CS:4350 Logic in Computer Science

Satisfiability and Randomization

Cesare Tinelli

Spring 2021

The University of Iowa

Credits

These slides are largely based on slides originally developed by **Andrei Voronkov** at the University of Manchester. Adapted by permission.

Outline

Satisfiability and Randomisation

- Randomly Generated Clause Sets
- Sharp Phase Transition
- Randomised Algoritms for Satisfiability-Checking

Random Clause Generation

How can one generate a random clause?

Random Clause Generation

How can one generate a random clause?

Let's first generate a *random literal*

Random Clause Generation

How can one generate a random clause?

Let's first generate a *random literal*

- Fix a number n of boolean variables

Random Clause Generation

How can one generate a random clause?

Let's first generate a *random literal*

- Fix a number n of boolean variables
- Select a literal among $p_1, \ldots, p_n, \neg p_1, \ldots, \neg p_n$ with an equal probability

Random Clause Generation

How can one generate a random clause?

Let's first generate a *random literal*

A *random clause* is a disjunctioin of random literals

- Fix a number n of boolean variables
- Select a literal among $p_1, \ldots, p_n, \neg p_1, \ldots, \neg p_n$ with an equal probability

Random Clause Generation

How can one generate a random clause?

Let's first generate a *random literal*

A *random clause* is a disjunctioin of random literals

- Fix a number n of boolean variables
- Select a literal among $p_1, \ldots, p_n, \neg p_1, \ldots, \neg p_n$ with an equal probability
- Fix the length k of the clause

Random Clause Generation

How can one generate a random clause?

Let's first generate a *random literal*

A *random clause* is a disjunctioin of random literals

- Fix a number n of boolean variables
- Select a literal among $p_1, \ldots, p_n, \neg p_1, \ldots, \neg p_n$ with an equal probability
- Fix the length k of the clause

Suppose we generate random clauses one after one

How does the set of models of this set change?

SAT and k-SAT

SAT is satisfiability checking for sets of clauses

k-SAT is satisfiability checking for sets of clauses of length k or less

SAT and *k*-SAT

SAT is satisfiability checking for sets of clauses

k-SAT is satisfiability checking for sets of clauses of length *k* or less

- SAT is NP-complete
- 2-SAT is decidable in linear time
- 3-SAT is NP-complete!

SAT and k-SAT

SAT is satisfiability checking for sets of clauses

k-SAT is satisfiability checking for sets of clauses of length k or less

- SAT is NP-complete
- 2-SAT is decidable in linear time
- 3-SAT is NP-complete!

SAT and k-SAT

SAT is satisfiability checking for sets of clauses

k-SAT is satisfiability checking for sets of clauses of length k or less

- SAT is NP-complete
- 2-SAT is decidable in linear time
- 3-SAT is NP-complete!

3-SAT is all you need

There is a simple reduction of SAT to 3-SAT based, again on naming:

1. Take a clause having more than 3 literals:

   $$L_1 \vee L_2 \vee L_3 \vee L_4 \vee \cdots \vee L_n$$

   and replace it by two clauses:

   $$L_1 \vee L_2 \vee n$$
   $$\neg n \vee L_3 \vee L_4 \vee \cdots \vee L_n$$

   where n is a fresh variable

2. Repeat until all clauses have at most 3 literals

3-SAT is all you need

There is a simple reduction of SAT to 3-SAT based, again on naming:

1. Take a clause having more than 3 literals:

$$L_1 \vee L_2 \vee L_3 \vee L_4 \vee \cdots \vee L_n$$

   and replace it by two clauses:

$$\begin{array}{l} L_1 \vee L_2 \vee n \\ \neg n \vee L_3 \vee L_4 \vee \cdots \vee L_n \end{array}$$

   where n is a fresh variable

2. Repeat until all clauses have at most 3 literals

3-SAT is all you need

There is a simple reduction of SAT to 3-SAT based, again on naming:

1. Take a clause having more than 3 literals:

   $$L_1 \vee L_2 \vee L_3 \vee L_4 \vee \cdots \vee L_n$$

   and replace it by two clauses:

   $$L_1 \vee L_2 \vee n$$
   $$\neg n \vee L_3 \vee L_4 \vee \cdots \vee L_n$$

   where n is a fresh variable

2. Repeat until all clauses have at most 3 literals

Example (obtained by a program) for $n = 5$ and $k = 2$

p_1	p_2	p_3	p_4	p_5
0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	0	1	0	0
0	0	1	0	1
0	0	1	1	0
0	0	1	1	1
0	1	0	0	0
0	1	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	0
0	1	1	0	1
0	1	1	1	0
0	1	1	1	1

p_1	p_2	p_3	p_4	p_5
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1
1	1	1	0	0
1	1	1	0	1
1	1	1	1	0
1	1	1	1	1

Number of models for 5 vars: 32

Example (obtained by a program) for $n = 5$ and $k = 2$

$\neg p_2 \vee \neg p_3$

p_1	p_2	p_3	p_4	p_5
0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	0	1	0	0
0	0	1	0	1
0	0	1	1	0
0	0	1	1	1
0	1	0	0	0
0	1	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	0
0	1	1	0	1
0	1	1	1	0
0	1	1	1	1

p_1	p_2	p_3	p_4	p_5
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1
1	1	1	0	0
1	1	1	0	1
1	1	1	1	0
1	1	1	1	1

Number of models for 5 vars: 32

Example (obtained by a program) for $n = 5$ and $k = 2$

$\neg p_2 \vee \neg p_3$

p_1	p_2	p_3	p_4	p_5
0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	0	1	0	0
0	0	1	0	1
0	0	1	1	0
0	0	1	1	1
0	1	0	0	0
0	1	0	0	1
0	1	0	1	0
0	1	0	1	1

p_1	p_2	p_3	p_4	p_5
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1

Number of models for 5 vars: 24

Example (obtained by a program) for $n = 5$ and $k = 2$

$\neg p_2 \vee \neg p_3$
$\neg p_2 \vee p_1$

p_1	p_2	p_3	p_4	p_5
0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	0	1	0	0
0	0	1	0	1
0	0	1	1	0
0	0	1	1	1
0	1	0	0	0
0	1	0	0	1
0	1	0	1	0
0	1	0	1	1

p_1	p_2	p_3	p_4	p_5
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1

Number of models for 5 vars: 24

Example (obtained by a program) for $n = 5$ and $k = 2$

$\neg p_2 \vee \neg p_3$
$\neg p_2 \vee p_1$

p_1	p_2	p_3	p_4	p_5
0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	0	1	0	0
0	0	1	0	1
0	0	1	1	0
0	0	1	1	1

