Portland State University

W'21 CS 584/684

Algorithm Design & Analysis

Fang Song

Lecture 18

- NPC

Central ideas in complexity

- Poly-time as "feasible"
  - Most natural problems either are easy (e.g., n^3) or no poly-time alg. known
- Reduction : relating hardness ($A \leq B \Rightarrow A$ no harder than B)
- Classify problems by "hardness"

Definition of class P

P. Decision problems for which there is a poly-time algorithm

Problem	Description	Algorithm	YES instance	No instance
Multiple	Is x a multiple of y?	Grade school	51,17	52,17
RELPRIME	Are x and y relatively prime?	Euclid (300 BCE)	34,39	34,51
PRIMES	Is x a prime?	AKS 2002	53	51
EDIT-DISTANCE	Is the edit distance between x and y less than 5?	Dynamic programming	neither either	algorithm quantum

Definition of class NP

NP. Decision problems for which there is a poly-time certifier

Idea of certifier

- Certifier checks a proposed proof π that $s \in X$
- Need not determine whether $s \in X$ on its own

N.B. $|t| = p(|s|)$ for some polynomial $p()$

Def. Algorithm $C(s,t)$ is a certifier for problem X if for every string s, $s \in X$ iff there exists a string t such that $C(s,t) = \text{yes}$

Equivalent def. $\mathbf{NP}$ = nondeterministic polynomial–time

not ~~polynomial~~–time

Certifiers and certificates: Composite

COMPOSITES. Given an integer s, is s composite?

- Certificate: A non-trivial factor t of s.
- Certifier.

```
CompositesCertifier(s,t)
 If (t ≤ 1 or t ≥ s)
     Return false
 Else if (s is a multiple of t)
     Return true
 Else
     Return false
```

- Instance. $s = 437{,}669$
  - Certificate. $t = 541\ or\ 809.\ 437{,}669 = 541 \times 809$

Conclusion. COMPOSITES ∈ **NP**

Certifiers and certificates: Hamiltonian cycle

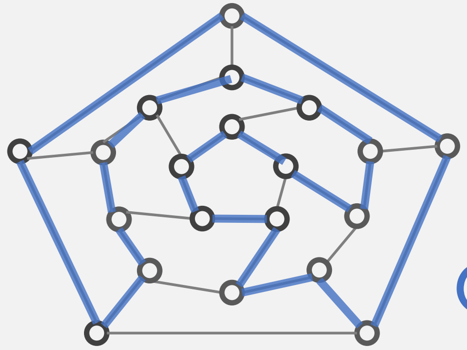
HAM-CYCLE. Given a graph $G = (V, E)$, does there exist a simple cycle that visits every node?

- Certificate: A permutation of n nodes
- Certifier.

```
HAM-CYCLE-Certifier(G, σ)
  If (∀i, j, σ_i ≠ σ_j &(σ_i, σ_{i+1}) ∈ E)
     Return true
```

Conclusion. $\text{HAM-Cycle} \in \mathbf{NP}$

Instance s



Certificate t

P,NP,EXP

P. Decision problems for which there is a poly-time algorithm

EXP. Decision problems for which ∃ an exponential-time algorithm

i.e., runs in time $O(2^{p(|s|)})$ for some polynomial $p()$

NP. Decision problems for which there is a poly-time certifier

- Claim. $\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{EXP}$

P ⊆ NP. Consider any $X \in \mathrm{P}$,

- ∃ poly−time A that solves X
- Certificate: $t = \epsilon$, certifier $C(s,t) = A(s)$

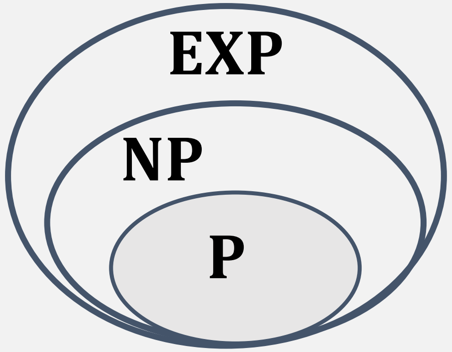
NP ⊆ EXP. Consider any $X \in \mathrm{NP}$,

- ∃ poly−time certifier $C(s,t)$
- To decide input s, run $C(s,t)$ on all strings t with $|t| \le p(|s|)$.
- Return yes, if $C(s,t)$ ever says yes.

Open question: P = NP?

The Millennium prize problems

- $1 million prize



- Consensus opinion on P = NP? Probably no.

Eight Signs A Claimed P≠NP Proof Is Wrong

As of this writing, Vinay Deolalikar still hasn't retracted his P≠NP c

https://www.scottaaronson.com/blog/?p=458

Millennium Problems

Yang–Mills and Mass Gap

Experiment and computer simulations suggest the existence of a "mass gap" in the
no proof of this property is known.

