INT201 Decision, Computation and Language

Lecture 11 –Reducibility and Rice's Theorem

Dr Yushi Li and Dr Chunchuan Lyu

Xi'an Jiaotong-Liverpool University
西交利物浦大學

Recap

- Cantor's Diagonalization Method
- Church-Turing Thesis and Universal Turing Machine
- Examples of decidable languages
- Existence of undecidable language and non-Turing recognizable language

Today

- Reducibility
- The Halting Problem and other undecidable languages
- Rice's Theorem

John von Neumann 1903-1957

Hungarian-American mathematician, physicist, computer scientist, engineer and polymath,

- The axiomatic construction of general set theory (PhD thesis, 1925)
- Private communication to Kurt Gödel about his independent discovery of the second theorem of incompleteness (1930)
- Stopped working foundation of mathematics after 1931
- Axiomatization of Quantum Physics and Functional Analysis
- Mean Ergodic Theorem
- Minimax Theorem and Duality in Linear Programming
- Von Neumann Architecture, Merge Sort, Cellular Automata
- The Computer and the Brain (1958) and Singularity

"Young man, in mathematics you don't understand things. You just get used to them."

Undecidable Problems

Definition

不可判定问题 相关语言无法被TM识别并最终一定能处于halt状态.

Undecidable problem. The associated language of a problem cannot be recognized by a TM that halts for all inputs.

不可识别问题 无法被TM识别

Unrecognizable problem. The associated language of a problem cannot be recognized by a TM.

L_{TM} 是不可判定的,但是可识别的

The languages of TMs are undecidable but recognizable

$$L_{TM} = \{\langle M, w\rangle : M \text{ is a Turing machine that accepts the string } w\}$$

We consider undecidable problems unsolvable (informal).

一个TM如果运行了非常久也不停 可能是因为过程太长 也可能它一直在loop.

Say the TM has run 6 weeks without giving a response. It is possible that it is not in the language, but it is also possible it is in the language and we only need to wait longer.

decidable language 对于等待时间有上限

For decidable languages, there is an upper bound of the waiting time.
For undecidable languages, the waiting has no time limit.

+ in Z
a+b in Z => + in R
addition is harder in R than Z

Reducibility

L_TM {(M,w) | M is a TM and M accepts w}
L_TM => L′ then L′ also undecidable

Definition

将一个问题转换为另一个问题

Reduction is a way of converting one problem to another problem, so that the solution to the second problem can be used to solve the first problem.

B的解也可以解决A.

If A **reduces** to B, then any solution of B solves A (Reduction always involves two problems, A and B).

如果A reducts B，A不可能比B难

- If A is reducible to B, then A cannot be harder than B.

如果A reducts B并且B是 decidable，A同样decidable

- If A is reducible to B and B is decidable, then A is also decidable.

如果A reducts B 并且 A是 undecidable，那么B同样 undecidable

- If A is reducible to B and A is undecidable, then B is also undecidable.

归约性

Reducibility

A common strategy for proving that a language L is undecidable is by reduction method, proceeding as follows:

通常用来辅助证明语言L是不可判定的

Typical approach to show L is undecidable via reduction from A to L:

- Find a problem A known to be undecidable 找到一个已知的undecidable的问题A
- Suppose L is decidable. 假设语言L是 decidable的
- Let R be a TM that decides L. TM R 可以判定L
- Using R as subroutine to construct another TM S that decides A. 将R用作子进程来构造另一个决定A的TM S.
- But A is not decidable. 但是A是 undecidable的
- Conclusion: L is not decidable. 所以 L也是 undecidable.

Mapping Reduction 是归约的一种具体体现，要求问题A和问题B之间的关系通过一个函数f来构建 并且该f是可计算的转换

Mapping Reduction

Definition

Suppose that A and B are two languages

- A is defined over alphabet Σ_1^*, so $A \subseteq \Sigma_1^*$
- B is defined over alphabet Σ_2^*, so $B \subseteq \Sigma_2^*$

Then A is **mapping reducible** to B, written

$$A \leq_m B$$

$\leq_m$ 映射归约

if there is a computable (Turing Machine can simulate through the tape) function

$$f : \Sigma_1^* \to \Sigma_2^*$$

补 封闭 结果取反

such that, for every $w \in \Sigma_1^*$

并 封闭 同时运行两个TM，如果有一个接受就接受

当且仅当 <=>

$$w \in A \Longleftrightarrow f(w) \in B$$

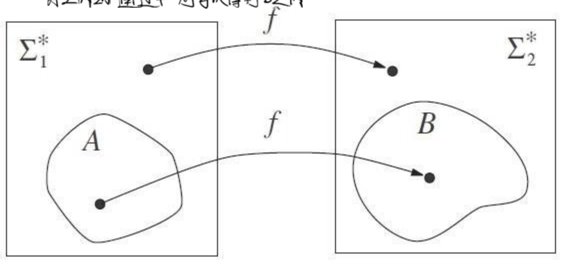
交 封闭 两个TM都接受

The function f is called a **reduction** of A to B.