p_1	p_2	p_3	p_4	p_5
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1

Number of models for 5 vars: 20

Example (obtained by a program) for $n = 5$ and $k = 2$

$\neg p_2 \vee \neg p_3$
$\neg p_2 \vee p_1$
$\neg p_2 \vee p_2$

p_1	p_2	p_3	p_4	p_5
0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	0	1	0	0
0	0	1	0	1
0	0	1	1	0
0	0	1	1	1

p_1	p_2	p_3	p_4	p_5
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1

Number of models for 5 vars: 20

Example (obtained by a program) for $n = 5$ and $k = 2$

$\neg p_2 \lor \neg p_3$
$\neg p_2 \lor p_1$
$\neg p_2 \lor p_2$
$p_1 \lor p_1$

p_1	p_2	p_3	p_4	p_5
0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	0	1	0	0
0	0	1	0	1
0	0	1	1	0
0	0	1	1	1

p_1	p_2	p_3	p_4	p_5
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1

Number of models for 5 vars: 20

Example (obtained by a program) for $n = 5$ and $k = 2$

p_1	p_2	p_3	p_4	p_5

$\neg p_2 \vee \neg p_3$
$\neg p_2 \vee p_1$
$\neg p_2 \vee p_2$
$p_1 \vee p_1$

p_1	p_2	p_3	p_4	p_5
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1

Number of models for 5 vars: 12

Example (obtained by a program) for $n = 5$ and $k = 2$

p_1	p_2	p_3	p_4	p_5

$\neg p_2 \vee \neg p_3$
$\neg p_2 \vee p_1$
$\neg p_2 \vee p_2$
$p_1 \vee p_1$
$\neg p_5 \vee p_5$

p_1	p_2	p_3	p_4	p_5
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1

Number of models for 5 vars: 12

Example (obtained by a program) for $n = 5$ and $k = 2$

p_1	p_2	p_3	p_4	p_5

$\neg p_2 \vee \neg p_3$
$\neg p_2 \vee p_1$
$\neg p_2 \vee p_2$
$p_1 \vee p_1$
$\neg p_5 \vee p_5$
$p_4 \vee p_5$

p_1	p_2	p_3	p_4	p_5
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1

Number of models for 5 vars: 12

Example (obtained by a program) for $n = 5$ and $k = 2$

$\neg p_2 \vee \neg p_3$
$\neg p_2 \vee p_1$
$\neg p_2 \vee p_2$
$p_1 \vee p_1$
$\neg p_5 \vee p_5$
$p_4 \vee p_5$

p_1	p_2	p_3	p_4	p_5

p_1	p_2	p_3	p_4	p_5
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1

Number of models for 5 vars: 9

Example (obtained by a program) for $n = 5$ and $k = 2$

$\neg p_2 \vee \neg p_3$
$\neg p_2 \vee p_1$
$\neg p_2 \vee p_2$
$p_1 \vee p_1$
$\neg p_5 \vee p_5$
$p_4 \vee p_5$
$\neg p_5 \vee \neg p_3$

p_1	p_2	p_3	p_4	p_5

p_1	p_2	p_3	p_4	p_5
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1

Number of models for 5 vars: 9

Example (obtained by a program) for $n = 5$ and $k = 2$

$\neg p_2 \vee \neg p_3$
$\neg p_2 \vee p_1$
$\neg p_2 \vee p_2$
$p_1 \vee p_1$
$\neg p_5 \vee p_5$
$p_4 \vee p_5$
$\neg p_5 \vee \neg p_3$

p_1	p_2	p_3	p_4	p_5

p_1	p_2	p_3	p_4	p_5
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1

Number of models for 5 vars: 7

Example (obtained by a program) for $n = 5$ and $k = 2$

p_1	p_2	p_3	p_4	p_5

$\neg p_2 \vee \neg p_3$

$\neg p_2 \vee p_1$

$\neg p_2 \vee p_2$

$p_1 \vee p_1$

$\neg p_5 \vee p_5$

$p_4 \vee p_5$

$\neg p_5 \vee \neg p_3$

$p_2 \vee \neg p_4$

p_1	p_2	p_3	p_4	p_5
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1

Number of models for 5 vars: 7

Example (obtained by a program) for $n = 5$ and $k = 2$

p_1	p_2	p_3	p_4	p_5

$\neg p_2 \vee \neg p_3$
$\neg p_2 \vee p_1$
$\neg p_2 \vee p_2$
$p_1 \vee p_1$
$\neg p_5 \vee p_5$
$p_4 \vee p_5$
$\neg p_5 \vee \neg p_3$
$p_2 \vee \neg p_4$

p_1	p_2	p_3	p_4	p_5
1	0	0	0	1
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1

Number of models for 5 vars: 4

Example (obtained by a program) for $n = 5$ and $k = 2$

$\neg p_2 \vee \neg p_3$
$\neg p_2 \vee p_1$
$\neg p_2 \vee p_2$
$p_1 \vee p_1$
$\neg p_5 \vee p_5$
$p_4 \vee p_5$
$\neg p_5 \vee \neg p_3$
$p_2 \vee \neg p_4$
$p_5 \vee \neg p_2$

p_1	p_2	p_3	p_4	p_5

p_1	p_2	p_3	p_4	p_5
1	0	0	0	1
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1

Number of models for 5 vars: 4

Example (obtained by a program) for $n = 5$ and $k = 2$

p_1	p_2	p_3	p_4	p_5

$\neg p_2 \vee \neg p_3$
$\neg p_2 \vee p_1$
$\neg p_2 \vee p_2$
$p_1 \vee p_1$
$\neg p_5 \vee p_5$
$p_4 \vee p_5$
$\neg p_5 \vee \neg p_3$
$p_2 \vee \neg p_4$
$p_5 \vee \neg p_2$

p_1	p_2	p_3	p_4	p_5
1	0	0	0	1
1	1	0	0	1
1	1	0	1	1

Number of models for 5 vars: 3

Example (obtained by a program) for $n = 5$ and $k = 2$

p_1	p_2	p_3	p_4	p_5

$\neg p_2 \vee \neg p_3$
$\neg p_2 \vee p_1$
$\neg p_2 \vee p_2$
$p_1 \vee p_1$
$\neg p_5 \vee p_5$
$p_4 \vee p_5$
$\neg p_5 \vee \neg p_3$
$p_2 \vee \neg p_4$
$p_5 \vee \neg p_2$
$p_5 \vee p_2$

