CSE 332 Autumn 2024
Lecture 29: Five Worlds

Nathan Brunelle

http://www.cs.uw.edu/332

(k-)CNF

- Conjunctive Normal Form (CNF) formula:
  - Logical AND of clauses
  - Each clause being an OR of variables
- k-CNF: Each clause has k variables

$$(x \vee y \vee z) \wedge (x \vee \bar{y} \vee y) \wedge (u \vee y \vee \bar{z}) \wedge (z \vee \bar{x} \vee u) \wedge (\bar{x} \vee \bar{y} \vee \bar{z})$$

Clause

Variables

1-SAT

- Given a 1-CNF formula (logical AND of clauses, each an OR of 1 variables), Is there an assignment of true/false to each variable to make the formula true?

$$(x) \land (y) \land (\bar{z}) \land (\bar{x})$$

1-SAT algorithm

Running Time:

2-SAT

- Given a 2-CNF formula (logical AND of clauses, each an OR of 2 variables), Is there an assignment of true/false to each variable to make the formula true?

$$(x \lor y) \land (x \lor \bar{y}) \land (y \lor \bar{z}) \land (z \lor u) \land (\bar{y} \lor \bar{z})$$

Clause

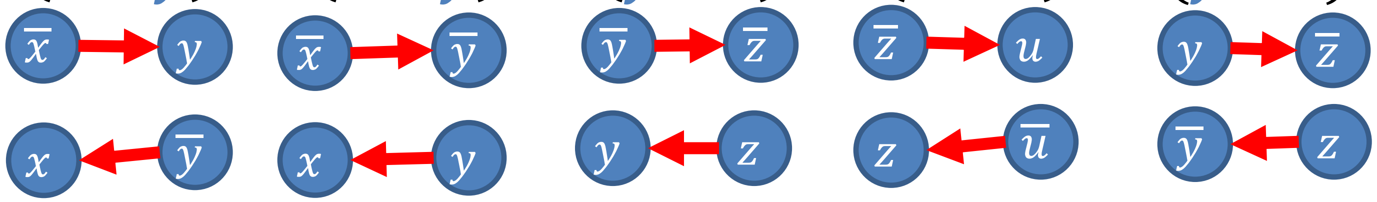
Variables

$$\boldsymbol{x = true}$$
$$\boldsymbol{y = false}$$
$$\boldsymbol{z = false}$$
$$\boldsymbol{u = true}$$

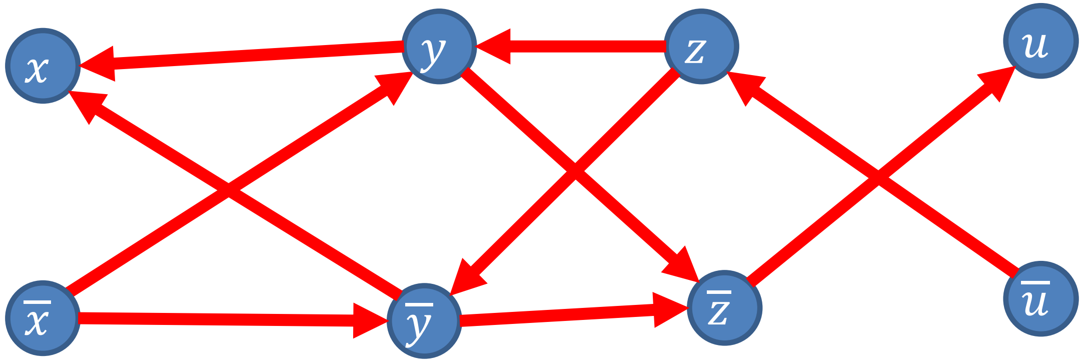
2-SAT in Polynomial Time

- Convert formula to an “implication graph”

$(x \lor y) \land (x \lor \bar{y}) \land (y \lor \bar{z}) \land (z \lor u) \land (\bar{y} \lor \bar{z})$



Are there any cycles with a variable and its negation?