映射归约

Mapping Reduction

A之外的内容通过f会得到B之外的内容
A之内的通过f则可以得到B之内

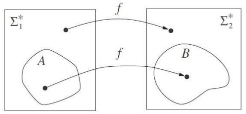


$$w \in A \Longleftrightarrow f(w) \in B$$

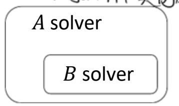
反之亦然

Mapping Reduction v.s. (General) Reduction

Mapping Reducibility of A to B:
Translate A-questions to B-questions.
可以将问题A通过可计算函数f转换为另一个问题B的实例



(General) Reducibility of A to B:
Use B solver to solve A.
如果B解法已知，可构造A解法
A的solve通过调用B的solver来判断A中实例是否被接受



$w \in A \Longleftrightarrow f(w) \in B$

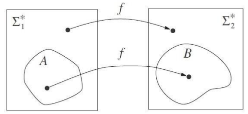
$A \leq_m B$

通过判断归约关系存在解决问题

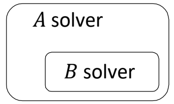
Clearly, we can use mapping reduction to construct general reduction.
By having A accepts w if B accepts f(w), rejects w if B rejects f(w) and loops if B loops on f(w).

Mapping Reduction v.s. (General) Reduction

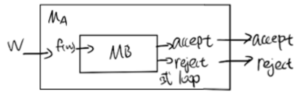
Mapping Reducibility of A to B:
Translate A-questions to B-questions.



(General) Reducibility of A to B:
Use B solver to solve A.



$w \in A \Longleftrightarrow f(w) \in B$



However, we have other options such as A accepts when B rejects, calling B multiple times or computing while conditioning on the B solution.

Theorem

If A $\leq_m$ B and B is decidable, then A is decidable.

Proof

- Let M_B be TM that decides B.
- Let f be reducing fcn from A to B.
- Consider the following TM:

  M_A = "On input w:
  1. Compute $f(w)$.
  2. Run M_B on input $f(w)$ and give the same result."
- Since f is a reducing function, $w \in A \iff f(w) \in B$.
  - If $w \in A$, then $f(w) \in B$, so M_B and M_A accept.
  - If $w \notin A$, then $f(w) \notin B$, so M_B and M_A reject.
- Thus, M_A decides A.

Corollary

If A $\leq_m$ B and A is undecidable, then B is undecidable also.

Theorem

If A $\leq_m$ B and B is Turing-recognizable, then A is Turing-recognizable.

Corollary

If A $\leq_m$ B and A is not Turing-recognizable, then B is not Turing-recognizable.

Halting problem for TMs is undecidable

L_{TM} (acceptance problem for TMs) is undecidable, where

$$L_{TM} = \{\langle M, w\rangle \mid \text{M is a TM and M accepts string w}\}$$

Define related problem:

$$HALT_{TM} = \{\langle M, w\rangle \mid \text{M is a TM and M halts on string w}\}$$

L_{TM} and $HALT_{TM}$ has same universe:

$$\Omega=\{\langle M, w\rangle \mid \text{M is TM, w is string}\}$$

Given a $\langle M, w\rangle \in \Omega$:

- if M halts on input w, then $\langle M, w\rangle \in HALT_{TM}$,
- if M doesn't halt on input w, then $\langle M, w\rangle \notin HALT_{TM}$.