p_1	p_2	p_3	p_4	p_5
1	0	0	0	1
1	1	0	0	1
1	1	0	1	1

Number of models for 5 vars: 3

Example (obtained by a program) for $n = 5$ and $k = 2$

$\neg p_2 \lor \neg p_3$
$\neg p_2 \lor p_1$
$\neg p_2 \lor p_2$
$p_1 \lor p_1$
$\neg p_5 \lor p_5$
$p_4 \lor p_5$
$\neg p_5 \lor \neg p_3$
$p_2 \lor \neg p_4$
$p_5 \lor \neg p_2$
$p_5 \lor p_2$

p_1	p_2	p_3	p_4	p_5

p_1	p_2	p_3	p_4	p_5
1	0	0	0	1

Number of models for 5 vars: 1

Example (obtained by a program) for $n = 5$ and $k = 2$

$\neg p_2 \vee \neg p_3$
$\neg p_2 \vee p_1$
$\neg p_2 \vee p_2$
$p_1 \vee p_1$
$\neg p_5 \vee p_5$
$p_4 \vee p_5$
$\neg p_5 \vee \neg p_3$
$p_2 \vee \neg p_4$
$p_5 \vee \neg p_2$
$p_5 \vee p_2$
$\neg p_1 \vee \neg p_4$

p_1	p_2	p_3	p_4	p_5

p_1	p_2	p_3	p_4	p_5
1	0	0	0	1

Number of models for 5 vars: 1

Example (obtained by a program) for $n = 5$ and $k = 2$

	p_1	p_2	p_3	p_4	p_5
$\neg p_2 \lor \neg p_3$					
$\neg p_2 \lor p_1$					
$\neg p_2 \lor p_2$					
$p_1 \lor p_1$					
$\neg p_5 \lor p_5$					
$p_4 \lor p_5$					
$\neg p_5 \lor \neg p_3$					
$p_2 \lor \neg p_4$					
$p_5 \lor \neg p_2$					
$p_5 \lor p_2$					
$\neg p_1 \lor \neg p_4$					
$p_5 \lor p_2$					

p_1	p_2	p_3	p_4	p_5
1	0	0	0	1

Number of models for 5 vars: 1

Example (obtained by a program) for $n = 5$ and $k = 2$

$\neg p_2 \vee \neg p_3$
$\neg p_2 \vee p_1$
$\neg p_2 \vee p_2$
$p_1 \vee p_1$
$\neg p_5 \vee p_5$
$p_4 \vee p_5$
$\neg p_5 \vee \neg p_3$
$p_2 \vee \neg p_4$
$p_5 \vee \neg p_2$
$p_5 \vee p_2$
$\neg p_1 \vee \neg p_4$
$p_5 \vee p_2$
$\neg p_1 \vee \neg p_5$

p_1	p_2	p_3	p_4	p_5

p_1	p_2	p_3	p_4	p_5
1	0	0	0	1

Number of models for 5 vars: 1

Example (obtained by a program) for $n = 5$ and $k = 2$

p_1	p_2	p_3	p_4	p_5

p_1	p_2	p_3	p_4	p_5

$\neg p_2 \vee \neg p_3$
$\neg p_2 \vee p_1$
$\neg p_2 \vee p_2$
$p_1 \vee p_1$
$\neg p_5 \vee p_5$
$p_4 \vee p_5$
$\neg p_5 \vee \neg p_3$
$p_2 \vee \neg p_4$
$p_5 \vee \neg p_2$
$p_5 \vee p_2$
$\neg p_1 \vee \neg p_4$
$p_5 \vee p_2$
$\neg p_1 \vee \neg p_5$

This set of 13 clauses is unsatisfiable

Number of models for 5 vars: 0

Random Clause Generation

What is the probability that a set of clauses of a given size is unsatisfiable?

Random Clause Generation

What is the probability that a set of clauses of a given size is unsatisfiable?

Fix:

- number n of propositional variables

Random Clause Generation

What is the probability that a set of clauses of a given size is unsatisfiable?

Fix:

- number n of propositional variables
- number k of literals per clause, so we will generate k-SAT instances

Random Clause Generation

What is the probability that a set of clauses of a given size is unsatisfiable?

Fix:

- number n of propositional variables
- number k of literals per clause, so we will generate k-SAT instances
- number m of the clauses

Random Clause Generation

What is the probability that a set of clauses of a given size is unsatisfiable?

Fix:

- number n of propositional variables
- number k of literals per clause, so we will generate k-SAT instances
- number m of the clauses

Generate m clauses, each with k literals chosen randomly with an equal probability from $\{p_1, \ldots, p_n, \neg p_1, \ldots, \neg p_n\}$

Random Clause Generation

What is the probability that a set of clauses of a given size is unsatisfiable?

Fix:

- number n of propositional variables
- number k of literals per clause, so we will generate k-SAT instances
- number m of the clauses

Generate m clauses, each with k literals chosen randomly with an equal probability from $\{p_1, \ldots, p_n, \neg p_1, \ldots, \neg p_n\}$

Note: Probability is a monotonic function: the larger the clause set, the higher the probability that it is unsatisfiable

Random Clause Generation

What is the probability that a set of clauses of a given size is unsatisfiable?

Fix:

- number n of propositional variables
- number k of literals per clause, so we will generate k-SAT instances
- real number r: ratio of clauses per variable

Generate $[r \cdot n]$ clauses, each with k literals chosen randomly with an equal probability from $\{p_1, \ldots, p_n, \neg p_1, \ldots, \neg p_n\}$

Note: Probability is a monotonic function: the larger the clause set, the higher the probability that it is unsatisfiable

Roulette

We will generate random instances of 2-SAT with 5-variables

SAT Roulette

We will generate random instances of 2-SAT with 5-variables

You will bet on whether the resulting set of clauses is **satisfiable** or **unsatisfiable**

SAT Roulette

We will generate random instances of 2-SAT with 5-variables

You will bet on whether the resulting set of clauses is **satisfiable** or **unsatisfiable**

- What will you bet on if we generate 5 clauses?
- 100 clauses?
- 15 clauses?

SAT Roulette

We will generate random instances of 2-SAT with 5-variables

You will bet on whether the resulting set of clauses is **satisfiable** or **unsatisfiable**

- What will you bet on if we generate 5 clauses?
- 100 clauses?
- 15 clauses?

