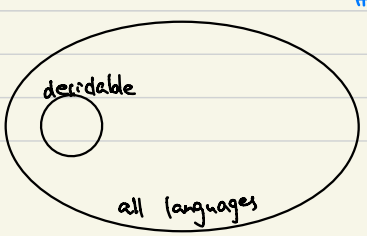
4.2 Undecidable languages

Thm 4.2 Almost all languages are undecidable.

Proof. # all languages = $\#\{L \subseteq \{0,1\}^*\} = 2^{\aleph_0} = \aleph_1$

decidable languages $\le$ # TM = $\aleph_0$

□



"Almost all": pick a random language, the probability of it being decidable is 0.

"diagonalization"

For example, there is no surjection from $\mathbb{N}$ to $\mathbb{R}$. (满射)

0 $\quad *.a_1a_2a_3\cdots$

1 $\quad *.b_1b_2b_3\cdots$

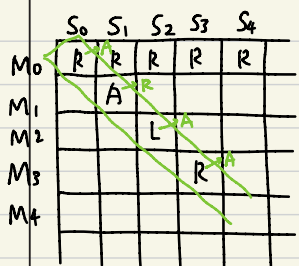
2 $\quad *.c_1c_2c_3\cdots$

$0\sim9$

$a_1 \to a_1+1 \quad 1\to 2 \quad 2\to 3 \quad \cdots \quad 9\to 0$

$b_2 \to b_2+1$

Constructed a different number.



$A \to R$

$R \to A$

$L \to A$

$M_i(S_j) \in \{A, R, L\}$

accept reject loop forever

Construct a new language that's not on this chart.

Let $L_{flip} = \{\alpha : M_\alpha \text{ does not accept } \alpha, \text{ i.e. } M_\alpha \text{ rejects } \alpha \text{ or loops forever}\}$.

Lem 4.3 L_{flip} is undecidable

Proof. Suppose L_{flip} is decidable by M_β.

Case 1. $\beta \in L_{flip}$. By the def. of L_{flip}, $M_\beta(\beta) \in \{R, L\}$. But M_β decides β, so it can't loop forever. M_β rejects β.

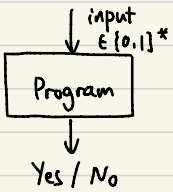
But M_β should accept β because $\beta \in L_{flip}$. Contradiction!

Case 2. $\beta \notin L_{flip}$. By def. M_β accepts β.

But M_β should reject β. Contradiction!

□

$\alpha, x \in \{0,1\}^*$. But (α, x) ? encode again!



Turing halting problem

$L_{halt} = \{(\alpha, x) : M_\alpha \text{ halts on input } x\}$

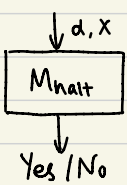
eg. Fermat's Last theorem

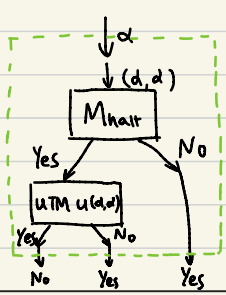
$a^n + b^n = c^n$ has no solutions for integers a, b, c and $n > 2$.

FLT is true iff. $(\alpha, \varepsilon) \notin L_{halt}$

TM M_α

```
T = 2
while true
    T = T + 1
    for d = 3 to T
        for a = 1 to T
            for b = 1 to T
                for c = 1 to T
                    if a^d + b^d = c^d then halt
```





Thm 4.4 L_{halt} is undecidable.

Proof. Assume for contradiction that L_{halt} is decidable, i.e. $\exists$ TM M_{halt} that decides L_{halt}. Construct a new TM M_{flip} as follows:

on input α, run M_{halt} on (α, α)

(1) If M_{halt} rejects (α, α), → loop forever let M_{flip} accept α.