$HALT_{TM}$ 给定一个图灵机M和一个输入w，判断M是否会在w上停机

Halting problem for TMs is undecidable

Proof by general reduction from L_{TM} L_{TM} reducts $HALT_{TM}$

We reduce L_{TM} to $HALT_{TM}$. We claim the following machine S decides L_{TM}

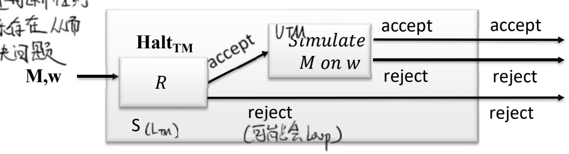
假设TM R判定$HALT_{TM}$ 则构造TM S判定L_{TM} S=在输入<M,w>上，其中M是TM，w是串

$S =$ "On input $\langle M, w\rangle$, an encoding of a TM M and a string w:

1. Run TM R on input $\langle M, w\rangle$. 在输入<M,w>上运行TM R
2. If R rejects, *reject*. 如果R拒绝，则拒绝
3. If R accepts, simulate M on w until it halts. 如果R接受，则在w上模拟M，直到halt
4. If M has accepted, *accept*; if M has rejected, *reject*." 如果M已接受，则接受；若M拒绝，则拒绝

Clearly, S accepts <M,w> iff M halts on w and M accept w.
S rejects, iff M loops on w or M halts and rejects w.

通过判断归约关系存在从而解决问题



如果S接受<M,w>，表明仅当M在w上停机且接受

如果M不停机，S无法进入模拟步骤，直接拒绝

Now, as L_{TM} is undecidable, so must $HALT_{TM}$

Halting problem for TMs is undecidable

假设$HALT_{TM}$ decidable：

Proof by contradiction 反证法

因此通过证明L_{TM}也是decidable的

Assume halting is decidable, we have that TM H, where

对于输入<M,w>

$$H(\langle M, w\rangle) = \begin{cases} \textit{accept} & \text{if } M \text{ halts on } w \\ \textit{reject} & \text{if } M \text{ does not halt on } w. \end{cases}$$

先判定M在w上是否停机

若M在w上不停机，则拒绝

若M在w上停机，则在w上模拟M

若M accept 则 accept

若M reject 则 reject.

Now, we use exactly the same strategy as in L_{TM} except we loop when we should reject, we have the following TM D:

$D =$ "On input $\langle M\rangle$, where M is a TM:

1. Run H on input $\langle M, \langle M\rangle\rangle$.
2. Output the opposite of what H outputs. That is, if H accepts, *loop*; and if H rejects, *accept*." 让D与H相反，如果H accept，D loop；如果H reject，D accept

We then ask, does D halts on <D>?
Clearly, if it halts, H accepts, then D loop.
If it does not halt, H rejects, then D accept (hence halt).
We have a contradiction.

空性问题，一个图灵机是否根本不接受任何串

Emptiness of TMs is undecidable

假设E_{TM}是可判定的，利用这个假设来证明L_{TM}是可判定的。输入<M,w>上，先把M修改为M_1，M_1拒绝除w以外所有其他输入，M_1在w上完全模拟M。

$$E_{TM} = \{< M > \mid M \text{ is a TM and } L(M) = \emptyset\}.$$

Intuitively, we need to show the Turing machine won't enter the accept state for any strings. However, unlike the CFG/DFA cases, we find it hard to enumerate over all possible derivations. 但是不能通过遍历所有例子的方法

然后判定$L(M_1)$是否为空
若$L(M_1)$为空，则 E_{TM} accept，L_{TM} reject
若不为空，则 E_{TM} reject，L_{TM} accept

Theorem: E_{TM} is undecidable.

Proof by reduction from : L_{TM}
For a given TM M and input w, we construct a TM M_w s.t. M accepts w iff $L(M_w) \neq \emptyset$.
We claim M_w = "On input x:

1. If $x \neq w$, *reject*.
2. If $x = w$, run M on input w and *accept* if M does."

If M accepts w, in step 1 M_w rejects all strings other than w, and in step 2, M_w accepts it, and hence $L(M_w) \neq \emptyset$.
If $L(M_w) \neq \emptyset$, then as it rejects all strings other than w, and accepts w only when M accepts w. Therefore M accepts w.

Emptiness of TMs is undecidable

$E_{TM} = \{< M > \mid M \text{ is a } TM \text{ and } L(M) = \emptyset\}$.

Mw Input x
If x=w
run M on w
If accept accept
elif reject reject
else loop loop
else x≠w
reject
if accept reject
elif reject reject
else loop loop

Theorem: E_{TM} is undecidable.

Proof by reduction from $\mathbf{L_{TM}}$:

M_w = "On input x:

1. If $x \neq w$, *reject*.
2. If $x = w$, run M on input w and *accept* if M does."

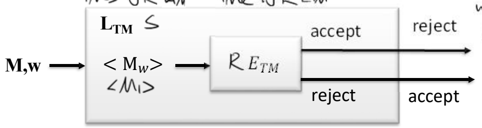
M accepts w iff $L(M_w) \neq \emptyset$

If M accepts w

=> L<Mw> not empty

Assume a decider for E_{TM} exists

TMS判定LTM TMe判定ETM



we want

L(<Mw>) not empty

也就是当 M accept w.

