Logic for Computer Science

Summer Semester
2019-2020

LECTURE 3

Lectures : LORENZO CLEMENTE
Tutorials : DARIA WALUKIEWICZ, JACEK CHRZĄSZCZ, JĘDRZEJ KOŁODZIEJSKI
Labs : DARIA, JACEK + PIOTR HOFMAN

PLAN

- SAT & CNF + applications.
- Unit propagation, Elimination of pure literals.
- Resolution ⟵ soundness, refutation completeness.
- Davis-Putnam algorithm DP '60: U + E + R.
- Splitting rule.
- Davis-Putnam-Loveland-Logeman DPLL '62: U + E + S.
- Modern SAT solvers. SAT competition.
- SAT phase transition.

THE SATISFIABILITY PROBLEM (SAT)

$$SAT(\varphi) \quad \text{iff} \quad \exists \rho : Var(\varphi) \to \{0,1\} \cdot [\![\varphi]\!]\rho = 1$$

Applications

- Validity: $\models \varphi$ iff not $SAT(\neg\varphi)$.
- Any NP problem: $L \in NP \Rightarrow L \leq_p SAT$.
  - Hardware & software verification
  - Cryptoanalysis.
  - Bioinformatics
  - Planning.
  - Factoring?
  - ⋮

SAT is the assembly programming of combinatorial problems

CONJUNCTIVE NORMAL FORM (CNF)

$$\underbrace{(p \vee q \vee \neg r)}_{\text{clause}} \wedge (\underbrace{\neg q}_{\text{literals}} \vee \underbrace{r}_{\text{literals}}) \wedge (p \vee r)$$

1) CNF formulas are expressively complete:

$\forall \varphi \cdot \exists \psi$ in CNF s.t. $\forall \rho \cdot [\![\varphi]\!]\rho = [\![\psi]\!]\rho$

ψ: exponential blow-up

2) Tseitin's method:

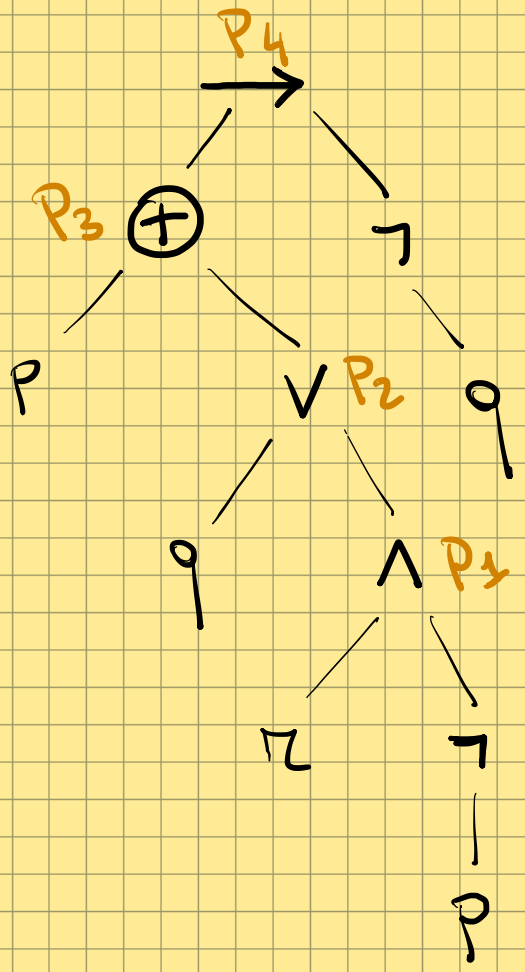
$\forall \varphi \exists \psi$ in CNF s.t. $SAT(\varphi)$ iff $SAT(\psi)$.

ψ: linear size

TSEITIN'S METHOD

INPUT: $(p \oplus (q \vee r \wedge \neg p)) \rightarrow \neg q$ (any connective allowed)

PARSE TREE



DEFINITIONS

$p_1 \leftrightarrow r \wedge \neg p$

$p_2 \leftrightarrow q \vee p_1$

$p_3 \leftrightarrow p \oplus p_2$

$p_4 \leftrightarrow p_3 \rightarrow \neg q$

Size 3

CNFs

ψ_1

ψ_2

ψ_3

ψ_4

Constant Size

OUTPUT

$p_1 \wedge \psi_1 \cdots \wedge \psi_4$

linear size

(I) UNIT PROPAGATION

$$p \wedge (p \vee \neg q) \wedge (\neg p \vee r)$$

p: unit clause (only one literal)

- It even preserves the semantics (Set of satisfying assignments).
- Always reduces the size of the formula.