(2) If M_{halt} accepts (α, α), simulate M_α on input α (using a UTM), and invert the output. ↘ A/R

Clearly, M_{flip} decides L_{flip}. Contradiction! □

Lem 4.5 Let $L_{accept} = \{(\alpha, x) : M_\alpha \text{ accepts } x\}$. L_{accept} is undecidable.

Proof. Assume for contradiction that L_{accept} is decidable by a TM M_{accept}.

Construct a new TM M_{halt} as follows:

1) On input (α, x), construct a new TM M_β which simulates M_α on input x and accepts iff it halts.
   If M_α runs forever, M_β runs forever as well.
2) Run M_{accept} on input (β, x), and forward the output.

Thus, M_{halt} decides L_{halt}. Contradiction! □

Lem 4.6 Let $L_{regular} = \{\alpha : L(M_\alpha) \text{ is a regular language}\}$.
$L_{regular}$ is undecidable.

Proof. Assume for contradiction that $L_{regular}$ is decidable. That is,
$\exists$ a TM $M_{regular}$ that decides $L_{regular}$.

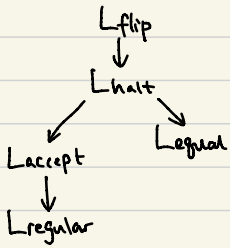
Construct a new TM M_{accept} with input (α, x) as follows:

1) Construct a TM M_β, where $\beta = \beta(\alpha, x)$, and the input of M_β is y.
   a) If y is of the form 0^n1^n, let M_β accept y.
   b) If y is not of the form 0^n1^n, simulate M_α on input x, and let M_β accept y iff. M_α accepts x.
2) Run $M_{regular}$ on β and forward its output

Note that $L(M_\beta) = \begin{cases} \{0^n1^n : n \geq 0\} & \text{if } M_\alpha \text{ does not accept } x. \\ \{0,1\}^*, & \text{otherwise} \end{cases}$

Therefore, M_{accept} decides L_{accept}. Contradiction! □

eg.
$\{0^n1^n : n \geq 0\}$ is not regular



Lem 4.7 Let $L_{equal} = \{(\alpha, \beta) : L(M_\alpha) = L(M_\beta)\}$ Then L_{equal} is undecidable.

Proof. Assume for contradiction that L_{equal} is decided by M_{equal}.

Fix a β such that $L(M_\beta) = \{0,1\}^*$.

Construct a new TM M_{halt} with input (α, x):

1) Create a new TM M_γ, where $\gamma = \gamma(\alpha, x)$ as follows:

on input y, run M_α on x, and accepts y whenever M_α halts.

2) Run M_{equal} on (γ, β), and forward the output.

Thus, M_{halt} decides L_{halt}. Contradiction! □

Def 4.8 (Nontrivial property of languages) Property $P \subseteq \{0,1\}^*$ is a property of languages recognized by TMs if, whenever $L(M_\alpha) = L(M_\beta)$, $\alpha \in P$ iff. $\beta \in P$.

Property P is nontrivial if $P \neq \emptyset$ and $P \neq \{0,1\}^*$

Thm 4.9 (Rice's Theorem) Any nontrivial property of the languages recognized by Turing machines is undecidable.

Proof. WLOG, assume $\varepsilon \notin P$ (Let M_ε always reject)

without loss of generality

Otherwise, take the complement of P.

Assume P is decided by a TM M_P.

Since P is nontrivial, pick an arbitrary $\beta \in P$. where $L(M_\beta) \neq \emptyset$

Construct a TM M_{accept} with input (α, x) as follows:

(1) Construct a TM M_γ, where $\gamma = \gamma(\alpha, x, \beta)$ with input y such that

(a) Simulate M_α on input x. If M_α accepts x, go to step (b); otherwise, loop forever

(b) Simulate M_β on input y, and forward the output.

(2) Run M_P on γ, and forward the output

We claim M_{accept} decides L_{accept}.

1) If M_α accepts x, $L(M_\gamma) = L(M_\beta)$

Since $\beta \in P$, $\gamma \in P$, Thus, M_P accepts.