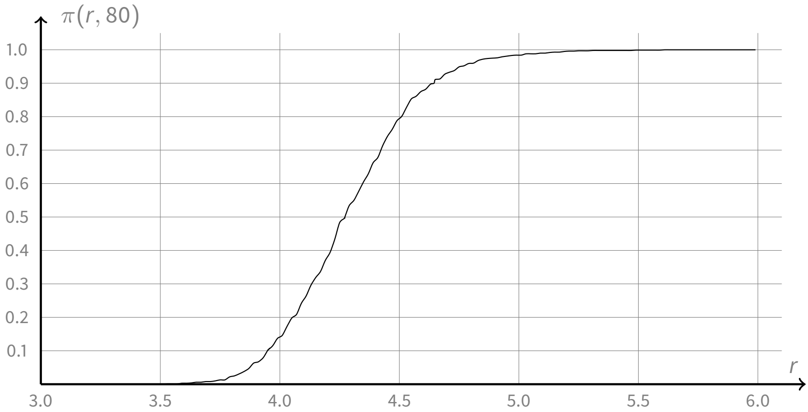
SAT Roulette

We will generate random instances of 2-SAT with 5-variables

You will bet on whether the resulting set of clauses is **satisfiable** or **unsatisfiable**

- What will you bet on if we generate 5 clauses?
- 100 clauses?
- 15 clauses?

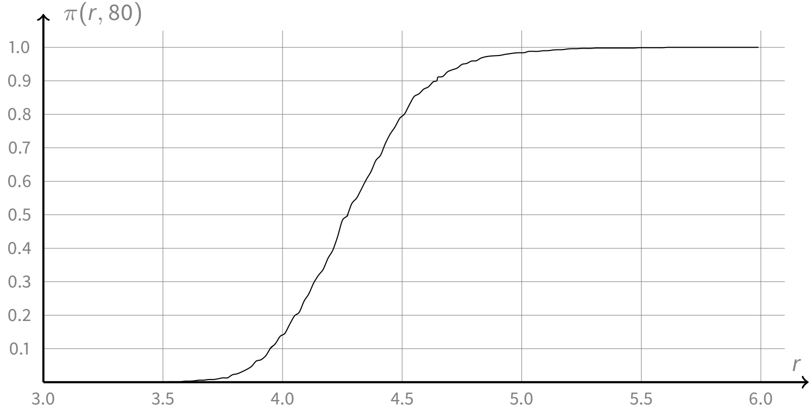
Probability of obtaining an unsatisfiable set



$\pi(r, n) =$ prob. that a randomly generated set of [rn] 3-clauses over n variables is unsat

Probability of obtaining an unsatisfiable set

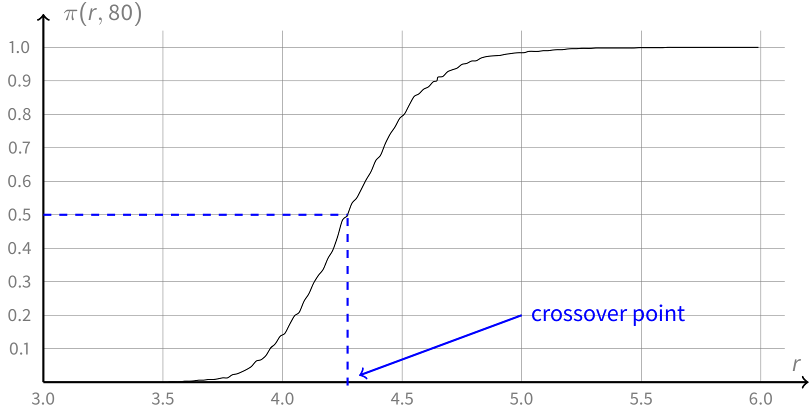
Crossover point: the value of r at which the probability crosses 0.5



$\pi(r, n) =$ prob. that a randomly generated set of [rn] 3-clauses over n variables is unsat

Probability of obtaining an unsatisfiable set

Crossover point: the value of r at which the probability crosses 0.5



$\pi(r, n) =$ prob. that a randomly generated set of [rn] 3-clauses over n variables is unsat

ϵ-window

For any (small) number $\epsilon > 0$, the ϵ-*window* is the interval of values of r where

$$\epsilon \leq \pi(r, n) \leq 1 - \epsilon$$

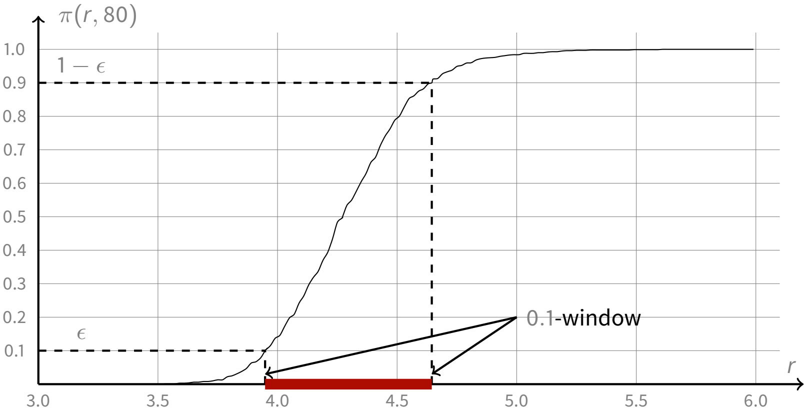
Example $\epsilon = 0.1$

ϵ-window

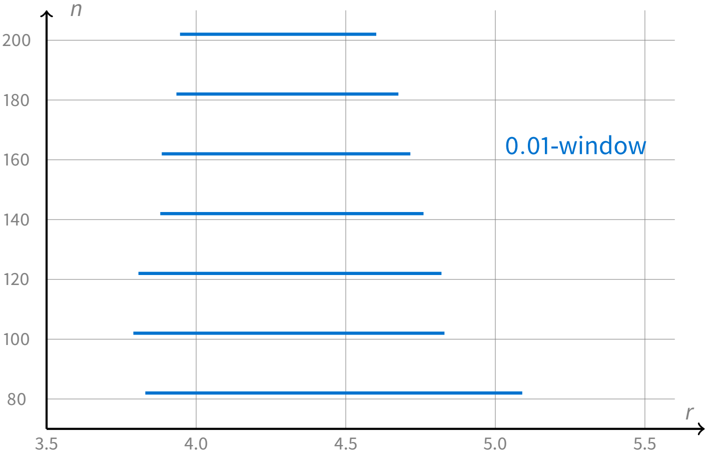
For any (small) number $\epsilon > 0$, the ϵ-*window* is the interval of values of r where