(II) ELIMINATION of PURE LITERALS

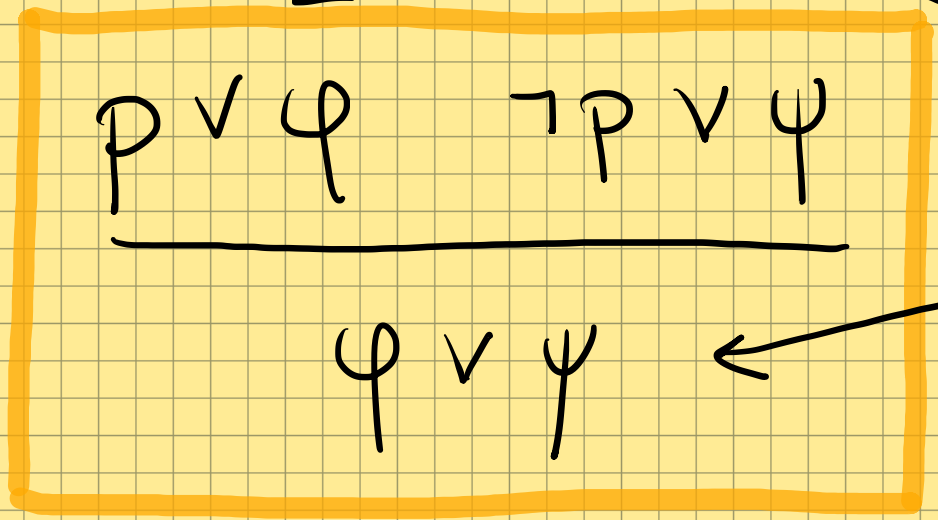
$$(p \vee q \vee \neg r) \wedge (\neg p \vee \neg r) \wedge (\neg q \vee s)$$

pure literal

(it appears either only positively or only negatively)

- It preserves satisfiability.
- Always reduces the size of the formula.

