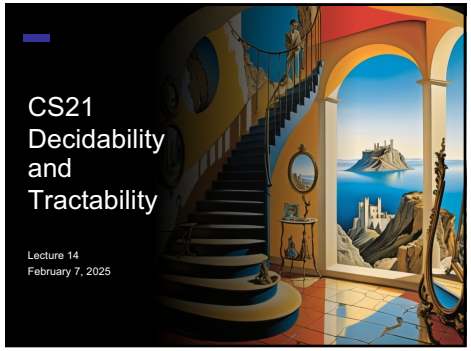


1

Outline

- undecidable problems
  - computation histories
  - surprising contrasts between decidable/undecidable
- Rice's Theorem
- Post Correspondence Problem (skip?)
- Beyond RE and co-RE
- Recursion Theorem

2

Dec. and undec. problems

Theorem: ALL_{CFG} is undecidable.

Proof:
- reduce from co-A_{TM} (i.e. show co-$A_{TM} \leq_m ALL_{CFG}$)
- what should f(<M, w>) produce?
- Idea:
  - produce CFG G that generates all strings that are ***not*** accepting computation histories of M on w

3

Dec. and undec. problems

Proof:
- build a NPDA, then convert to CFG
- want to accept strings not of this form,

$$\#C_1\#C_2\#C_3\#\ldots\#C_k\#$$

plus strings of this form but where
- C_1 is not the start config. of M on input w, or
- C_k is not an accept. config. of M on input w, or
- C_i does not yield in one step C_{i+1} for some i

4

Dec. and undec. problems

Proof:
- our NPDA nondeterministically checks one of:
  - C_1 is not the start config. of M on input w, or
  - C_k is not an accept. config. of M on input w, or
  - C_i does not yield in one step C_{i+1} for some i
  - input has fewer than two #'s
- details of first two?
- to check third condition:
  - nondeterministically guess C_i starting position
  - how to check that C_i doesn't yield in 1 step C_{i+1} ?

5

Dec. and undec. problems

Proof:
- checking:
  - C_i does not yield in one step C_{i+1} for some i
- push C_i onto stack
- at #, start popping C_i and compare to C_{i+1}
  - accept if mismatch away from head location, or
  - symbols around head changed in a way inconsistent with M's transition function.
- is everything described possible with NPDA?

6

Dec. and undec. problems

Proof:

– Problem: cannot compare C_i to C_{i+1}

– could prove in same way that proved {ww: w ∈ Σ*} not context-free

– recall that {ww^R: w ∈ Σ*} is context-free

– free to tweak construction of G in the reduction

– solution: write computation history:

$\#C_1\#C_2^R\#C_3\#C_4^R...\#C_k\#$

7

Dec. and undec. problems

Proof:

– f(<M, w>) = <G> equiv. to NPDA below:

on input x, accept if not of form: $\#C_1\#C_2^R\#C_3\#C_4^R...\#C_k\#$

• accept if C_1 is the not the start configuration for M on input w

• accept if check that C_i does not yield in one step C_{i+1}

• accept if C_k is not an accepting configuration for M

• is f computable?

• YES maps to YES?

<M, w> ∈ co-A_{TM} ⇒ f(M, w) ∈ ALL_{CFG}

• NO maps to NO?

<M, w> ∉ co-A_{TM} ⇒ f(M, w) ∉ ALL_{CFG}

8

Rice's Theorem

- We have seen that the following properties of TM's are undecidable:
  - TM accepts string w
  - TM halts on input w
  - TM accepts the empty language
  - TM accepts a regular language
- Can we describe a single generic reduction for all these proofs?
- Yes. *Every* property of TMs undecidable!

9

Rice's Theorem

- A TM property is a language P for which
  – if $L(M_1) = L(M_2)$ then $<M_1> \in P$ iff $<M_2> \in P$
- TM property P is nontrivial if
  – there exists a TM M_1 for which $<M_1> \in P$, and
  – there exists a TM M_2 for which $<M_2> \notin P$.

Rice's Theorem: Every nontrivial TM property is undecidable.

10

Rice's Theorem

- The setup:
  – let $T_\emptyset$ be a TM for which $L(T_\emptyset) = \emptyset$
    - technicality: if $<T_\emptyset> \in P$ then work with property co-P instead of P.
    - conclude co-P undecidable; therefore P undec. due to closure under complement
  – so, WLOG, assume $<T_\emptyset> \notin P$
  – non-triviality ensures existence of TM M_1 such that $<M_1> \in P$

11

Rice's Theorem

Proof:

– reduce from A_{TM} (i.e. show $A_{TM} \leq_m P$)

– what should f(<M, w>) produce?

– f(<M, w>) = <M'> described below:

on input x,

• accept iff M accepts w and M_1 accepts x

(intersection of two RE languages)

• f computable?

• YES maps to YES?

<M, w> ∈ A_{TM} ⇒ $L(f(M, w)) = L(M_1)$ ⇒ f(M, w) ∈ P

12

Rice's Theorem

Proof:

– reduce from A_{TM} (i.e. show $A_{TM} \leq_m P$)

– what should f(<M, w>) produce?

