P vs NP

…or why I hate Sudoku

k-SAT

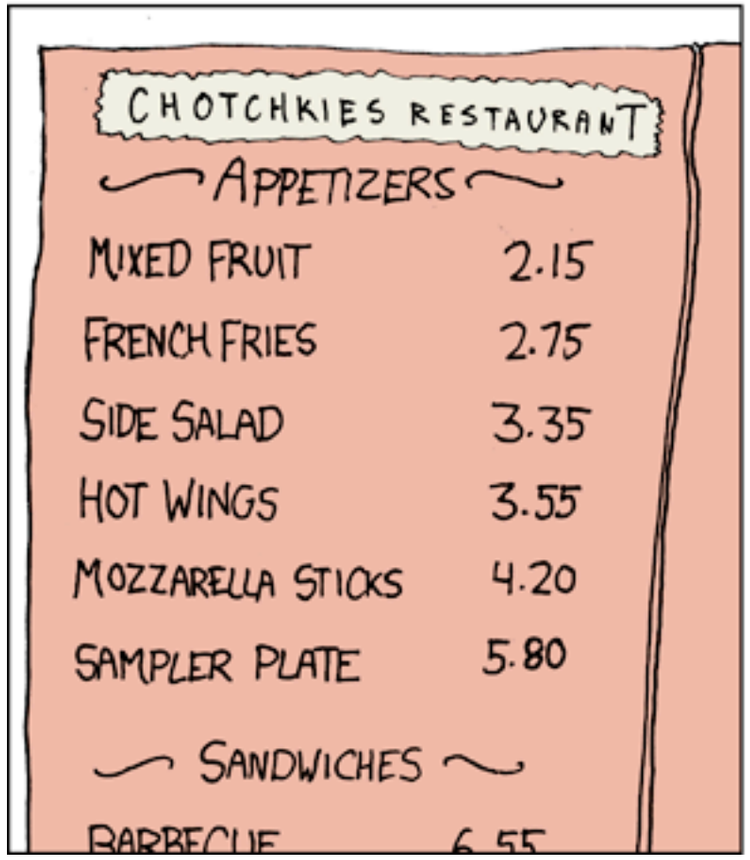
$$E = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_1 \vee x_2 \vee x_4)$$