Riemann Hypothesis

The prime number theorem determines the average distribution of the primes. Th
average. Formulated in Riemann's 1859 paper, it asserts that all the 'non-obvious'

P vs NP Problem

If it is easy to check that a solution to a problem is correct, is it also easy to solve th
the NP problems is that of the Hamiltonian Path Problem: given N cities to visit, ho
solution, I can easily check that it is correct. But I cannot so easily find a solution.

Navier–Stokes Equation

This is the equation which governs the flow of fluids such as water and air. Howeve
solutions exist, and are they unique? Why ask for a proof? Because a proof gives no

Hodge Conjecture

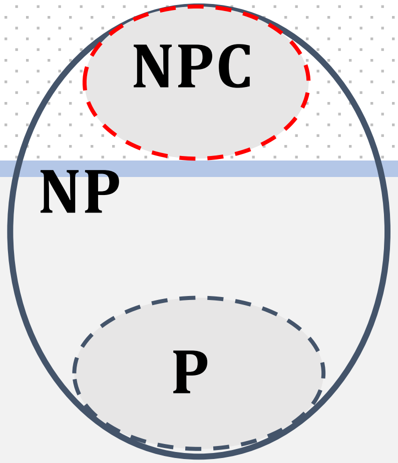
The answer to this conjecture determines how much of the topology of the solutio
further algebraic equations. The Hodge conjecture is known in certain special case
dimension four it is unknown.

Poincaré Conjecture

In 1904 the French mathematician Henri Poincaré asked if the three dimensional s
manifold. This question, the Poincaré conjecture, was a special case of Thurston's g
three manifold is built from a set of standard pieces, each with one of eight well-ur

Birch and Swinnerton-Dyer Conjecture

NP-Completeness



Def. A problem Y is NP-Complete if

1. $Y \in \mathbf{NP}$
2. $\forall X \in \mathbf{NP}, X \leq_{P,Karp} Y$

Theorem. Suppose Y is NP-Complete, then Y is solvable in poly-time iff. $\mathbf{P} = \mathbf{NP}$

Pf.

- (⇐) If $\mathbf{P} = \mathbf{NP}$, then Y can be solved in poly-time since $Y \in \mathbf{NP}$
- (⇒) If Y is solvable in poly-time, consider any $X \in \mathbf{NP}$.
  Since $X \leq_{P,Karp} Y$, X has a poly-time algorithm as well
  I.e., $\mathbf{NP} \subseteq \mathbf{P} \rightarrow \mathbf{P} = \mathbf{NP}$

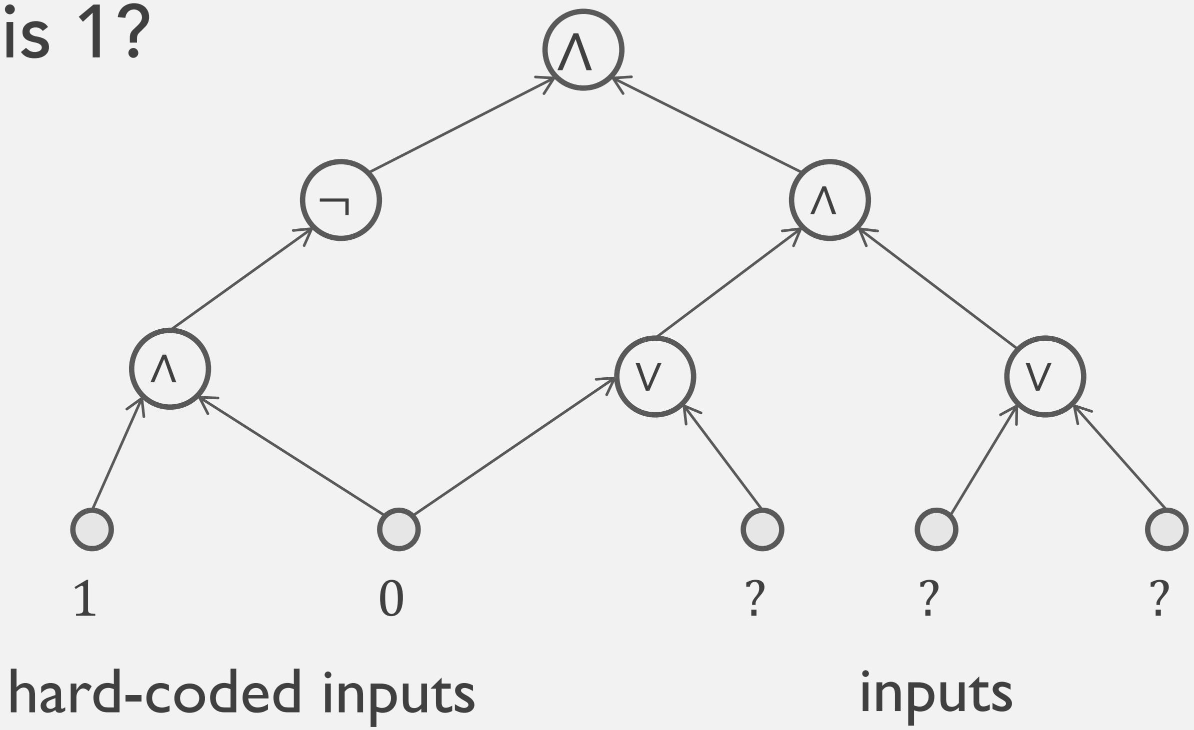
Fundamental question: Are there natural NP-complete problems?