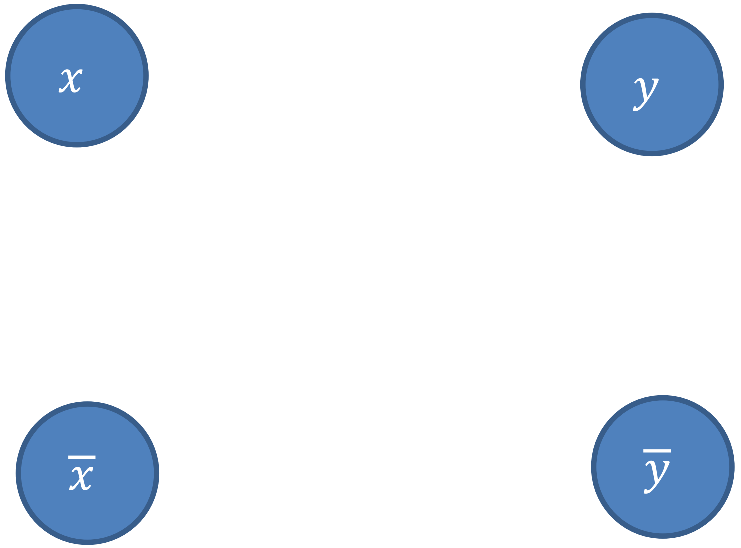


2-SAT in Polynomial Time

- Convert formula to an "implication graph"

$$(x \vee y) \wedge (\bar{x} \vee y) \wedge (x \vee \bar{y}) \wedge (\bar{x} \vee \bar{y})$$

Are there any cycles with a variable and its negation?



3-SAT

- Given a 3-CNF formula (logical AND of clauses, each an OR of 3 variables), Is there an assignment of true/false to each variable to make the formula true?

$$(x \lor y \lor z) \land (x \lor \bar{y} \lor y) \land (u \lor y \lor \bar{z}) \land (z \lor \bar{x} \lor u) \land (\bar{x} \lor \bar{y} \lor \bar{z})$$

Clause

Variables

$$\boldsymbol{x = true}$$
$$\boldsymbol{y = false}$$
$$\boldsymbol{z = false}$$
$$\boldsymbol{u = true}$$

3-SAT algorithm

- Given a 3-CNF formula with n variables and m clauses, try all combinations of True/False, check to see if any combinations evaluate to True.

3-SAT algorithm

- Given a 3-CNF formula with n variables and m clauses, try all combinations of True/False, check to see if any combinations evaluate to True.

Running Time: $O(2^n)$

Other ideas related to P and NP

- One-Way function
  - $f: \{0,1\}^{\mathrm{k}} \rightarrow \{0,1\}^k$ is a one-way function provided that there is an algorithm to compute $f(x)$ in polynomial time, but $f^{-1}(x)$ requires exponential time
  - Note that computing f^{-1} belongs to NP
    - To verify that $f^{-1}(x) = y$, compute $f(y)$
- Public Key Cryptography
  - Two keys: public key, private key
  - To encrypt a message: run $E(m, k_{pub})$ in polynomial time
  - To decrypt a ciphertext: run $D(c, k_{priv})$ in polynomial time
  - If you don't know the private key $D(c, k_{pub})$ requires exponential time
    - $E^{-1}(m, k_{pub})$ is $D(c, k_{pub})$, which we need to be a one-way function

Impagliazzo's 5 Worlds

Describes what computer science might look like depending on how certain open questions are answered.

