CS21
Decidability and Tractability

Lecture 12

February 2, 2018

Outline

- undecidable problems
  - computation histories
  - surprising contrasts between decidable/undecidable
- Rice's Theorem
- Post Correspondence problem
- a non-RE and non-co-RE language
- the Recursion Theorem

Dec. and undec. problems

- the boundary between decidability and undecidability is often quite delicate
  - seemingly related problems
  - one decidable
  - other undecidable

- We will see two examples of this phenomenon next.

Computation histories

- Recall configuration of a TM: string uqv with u,v $\in \Gamma^*$, q $\in$ Q
- The sequence of configurations M goes through on input w is a computation history of M on input w
  - may be *accepting*, or *rejecting*
  - reserve the term for halting computations
  - nondeterministic machines may have several computation histories for a given input.

Linear Bounded Automata

LBA definition: TM that is prohibited from moving head off right side of input.

- machine prevents such a move, just like a TM prevents a move off left of tape

- How many possible configurations for a LBA M on input w with |w| = n, m states, and p = $|\Gamma|$?
  - counting gives: mnp^n

Dec. and undec. problems

- two problems we have seen with respect to TMs, now regarding LBAs:
  - LBA acceptance:

    A_{LBA} = {<M, w> : LBA M accepts input w}
  - LBA emptiness:

    E_{LBA} = {<M> : LBA M has L(M) = Ø}
- Both decidable? both undecidable? one decidable?

Dec. and undec. problems

Theorem: A_{LBA} is decidable.

Proof:

- input <M, w> where M is a LBA
- key: only mnp^n configurations
- if M hasn't halted after this many steps, it must be looping forever.
- simulate M for mnp^n steps
- if it halts, accept or reject accordingly,
- else reject since it must be looping

Dec. and undec. problems

Theorem: E_{LBA} is undecidable.

Proof:

- reduce from co-A_{TM} (i.e. show co-$A_{TM} \leq_m E_{LBA}$)
- what should f(<M, w>) produce?
- Idea:
  - produce LBA B that accepts exactly the accepting computation histories of M on input w

Dec. and undec. problems

Proof:

– f(<M, w>) = <B> described below

on input x, check if x has form

$$\#C_1\#C_2\#C_3\#...\#C_k\#$$

• check that C_1 is the start configuration for M on input w

• check that $C_i \Rightarrow^1 C_{i+1}$

• check that C_k is an accepting configuration for M

• is B an LBA?

• is f computable?

• YES maps to YES?

$<M, w> \in \text{co-}A_{TM} \Rightarrow f(M, w) \in E_{LBA}$

• NO maps to NO?

$<M, w> \notin \text{co-}A_{TM} \Rightarrow f(M, w) \notin E_{LBA}$

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!

Rice's Theorem

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

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

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 $<M_1> \in P$

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?

$\langle M, w \rangle \in A_{TM} \Rightarrow$
$L(f(M, w)) = L(M_1) \Rightarrow$
$f(M, w) \in P$

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?

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

Post Correspondence Problem

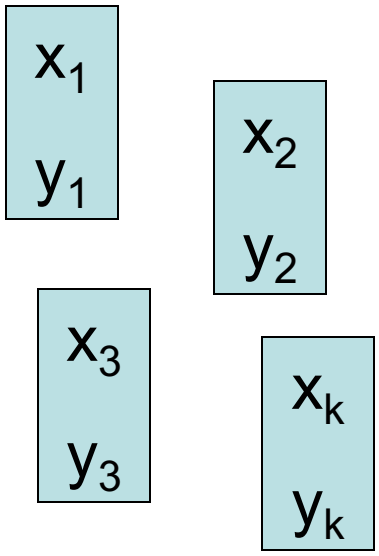
- many undecidable problems unrelated to TMs and automata

- classic example: Post Correspondence Problem

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

Post Correspondence Problem

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



"tiles"

x_2	x_1	x_5	x_2	x_1	x_3	x_4	x_4
y_2	y_1	y_5	y_2	y_1	y_3	y_4	y_4

$x_2x_1x_5x_2x_1x_3x_4x_4 = y_2y_1y_5y_2y_1y_3y_4y_4$

"match"

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

Post Correspondence Problem

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

Proof of $MPCP \leq_m PCP$:

– notation: for a string $u = u_1 u_2 u_3 \ldots u_m$

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

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:

  $(*x_1, *y_1*)$, $(*x_1, y_1*), (*x_2, y_2*), \ldots, (*x_k, y_k*), (*\blacklozenge, \blacklozenge)$
- 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

Post Correspondence Problem

– YES maps to YES?