$$\epsilon \leq \pi(r, n) \leq 1 - \epsilon$$

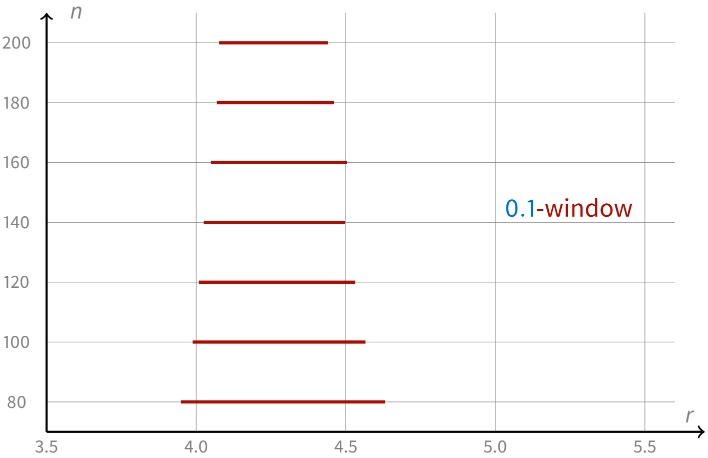
Example $\epsilon = 0.1$



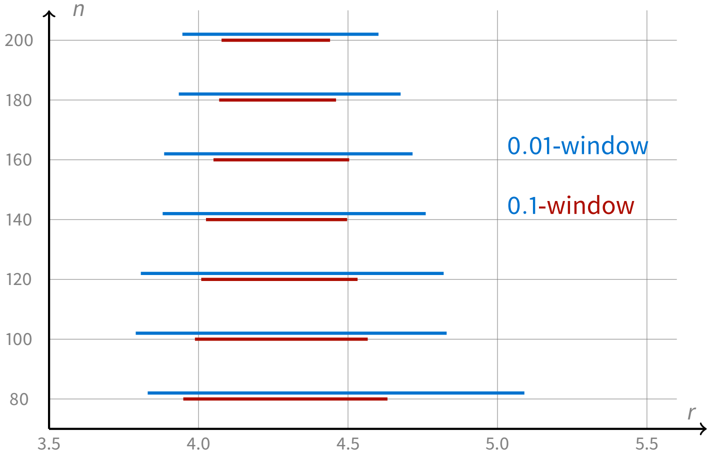
Scaling Window Effect



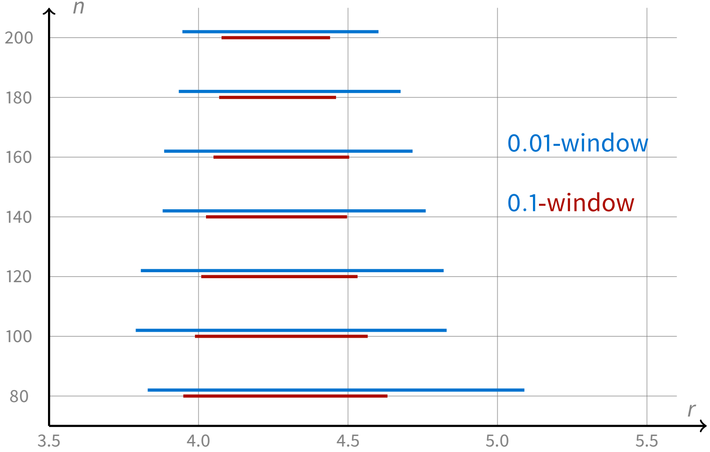
Scaling Window Effect



Scaling Window Effect

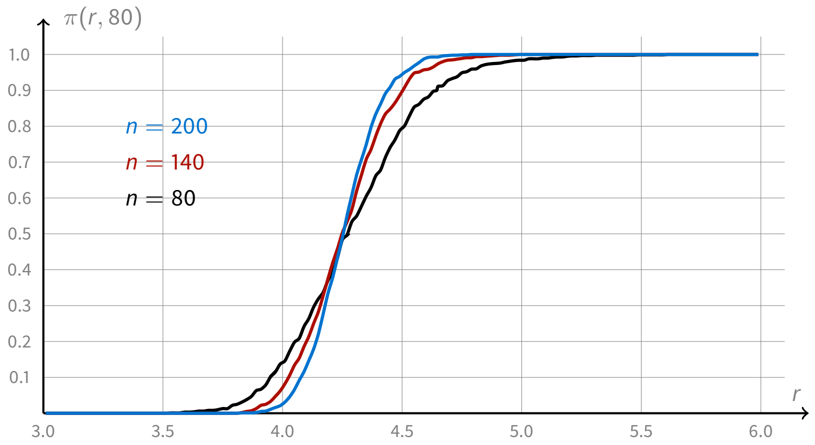


Scaling Window Effect



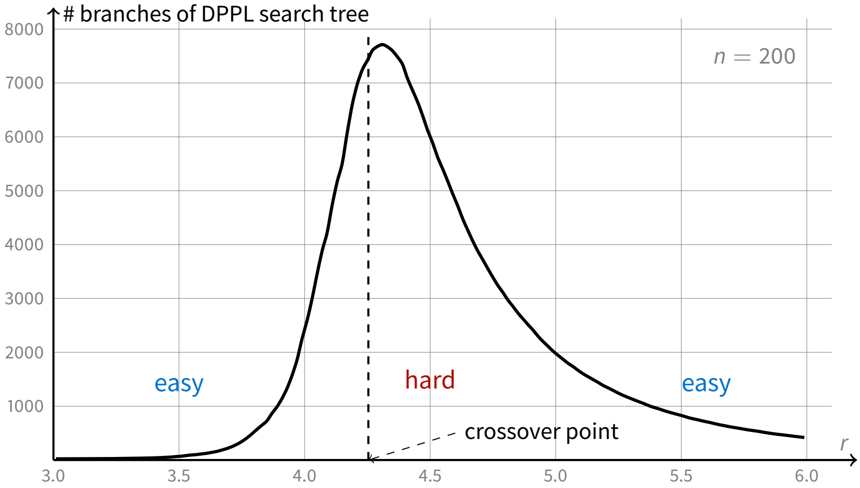
Conjecture: for $n \to \infty$ every ϵ-window degenerates into a point

Sharp Phase Transition



$\pi(r, n) =$ prob. that a randomly generated set of [rn] 3-clauses over n variables is unsat

Easy-Hard-Easy Pattern



Satisfiability Algorithm that Cannot Establish Unsatisfiability

procedure *CHAOS*(S)
input: set of clauses S
output: interpretation $\mathcal{I}$ such that $\mathcal{I} \models S$ or *"don't know"*