- Algorithmica
- Heuristica
- Pessiland
- Minicrypt
- Cryptomania

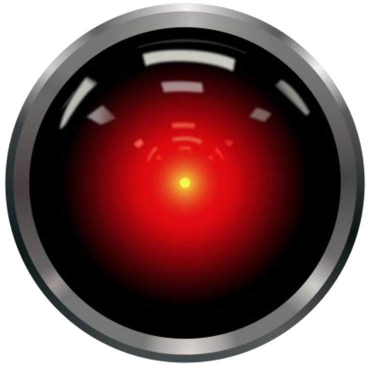
Gauss vs. Büttner

Büttner's goal: embarrass Gauss

Come up with a problem which Gauss finds difficult but Büttner can solve quickly

1. Come up with a 3-CNF formula and a satisfying assignment together
2. Give the formula to Gauss
3. When Gauss is stumped show the satisfying assignment

Algorithmica



P=NP

NP problems solvable efficiently

Gauss can quickly find the solution to Buttner’s problem

Gauss is not embarrassed – he can solve any problem Buttner gives

Advantages:

- VLSI Design
- Strong AI
- Cure for cancer?

Disadvantages:

- No privacy
- Computers take over

Heuristica

P≠NP in worst case, P=NP on average

Time to come up with a problem ≈ time to solve it

Büttner can give hard problems, but it’s hard to find them

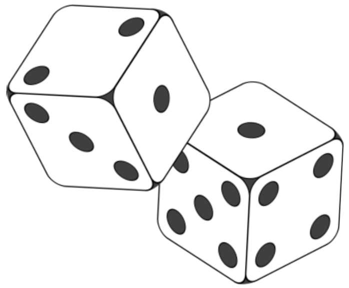
Gauss is not embarrassed – Most formulas Buttner gives are easy

Advantages:

- Maybe similar to Algorithmica
- Depends on real-world distributions

Disadvantages:

- Bad real world distributions could make things hard to solve



Pessiland

P≠NP on average, one-way functions don’t exist

Hard problems easy to find, but *solved* hard problems difficult to find

Gauss can be stumped, but Büttner does no better – The only way Buttner wins is to first find a hard problem, then solve it on his own.

Advantages:

- Universal Compression
- Derandomization
- Quantum computing doesn’t matter

Disadvantages:

- No crypto
- No algorithmic advantages
- Progress is slow

Minicrypt

One-way functions exist, no public key cryptography

Büttner can give hard problems to Gauss and also know their solutions

Gauss is embarrassed – Using a one-way function, Buttner can give $f(x)$ and ask Gauss to identify x

Advantages:

- Private key crypto
- Can prove identity (digital signatures)

Disadvantages:

- No electronic currencies

Cryptomania

Public Key Crypto Exists

Büttner can come up with problems and solutions, then share the solution with all other students

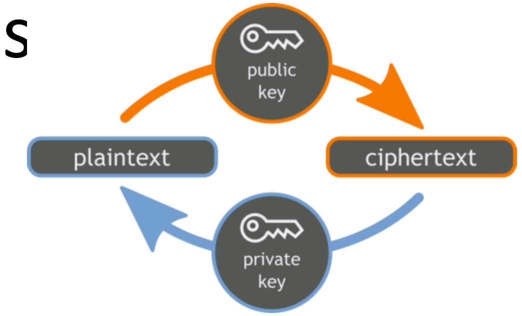
Gauss is very embarrassed – Buttner can share the private key with all students, then ask Gauss to decrypt ciphertexts. Gauss is the only one in the room who won't be able to do it.

Advantages:

- Secure computation
- Signatures
- Bitcoin, etc.

Disadvantages:

- Algorithmic progres will be slow



Does P=NP?

	P≠NP	P=NP	Ind	DC	DK	DK and DC	other
2002	61 (61%)	9 (9%)	4 (4%)	1 (1%)	22 (22%)	0 (0%)	3 (3%)
2012	126 (83%)	12 (9%)	5 (3%)	5 (3%)	1 (0.66%)	1 (0.66%)	1 (0.66%)
2019	109 (88%)	15 (12%)	0	0	0	0	0

When Will P=NP be resolved?