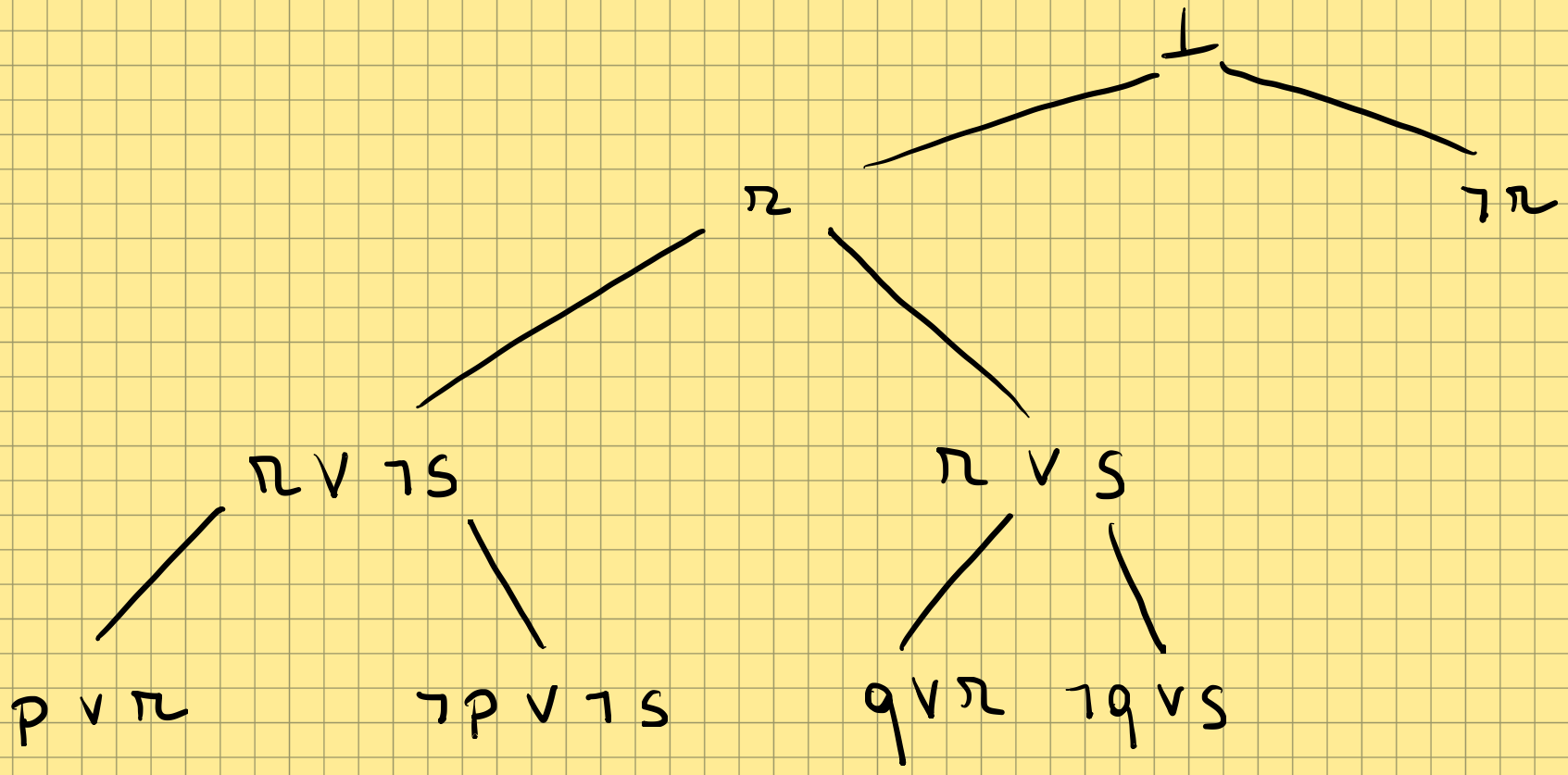
(III) RESOLUTION



(Generalisation of modus ponens: $\dfrac{p \quad p \to \psi}{\psi}$ (when applied to clauses))

- $\Delta \vdash \varphi$ iff there is a proof of clause φ from clauses Δ:
  $\exists$ sequence of formulas $\varphi_1, \ldots, \varphi_m \equiv \varphi$ and $\forall\, 1 \leq i \leq m$
  either $\varphi_i \in \Delta$ or $\exists j, k < i \,.\, (R)\ \dfrac{\varphi_j \quad \varphi_k}{\varphi_i}$
- REFUTATION: proof of $\bot$ (empty clause).

REFUTATION EXAMPLE



CNF $\varphi \equiv (p \vee r) \wedge (\neg p \vee \neg s) \wedge (q \vee r) \wedge (\neg q \vee s) \wedge \neg r$

SOUNDNESS of RESOLUTION

$$\Delta \vdash \varphi \quad \text{implies} \quad \Delta \models \varphi$$

Proof: By complete induction on the length of proofs.

Let φ have a proof of length n $\varphi_1, \ldots, \varphi_n \equiv \varphi$.

Assume $\forall \psi \in \Delta \cdot [\![\psi]\!] \rho = 1$. Two cases:

1) $\varphi \in \Delta$: then $[\![\varphi]\!] \rho = 1$ by assumption.

2) $\dfrac{\varphi_i \quad \varphi_j}{\varphi}$, $i, j < n$: By induction, $[\![\varphi_i]\!] \rho = [\![\varphi_j]\!] \rho = 1$.

$\varphi_i \equiv p \vee \psi$, $\varphi_j \equiv \neg p \vee \theta$, $\varphi \equiv \psi \vee \theta$.

Then, $[\![\varphi]\!] \rho = 1$.

IS RESOLUTION COMPLETE?

$\Delta \models \varphi$ implies $\Delta \vdash \varphi$

No! Counterexample: $p \not\models p \vee q$.

$\Rightarrow$ Resolution cannot produce all logical consequences of Δ.

However, if $\perp$ is a logical consequence, then it will prove it.

