Reducibility among Combinatorial Problems

Richard Karp

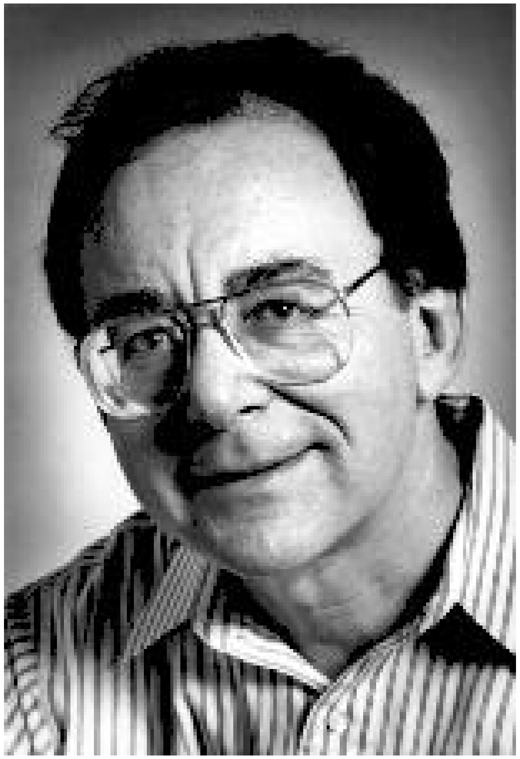
Presented by

Chaitanya Swamy

Richard Manning Karp

- Born in Boston, MA on January 3, 1935.
- AB in 1955, SM in 1956 and Ph.D. in 1959 from Harvard.
- 1959 - 1968 : IBM TJ Watson Research Center.
- 1968 - 1995 : UC Berkeley.
- 1972 : Wrote this paper.
- 1995 - 1999 : U. Washington, Seattle.
- Since 1999 at UC Berkeley. Currently works in Computational Biology – sequencing the human genome, analyzing gene expression data, other combinatorial problems

Awards and Honors



Richard Karp

- 1996 : National Medal of Science
- 1995 : Babbage Prize
- 1990 : John von Neumann Theory Prize, ORSA-TIMS
- 1986 : Distinguished Teaching Award, UC Berkeley
- 1985 : ACM Turing Award
- 1979 : Fulkerson Prize, AMS
- 1977 : Lanchester Prize, ORSA

History of *NP*-Completeness

- Stephen Cook, 1971, showed that formula Satisfiability is *NP*-Complete.
- Karp's paper showed that computational intractability is the rule rather than the exception.
- Together Cook & Karp, and independently Levin laid the foundations of the theory of *NP*-Completeness.
- "… Karp introduced the now standard methodology for proving problems to be *NP*-Complete …" – Turing Award citation.

Definitions

Given an alphabet Σ,

A problem Q is a set of 'yes' instances e.g..
$SAT = \{F \mid F \text{ is satisfiable }\}$, $(x1 \vee x2) \in SAT$

An algorithm A solves problem Q if, $A(x)=\text{'yes'} \Leftrightarrow x \in Q$.

A certifier B is an efficient certifier for problem Q if,
$\forall x.\ (x \in Q \Leftrightarrow \exists y.\ |y| \leq$ poly($|x|$) s.t. $B(x,y)=\text{'yes'}$ and
running time of $B \leq$ poly($|x|+|y|$))

P $= \{Q \mid Q$ has a polynomial time algorithm $A\}$

NP $= \{Q \mid Q$ has an efficient certifier $B\}$

Defn's. (contd.)

Let L, M be languages.

$L \leq_P M$ if $\exists$a polynomial time computable function f s.t.
$x \in L \Leftrightarrow f(x) \in M$.

The relation $\leq_P$ is symmetric and transitive.
Also, $L \leq_P M$ and $M \in P \Rightarrow L \in P$
$L \leq_P M$ and $M \in NP \Rightarrow L \in NP$.

L is said to be complete for NP w.r.t $\leq_P$, if
i. $\forall M \in NP$, $M \leq_P L$ ($\Rightarrow$ L is NP-Hard), and
ii. $L \in NP$

Classification of NP-Complete Problems

1. Constraint Satisfaction : SAT, 3SAT
2. Covering : Set Cover, Vertex Cover, Feedback Set, Clique Cover, Chromatic Number, Hitting Set
3. Packing : Set Packing
4. Partitioning : 3D-Matching, Exact Cover
5. Sequencing : Hamilton Circuit, Sequencing
6. Numerical Problems : Subset Sum, Max Cut

Some *NP*-Complete Problems

3SAT : Given $F(x_1, \ldots, x_n)$ in 3-CNF i.e. $F = C_1 \wedge \ldots \wedge C_m$, $C_i = (x_{i1} \vee x_{i2} \vee x_{i3})$, is F satisfiable ?

Clique : Given a graph G, a number k, does G have a complete subgraph of size k ?

Vertex Cover : Given $G=(V,E)$, l, is there a subset U of V s.t. $|U| = l$ and for every $e=(u,v)$, at least one of u,v is in U ?

3D-Matching : Given finite disjoint sets X, Y, Z of size n, and a set of triples $\{t_i\} \subseteq X \times Y \times Z$, are there n pairwise disjoint triples ?