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

x_1	x_4	x_5	x_2	x_1	x_3	x_4	x_4
y_1	y_4	y_5	y_2	y_1	y_3	y_4	y_4

$*x_1$	$*x_4$	$*x_5$	$*x_2$	$*x_1$	$*x_3$	$*x_4$	$*x_4$	$*\blacklozenge$
$*y_1*$	y_4*	y_5*	y_2*	y_1*	y_3*	y_4*	y_4*	$\blacklozenge$

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

$*x_1$	$*x_4$	$*x_5$	$*x_2$	$*x_1$	$*x_3$	$*x_4$	$*x_4$	$*\blacklozenge$
$*y_1*$	y_4*	y_5*	y_2*	y_1*	y_3*	y_4*	y_4*	$\blacklozenge$

x_1	x_4	x_5	x_2	x_1	x_3	x_4	x_4
y_1	y_4	y_5	y_2	y_1	y_3	y_4	y_4

"*" symbols must align

can only appear at the end

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 

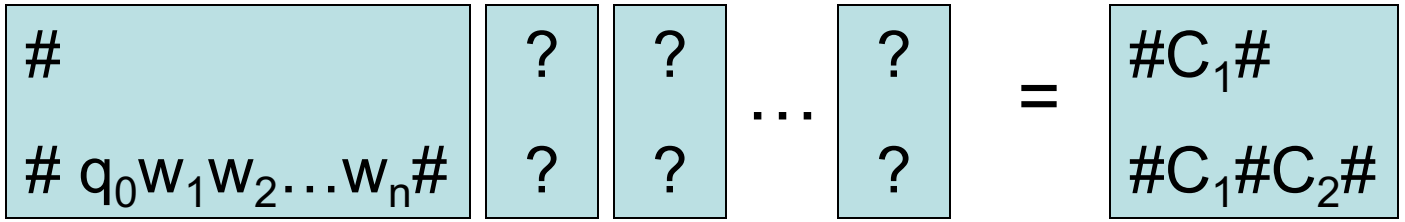
Post Correspondence Problem

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 (#, #$q_0w_1w_2w_3\ldots w_n$#)

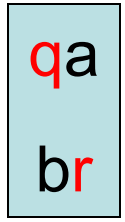
#		#
#$q_0w_1w_2\ldots w_n$#	=	#C_1#

Post Correspondence Problem



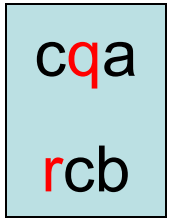
– tiles for head motions to the right:

- for all a,b $\in \Gamma$ and all q, r $\in$ Q with q $\neq q_{reject}$, if δ(q, a) = (r, b, R), add tile (qa, br)

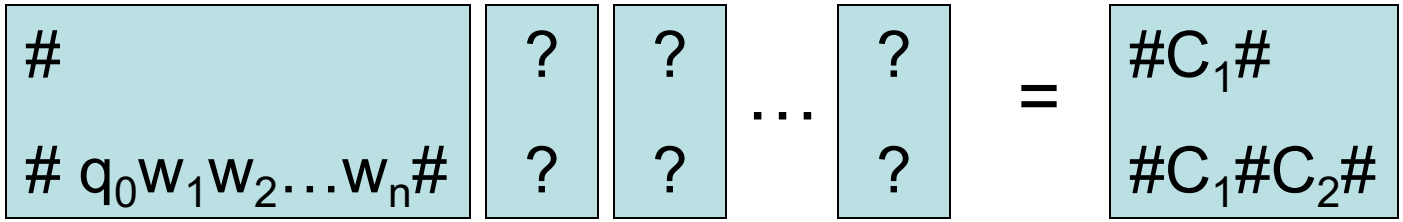


– tiles for head motions to the left:

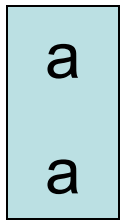
- for all a,b,c $\in \Gamma$ and all q, r $\in$ Q with q $\neq q_{reject}$, if δ(q, a) = (r, b, L), add tile (cqa, rcb)



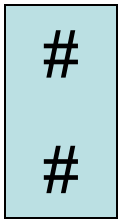
Post Correspondence Problem



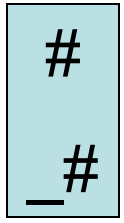
– tiles for copying (not near head)
  - for all a ∈ Γ, add tile (a, a)