The "first" NP-Complete problem

Theorem. Circuit−SAT is NP-Complete [Cook 1971,Levin 1973]

Input. A combinational circuit built out of AND/OR/NOT gates

Goal. Decide if there is a way to set the circuit inputs so that the output is 1?

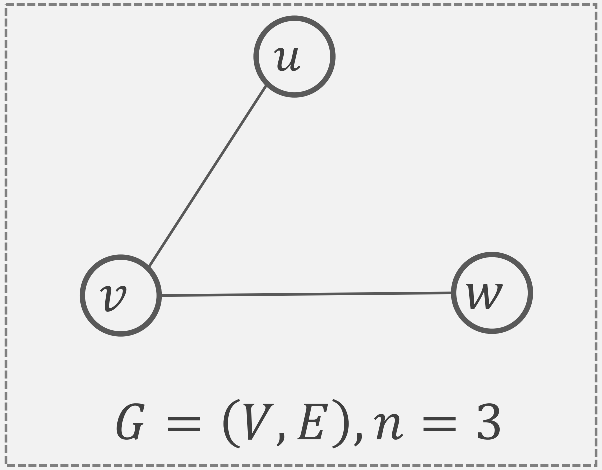


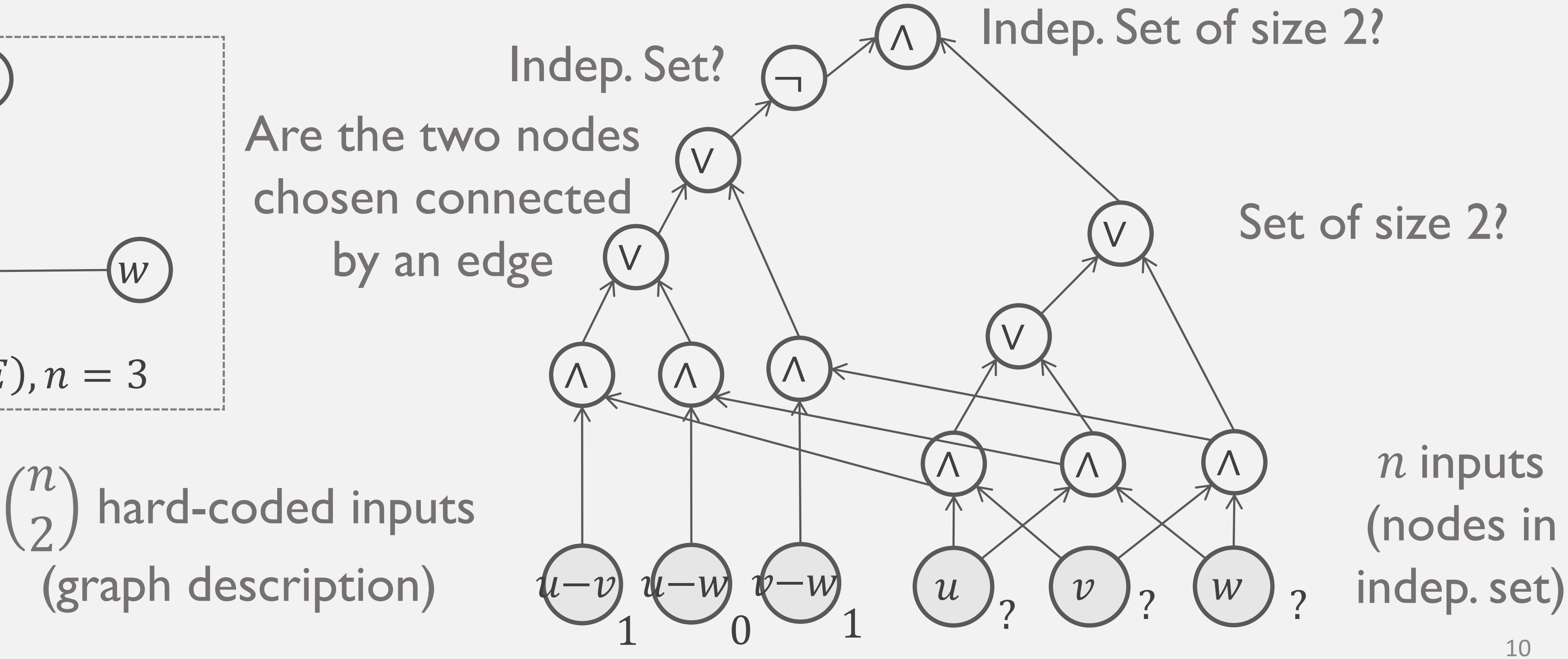
Stephen Cook Leonid Levin

Example

Given. Graph G

Construction. Circuit K whose inputs can be set so that K outputs true iff. graph G has an independent set of size 2





