More undecidable problems

From flickr user pinprick, CC-BY-NC-ND

HW1 due today, HW2 out "Soon"

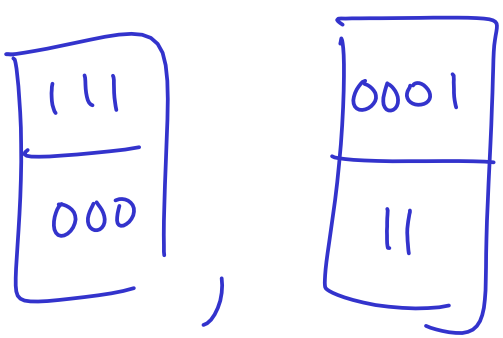
Today's class

- The following are undecidable:
  - L_{acc} = { <M,x> | M is a TM that accepts x }
  - L_{empty} = { <M> | M is a TM, L(M) = $\varnothing$ }
  - L_{finite} = { <M> | M is a TM, L(M) is finite }

- These are all problems about understanding behavior of arbitrary TMs
  - What about more **natural** problems that don't (seem to) involve TMs / computation?

Post Correspondence Problem (PCP)

- **Input:** finite set of “domino” types
- **Question:** given an infinite supply of each domino type, can you make top row = bottom row? ≠ empty



string on top

string on bottom

PCP examples

- Input:

0	01	110
100	00	11

yes, solvable

1 1 0 0 1 1 1 0 0

1 1 0 0 1 1 1 0 0

- Input:

0	11	01
100	001	101

no, not solvable

$|\text{top string}| < |\text{bottom string}|$

every domino has 1st bits of top/bottom disagree.

Theorem: PCP is undecidable!

- $L_{acc} \leq$ PCP: write subroutine for L_{acc}, using a subroutine for PCP
- Need to convert <M,x> into a **set of dominos** that somehow captures the computation of M on x!

Need to deal with TM details

- “Source code” of a TM is a finite list of simple **transition rules**:

$$(q,c) \rightarrow (q',c',L/R)$$

if in state q reading character c

write character c'

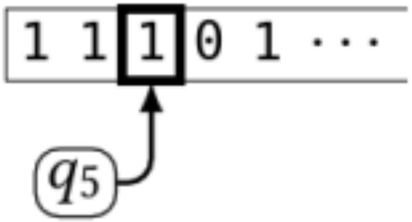
move to state q'

move tape head {left,right}

Need to deal with TM details

- TM **configuration** specifies:
  - Current contents of tape
  - Current state
  - Current tape head location

Note:
"state" ≠ "configuration"

- **Example:** 

$\Rightarrow$ write as "1 1 q_5 1 0 1 …"

Need to deal with TM details

- **Observation:** M accepts x ⇔ there is a sequence of configurations where:
  - First config is "q_{start} x"
  - Last config is in q_{accept}
  - Config i follows from config i+1 by a valid transition

accepting computation history

- **Example:** write a sequence of configurations separated by # symbols:

" … # 1 1 q_5 1 0 1 # 1 1 0 q_8 0 1 # … "

PCP reduction idea:

$conf_1$ # $conf_2$ # … # $conf_i$
$conf_1$ # $conf_2$ # … # $conf_i$ # $conf_{i+1}$

- **Idea/invariant:**
  - Top/bottom rows are sequence of **TM configurations**
  - Bottom row always “1 step ahead” of top row
  - Can't get top row to match without adding **correct next TM configuration** to bottom row
  - Top row can only “catch up” when TM is in accepting state

PCP proof <u>sketch</u> (III)

- Given TM M & input x, generate set of dominoes as follows:

(#1) Include a domino

#
q_0 x

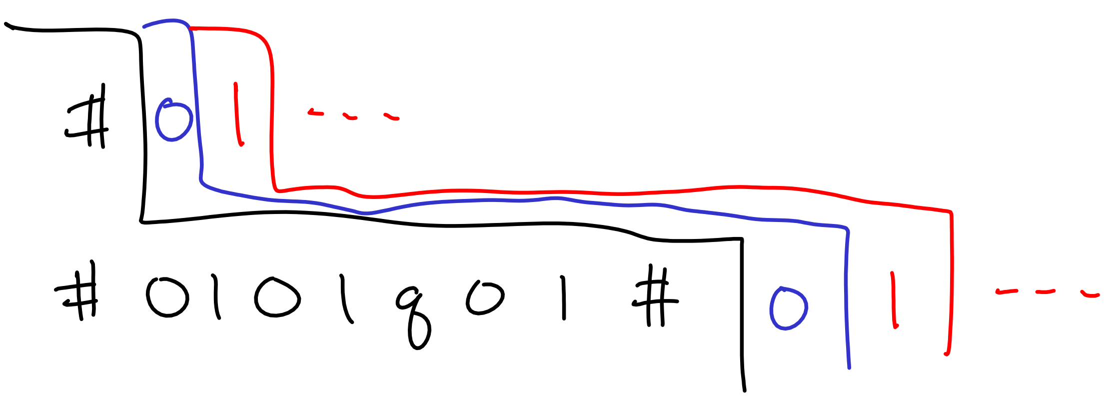
(let's assume we can force this to be the first domino in PCP solution)

PCP proof sketch (IV)

- Given TM M & input x, generate set of dominoes as follows:

(#2) Include dominos that "copy" tape contents to next configuration

0	1	#
0	1	#



PCP proof <u>sketch</u> (V)

- Given TM M & input x, generate set of dominoes as follows:

(#3) Include dominos that advance TM one step:

(q,1) → (q',0,L) becomes

0 q 1
q' 0 0

(edge case: make blank chars appear at right end of tape)

0 1 0 0 q 1 1

0 1 0 0 q 1 1 # 0 1 0 q' 0 0 1

PCP proof sketch (VI)

- Given TM M & input x, generate set of dominoes as follows:

(#4) Include dominos that let the accept state "eat" neighboring characters & "catch up" to bottom row

q_{accept} 0	q_{accept} 1	0 q_{accept}	1 q_{accept}	# q_{accept} # #
q_{accept}	q_{accept}	q_{accept}	q_{accept}	# #

0 1 0 1 q_{accept} 1 # . . . . .

0 1 0 1 q_{accept} 1 # 0 1 0 1 q_{accept} # # q_{accept}

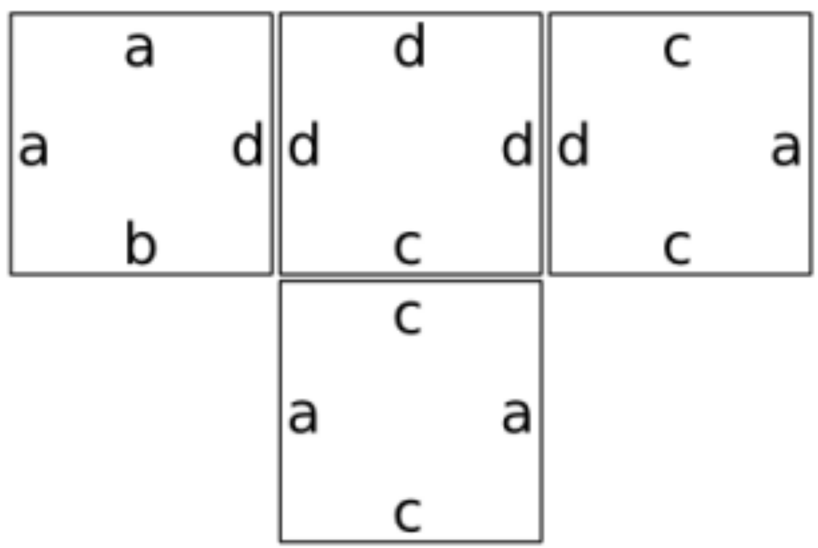
PCP proof <u>sketch</u> (overview)

- Given TM M & input x, generate set of dominoes as follows:

1) “seed” domino with initial TM configuration
2) dominoes that copy tape chars to next configuration
3) dominoes that advance tape head + state one step
4) dominoes that allow accepting configuration to “catch up” to bottom row

Another example: Wang tiles

- **Wang tile:** square with specific “flavor of glue” on each edge
- Adjacent tiles must have matching glue flavors on shared edge



Another example: Wang tiles

- **Input:**
  - finite set of Wang tile types
  - Initial arrangement of tiles
- **Question:**
  - Can the configuration be extended to tile the entire plane?
- **Theorem:** this problem is undecidable!

Wang tile reduction: $L_{acc} \leq \text{WANG}$

- Write subroutine for L_{acc}, using a subroutine for Wang tile question
- Need to convert <M,x> into a **set of tiles** that somehow captures the computation of M on x!

Wang tile reduction idea