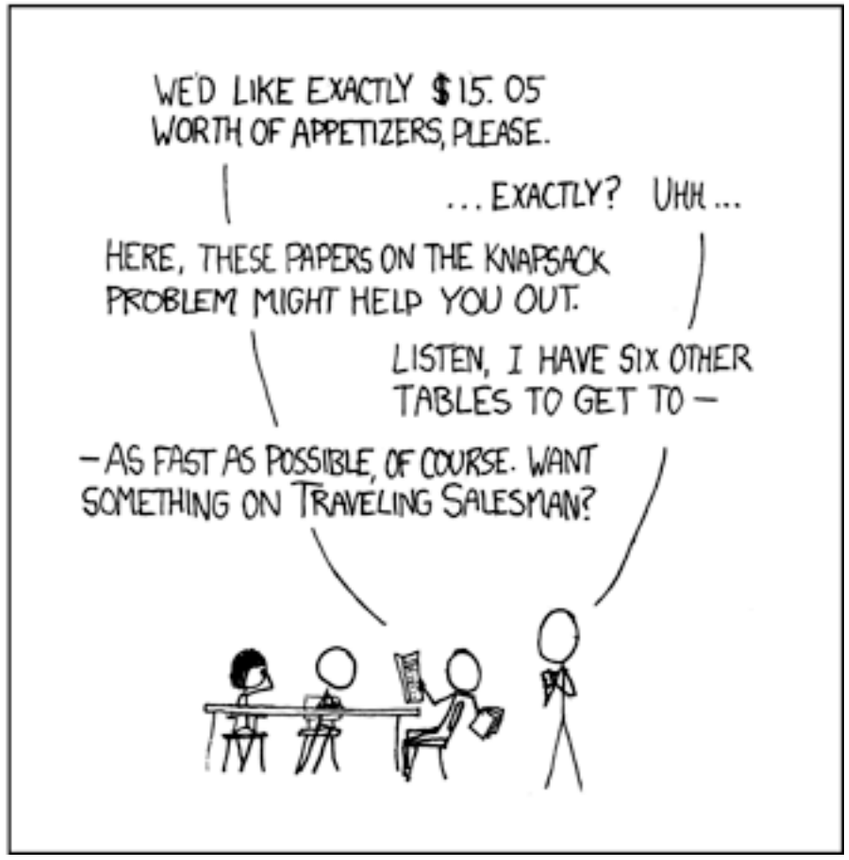
Sudoku

8								
		3	6					
	7			9		2		
	5				7			
				4	5	7		
			1				3	
		1					6	8
		8	5				1	
	9					4		

MY HOBBY:

EMBEDDING NP-COMPLETE PROBLEMS IN RESTAURANT ORDERS





Hallmarks of NP

- Combinatorial solution space
  - (exponential in size)

Hallmarks of NP

- Combinatorial solution space
  - (exponential in size)
- "Right" choice determined later in graph

Hallmarks of NP

- Combinatorial solution space
  - (exponential in size)
- “Right” choice determined later in graph
- Polynomial number of valid states

Hallmarks of NP

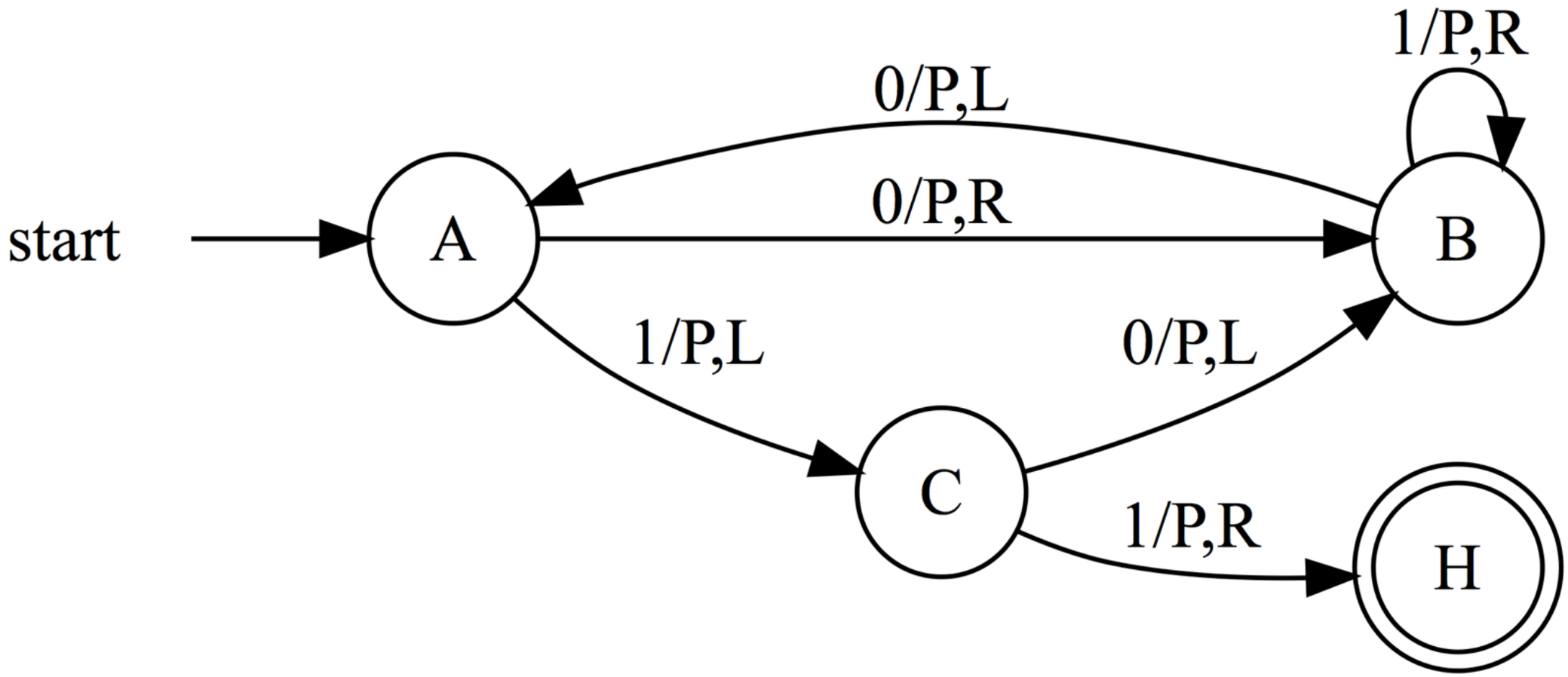
- Combinatorial solution space
  - (exponential in size)
- "Right" choice determined later in graph
- Polynomial number of valid states
- Polynomial-time verification