Establishing NP-Completeness

Once we establish first "natural" NP-complete problem, others fall like dominoes …

Recipe to establish NP-Completeness of problem Y

1. Show that $Y \in \mathbf{NP}$
2. Choose an $\mathrm{NP-complete}$ problem X
3. Prove that $X \leq_{P,Karp} Y$

Justification. If X is an NP-complete problem, and Y is a problem in $\mathbf{NP}$ with the property that $X \leq_{P,Karp} Y$ then Y is NP-complete (by transitivity)

Practicing reductions

- $\text{Circuit–SAT} \leq \text{3–SAT}$

+

$$\text{3–SAT} \leq_P \text{INDEPENDENT–SET} \leq_P \text{VERTEX–COVER} \leq_P \text{SET–COVER}$$

- $\text{3–SAT} \leq \text{HAM–CYCLE}$

⇒ They are all NP-Complete!

Richard M. Karp

REDUCIBILITY AMONG COMBINATORIAL PROBLEMS†

Richard M. Karp

University of California at Berkeley

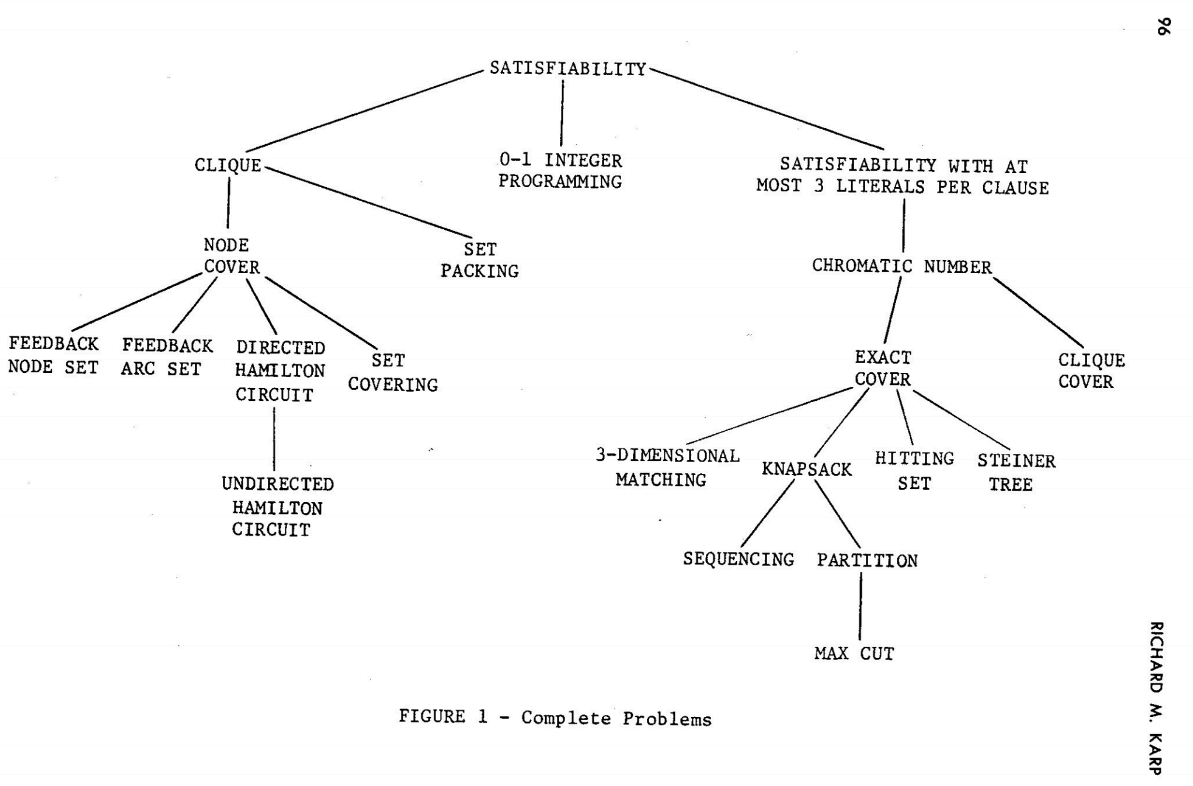
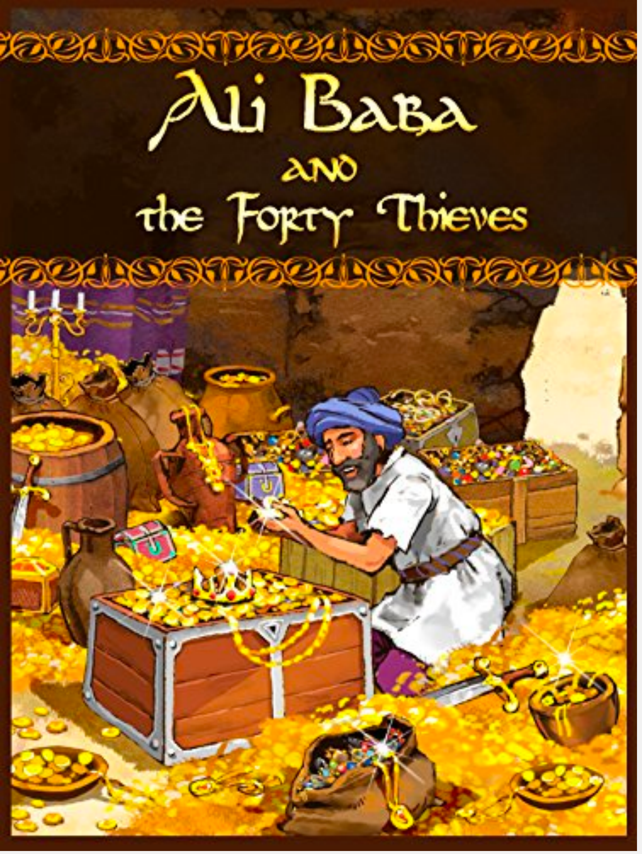
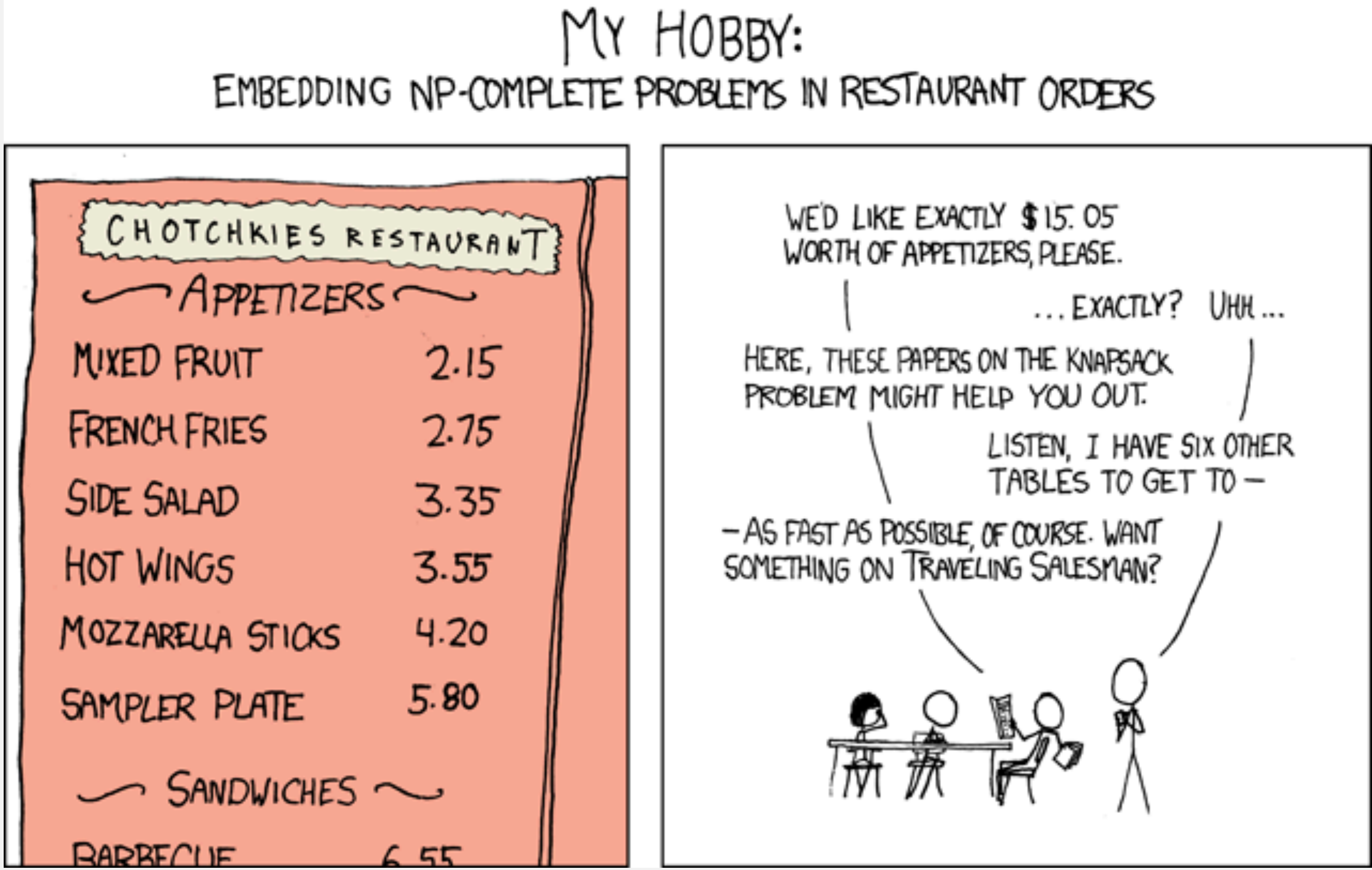