RESOLUTION is REFUTATION COMPLETE

unsat ↪ $\Delta \models \bot$ implies $\Delta \vdash \bot$ ↩ refutation

- This is sufficient: $\Delta \models \varphi$ iff $\Delta \cup \{\neg\varphi\} \not\models \bot$.

RESOLUTION is REFUTATION COMPLETE

unsat

$\Delta \models \bot$ implies $\Delta \vdash \bot$

refutation

Proof: By induction on the number of variables of Δ.

Can assume:

1) $\bot \notin \Delta$ (otherwise we are done).

2) Δ does not contain any tautological clause (can remove them preserving unsatisfiability).

RESOLUTION is REFUTATION COMPLETE

unsat ↪ $\Delta \models \bot$ implies $\Delta \vdash \bot$ ← refutation

Proof: By induction on the number of variables of Δ.

There is a variable p appearing positively & negatively.

$$\Delta = \Delta_p \cup \Delta_{\neg p} \cup \Delta'.$$ (disjoint)

Take $\Gamma := \{\varphi \vee \psi \mid p \vee \varphi \in \Delta_p, \neg p \vee \psi \in \Delta_{\neg p}\} \cup \Delta'$

By soundness, $\Gamma \models \bot$. Γ does not contain p.

By induction, $\Gamma \vdash \bot$, and thus $\Delta \vdash \bot$.

DAVIS-PUTNAM ALGORITHM '60

1. If $\varphi \equiv \top$, then return SAT.
2. If $\bot \in \varphi$, then return UNSAT.
3. Remove tautological clauses $(\ldots \vee p \vee \ldots \vee \neg p \vee \ldots) \in \varphi$.
4. Apply (I) UNIT: If changed, GOTO 1. }
5. Apply (II) PURE: If changed, GOTO 1. }
6. Choose p appearing positively and negatively:

{ Apply (III) RESOLUTION globally, GOTO 1.

BFS { $\varphi := \{\psi \vee \theta \mid p \vee \psi, \neg p \vee \theta \in \varphi\} \cup \{\psi \in \varphi \mid p, \neg p \notin \psi\}$

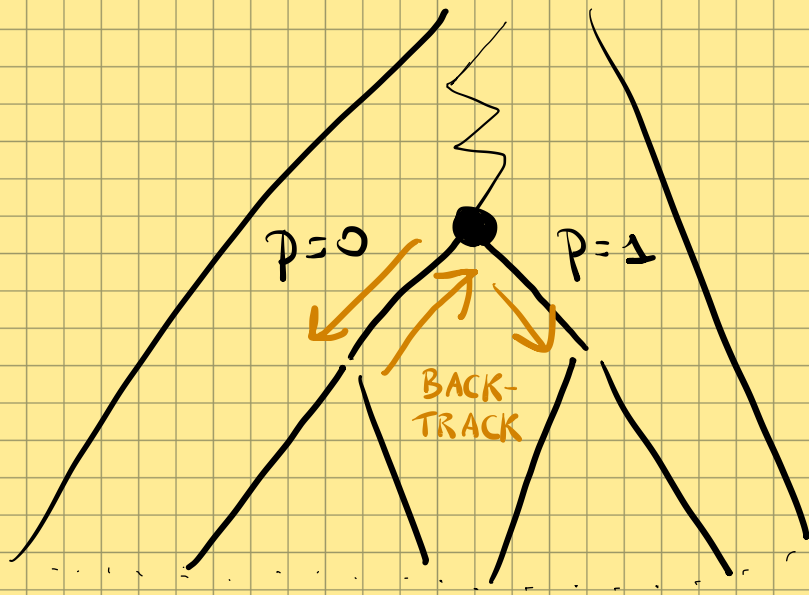
Removes p entirely. High memory consumption!

DAVIS-LOVELAND-LOGEMANN (DPLL) '62

1. If $\varphi \equiv \top$, then return SAT.
2. If $\bot \in \varphi$, then BACKTRACK.
3. Remove tautological clauses $(\ldots \vee p \vee \ldots \vee \neg p \vee \ldots) \in \varphi$.
4. Apply (I) UNIT: If changed, GOTO 1. }
5. Apply (II) PURE: If changed, GOTO 1. }
6. Choose p appearing positively and negatively:

DFS {
- Branch — $\varphi :\equiv p \wedge \varphi$. / $\varphi :\equiv \neg p \wedge \varphi$. (SPLITTING RULE)
- Return UNSAT.