“N”? “P”??

NP = Nondeterministically Polynomial

“N”? “P”??

NP = Nondeterministically Polynomial


start
A
B
C
H
0/P,L
1/P,R
0/P,R
1/P,L
0/P,L
1/P,R

$O(rly?)$

- Count state transitions of your TM exec

$O(rly?)$

- Count state transitions of your TM exec
- Worst case as a function of input tape

$O(rly?)$

- Count state transitions of your TM exec
- Worst case as a function of input tape
- Undecidable problems are $O(\infty)$

$O(rly?)$

- Count state transitions of your TM exec
- Worst case as a function of input tape
- Undecidable problems are $O(\infty)$
- We can group problems by complexity

Complexity

Complexity

- Polynomial time for DTM

Complexity

- Polynomial time for DTM
- Polynomial time for NDTM

Complexity

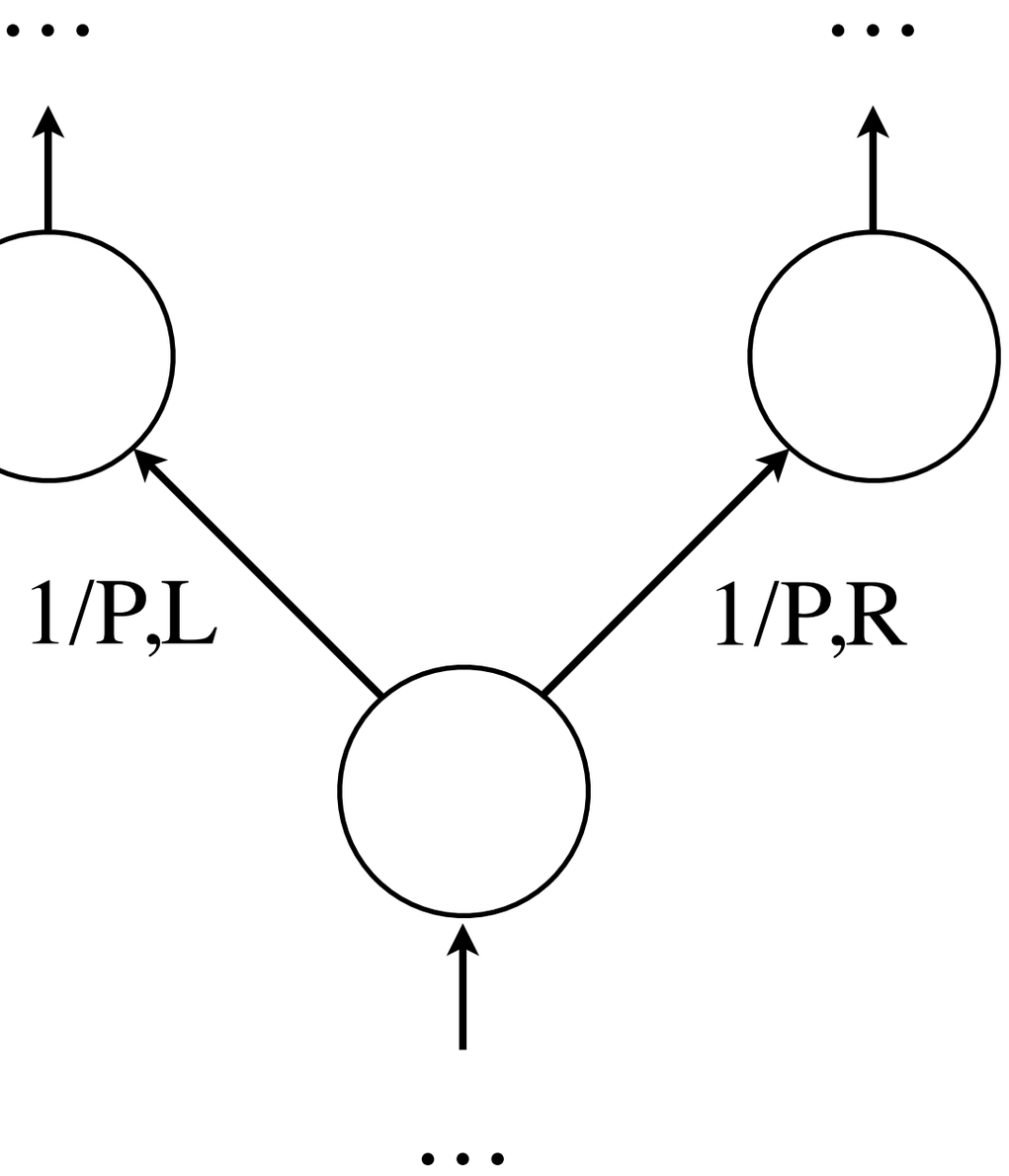
- Polynomial time for DTM
- Polynomial time for NDTM
- Non-deterministic automata