<=> Mw accepts only if M accepts w

But $\mathbf{L_{TM}}$ is undecidable, so must E_{TM}.

Other Undecidable Problems

Define T to be the set of all Turing machines descriptions, i.e.,

T = {⟨M⟩: M is a Turing machine}

- $E_{TM} = \{< M > \mid M \text{ is a } TM \text{ and } L(M) = \emptyset\}$.
- $INFINITE_{TM} = \{< M > \mid M \text{ is a } TM \text{ and } |L(M)| = \infty\}$.
- $LT_{TM} = \{< M > \mid M \text{ is a } TM \text{ and } L(M) = L(T)\}$.
- $FINITE_{TM} = \{< M > \mid M \text{ is a } TM \text{ and } \exists n \in N, |L(M)| = n\}$.
- $ALL_{TM} = \{< M > \mid M \text{ is a } TM \text{ and } L(M) = \Sigma^*\}$

Is it possible to prove them all at once?

Non-trivial Properties and Rice's Theorem

- $E_{TM} = \{< M > \mid M \text{ is a } TM \text{ and } L(M) = \emptyset\}$.
- $INFINITE_{TM} = \{< M > \mid M \text{ is a } TM \text{ and } |L(M)| = \infty\}$.
- $LT_{TM} = \{< M > \mid M \text{ is a } TM \text{ and } L(M) = L(T)\}$.
- $FINITE_{TM} = \{< M > \mid M \text{ is a } TM \text{ and } \exists n \in N, |L(M)| = n\}$.
- $ALL_{TM} = \{< M > \mid M \text{ is a } TM \text{ and } L(M) = \Sigma^*\}$

(Informal) Non-trivial properties in common:

1. none of them are empty set
2. none of them includes all Turing machines.
3. <M> ∈ P iff L(M) satisfy some properties,

   i.e., $P = \{< M > \mid M \text{ is a } TM$ and p(L(M))==1$\}$ *where* $p: \{L(M) \mid M \in T\} \to \{1,0\}$

(Informal) Rice's Theorem:

Any non-trivial property of Turing machines is undecidable

Rice's Theorem

非平凡性质P只适用于一部分图灵机

T是所有图灵机描述的集合 <M>代表图灵机M的编码

Define T to be the set of all Turing machines descriptions, i.e.,

T = {⟨M⟩: M is a Turing machine}

$P \neq \emptyset$, $P \neq \Sigma^*$

Let P be a subset of T such that

非空性质：P中至少包含一个图灵机的描述<M>.

1. $P \neq \emptyset$, i.e., there exists a Turing machine M such that ⟨M⟩ ∈ P,

P是T的真子集 P⊂T，P不能包含所有图灵机描述

2. P is a proper subset of T , i.e., there exists a Turing machine N such that ⟨N⟩ ∉ P,

3. for any two Turing machines M_1 and M_2 with $L(M_1) = L(M_2)$, 如果两台图灵机M_1M_2的语言相同

   (b) (a) either both ⟨M_1⟩ and ⟨M_2⟩ are in P or
   none of ⟨M_1⟩ and ⟨M_2⟩ is in P.

This is a more operational for checking the condition than $\exists\, p: \{L(M) \mid M \in T\} \to \{1,0\}$, such that ⟨M⟩ ∈ P iff $p(L(M)) = 1$.

Then the language P is undecidable 则(满足上述条件)

性质P仅与图灵机L(M)有关，而不是图灵机实现方式

Rice’s Theorem and Applications

Let P be a subset of T such that

1. $P \neq \emptyset$,
2. P is a proper subset of T ,
3. for any two Turing machines M_1 and M_2 with $L(M_1) = L(M_2)$,
   (a) either both $\langle M_1 \rangle$ and $\langle M_2 \rangle$ are in P or
   (b) none of $\langle M_1 \rangle$ and $\langle M_2 \rangle$ is in P.

$E_{TM} = \{< M > | M \text{ is a } TM \text{ and } L(M) = \emptyset\}$.

一个拒绝所有输入的TM就够了

1. A TM rejects all inputs will suffice is in the set.
2. A TM accepts all inputs is not in the set. 一个接受所有输入的TM不在集合中
3. If $L(M_1) = L(M_2)$, they are both either $\emptyset$ or non-empty, either both in the set or not in the set respectively.

