

# Undecidability

Lecture 25

# Today

Undecidable Problems

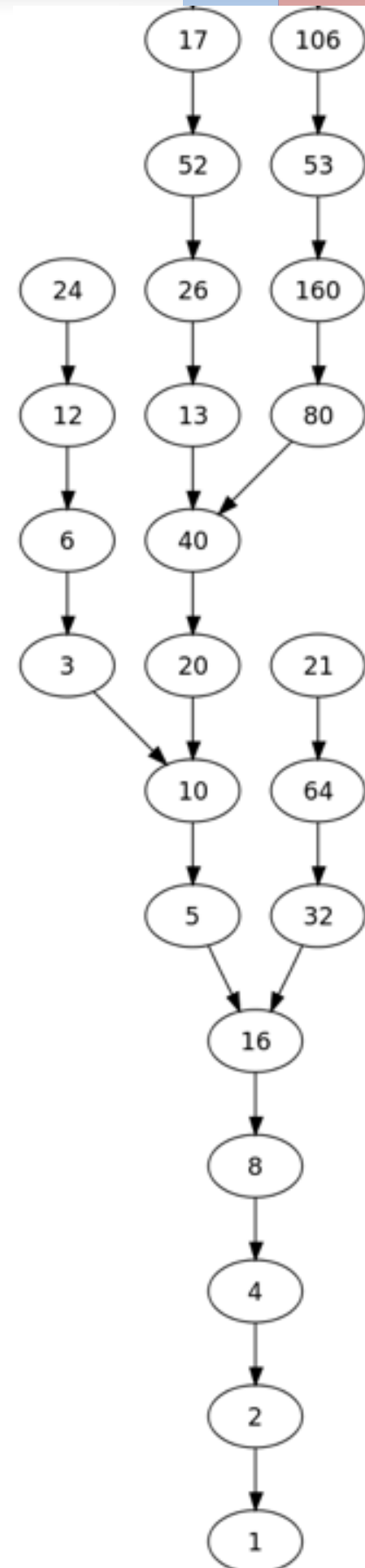
Proving undecidability

Using reductions to prove more undecidability



# A Puzzle

```
Program Collatz (n:integer)
  while n > 1 {
    if Even(n) then n := n/2
    else n := 3n+1
  }
```



**Q:** Is there  $n$  that makes this program run forever?

**Conjecture:** Collatz( $n$ ) halts for every  $n > 0$

We know that up to numbers with several hundred digits ( $10^{120}$ ) it halts.

# Can we write a program?



Could ask: isn't there a program that can help us decide if it is possible that it gets stuck in infinite loop?

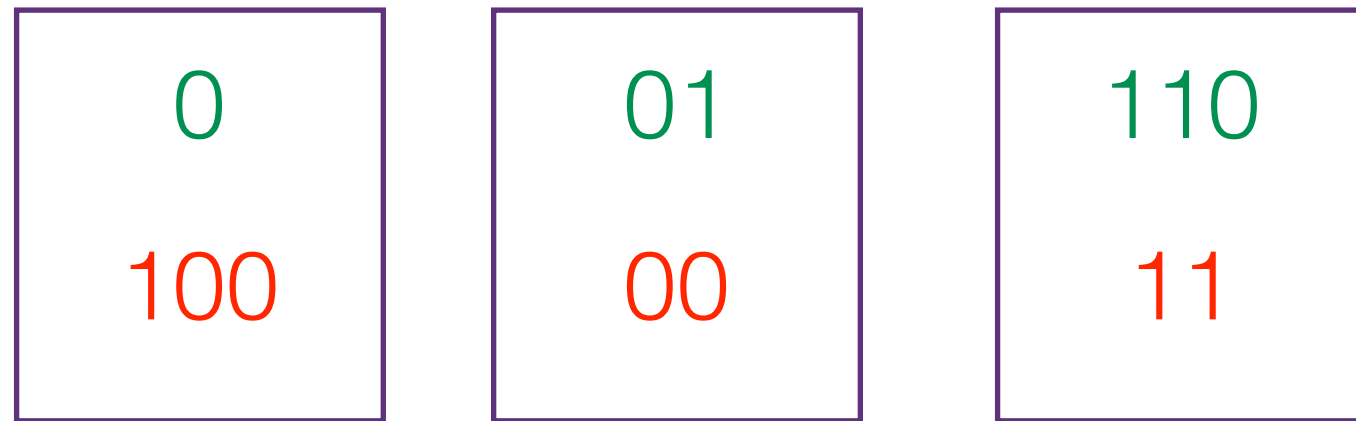
There is no program that takes input an arbitrary piece of code and report correctly if it halts or not!