Complexity

- Polynomial time for DTM
- Polynomial time for NDTM
- Non-deterministic automata
  - $f : (S \times \Sigma) \rightarrow (S \times (\Sigma + Unit) \times \{\mathrm{R}, \mathrm{L}\})$

Complexity

- Polynomial time for DTM
- Polynomial time for NDTM
- Non-deterministic automata
  - $f : (S \times \Sigma) \to (S \times (\Sigma + \mathit{Unit}) \times \{\mathrm{R}, \mathrm{L}\})$
  - $f : (S \times \Sigma) \to \mathcal{P}(S \times (\Sigma + \mathit{Unit}) \times \{\mathrm{R}, \mathrm{L}\})$


1/P,L
1/P,R

Oracles

- At every ambiguity, we must "guess"
- A true NDTM never guesses wrong!
- Can we encode this deterministically?
  - (hint: yes)
- How expensive is this?

The Question

Does there exist a polynomial time reduction from the class NP-TIME to the class P-TIME?

The Question

Can we guess accurately *and* efficiently?

Proof?

- Assume P = NP and derive a contradiction

Proof?

- Assume P = NP and derive a contradiction
- Prune the search space in P-TIME

Proof?

- Assume P = NP and derive a contradiction
- Prune the search space in P-TIME
- Is it possible? Is it not?

Proof?

- Assume P = NP and derive a contradiction
- Prune the search space in P-TIME
- Is it possible? Is it not?
- Is there another way?

Proof?

- Assume P = NP and derive a contradiction
- Prune the search space in P-TIME
  - Is it possible? Is it not?
  - Is there another way?
- Nothing about it seems *definitively* wrong

What if?

What if?

- Traveling salesmen everywhere rejoice

What if?

- Traveling salesmen everywhere rejoice
- Mathematical research becomes trivial

What if?

- Traveling salesmen everywhere rejoice
- Mathematical research becomes trivial
- RSA does *not* collapse
  - Factorization is not NP-complete

What if?

- Traveling salesmen everywhere rejoice
- Mathematical research becomes trivial
- RSA does *not* collapse
  - Factorization is not NP-complete
- Industrial applications…

Usefulness

$$O(k^n) \approx O(n^{42{,}000{,}000})$$

Questions?