如果$L(M_1)=L(M_2)$，要么都是$\emptyset$，要么都不为空，要么都在集合中，要么都不在

Therefore E_{TM} is undecidable by Rice’s theorem

Rice’s Theorem and Applications

Let P be a subset of T such that

1. $P \neq \emptyset$,
2. P is a proper subset of T ,
3. for any two Turing machines M_1 and M_2 with $L(M_1) = L(M_2)$,
   (a) either both $\langle M_1 \rangle$ and $\langle M_2 \rangle$ are in P or
   (b) none of $\langle M_1 \rangle$ and $\langle M_2 \rangle$ is in P.

$INFINITE_{TM} = \{< M > | M \text{ is a } TM \text{ and } |L(M)| = \infty\}$.

1. A TM accepts all inputs will suffice is in the set.
2. A TM rejects all inputs is not in the set.
3. If $L(M_1) = L(M_2)$, $|L(M_1)| = |L(M_2)|$ so they are both either infinite or finite, either both in the set or not in the set respectively.

Therefore $INFINITE_{TM}$ is undecidable by Rice’s theorem

Rice’s Theorem and Applications

Let P be a subset of T such that

1. $P \neq \emptyset$,
2. P is a proper subset of T ,
3. for any two Turing machines M_1 and M_2 with $L(M_1) = L(M_2)$,
   (a) either both $\langle M_1 \rangle$ and $\langle M_2 \rangle$ are in P or
   (b) none of $\langle M_1 \rangle$ and $\langle M_2 \rangle$ is in P.

$INFINITE_{TM} = \{< M > | M \text{ is a } TM \text{ and } |L(M)| = \infty\}$.

Note that there might be $\boldsymbol{M_1, M_2} \in \boldsymbol{INFINITE_{TM}}$ but $\mathbf{L(M_1) \neq L(M_2)}$

Rice’s Theorem and Non-Applications

补 封闭 结果取反

并封闭 同时运行两个TM，如果有一个接受就接受

交 封闭 两个TM都接受

Let P be a subset of T such that

1. $P \neq \emptyset$,
2. P is a proper subset of T ,
3. for any two Turing machines M_1 and M_2 with $L(M_1) = L(M_2)$,
   (a) either both $\langle M_1 \rangle$ and $\langle M_2 \rangle$ are in P or
   (b) none of $\langle M_1 \rangle$ and $\langle M_2 \rangle$ is in P.

$FIVE_{TM} = \{< M > | M \text{ is a } TM \text{ and } M \text{ has } 5 \text{ states}\}$.

We cannot apply Rice’s theorem here, as clearly we can have a UTM with 3 states recognize the same language as a given 5 states TM by simulating that machine.

In general, a property is about the language not about the TMs

性质不依赖于语言，而是与图灵机的结构相关

$\langle M\rangle$图灵机 M 的编码.

P是TM编码的集合 $S=\{\langle M_1\rangle, \langle M_2\rangle, \dots\}$ $\langle M_1\rangle \in P \wedge L(M_1)=L(M_2) \Rightarrow \langle M_2\rangle \in P$

Rice's Theorem and Proof

Rice's Theorem

$P\neq\emptyset$; $P\neq\Sigma^*$ (一定有至少一个TM N不满足P $\langle N\rangle\notin P$)

Let P be a proper non-empty subset of TM descriptions such that for M_1 and M_2 with $L(M_1) = L(M_2)$, $\langle M_1\rangle \in P$ iff $\langle M_2\rangle \in P$. Then, P is undecidable.

两者接受语言相同，那 $\langle M_1\rangle\in P \Longleftrightarrow \langle M_2\rangle\in P$，则 P是不可判定的

Proof attempt by reduction from L_{TM}:

将L_{TM}归约到一个非平凡的问题上 (例如P)

We will reduce L_{TM} to all non-trivial property problems. Denote any given problem P, we have a **decider R_P** and a Turing machine **T such that <T> $\in$ P**. For a given TM M and input w, ideally we construct a TM M_w s.t. M accepts w iff $<M_w> \in P$.

$M_w^{(T)}$ = "On input x:
1. Simulate M on w. If it halts and rejects, *reject*. If it accepts, proceed to stage 2.
2. Simulate T on x. If it accepts, *accept*."