FIGURE 1 - Complete Problems



https://images.app.goo.gl/pwGFyw2pp6Xmx6CB8



https://xkcd.com/287/

Quiz

For each of the following statements, decide T/F/Unknown.

a) All problems in $\mathbf{P}$ can be solved in n^{2019} time.
b) If a problem is in $\mathbf{NP}$, then it cannot be solved in n^{2019} time.
c) If a problem is $\mathbf{NP-Complete}$, then the best algorithm for it takes $2^{\Omega(n)}$ time.
d) There exists a problem in $\mathbf{NP}$ but not in $\mathbf{P}$.

$\mathbf{3-SAT}$ is NP-Complete

Theorem. $\mathbf{3-SAT}$ is NP-Complete

Pf. We show $\text{Circuit}-\text{SAT} \leq_P 3-\text{SAT}$

- Given a circuit K, create a $3-\text{SAT}$ variable x_i for each gate
- Make circuit compute correct values at each node

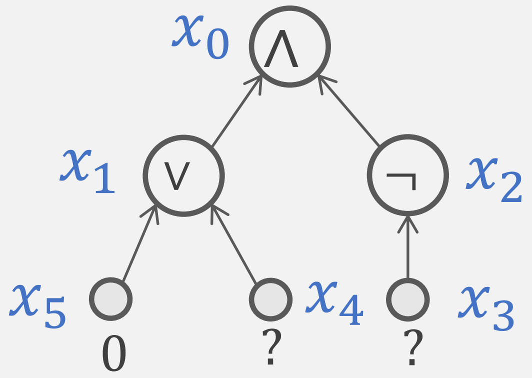
$x_2 = \neg x_3$	$\Rightarrow (x_2 \vee x_3) \wedge (\overline{x_2} \vee \overline{x_3})$
$x_1 = x_4 \vee x_5$	$\Rightarrow (x_1 \vee \overline{x_4}) \wedge (x_1 \vee \overline{x_5}) \wedge (\overline{x_1} \vee x_4 \vee x_5)$
$x_0 = x_1 \wedge x_2$	$\Rightarrow (\overline{x_0} \vee x_1) \wedge (\overline{x_0} \vee x_2) \wedge (x_0 \vee \overline{x_1} \vee \overline{x_2})$

- Hard-coded input values and output value

$x_5 = 0 \Rightarrow \overline{x_5}$ $\quad$ $x_0 = 1 \Rightarrow x_0$

- Final step: turn clauses into exactly 3 literals by adding dummy variables

EX. $x_1 \vee x_2 \Rightarrow (x_1 \vee x_2 \vee y) \wedge (x_1 \vee x_2 \vee \overline{y})$



Circuit K satisfiable iff. $\exists$ assignment satisfying all clauses constructed

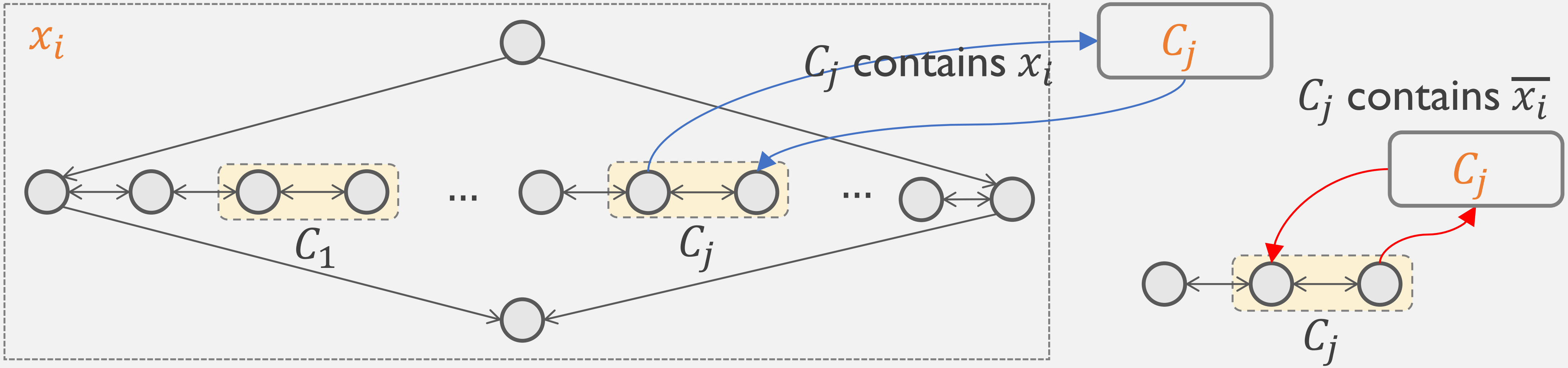
! Don't forget to show $\mathbf{3-SAT} \in \mathbf{NP}$