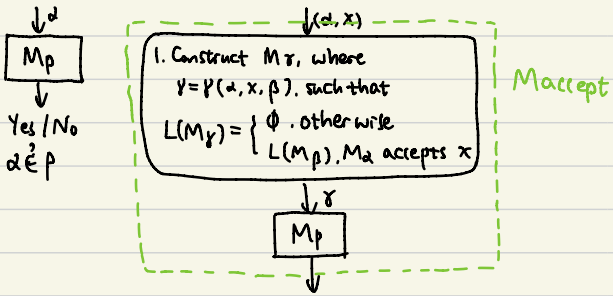
2) If M_α does not accept x, $L(M_\gamma) = \phi$.

Since $\varepsilon \notin P$. $\gamma \notin P$. Thus, M_P rejects.

Contradiction!

□

P WLOG, $\varepsilon \notin P$, $\beta \in P$.



4.3 Post correspondence problem

domino $\left[\frac{a}{ab}\right]$ $\left[\frac{a}{ab}\right]\left[\frac{b}{ca}\right]\left[\frac{ca}{a}\right]\left[\frac{a}{ab}\right]\left[\frac{abc}{c}\right]$

a collection of dominos → ∞ 개

$\left\{\left[\frac{b}{ba}\right], \left[\frac{a}{ab}\right], \left[\frac{ca}{a}\right], \left[\frac{abc}{c}\right]\right\}$

The task is to make a list that forms a match.

Given an alphabet, and a collection of dominos, the Post correspondence problem is to determine whether there is a match.

Formally, let Σ be a finite alphabet of size at least 2. Let

top, bottom $t_i, b_i \in \Sigma^*$, $1 \le i \le n$. An instance is $P = \left\{\left[\frac{t_1}{b_1}\right], \cdots, \left[\frac{t_k}{b_k}\right]\right\}$

4.3 Post correspondence problem

Def 4.11 Let $\Sigma = \{0,1\}$. Given $t_i, b_i \in \Sigma^*$, $i = 1, \cdots, k$, a **match** is a sequence $s_1, \cdots, s_m \in \{1, \cdots, k\}$, $m \geq 1$, such that

$$t_{s_1} t_{s_2} \cdots t_{s_m} = b_{s_1} b_{s_2} \cdots b_{s_m}.$$

Given $P = \left\{ \left[\frac{t_1}{b_1}\right], \cdots, \left[\frac{t_k}{b_k}\right] \right\}$, the problem is to determine if there is a match.

That is, $L_{pcp} = \{P = (t_1, b_1, \cdots, t_k, b_k) : P \text{ has a match}\}$

For example, $P = \left\{ \left[\frac{b}{ca}\right], \left[\frac{a}{ab}\right], \left[\frac{ca}{a}\right], \left[\frac{abc}{c}\right] \right\}$

$\left[\frac{a}{ab}\right]\left[\frac{b}{ca}\right]\left[\frac{ca}{a}\right]\left[\frac{a}{ab}\right]\left[\frac{abc}{c}\right]$

$\langle P \rangle \in L_{pcp}$ // $\langle P \rangle$: encoding of P

Thm 4.12 (Emil Post 1946) L_{pcp} is undecidable.

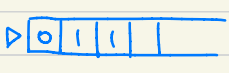
Proof. We "reduce" L_{accept} to L_{pcp}. Given a TM M and input w, construct a PCP instance $P = P(M, w)$ such that

M accept $w \Leftrightarrow \langle P \rangle \in L_{pcp}$.

Since L_{accept} is undecidable, so is L_{pcp}

WLOG, consider a single-tape one-way infinite TM.

Assume the behaviour of M satisfies the following:



放一个特殊符号，遇到就右移

1) M never attempts to move its head off the left-hand end of the tape.

2) If $w = \varepsilon$, use $\sqcup$ instead of w.