Satisfiability Algorithm that Cannot Establish Unsatisfiability

procedure *CHAOS*(S)
input: set of clauses S
output: interpretation $\mathcal{I}$ such that $\mathcal{I} \models S$ or *"don't know"*
parameters: positive integer *MAX-TRIES*
begin
  repeat *MAX-TRIES* times

end

Satisfiability Algorithm that Cannot Establish Unsatisfiability

procedure *CHAOS*(S)
input: set of clauses S
output: interpretation $\mathcal{I}$ such that $\mathcal{I} \models S$ or *"don't know"*
parameters: positive integer *MAX-TRIES*
begin
  repeat *MAX-TRIES* times
    $\mathcal{I} := $ random interpretation
    if $\mathcal{I} \models S$ **then return** $\mathcal{I}$
  return *"don't know"*
end

Satisfiability Algorithm that Cannot Establish Unsatisfiability

procedure *CHAOS*(S)
input: set of clauses S
output: interpretation $\mathcal{I}$ such that $\mathcal{I} \models S$ or *"don't know"*
parameters: positive integer *MAX-TRIES*
begin
  repeat *MAX-TRIES* times
    $\mathcal{I} \ := \$ random interpretation
    if $\mathcal{I} \models S$ **then return** $\mathcal{I}$
  return *"don't know"*
end

Satisfiability has short witnesses: interpretations

Randomised Algorithms for SAT

1. Choose a random interpretation
2. If this interpretation is not a model, repeatedly choose a variable and change its value in the interpretation (*flip* the variable)

$$flip(\mathcal{I}, p)(q) = \begin{cases} \mathcal{I}(q) & \text{if } p \neq q \\ 1 & \text{if } p = q \text{ and } \mathcal{I}(p) = 0 \\ 0 & \text{if } p = q \text{ and } \mathcal{I}(p) = 1 \end{cases}$$

The flipped variables are chosen using heuristics or randomly, or both

Randomised Algorithms for SAT

1. Choose a random interpretation
2. If this interpretation is not a model,
   repeatedly choose a variable and
   change its value in the interpretation (*flip* the variable)

$$flip(\mathcal{I}, p)(q) = \begin{cases} \mathcal{I}(q) & \text{if } p \neq q \\ 1 & \text{if } p = q \text{ and } \mathcal{I}(p) = 0 \\ 0 & \text{if } p = q \text{ and } \mathcal{I}(p) = 1 \end{cases}$$

The flipped variables are chosen using heuristics or randomly, or both

Randomised Algorithms for SAT

1. Choose a random interpretation
2. If this interpretation is not a model,
   repeatedly choose a variable and
   change its value in the interpretation (*flip* the variable)

$$flip(\mathcal{I}, p)(q) = \begin{cases} \mathcal{I}(q) & \text{if } p \neq q \\ 1 & \text{if } p = q \text{ and } \mathcal{I}(p) = 0 \\ 0 & \text{if } p = q \text{ and } \mathcal{I}(p) = 1 \end{cases}$$

The flipped variables are chosen using heuristics or randomly, or both

Randomised Algorithms for SAT

1. Choose a random interpretation
2. If this interpretation is not a model,
   repeatedly choose a variable and
   change its value in the interpretation (*flip* the variable)

$$\mathit{flip}(\mathcal{I}, p)(q) = \begin{cases} \mathcal{I}(q) & \text{if } p \neq q \\ 1 & \text{if } p = q \text{ and } \mathcal{I}(p) = 0 \\ 0 & \text{if } p = q \text{ and } \mathcal{I}(p) = 1 \end{cases}$$

The flipped variables are chosen using heuristics or randomly, or both

GSAT

procedure $GSAT(S)$
input: set of clauses S
output: interpretation $\mathcal{I}$ such that $\mathcal{I} \models S$ or *"don't know"*

GSAT

procedure $GSAT(S)$
input: set of clauses S
output: interpretation $\mathcal{I}$ such that $\mathcal{I} \models S$ or *"don't know"*
parameters: integers *MAX-TRIES*, *MAX-FLIPS*

GSAT

procedure *GSAT*(S)
input: set of clauses S
output: interpretation $\mathcal{I}$ such that $\mathcal{I} \models S$ or *"don't know"*
parameters: integers *MAX-TRIES*, *MAX-FLIPS*
begin
 repeat *MAX-TRIES* times
  $\mathcal{I}$:= random interpretation
  if $\mathcal{I} \models S$ **then return** $\mathcal{I}$

end

GSAT

```
procedure GSAT(S)
input: set of clauses S
output: interpretation 𝓘 such that 𝓘 ⊨ S or "don't know"
parameters: integers MAX-TRIES, MAX-FLIPS
begin
 repeat MAX-TRIES times
  𝓘 := random interpretation
  if 𝓘 ⊨ S then return 𝓘
  repeat MAX-FLIPS times
   p := a variable such that flip(𝓘, p) satisfies
        the maximal number of clauses in S
   𝓘 = flip(𝓘, p)
   if 𝓘 ⊨ S then return 𝓘
 return "don't know"
end
```

GSAT example

0		0		1
p_1	$\vee$	$\neg p_2$	$\vee$	p_3
		$\neg p_2$	$\vee$	$\neg p_3$
$\neg p_1$			$\vee$	$\neg p_3$
$\neg p_1$	$\vee$	p_2		
p_1	$\vee$	p_2		

GSAT example

0		0		1	
p_1	$\vee$	$\neg p_2$	$\vee$	p_3	✓
		$\neg p_2$	$\vee$	$\neg p_3$	✓
$\neg p_1$			$\vee$	$\neg p_3$	✓
$\neg p_1$	$\vee$	p_2			✓
p_1	$\vee$	p_2			

flip no.	interpretation			satisfied clauses				candidates for flipping	flipped variable
	p_1	p_2	p_3		p_1	p_2	p_3		
1	0	0	1	4					

GSAT example

0		0		1	
p_1	$\vee$	$\neg p_2$	$\vee$	p_3	✓
		$\neg p_2$	$\vee$	$\neg p_3$	✓
$\neg p_1$			$\vee$	$\neg p_3$	
$\neg p_1$	$\vee$	p_2			
p_1	$\vee$	p_2			✓

flip no.	interpretation			satisfied clauses				candidates for flipping	flipped variable
	p_1	p_2	p_3		p_1	p_2	p_3		
1	0	0	1	4	3				

GSAT example

0		1		1	
p_1	$\vee$	$\neg p_2$	$\vee$	p_3	✓
		$\neg p_2$	$\vee$	$\neg p_3$	
$\neg p_1$			$\vee$	$\neg p_3$	✓
$\neg p_1$	$\vee$	p_2			✓
p_1	$\vee$	p_2			✓

flip no.	interpretation p_1	p_2	p_3	satisfied clauses	p_1	p_2	p_3	candidates for flipping	flipped variable
1	0	0	1	4	3	4			