	02–09	10–19	20–29	30–39	40–49	50–59	60–69	70–79
2002	5 (5%)	12 (12%)	13 (13%)	10 (10%)	5 (5%)	12 (12%)	4 (4%)	0 (0%)
2012	0 (0%)	2 (1%)	17 (11%)	18 (12%)	5 (3%)	10 (6.5%)	10 (6.5%)	9 (6%)
2019	0 (0%)	0 (0%)	26 (22%)	20 (17%)	14 (12%)	9 (7%)	7 (6%)	5 (4%)

	80–89	90–99	100–109	110–119	150–159	2200–3000	4000–4100
2002	1 (1%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	5 (5%)	0 (0%)
2012	4 (3%)	5 (3%)	2 (1.2%)	5 (3%)	2 (1.2%)	3 (2%)	3 (2%)
2019	0 (0%)	0 (0%)	1 (0.8%)	10 (12%)	10 (12%)	1 (0.8%)	11 (9%)

	Long Time	Never	Don't Know	Sooner than 2100	Later than 2100
2002	0 (0%)	5 (5%)	21 (21%)	62 (62%)	17 (17%)
2012	22 (14%)	5 (3%)	8 (5%)	81 (53%)	63 (41%)
2019	7 (6%)	11 (9%)	0 (0%)	84 (66%)	40 (34%)

If P$\neq$NP, will that have large practical impact?

If P≠NP, will that have large practical impact?

116 responses.

- YES: 22 (19%)
- NO: 94 (81%)

Dmytro Taranovsky thinks yes:
Given enough time, fundamental breakthroughs tend to have a big practical impact.

Peter Gerdes thinks yes:
Well the proof won't but the fact that it's true will.

Hal Gabow thinks not:
We already have put our faith in P≠NP .

If P=NP, will that have large practical impact?

118 responses.

- YES: 68 (58%)
- NO: 50 (42%)

YES:

Dmytro Taranovsky:

While it is possible the solution will be ineffective, the consequences of a fully effective P=NP *would be enormous. It can lead to human immortality in 5 years, or if held secret by a power-seeking group, world government in 2 years.*

Peter Gerdes:

Indirectly, the proof will inevitably involve powerful ideas that will have an effect.

John Tromp:

Crypto will be all but dead. [Contrast this to Mitch Harris' NO answer.]

Scott Aaronson:

The practical impact would come not from the result itself, but from the new ideas needed to achieve it.

If P=NP, will that have large practical impact?

118 responses.

- YES: 68 (58%)
- NO: 50 (42%)

NO:

Richard Lorentz:

Probably not. I might be wrong but, e.g., I don't think putting linear programming in P *really had much of a practical effect.*

Clyde Kruskal:

There will probably be something special about NP*-complete problems that still makes them hard to solve.*

Lenwood Heath:

I believe that the problems that we have been kicking around for years as NP*-hard will still be hard to solve in some theoretically describable sense.*

Mitch Harris:

Only a small effect. The constants won't be huge, but physical limits to Moore's law will mean the cross over point is pretty impractical. Not galactic [Lipton and Regan in a Blog Post coined "Galactic" to mean an algorithm in poly time but you would never actually run it either due to large degree or large constants] but let's say interplanetary. Also the algorithms would be extremely non-trivial. As for cryptography, there will still be hard problems with one-way functions, just at the next higher level in the hierarchy. [Contrast with John Tromp's YES answer.]

OTHER:

András Salamon:

If someone produces an algorithm that decides SAT *in quadratic time, yes (because we already have efficient reductions to* SAT *for many problems of interest). If someone gives a nonconstructive proof, or one with a polynomial with degree that depends on the cardinality of some large finite group, not so much.*

Ryan Krusinga:

Some problems may just have ridiculously impractical polynomial-time solutions, even in the best case. Maybe there will be some creative algorithms that work some of the time, but I don't think most problems will be affected much.