Subset Sum(Knapsack) : Given n elements, $\{w_1, \ldots, w_n\}$ and a target B, is there a subset of elements which adds up exactly to B ?

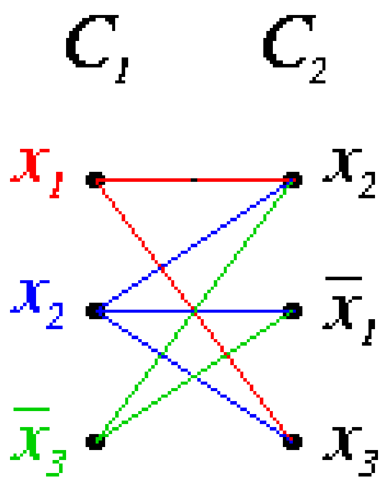
3SAT $\leq_P$ Clique

Construct $V=\{\langle\sigma, i\rangle \mid \sigma$ is a literal and occurs in $C_i\}$

$E=\{(\langle\sigma, i\rangle, \langle\delta, j\rangle) \mid$ *i* ¹ *j* and *s* ¹ `*d* }

$k = m$

e.g. $F = (x_1 \vee x_2 \vee \overline{x_3}) \wedge (x_2 \vee \overline{x_1} \vee x_3)$



Suppose $F = C_1 \wedge \ldots \wedge C_m$ is satisfiable, then at least one literal σ_i in every C_i is true, also both σ_i and $\overline{\sigma}_i$ are not true

$\Rightarrow$ the nodes $\{\langle\sigma_1, 1\rangle, \ldots, \langle\sigma_m, m\rangle\}$ form a clique of size $m=k$.

Conversely if $\exists$ a clique of size m, then we must have a node $\langle\sigma_i, i\rangle$ for each i, since two literals in the same clause do not have an edge between them. Also both σ, $\overline{\sigma}$ cannot be in the clique.

$\Rightarrow$setting the corresponding literals to true satisfies F.

$\therefore F \in$ 3SAT $\Leftrightarrow (G, m) \in$ Clique

Clique $\leq_P$ Vertex Cover

Construct $G^C = (V,E^C)$, where $E^C = \{(u,v) \mid (u, v) \notin E \}$

$$l = |V| - k = n\text{-}k$$

Suppose G has a clique K of size k. Then in G^C, no two vertices in K are connected $\Rightarrow$ V-K is a vertex cover for G^C since for any edge $e=(u,v) \in E^C$, both u, v cannot be in K

$\Rightarrow$ V-K is a vertex cover of size n-k.

Conversely if G^C has a vertex cover U of size n-k. Then no two vertices in V-U are connected in G^C

$\Rightarrow$ V-U forms a clique of size k in G.

3D-Matching $\leq_P$ Subset Sum

Let $m = |\{t_i\}|+1$. Encode each triple as a number in base m.

Each triple written as a 'bit' string of length $3n$ in base m.

$x_j \mapsto$ position $j' = j-1,\ \ 0 \leq j' < n$

$y_k \mapsto$ position $k' = n+k-1,\ \ n \leq k' < 2n$

$z_l \mapsto$ position $l' = 2n+l-1,\ \ 2n \leq l' < 3n$

For each $t_i = (x_j, y_k, z_l)$, we have $w_i = m^{j'} + m^{k'} + m^{l'}$ ie. w_i is the string which has 1s at positions j', k' and l'.

z_n	. . .	z_l	. . .	z_1	y_n	. . .	y_k	. . .	y_1	x_n	. . .	x_j	. . .	x_1
0	…	1	…	0	0	…	1	…	0	0	…	1	…	0

Finally we let B = string of all 1s = $(m^{3n}-1)/(m-1)$.

If we have a 3D-Matching, then since there are n pairwise disjoint triples, each x_j, y_k, z_l is present in exactly one triple

∴ adding w_i's corresponding to the triples gives a string of 1s

⇒ there is a subset with sum = B.

Conversely if there is a subset adding up to B, then by construction the triples corresponding to the elements cover each x_j, y_k, z_l exactly once

⇒ there are n pairwise disjoint triples.

Impact of the paper

- Along with Cook's paper laid the foundations of the theory of *NP*-Completeness.
- Showed that all these different looking problems are essentially the same problem in disguise.
- Since Karp's paper there have been a plethora of papers on proving problems *NP*-Complete or *NP*-Hard.

  Gary & Johnson, "Computers and Intractability : A Guide to the Theory of NP-Completeness" has an extensive catalogue of these.
- An AltaVista search for *NP* Completeness gave 227,598 hits.

Discussion

- In the face of computational intractability, how do we approach *NP*-Complete problems?
- Are all *NP*-Complete and *NP*-Hard problems equally hard?
- Are all instances of *NP*-Complete problems equally hard?
- *PCP* model(Arora, Lund, Motwani et al.) – Proof, Verifier model.

  Given a string x, a proof of membership y, a probabilistic $(r(n), q(n))$ verifier uses $O(r(n))$ random bits to compute $O(q(n))$ addresses in the proof. Then using random access it queries those addresses and decides membership.

  Main Theorem : $NP = PCP(\log n, 1)$
- Karp anecdotes?
- $P = NP$?