– f(<M, w>) = <M'> described below:

on input x,

• accept iff M accepts w and M_1 accepts x

(intersection of two RE languages)

• NO maps to NO?

$\langle M, w\rangle \notin A_{TM} \Rightarrow L(f(M, w)) = L(T\emptyset) \Rightarrow f(M, w) \notin P$

13

Post Correspondence Problem

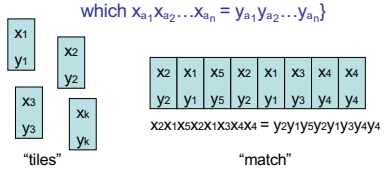
- many undecidable problems unrelated to TMs and automata
- classic example: Post Correspondence Problem

PCP = {<$(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$> : $x_i, y_i \in \Sigma^*$ and there exists $(a_1, a_2, \ldots, a_n)$ for which $x_{a_1}x_{a_2}\ldots x_{a_n} = y_{a_1}y_{a_2}\ldots y_{a_n}$}

14

Post Correspondence Problem

PCP = {<$(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$> : $x_i, y_i \in \Sigma^*$ and there exists $(a_1, a_2, \ldots, a_n)$ for which $x_{a_1}x_{a_2}\ldots x_{a_n} = y_{a_1}y_{a_2}\ldots y_{a_n}$}



15

Post Correspondence Problem

Theorem: PCP is undecidable.

Proof:

– reduce from A_{TM} (i.e. show $A_{TM} \leq_m$ PCP)

– two step reduction makes it easier

– first, show $A_{TM} \leq_m$ MPCP

(MPCP = "modified PCP")

– next, show MPCP $\leq_m$ PCP

16

Post Correspondence Problem

MPCP = {<$(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$> : $x_i, y_i \in \Sigma^*$ and there exists $(a_1, a_2, \ldots, a_n)$ for which $x_1x_{a_1}x_{a_2}\ldots x_{a_n} = y_1y_{a_1}y_{a_2}\ldots y_{a_n}$}

Proof of MPCP $\leq_m$ PCP:

– notation: for a string $u = u_1u_2u_3\ldots u_m$

- $\star u$ means the string $\star u_1\star u_2\star u_3\star u_4\ldots\star u_m$
- $u\star$ means the string $u_1\star u_2\star u_3\star u_4\ldots\star u_m\star$
- $\star u\star$ means the string $\star u_1\star u_2\star u_3\star u_4\ldots\star u_m\star$

17

Post Correspondence Problem

Proof of MPCP $\leq_m$ PCP:

– given an instance $(x_1, y_1), \ldots, (x_k, y_k)$ of MPCP

– produce an instance of PCP:

$(\star x_1, \star y_1\star), (\star x_1, y_1\star), (\star x_2, y_2\star), \ldots, (\star x_k, y_k\star), (\star \square, \square)$

– YES maps to YES?

- given a match in original MPCP instance, can produce a match in the new PCP instance

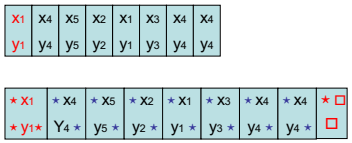
– NO maps to NO?

- given a match in the new PCP instance, can produce a match in the original MPCP instance

18

Post Correspondence Problem

– YES maps to YES?
- given a match in original MPCP instance, can produce a match in the new PCP instance

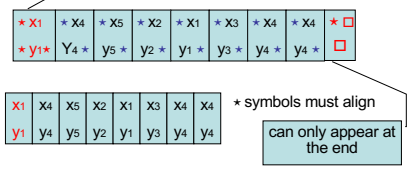


19

Post Correspondence Problem

– NO maps to NO?
- given a match in the new PCP instance, can produce a match in the original MPCP instance

can't match unless start with this tile



⋆ symbols must align

can only appear at the end

20

Post Correspondence Problem

Theorem: PCP is undecidable.

Proof:

– show $A_{TM} \leq_m$ MPCP

MPCP = $\{<(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)> :$ $x_i, y_i \in \Sigma^*$ and there exists $(a_1, a_2, \ldots, a_n)$ for which $x_1 x_{a_1} x_{a_2} \ldots x_{a_n} = y_1 y_{a_1} y_{a_2} \ldots y_{a_n}\}$

– show MPCP $\leq_m$ PCP

21

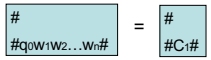
Post Correspondence Problem

Proof of $A_{TM} \leq_m$ MPCP:

– given instance of A_{TM}: <M, w>

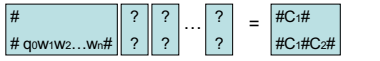
– idea: a match will record an accepting computation history for M on input w

– start tile records starting configuration:
- add tile (#, #$q_0 w_1 w_2 w_3 \ldots w_n$#)



22

Post Correspondence Problem



– tiles for head motions to the right:
- for all a,b ∈ Γ and all q, r ∈ Q with q ≠ q_{reject}, if δ(q, a) = (r, b, R), add tile (qa, br)