Modify L_{pcp} that any match is required to begin with the first domino $\left[\frac{t_1}{b_1}\right]$. Call it L_{mpcp}

$L_{accept} \longrightarrow L_{mpcp} \longrightarrow L_{pcp}$

Runtime configuration (snapshot)

tape: $x_1\ x_2 \cdots x_n \sqcup \cdots$ (head at x_1)

configuration: $x_1 q x_2 x_3 \cdots x_n$ (head at x_1) $\quad x_1 \cdots x_{i-1} q x_i$ (head at x_i)

状态q也是一种特殊的符号.

"computation is local"

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$

We construct an instance P (of PCP) s.t. P has a match iff. M accepts w. First, we construct instance P' of MPCP.

C: TM的snapshot

$\#C_1\#C_2\#\cdots\#C_T\#$

$\#C_1\#C_2\#\cdots\#C_T\#$

Part 1. (initial configuration) Put

$$\left[\frac{t_1}{b_1}\right] = \left[\frac{\#}{\#q_0 w_1 w_2 \cdots w_n \#}\right]$$

$\left[\frac{\#}{\#q_0 0 1 0\#}\right]\left[\frac{q_0 0}{1 q_7}\right]\left[\frac{1}{1}\right]\left[\frac{0}{0}\right]$

$\delta(q_0, 0) = (q_7, 1, R)$

☆ Part 2. (right move) For every $a, b \in \Gamma$ and every $q, r \in Q$, where $r \neq q_{reject}$, if $\delta(q, a) = (r, b, R)$, then put $\left[\frac{qa}{br}\right]$ into P'.

☆ Part 3. (Left move) For every $a, b, c \in \Gamma$ and every $q, r \in Q$, where $q \neq q_{reject}$, if $\delta(q, a) = (r, b, L)$, then put $\left[\frac{cqa}{rcb}\right]$ into P'. 当c: q在左边, 左移.

$\#q_0 0 1 0\#$

↓ 改变snapshot

$\#1q_7 1 0\#$

Part 4. (Copy) For every $a \in \Gamma$, put $\left[\frac{a}{a}\right]$ into P'. Put $\left[\frac{\#}{\#}\right]$ into P'.

$\#|q_0 0|1|0|\#|1|q_7|1|0|\#|$

$\#|q_0 0|1|0|\#|1|q_7|1|0|\#|q_8|1|0|\#|$

$\delta(q_0, 0) = (q_7, 1, R)$

$\delta(q_7, 1) = (q_8, 0, L)$ $c = 1$

$\frac{01\#}{01\#\ \sqcup\ \#}$

Part 5 (Simulate infinitely many blanks) Put $\left[\frac{\#}{\sqcup\#}\right]$ into P'.

Part 6 (Complete the accept configuration, eliminating one symbol at a time) For every $a \in \Gamma$, Put $\left[\frac{a\ q_{accept}}{q_{accept}}\right]$ and $\left[\frac{q_{accept}\ a}{q_{accept}}\right]$ into P'.

Part 7. Put $\left[\frac{q_{accept}\#}{\#}\right]$ into P'.

0 | q_{accept} | # 0 | q_{accept} # 0 q_{accept} # q_{accept} #

0 | q_{accept} | # 0 | q_{accept} # 0 q_{accept} # q_{accept} #

If M accepts w, then consider the runtime configurations $C_1, C_2, \cdots, C_T$. Then there is a match

$$\frac{\# C_1 \# C_2 \# \cdots \# C_T \# E_5 \# \cdots \# E_0 \#}{\# C_1 \# C_2 \# \cdots \# C_T \# E_5 \# \cdots \# E_0 \#}$$

"forcing"

If there is a match, then M accepts w.

(MPCP)

$$\left[\frac{\#}{\# C_1 \#}\right]\left[\frac{q a}{b r}\right] \overset{copy}{\cdots}\left[-\right]\left[-\right]\cdots = \frac{\# C_1 \#}{\# C_1 \# C_2 \#}$$

It suffices to prove, two adjacent configurations are valid.