$\text{(DIR}-\text{)HAM}-\text{CYCLE}$ is NP-Complete

$\text{(DIR}-\text{)HAM}-\text{CYCLE}$. Given a directed graph $G = (V, E)$, does there exist a directed cycle Γ that visits every node exactly once?

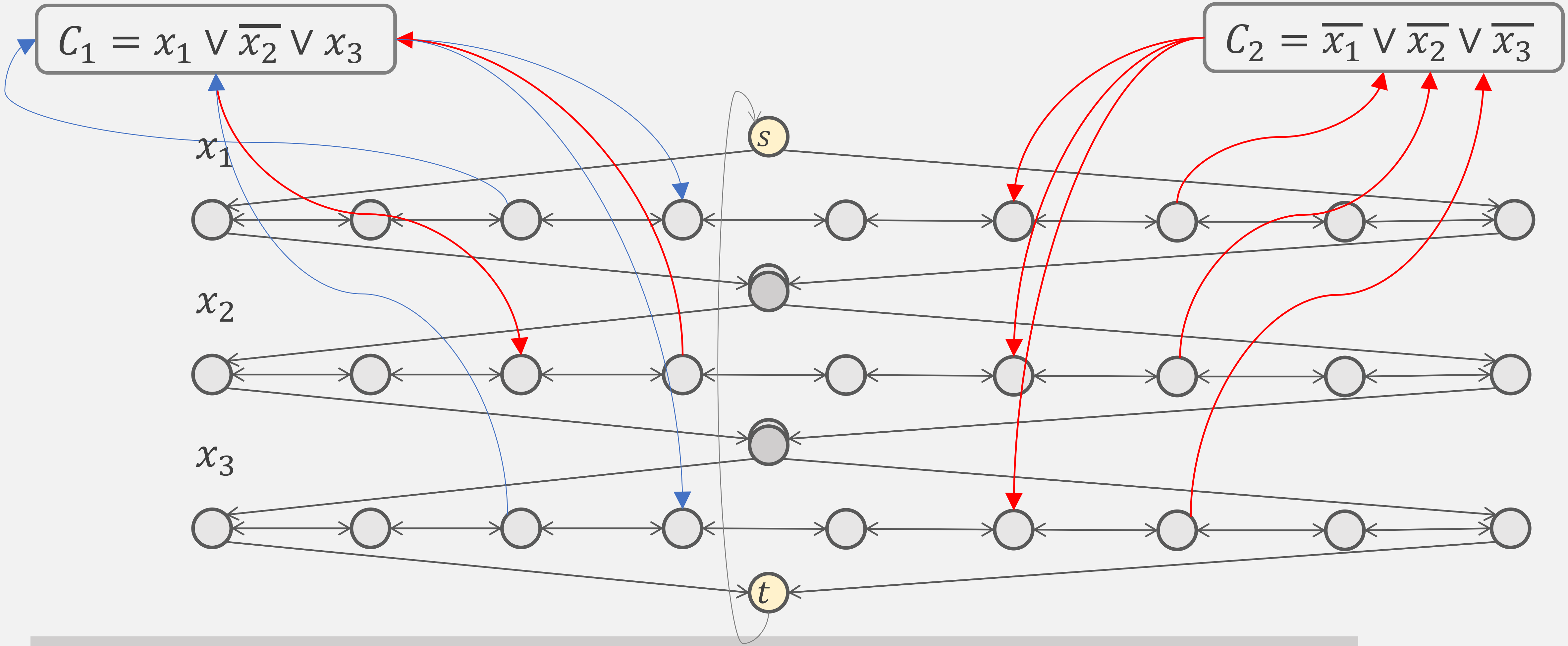
Theorem. $3-\text{SAT} \leq_P \text{(DIR}-\text{)HAM}-\text{CYCLE}$

Pf. Given $3-\text{SAT}$ instance Φ in CNF: n variables x_i and k clauses C_j



Intuition: traverse row i from left to right $\Leftrightarrow$ set variable $x_i = \text{true}$