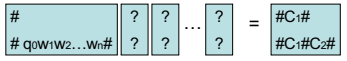
qa
br

– tiles for head motions to the left:
- for all a,b,c ∈ Γ and all q, r ∈ Q with q ≠ q_{reject}, if δ(q, a) = (r, b, L), add tile (cqa, rcb)

cqa
rcb

23

Post Correspondence Problem



– tiles for copying (not near head)
- for all a ∈ Γ, add tile (a, a)

a
a

– tiles for copying # marker
- add tile (#, #)

#
#

– tiles for copying # marker and adding _ to end of tape
- add tile (#, _#)

#
_#

24

Post Correspondence Problem

# / #$uaq_{accept}v$#	? / ?	…	? / ?	= #$uaq_{accept}v$# / #$uaq_{accept}v\#uq_{accept}v$#

– tiles for deleting symbols to left of q_{accept}
- for all $a \in \Gamma$, add tile (aq_{accept}, q_{accept})

aq_{accept}
q_{accept}

25

Post Correspondence Problem

# / #$q_{accept}av$#	? / ?	…	? / ?	= #$q_{accept}av$# / #$q_{accept}av\#q_{accept}v$#

– tiles for deleting symbols to right of q_{accept}
- for all $a \in \Gamma$, add tile ($q_{accept}a$, q_{accept})

$q_{accept}a$
q_{accept}

26

Post Correspondence Problem

# / #q_{accept}#	? / ?	…	? / ?	= #q_{accept}## / #q_{accept}##

– tiles for completing the match
- for all $a \in \Gamma$, add tile (q_{accept}##, #)

q_{accept}##
#

27

Post Correspondence Problem

– YES maps to YES?
- by construction, if M accepts w, there is a way to assemble the tiles to achieve this match:

$\#C_1\#C_2\#C_3\#...\#C_m\#$
$\#C_1\#C_2\#C_3\#...\#C_m\#$

where $\#C_1\#C_2\#C_3\#...\#C_m\#$ is an accepting computation history

– NO maps to NO?
- sketch: at any step if the "intended" next tile is not used, then it is impossible to recover and produce a match in the end (case analysis)

28

Post Correspondence Problem

We have proved:

Theorem: PCP is undecidable.

by showing:
– $A_{TM} \leq_m$ MPCP
– MPCP $\leq_m$ PCP
– conclude $A_{TM} \leq_m$ PCP

29

Beyond RE and co-RE

- We saw (by a counting argument) that there is *some* language that is neither RE or co-RE.
- We will prove this for a natural language:
  $EQ_{TM} = \{\langle M_1, M_2\rangle : L(M_1) = L(M_2)\}$
- Recall:
  – A_{TM} is undecidable, but RE
  – co-A_{TM} is undecidable, but coRE

Therefore, not in co-RE

Therefore, not in RE

30

Beyond RE and co-RE

Theorem: EQ_{TM} is neither RE nor coRE.

Proof:

– not RE:
- reduce from co-A_{TM} (i.e. show co-$A_{TM} \leq_m EQ_{TM}$)
- what should f(<M, w>) produce?

– not co-RE:
- reduce from A_{TM} (i.e. show $A_{TM} \leq_m EQ_{TM}$)
- what should f(<M, w>) produce?

31

Beyond RE and co-RE

Proof ($A_{TM} \leq_m EQ_{TM}$)

– f(<M, w>) = <M_1, M_2> described below:

TM M_1: on input x,
- accept

TM M_2: on input x,
- simulate M on input w
- accept if M accepts w

•YES maps to YES?

<M, w> $\in A_{TM} \Rightarrow L(M_1) = \Sigma^*$ and $L(M_2) = \Sigma^* \Rightarrow$ f(<M, w>) $\in EQ_{TM}$

• NO maps to NO?

<M, w> $\notin A_{TM} \Rightarrow L(M_1) = \Sigma^*$ and $L(M_2) = \emptyset \Rightarrow$ f(<M, w>) $\notin EQ_{TM}$

32

Beyond RE and co-RE

Proof (co-$A_{TM} \leq_m EQ_{TM}$)

– f(<M, w>) = <M_1, M_2> described below:

TM M_1: on input x,
- reject

TM M_2: on input x,
- simulate M on input w
- accept if M accepts w

•YES maps to YES?

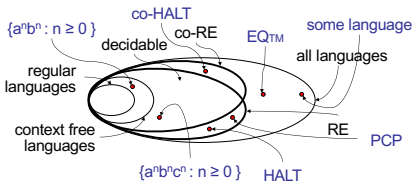
<M, w> $\in$ co-A_{TM} $\Rightarrow L(M_1) = \emptyset$ and $L(M_2) = \emptyset \Rightarrow$ f(<M, w>) $\in EQ_{TM}$

• NO maps to NO?

<M, w> $\notin$ co-A_{TM} $\Rightarrow L(M_1) = \emptyset$ and $L(M_2) = \Sigma^* \Rightarrow$ f(<M, w>) $\notin EQ_{TM}$

33

Summary



34