GSAT example

0		1		1	
p_1	$\vee$	$\neg p_2$	$\vee$	p_3	✓
		$\neg p_2$	$\vee$	$\neg p_3$	✓
$\neg p_1$			$\vee$	$\neg p_3$	✓
$\neg p_1$	$\vee$	p_2			✓
p_1	$\vee$	p_2			

flip no.	interpretation			satisfied clauses				candidates for flipping	flipped variable
	p_1	p_2	p_3		p_1	p_2	p_3		
1	0	0	1	4	3	4	4		

GSAT example

0		1		1
p_1	$\vee$	$\neg p_2$	$\vee$	p_3
		$\neg p_2$	$\vee$	$\neg p_3$
$\neg p_1$			$\vee$	$\neg p_3$
$\neg p_1$	$\vee$	p_2		
p_1	$\vee$	p_2		

flip no.	interpretation p_1	p_2	p_3	satisfied clauses	p_1	p_2	p_3	candidates for flipping	flipped variable
1	0	0	1	4	3	4	4	p_2, p_3	

GSAT example

0		1		1
p_1	$\vee$	$\neg p_2$	$\vee$	p_3
		$\neg p_2$	$\vee$	$\neg p_3$
$\neg p_1$			$\vee$	$\neg p_3$
$\neg p_1$	$\vee$	p_2		
p_1	$\vee$	p_2		

flip no.	interpretation			satisfied clauses				candidates for flipping	flipped variable
	p_1	p_2	p_3		p_1	p_2	p_3		
1	0	0	1	4	3	4	4	p_2, p_3	p_2

GSAT example

0		1		1	
p_1	$\vee$	$\neg p_2$	$\vee$	p_3	✓
		$\neg p_2$	$\vee$	$\neg p_3$	
$\neg p_1$			$\vee$	$\neg p_3$	✓
$\neg p_1$	$\vee$	p_2			✓
p_1	$\vee$	p_2			✓

flip no.	interpretation p_1	p_2	p_3	satisfied clauses	p_1	p_2	p_3	candidates for flipping	flipped variable
1	0	0	1	4	3	4	4	p_2, p_3	p_2
2	0	1	1	4	3	4	4		

GSAT example

0		1		0	
p_1	$\vee$	$\neg p_2$	$\vee$	p_3	
		$\neg p_2$	$\vee$	$\neg p_3$	✓
$\neg p_1$			$\vee$	$\neg p_3$	✓
$\neg p_1$	$\vee$	p_2			✓
p_1	$\vee$	p_2			✓

flip no.	interpretation p_1	p_2	p_3	satisfied clauses	p_1	p_2	p_3	candidates for flipping	flipped variable
1	0	0	1	4	3	4	4	p_2, p_3	p_2
2	0	1	1	4	3	4	4	p_2, p_3	p_3
3	0	1	0						

GSAT example

0		1		0	
p_1	$\vee$	$\neg p_2$	$\vee$	p_3	
		$\neg p_2$	$\vee$	$\neg p_3$	✓
$\neg p_1$			$\vee$	$\neg p_3$	✓
$\neg p_1$	$\vee$	p_2			✓
p_1	$\vee$	p_2			✓

flip no.	interpretation p_1	p_2	p_3	satisfied clauses	p_1	p_2	p_3	candidates for flipping	flipped variable
1	0	0	1	4	3	4	4	p_2, p_3	p_2
2	0	1	1	4	3	4	4	p_2, p_3	p_3
3	0	1	0	4					

GSAT example

0		1		0
p_1	$\vee$	$\neg p_2$	$\vee$	p_3
		$\neg p_2$	$\vee$	$\neg p_3$
$\neg p_1$			$\vee$	$\neg p_3$
$\neg p_1$	$\vee$	p_2		
p_1	$\vee$	p_2		

flip no.	interpretation			satisfied clauses				candidates for flipping	flipped variable
	p_1	p_2	p_3		p_1	p_2	p_3		
1	0	0	1	4	3	4	4	p_2, p_3	p_2
2	0	1	1	4	3	4	4	p_2, p_3	p_3
3	0	1	0	4	5	4	4		

GSAT example

1		1		0	
p_1	$\vee$	$\neg p_2$	$\vee$	p_3	✓
		$\neg p_2$	$\vee$	$\neg p_3$	✓
$\neg p_1$			$\vee$	$\neg p_3$	✓
$\neg p_1$	$\vee$	p_2			✓
p_1	$\vee$	p_2			✓

flip no.	interpretation p_1	p_2	p_3	satisfied clauses	p_1	p_2	p_3	candidates for flipping	flipped variable
1	0	0	1	4	3	4	4	p_2, p_3	p_2
2	0	1	1	4	3	4	4	p_2, p_3	p_3
3	0	1	0	4	5	4	4	p_1	p_1
	1	1	0	5					

GSAT with random walks

procedure *GSATwithWalks*(S)
input: set of clauses S
output: interpretation $\mathcal{I}$ such that $\mathcal{I} \models S$ or *"don't know"*

GSAT with random walks

procedure *GSATwithWalks*(S)
input: set of clauses S
output: interpretation $\mathcal{I}$ such that $\mathcal{I} \models S$ or *"don't know"*
parameters: integers *MAX-TRIES*, *MAX-FLIPS*
real number $0 \leq \pi \leq 1$ (probability of a sideways move)

GSAT with random walks

procedure *GSATwithWalks*(S)
input: set of clauses S
output: interpretation $\mathcal{I}$ such that $\mathcal{I} \models S$ or *"don't know"*
parameters: integers *MAX-TRIES*, *MAX-FLIPS*
real number $0 \leq \pi \leq 1$ (probability of a sideways move)
begin
repeat *MAX-TRIES* times
$\mathcal{I}$:= random interpretation
if $\mathcal{I} \models S$ **then return** $\mathcal{I}$

end

GSAT with random walks

procedure $GSATwithWalks(S)$
input: set of clauses S
output: interpretation $\mathcal{I}$ such that $\mathcal{I} \models S$ or *"don't know"*
parameters: integers *MAX-TRIES*, *MAX-FLIPS*
real number $0 \leq \pi \leq 1$ (probability of a sideways move)
begin
repeat *MAX-TRIES* times
$\mathcal{I}$:= random interpretation
if $\mathcal{I} \models S$ **then return** $\mathcal{I}$
repeat *MAX-FLIPS* times
with probability π
p := a variable such that $flip(\mathcal{I}, p)$ satisfies the maximal number of clauses in S
with probability $1 - \pi$
randomly select p among all variables occurring in clauses falsified by $\mathcal{I}$
$\mathcal{I} = flip(\mathcal{I}, p)$
if $\mathcal{I} \models S$ **then return** $\mathcal{I}$
return *"don't know"*
end