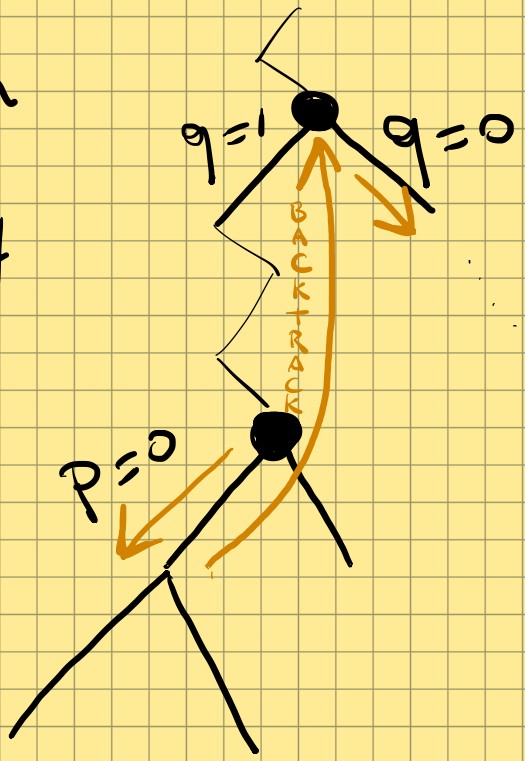
MODERN SAT SOLVERS



- Chronological backtracking (DPLL)

- Nonchronological backtracking:

  Conflict-driven clause learning (CDCL)