Finally, we show how to convert MPCP P' to PCP P by a trick.

Let $u = u_1 u_2 \cdots u_n$ be a string of length n. Define

$$*u = * u_1 * u_2 * \cdots * u_n$$

$$u* = u_1 * u_2 * \cdots * u_n *$$

$$*u* = * u_1 * u_2 * \cdots * u_n *$$

If P' were $\left\{\left[\frac{t_1}{b_1}\right], \left[\frac{t_2}{b_2}\right], \cdots, \left[\frac{t_k}{b_k}\right]\right\}$,

then P be

$$\left\{\left[\frac{* t_1}{* b_1 *}\right], \left[\frac{* t_1}{b_1 *}\right], \left[\frac{* t_2}{b_2 *}\right], \cdots, \left[\frac{* t_k}{b_k *}\right], \left[\frac{* \diamond}{\diamond}\right]\right\}$$

又有第一张牌来一个symbol匹配.

Thus, $\langle P' \rangle \in L_{MPCP}$ iff. $\langle P \rangle \in L_{PCP}$. □

Turing 1936
Shannon 1948

4.4 Gödel's first incompleteness theorem

First-order Peano Arithmetic (PA) 一阶Peano算术：素数论命题

PA is a formal theory of natural numbers.

$\mathbb{N}$ formulated in first-order logic with equality.

$L_{PA} = \{0, S, +, \times, =\}$, where

1) 0 constant zero

2) S unary function symbol (successor) 后继

$S(0) = 1$, $S(S(S(S(0)))) = 4$

3) $+$, $\times$ binary function (addition and multiplication)

4) $=$ binary relation (equality)

Formulas of PA are first-order built from symbols, variables, logical connectives ($\neg, \vee, \wedge, \rightarrow$) and quantifiers ($\forall, \exists$) 量词

For example,

1) $5 = S(S(S(S(S(0)))))$

2) "x divides y" $y|x$.

$DIVIDES(x,y) \equiv (\exists z)(y = x \times z)$

3) "p is a prime"

$PRIME(p) \equiv (\forall x)(x = 1 \vee x = p \vee \neg DIVIDES(x, p))$ $\in \mathbb{N}$

4) Goldbach conjecture

$(\forall x)(x \geq 2 \rightarrow (\exists y)(\exists z)(2 \times x = y + z \wedge PRIME(y) \wedge PRIME(z))$ $2 = S(S(0))$.

Exercise.

1) n is even $(\exists x)(n = x + x)$

2) n is positive $(\exists x)(n = S(x))$

因为0不是任何数的后继.

3) n, m is coprime 互质

$x \neq 0 \wedge x \neq 1$

$(\forall x)(\text{"}x>1\text{"} \rightarrow ((\text{"}x|n\text{"} \rightarrow \text{"}x \nmid m\text{"}) \wedge (\text{"}x|m\text{"} \rightarrow \text{"}x \nmid n\text{"})))$

4) Lagrange's four-square thm.

$(\forall n)(\exists x, y, u, v)(n = x \times x + y \times y + u \times u + v \times v)$.

Thm 4.13 (Gödel first incompleteness theorem)

Any consistent formal system F containing Peano arithmetic has undecidable statements. That, there exists a formula ϕ such that neither ϕ nor $\neg\phi$ is provable in F.

Proof. Assuming the formal system F is complete, i.e., any statement can be proved or disproved, we will prove L_{pcp} is decidable, which would be a contradiction.

Let $P = \{(t_1, b_1), \cdots, (t_k, b_k)\}$ be a collection of dominos, where $t_i, b_i \in \Sigma^*$. WLOG, Let $\Sigma = \{1, 2\}$.

