April 17, 2019

Diagonalization and Halting Problem.

Def^n: (Halting Problem)

$L_{Halt} = \{ \langle M, x \rangle$: M is a SJava program that halts on input $x \}$.

① L_{Halt} is recognizable.

```
bool recogHaltchecker (string M, string x){
    return interpreter (M, x);
}
```

// Code for universal SJava program.

Is this code correct?

(a) Yes (b) NO

```
bool recogHaltchecker (string M, string x){
    bool b = interpreter (M, x);
    return true;
}
```

// code for universal sJava program.

Thm: L_{Halt} is not decidable.

Detour:

Diagonalization.

Proof technique invented by Georg Cantor.

Def^n: S is countable if $\exists$ a bijection from S to $\mathbb{N}$ ← set of natural numbers.

Notation: $\{0,1\}^\infty$: set of infinite length binary strings.

Lemma: $\{0,1\}^\infty$ is not countable.

Pf: Let f be a bijection from $\{0,1\}^\infty$ to $\mathbb{N}$.

Let s_i be the string in $\{0,1\}^\infty$ thats mapped to i (by f).

s_1 0 1 1 0 0 1 1 - - - - -

s_2 1 0 1 1 1 0 0 - - -

s_3 1 1 1 0 0 0 0 - - -

s_4 0 0 0 1 - - - -

⋮

Define a string w, such that

$w(i)$: obtained by flipping i^{th} coordinate of S_i.

$w \in \{0,1\}^{\infty}$.

But $w \neq S_i$ for any i.

Contradiction $\Rightarrow$ f cannot exist.

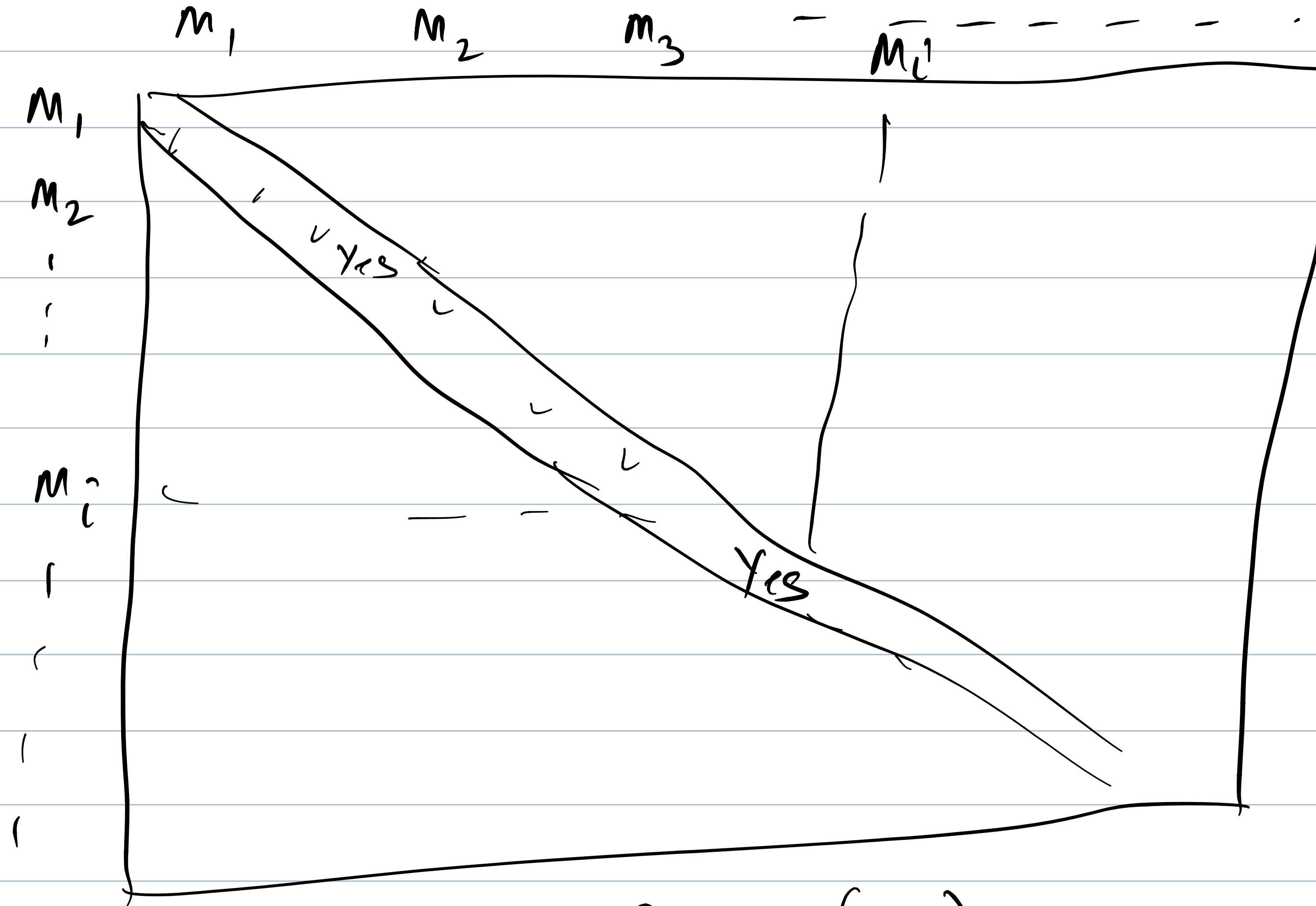
$\square$.

Thm: L_{Halt} is undecidable.

Pf:

Recall: any string w is an SJava program.

Rows, colums are strings.



Program (row)

Behavior of M_i given as input the source code of M_i?

Does M_i halt on input M_i?

Suppose L_{Halt} is decidable.

Let 'haltchecker' be a SJava program that decides it.

We can use:

halfchecker (M_i, M_i) ;

```
bool P( string w) {
  if ( haltcheeker (w, w)) {
      return infinite looper();
  }
  else {
      return true;
  }
}.

// code of haltchecker

bool infinite looper () {
  while (true ) {
```

}

}

<u>Claim:</u> $P \neq M_i$ for any i.

Pf:

If M_i halts on M_i,

P on input M_i goes into an infinite loop.

If M_i does not halt on M_i,

P halts on input M_i.

$\square$.

What's P's behavior on input P?

(1) If P halts on input P

$\Rightarrow$ haltchecker (P, P) is false

$\Rightarrow$ P loops infinitely on input P.

(ii) If P loops inf. on input P

$\Rightarrow$ haltchecker(P, P) is true

$\Rightarrow$ P halts on input P.

Contradiction. $\square$.