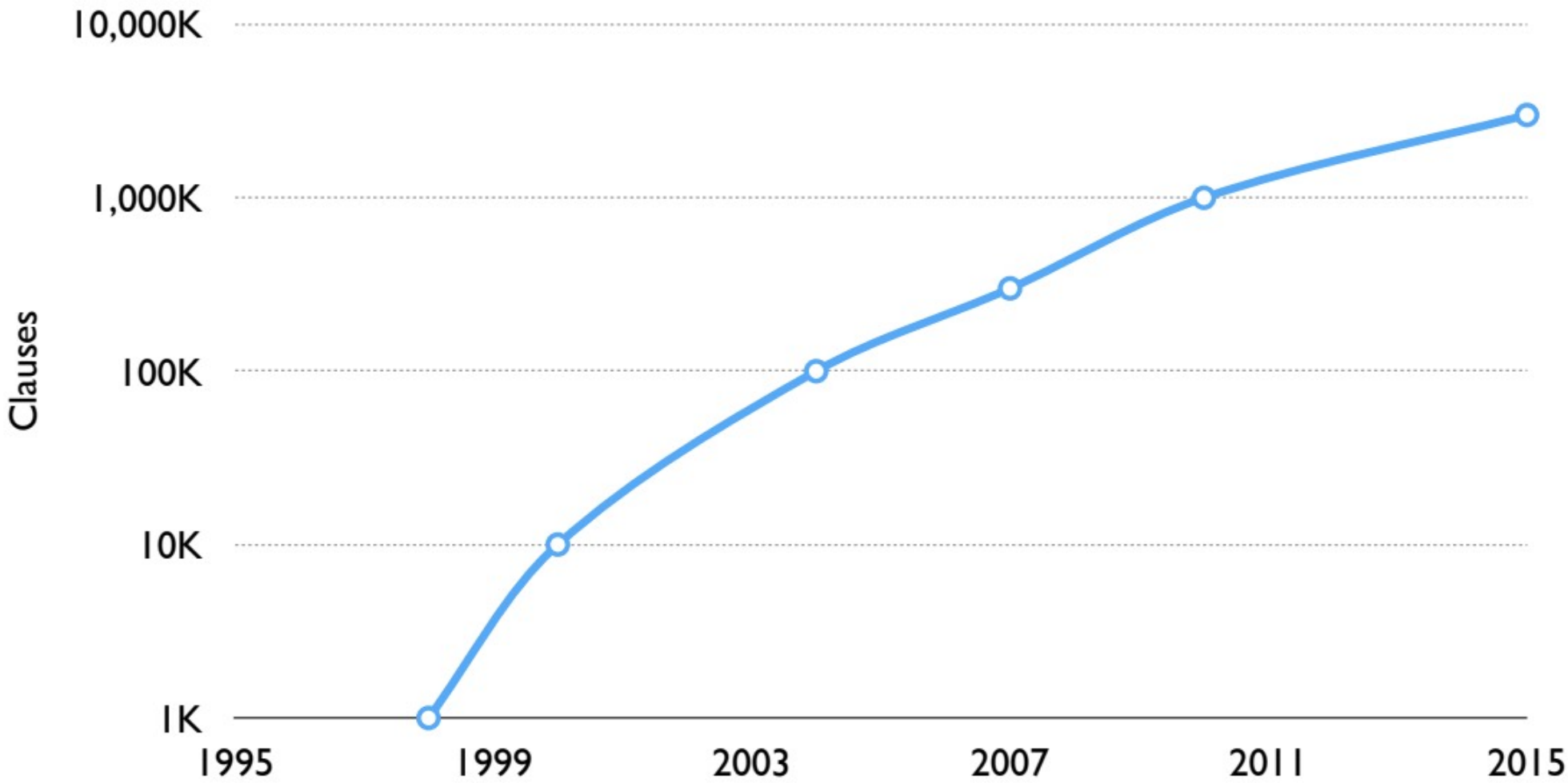


Setting p=0 generates a conflict, backtrack to the oldest "cause" of the conflict (implication graph)

MODERN SAT SOLVERS

- Lazy data structures (more efficient).
- Random search restarts (may become incomplete)
- Heuristics
- Examples: Minisat, zChaff, Z3, ...

IS SAT EASY OR HARD?



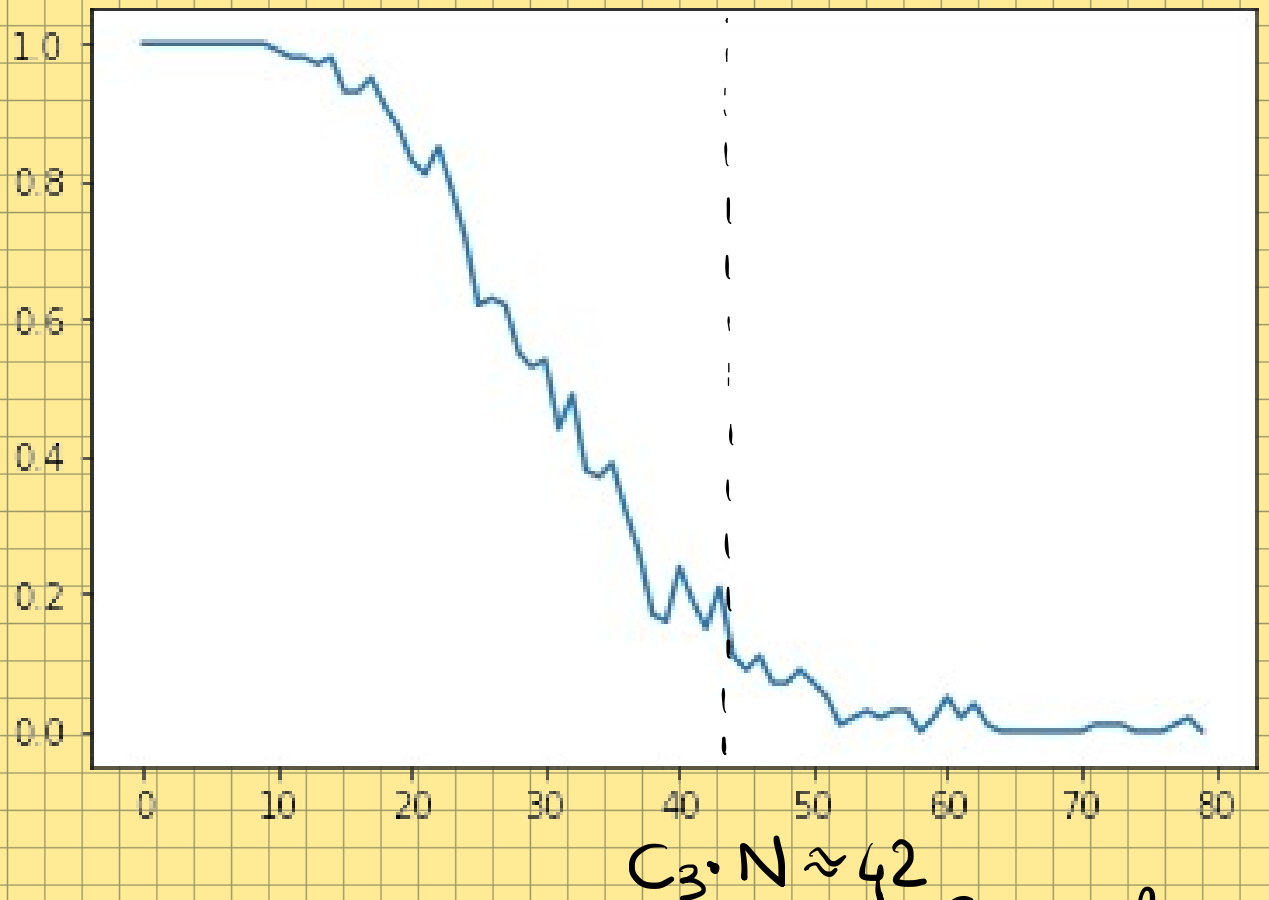
Based on a slide from Vijay Ganesh + Emina Torlak