Rice's Theorem and Proof

Proof attempt by reduction from L_{TM}: $\boxed{M \text{ accepts w} \rightarrow\ <M_w(T)> \in P.}$

$M_w^{(T)}$ = "On input x:
1. Simulate M on w. If it halts and rejects, *reject*. If it accepts, proceed to stage 2.
2. Simulate T on x. If it accepts, *accept*."

If M accepts w, we have $L(M_w(T))=L(T)$. As <T> $\in P$, we have $<M_w(T)> \in P$.

If M rejects or loops on w, we have $L(M_w(T))= \emptyset$. Now $<M_w(T)> \notin P$ iff Turing machines with empty language is not in the property.

Obviously, this is not true for all property.

Can we do something differently when empty language is not in the property?

Rice's Theorem and Proof

Proof by reduction from L_{TM}:

$M_w^{(T)}$ = "On input x:
1. Simulate M on w. If it halts and rejects, *reject*. If it accepts, proceed to stage 2.
2. Simulate T on x. If it accepts, *accept*."

Take any P, let $M_\emptyset$ be a TM such that $L(M_\emptyset) = \emptyset$. If $<M_\emptyset> \in P$, we use its' complement $\overline{P}$.

Obviously, P is a decider iff $\overline{P}$ is a decider.

Moreover, there always exists $<\overline{T}> \in \overline{P}$ as P is a proper subset.

If M accepts w, we have $L(M_w(\overline{T}))=L(\overline{T})$. As $<\overline{T}> \in \overline{P}$, we have $<M_w> \in \overline{P}$.

If M rejects or loops w, we have $L(M_w(\overline{T}))= \emptyset$. As we have $<M_\emptyset> \in P$, so $<M_w(\overline{T})> \notin \overline{P}$

So, we have if $<M_\emptyset> \in P$, M accepts w iff $<M_w(\overline{T})> \in \overline{P}$

Rice's Theorem and Proof

Proof by reduction from L_{TM}:

$M_w^{(T)}$ = "On input x:
1. Simulate M on w. If it halts and rejects, *reject*. If it accepts, proceed to stage 2.
2. Simulate T on x. If it accepts, *accept*."

In all, we have the following results:

If $<M_\emptyset> \notin P$, M accepts w iff $<M_w(T)> \in P$.

If $<M_\emptyset> \in P$, M accepts w iff $<M_w(\overline{T})> \in \overline{P}$

在w上运行M，分情况讨论，如果在w上M上运行，或者w在M上不被接受

Rice’s Theorem and Proof

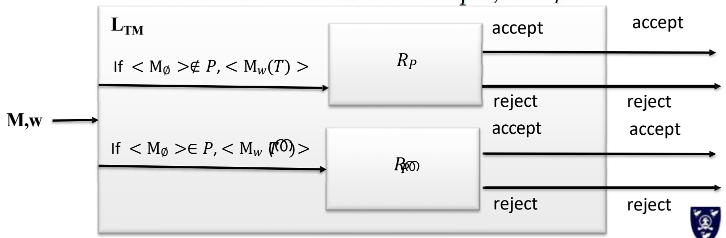
If $< M_\emptyset > \notin P$, M accepts w iff $< M_w(T) > \in P$.

If $< M_\emptyset > \in P$, M accepts w iff $< M_w(\overline{T}) > \in \overline{P}$

Proof by reduction from L_{TM}:

$M_w^{(T)}$ = “On input x:

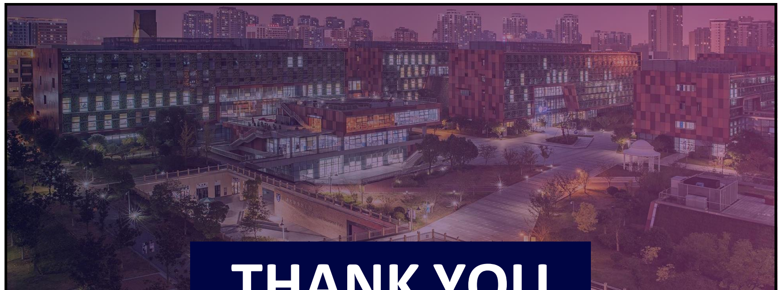
1. Simulate M on w. If it halts and rejects, *reject*. If it accepts, proceed to stage 2.
2. Simulate T on x. If it accepts, *accept*.”



Now, as L_{TM} is undecidable, so must all the non-trivial properties.

Quick review

- Reducibility shows a problem can only be reduced to problems that is at least as hard as itself.
- The Halting Problem and other undecidable languages
- Rice’s Theorem states non-trivial properties is undecidable



THANK YOU