There is no automatic/systematic way to answer that question in general, but maybe that question has a particular answer.



# Another Puzzle

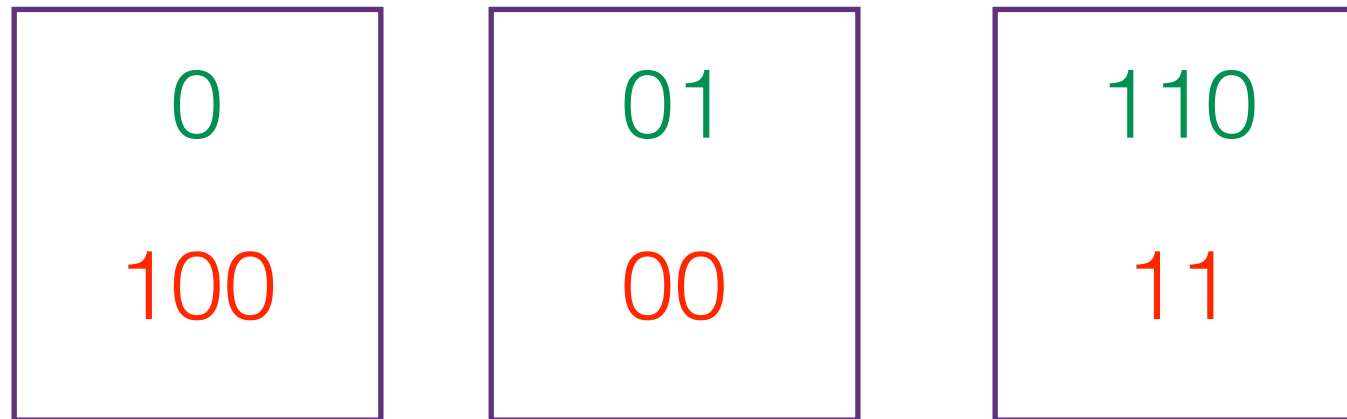
Three stacks of cards, all cards in a stack identical. Two strings written on each card, one in green and one in red.



I want to choose a sequence of cards, each from a different stack so that when I line up the strings in green and red they are the same string.

# Another Puzzle

There is  
solution!



I want to choose a sequence of cards, each from a different stack so that when I line up the strings in green and red they are the same string.

# Another Puzzle



There is solution that uses 451 cards!

I want to choose a sequence of cards, each from a different stack so that when I line up the strings in green and red they are the same string.

# Can we write a program?



Could ask: isn't there a program that can help us decide if there is solution?

Input is finite set of pairs of strings ( $2n$ )

Yes/No Question

There is no such algorithm!

Post correspondance problem has no solution!

# How to show there is no algorithm?

- A TM on input a string  $w$  can : accept  $w$ , reject  $w$ , diverge on  $w$  (run forever)

Language of a TM  $T$ :

$$ACCEPT(T) = \{ w \mid T \text{ accepts } w \}$$

$$REJECT(T) = \{ w \mid T \text{ rejects } w \}$$

$$HALT(T) = \{ w \mid T \text{ halts on } w \}$$

$$DIVERGE(T) = \{ w \mid T \text{ diverges on } w \}$$

- We say  $T$  accepts  $L$  iff  $L=ACCEPT(T)$
- What happens to the strings that are not accepted?
- We say  $T$  decides  $L$  iff  $L=ACCEPT(T)$  and  $DIVERGE(T)=\emptyset$

# Decidability

- A language  $L$  is decidable iff some TM  $T$  decides  $L$
- A language  $L$  is acceptable iff some TM  $T$  accepts  $L$
- A language  $L$  is undecidable iff no TM  $T$  decides  $L$
- A language  $L$  is unacceptable iff no TM  $T$  accepts  $L$
- TM decides a language = algorithm
- if TM does not halt in some inputs its not really an algorithm
- Getting the YES answers right is not enough!



# No algorithm for a problem?

- There is no algorithm for a problem means the language associated with this problem is undecidable.
- E.g. Post correspondence problem: input is a string. Output is YES/NO. Language of all the instances where the answer is yes is a language  $L$ . No solution means  $L$  is not decidable.
- Same for the language  $L$  of programs that never enter an infinite loop.  $L$  is undecidable.
- Not hard to show that there are undecidable languages (uncountable number of languages, countable number of TM).