P has a matching if $\exists s_1, \cdots, s_m \in \{1, 2, \cdots, k\}$, s.t.

$$t_{s_1} t_{s_2} \cdots t_{s_m} = b_{s_1} b_{s_2} \cdots b_{s_m}. \quad (1)$$

Construct a formula φ_P such that $\langle P \rangle \in L_{pcp}$ iff. φ_P is true

We express "P has a matching" using a Peano arithmetic formula $\varphi = \varphi(P)$

After that we are done.

Consider the following TM: (that decides L_{pcp})

1) On input P, construct a formula φ s.t. $\langle P \rangle \in L_{pcp} \in$ iff φ is true.

2) For $L = 1, 2, \cdots$, enumerate all proofs π of length L. Check if

π is a proof of φ or $\neg\varphi$. Once found, return true or false.

View $t_i, b_i \in \{1,2\}^*$ as decimal numbers.

Then (1) is $\exists s_1, \cdots, s_m$:

$$\sum_{i=1}^{m} t_{s_i} \times 10^{\sum_{j=i+1}^{m} Len(t_{s_j})} = \sum_{i=1}^{m} b_{s_i} \times 10^{\sum_{j=i+1}^{m} Len(b_{s_j})} \quad (2)$$

$t_1 t_2 = b_1 b_2$

$t_1 \times 10^{Len(t_2)} + t_2 = b_1 \times 10^{Len(b_2)} + b_2$

Represent $Len(t) = L$.

$(\exists L)(10^L \le t \wedge 10^{L+1} > t)$ $\quad Len(2) = 1 \quad Len(221) = 3.$

Represent Σ in (2). Let $F(0) = 0$. $F(i) = F(i-1)\,10^{Len(t_{s_i})} + t_{s_i}$.

$(\exists F)(F(0) = 0 \wedge (\forall i)(i \le 0 \vee i > m \vee F(i) = F(i) \times 10^{Len(t_{s_i})} + t_{s_i}))$

RHS of (2) is

$(\exists G)(G(0) = 0 \wedge (\forall i)(i \le 0 \vee i > m \vee G(i) = G(i-1) \times 10^{Len(b_{s_i})} + b_{s_i}))$

The Chinese Remainder Theorem

Let $m_1, \cdots, m_n \in \mathbb{N}^+$ be pairwise coprime. For any given $a_1, \cdots, a_n \in \mathbb{N}$ the following equations have a solution:

$x \equiv a_1 \pmod{m_1}$

$\cdots$

$x \equiv a_n \pmod{m_n}$

How to represent a function $F: \{0, \cdots, m\} \to \mathbb{N}$?

Gödel's beta function. ☆ $\quad b = (m!)^l$

$\beta(a,b,i) = a \bmod (1 + (i+1) \times b)$ // can be expressed by first-order PA

Function F is represented as $f \in \mathbb{N}$. s.t.

$F(0) = \beta(f, b, 0)$

$F(1) = \beta(f, b, 1)$

$\cdots$

$F(m) = \beta(f, b, m)$

How to define exponentiation:

$pow(a,b,c)$ iff $c = a^b$.

Use $f \in \mathbb{N}$ to "encode" F

$F(0) = f \bmod (1+b)$

$F(1) = f \bmod (1+2b)$

$\vdots$

$F(m) = f \bmod (1+mb)$

$gcd(1+ib, 1+jb) = gcd((i-j)b, 1+jb) = 1.$

$1+b > \max F(i) \quad 0 \le i \le m.$

$b = (m!)^l$

$\mathrm{mod}(a,b,c) \equiv (\exists n)(a = b \times n + c \wedge c < b)$.

How to express "$n = 10^m$" ?

$F: \{0, \cdots, m\} \to \mathbb{N}$

$F(0) = 1,\ F(1) = 10,\ F(2) = 10^2, \cdots, F(m) = 10^m$

$(\forall i)\Big(\underbrace{(i = 0 \wedge F(0) = 1)}_{F(0)} \vee \underbrace{(i \geq 1 \wedge i \leq m \wedge F(i) = 10 \times F(i-1))}_{\text{recursive def.}}\Big)$ □