SAT: PHASE TRANSITION

Gent & Walsh '94

K = 3SAT, N = 10 variables, 100 samples per point

$\frac{\#SAT}{100}$



$c_2 = 1$

$c_3 \approx 4.24$

experimentally

$c_3 \cdot N \approx 42$

$\#$ clauses L

$$\lim_{N\to\infty} P(k\text{-SAT}(N, \underbrace{c \cdot N}_{L})) = \begin{cases} 0 & \text{for } c > c_k \\ 1 & \text{for } c < c_k \end{cases}$$

RESOLUTION ⇒ POLYNOMIAL INTERPOLANTS

Recall: $\varphi \wedge \psi \vdash \bot$ implies $\exists$ interpolant θ:

$$\varphi \vdash \theta \;,\; \theta \wedge \psi \vdash \bot \;,\; Var(\theta) \subseteq Var(\varphi) \cap Var(\psi).$$

let $\varphi \equiv \varphi_1 \wedge \cdots \wedge \varphi_m$, $\psi \equiv \psi_1 \wedge \cdots \wedge \psi_n$ in CNF.

hit: $\varphi_i[\bot]$, $\psi_j[\top]$. Enhanced resolution rule:

$$\frac{p \vee \varphi'[\theta] \qquad \neg p \vee \psi'[\theta']}{\varphi' \vee \psi' \;\; [\theta'']}\;,\quad \theta'' \equiv \begin{cases} \theta \vee \theta' & \text{if } p \in Var(\varphi) \setminus Var(\psi) \\ \theta \wedge \theta' & \text{if } p \in Var(\psi) \setminus Var(\varphi) \\ (p \vee \theta) \wedge (\neg p \vee \theta') & \text{otherwise} \end{cases}$$

invariant: $\vdash \varphi'[\theta]$ implies $\varphi \wedge \neg\varphi' \vdash \theta$, $\theta \wedge \psi \vdash \varphi'$

preinterpolant ↗ (when $\varphi' \equiv \bot$ we get an interpolant)

EXAMPLE

$\varphi \equiv (p \vee q) \wedge \neg q$

$\psi \equiv (\neg p \vee r) \wedge \neg r$

$$\dfrac{\dfrac{p \vee q\ [\bot] \quad \neg q\ [\bot]}{p\ [\bot \vee \bot]} \qquad \dfrac{\neg p \vee r\ [\top] \quad \neg r\ [\top]}{\neg p\ [\top \wedge \top]}}{\bot\ [\underbrace{(p \vee \bot) \wedge (\neg p \vee \top)}_{p}]}$$

Thus $\Theta \equiv p$ is an interpolant:

$\varphi \vdash p$ and $\psi \wedge p \vdash \bot$