$3-\mathrm{SAT} \leq_P (\mathrm{DIR}-)\mathrm{HAM}-\mathrm{CYCLE}$



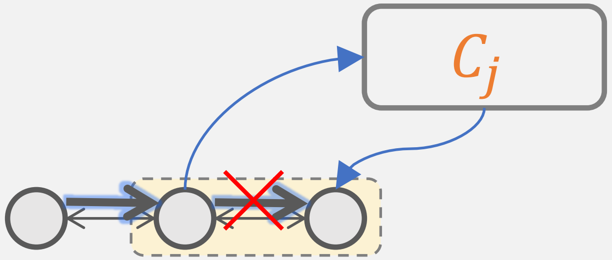
Claim. Φ is satisfiable iff. G has a Hamiltonian cycle

$3\text{−SAT} \leq_P (\text{DIR−})\text{HAM−CYCLE}$

Claim. Φ is satisfiable iff. G has a Hamiltonian cycle

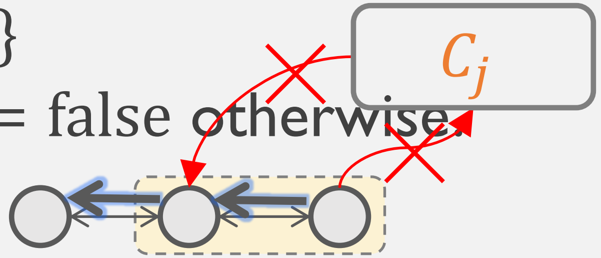
(⇒) Suppose Φ has a satisfying assign. $\boldsymbol{x}^*$. Define an H-Cycle in G:

- if $x_i^* = \text{true}$, traverse row x_i from left to right
- if $x_i^* = \text{false}$, traverse row x_i from right to left
- For each clause C_j pick (only) one row i and take a detour



(⇐) Suppose G has a H-Cycle Γ. Define a satisfying assign. in Φ:

- In Γ, replace edges going/leaving C_j with the edge of the corresponding two nodes in some row. This gives a new cycle Γ' in $G - \{C_1, C_2, \ldots, C_k\}$
- In Γ', set $x_i = \text{true}$ if Γ' traverses row i left-to-right; set $x_i = \text{false}$ otherwise.

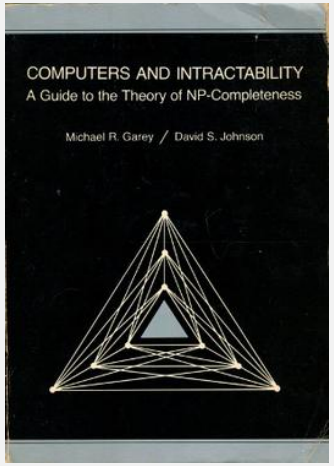


Hard computational problems cont'd

- Aerospace engineering: optimal mesh partitioning for finite elements.
- Chemical engineering: heat exchanger network synthesis
- Civil engineering: equilibrium of urban traffic flow
- Electrical engineering: VLSI layout.
- Mechanical engineering: structure of turbulence in sheared flows
- Biology: protein folding
- Physics: partition function of 3-D Ising model in statistical mechanics.
- Economics: computation of arbitrage in financial markets with friction
- Financial engineering: find minimum risk portfolio of given return
- Politics: Shapley-Shubik voting power
- Pop culture: Sudoku (http://www-imai.is.s.u-tokyo.ac.jp/~yato/data2/SIGAL87-2.pdf)

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

Want to learn more?

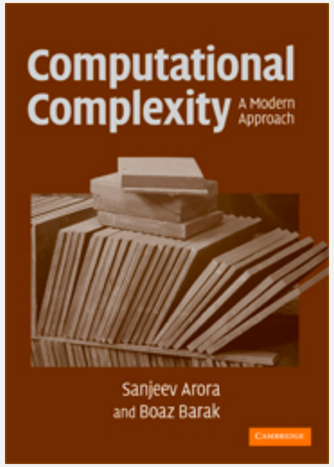


Computers and Intractability: A Guide to the Theory of NP-Completeness.
Michael Garey and David S. Johnson

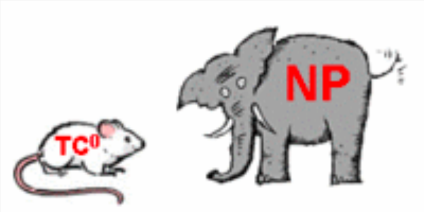
Most Cited Computer Science Citations

This list is generated from documents in the CiteSeer^{x} database as of March 19, 2015. This list is automaticall
mode and citation counts may differ from those currently in the CiteSeer^{x} database, since the database is con
All Years | 1990 | 1991 | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 | 1998 | 1999 | 2000 | 2001 | 2002 | 2003 | 20
| 2015

1. M R Garey, D S Johnson
Computers and Intractability: A Guide to the Theory of NPCompleteness" W.H. Feeman and 1979
11468



Computational Complexity: A Modern Approach
Sanjeev Arora & Boaz Barak



Complexity Zoo
There are now 544 classes and counting!