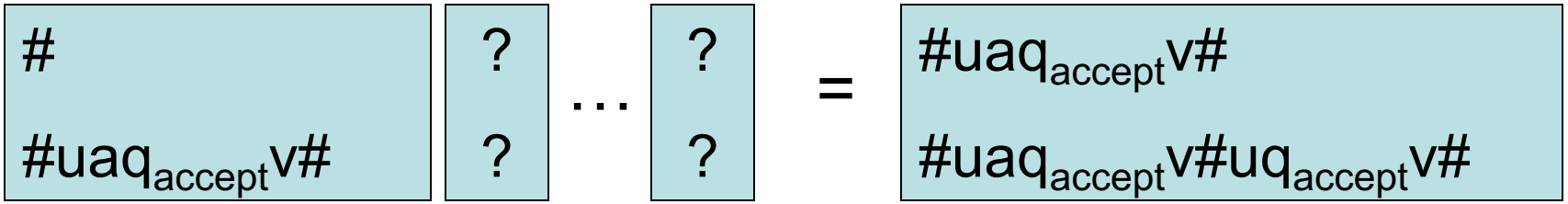
– tiles for copying # marker
  - add tile (#, #)



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

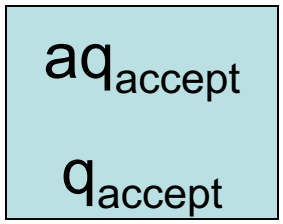


Post Correspondence Problem

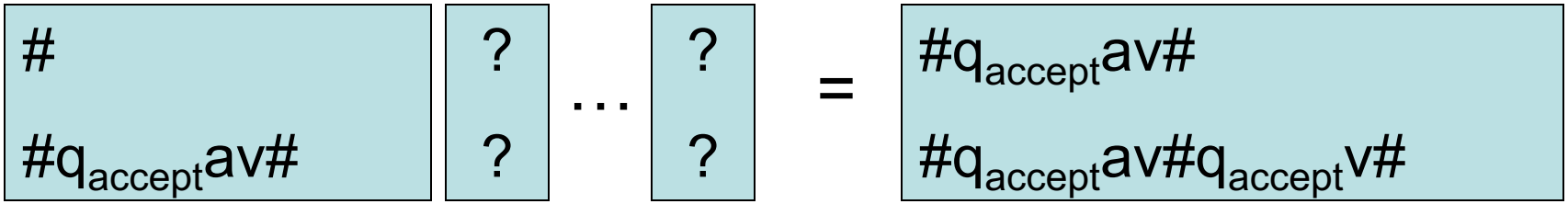


– tiles for deleting symbols to left of q_{accept}

- for all $a \in \Gamma$, add tile (aq_{accept}, q_{accept})

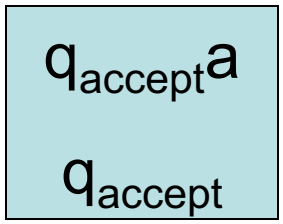


Post Correspondence Problem

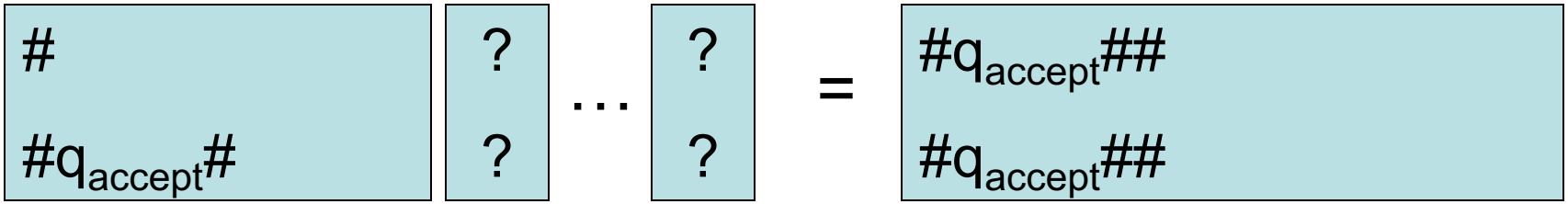


– tiles for deleting symbols to right of q_{accept}

- for all $a \in \Gamma$, add tile ($q_{accept}a$, q_{accept})

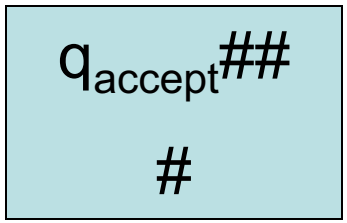


Post Correspondence Problem



– tiles for completing the match

- for all a $\in \Gamma$, add tile (q_{accept}##, #)



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)

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