# Example of Undecidable Language

$$SELFREJECT = \{ \langle M \rangle \mid M \text{ rejects } \langle M \rangle \}$$

$M$  = Turing Machine (piece of executable code)

$\langle M \rangle$  = encoding of  $M$  as a string (source code for  $M$ )

$\langle M \rangle$  is what you would feed to a universal TM, that would allow it to simulate  $M$ .

(e.g. TM that rejects everything. TM that rejects every description of a TM are in that language)





# Example of Undecidable Language

$$SELFREJECT = \{ \langle M \rangle \mid M \text{ rejects } \langle M \rangle \}$$

**Claim:** *SELFREJECT* is undecidable

Proof:

Suppose (towards contradiction) that there is a TM SR that decides *SELFREJECT*.

What happens if we feed  $\langle SR \rangle$  to SR?

- SR accepts  $\langle SR \rangle$
- SR rejects  $\langle SR \rangle$
- SR diverges on  $\langle SR \rangle$



# Example of Undecidable Language

$$SELFREJECT = \{ \langle M \rangle \mid M \text{ rejects } \langle M \rangle \}$$

**Claim:** *SELFREJECT* is undecidable

Proof:

Suppose (towards contradiction) that there is a TM SR that decides SELFREJECT.

What happens if we feed  $\langle SR \rangle$  to SR?

- SR accepts  $\langle SR \rangle$  :  $\langle SR \rangle$  is in SELFREJECT = SR rejects  $\langle SR \rangle$ !
- SR rejects  $\langle SR \rangle$  :  $\langle SR \rangle$  not in SELFREJECT = SR accepts or diverges
- SR diverges on  $\langle SR \rangle$  : impossible because this TM makes decisions about every string!

# Russel's barbers paradox



- On a small town, on a certain day everyone gets a haircut
- Everyone either cuts their own hair or has their hair cut by someone else.
- Barber = cuts your hair iff you don't
- You get a barber's license if you can prove that everyone in town either cuts their own hair or you cut it
- There is no barber

# Russel's barbers paradox



- On a small town, on a certain day everyone gets a haircut
- Everyone either cuts their own hair or has their hair cut by someone else.



Every TM

$A$  cuts  $B$ 's hair =  $A$  accepts  $\langle B \rangle$

- Barber = cuts your hair iff you don't

Barber is TM that accepts  $\langle M \rangle$  iff  $M$  does not accept  $\langle M \rangle$

- You get a barber's license if you can prove that everyone in town either cuts their own hair or you cut it

- There is no barber

# Cantor

Let  $X$  be any set and  $2^X$  its powerset (Set of subsets)

$f: X \rightarrow 2^X$  is NOT onto

Meaning there is a set  $S$  in  $2^X$  that has no preimage.

- $x$  is SAD if  $x$  is not in  $f(x)$
- $Y = \{x \text{ in } X \mid x \text{ is SAD}\}$
- There is no  $y$  in  $X$  such that  $f(y) = Y$
- maps to barber paradox ( $X$  is the set of people,  $f(x)$  is the set of people whose hair  $x$  cuts). A person is sad if they don't cut their own hair
- maps to SELFREJECT :  $X$  is the set of TM that halt on all inputs.  $f(x)$  is the set of all TM that  $x$  accepts. A TM is sad if it rejects its own encoding



# Cantor's Diagonal Slash



0 1 0 0 1 1 1 . .

Is the set of all infinitely long binary strings countable?

Suppose it was: consider *enumerating* them in a table

Consider the string corresponding to the “flipped diagonal”

It doesn't appear in this table!

$S_i$									
$S_1 =$	1	0	0	1	0	0	0	0	1
$S_2 =$	0	0	1	0	1	0	0	1	1
$S_3 =$	1	1	1	1	1	1	1	0	0
$S_4 =$	1	1	0	1	0	1	0	1	1
$S_5 =$	1	1	0	0	0	0	1	0	0
$S_6 =$	0	0	0	0	0	0	1	1	0
$S_7 =$	0	1	0	1	0	1	0	1	1



# Showing Undecidability



To show  $L$  is undecidable, reduce some undecidable language to  $L$

$$SELFHALT = \{ \langle M \rangle \mid M \text{ halts on } \langle M \rangle \}$$

**Claim:**  $SELFHALT$  is undecidable

More general looking problem:

$$HALT = \{ \langle M, w \rangle \mid M \text{ halts on } w \}$$

**Claim:**  $HALT$  is acceptable

The halting problem

**Claim:**  $HALT$  is undecidable