WSAT

procedure *WSAT*(S)
input: set of clauses S
output: interpretation $\mathcal{I}$ such that $\mathcal{I} \models S$ or *"don't know"*
parameters: integers *MAX-TRIES*, *MAX-FLIPS*

WSAT

procedure *WSAT*(S)
input: set of clauses S
output: interpretation $\mathcal{I}$ such that $\mathcal{I} \models S$ or *"don't know"*
parameters: integers *MAX-TRIES*, *MAX-FLIPS*
begin
 repeat *MAX-TRIES* times
  $\mathcal{I}$:= random interpretation
  if $\mathcal{I} \models S$ **then return** $\mathcal{I}$

end

WSAT

procedure *WSAT*(S)
input: set of clauses S
output: interpretation $\mathcal{I}$ such that $\mathcal{I} \models S$ or *"don't know"*
parameters: integers *MAX-TRIES*, *MAX-FLIPS*
begin
  repeat *MAX-TRIES* times
    $\mathcal{I} :=$ random interpretation
    if $\mathcal{I} \models S$ **then return** $\mathcal{I}$
    repeat *MAX-FLIPS* times
      randomly select a clause $C \in S$ such that $\mathcal{I} \not\models C$
      randomly select a variable p in C
      $\mathcal{I} = flip(\mathcal{I}, p)$
      if $\mathcal{I} \models S$ **then return** $\mathcal{I}$
  return *"don't know"*
end

WSAT example

0		0		1
p_1	$\vee$	$\neg p_2$	$\vee$	p_3
		$\neg p_2$	$\vee$	$\neg p_3$
$\neg p_1$			$\vee$	$\neg p_3$
$\neg p_1$	$\vee$	p_2		
p_1	$\vee$	p_2		

WSAT example

0		0		1	
p_1	$\vee$	$\neg p_2$	$\vee$	p_3	✓
		$\neg p_2$	$\vee$	$\neg p_3$	✓
$\neg p_1$			$\vee$	$\neg p_3$	✓
$\neg p_1$	$\vee$	p_2			✓
p_1	$\vee$	p_2			

flip no.	interpretation			unsatisfied clauses	candidates for flipping	flipped variable
	p_1	p_2	p_3			
1	0	0	1			

WSAT example

0		0		1
p_1	$\vee$	$\neg p_2$	$\vee$	p_3
		$\neg p_2$	$\vee$	$\neg p_3$
$\neg p_1$			$\vee$	$\neg p_3$
$\neg p_1$	$\vee$	p_2		
p_1	$\vee$	p_2		

flip no.	interpretation			unsatisfied clauses	candidates for flipping	flipped variable
	p_1	p_2	p_3			
1	0	0	1	$p_1 \vee p_2$	p_1, p_2	

WSAT example

1		0		1	
p_1	$\vee$	$\neg p_2$	$\vee$	p_3	✓
		$\neg p_2$	$\vee$	$\neg p_3$	✓
$\neg p_1$			$\vee$	$\neg p_3$	
$\neg p_1$	$\vee$	p_2			
p_1	$\vee$	p_2			✓

flip no.	interpretation			unsatisfied clauses	candidates for flipping	flipped variable
	p_1	p_2	p_3			
1	0	0	1	$p_1 \vee p_2$	p_1, p_2	p_1
2	1	0	1			

WSAT example

1		0		1
p_1	$\vee$	$\neg p_2$	$\vee$	p_3
		$\neg p_2$	$\vee$	$\neg p_3$
$\neg p_1$			$\vee$	$\neg p_3$
$\neg p_1$	$\vee$	p_2		
p_1	$\vee$	p_2		

flip no.	interpretation			unsatisfied clauses	candidates for flipping	flipped variable
	p_1	p_2	p_3			
1	0	0	1	$p_1 \vee p_2$	p_1, p_2	p_1
2	1	0	1	$\neg p_1 \vee \neg p_3$ $\neg p_1 \vee p_2$	p_1, p_2, p_3	

WSAT example

1		1		1	
p_1	$\lor$	$\neg p_2$	$\lor$	p_3	✓
		$\neg p_2$	$\lor$	$\neg p_3$	
$\neg p_1$			$\lor$	$\neg p_3$	
$\neg p_1$	$\lor$	p_2			✓
p_1	$\lor$	p_2			✓

flip no.	interpretation			unsatisfied clauses	candidates for flipping	flipped variable
	p_1	p_2	p_3			
1	0	0	1	$p_1 \lor p_2$	p_1, p_2	p_1
2	1	0	1	$\neg p_1 \lor \neg p_3$ $\neg p_1 \lor p_2$	p_1, p_2, p_3	p_2
3	1	1	1			

WSAT example

1		1		1
p_1	$\lor$	$\neg p_2$	$\lor$	p_3
		$\neg p_2$	$\lor$	$\neg p_3$
$\neg p_1$			$\lor$	$\neg p_3$
$\neg p_1$	$\lor$	p_2		
p_1	$\lor$	p_2		

flip no.	interpretation			unsatisfied clauses	candidates for flipping	flipped variable
	p_1	p_2	p_3			
1	0	0	1	$p_1 \lor p_2$	p_1, p_2	p_1
2	1	0	1	$\neg p_1 \lor \neg p_3$ $\neg p_1 \lor p_2$	p_1, p_2, p_3	p_2
3	1	1	1	$\neg p_2 \lor \neg p_3$ $\neg p_1 \lor \neg p_3$	p_1, p_2, p_3	

WSAT example

1		1		0
p_1	$\lor$	$\neg p_2$	$\lor$	p_3
		$\neg p_2$	$\lor$	$\neg p_3$
$\neg p_1$			$\lor$	$\neg p_3$
$\neg p_1$	$\lor$	p_2		
p_1	$\lor$	p_2		

flip no.	interpretation			unsatisfied clauses	candidates for flipping	flipped variable
	p_1	p_2	p_3			
1	0	0	1	$p_1 \lor p_2$	p_1, p_2	p_1
2	1	0	1	$\neg p_1 \lor \neg p_3$ $\neg p_1 \lor p_2$	p_1, p_2, p_3	p_2
3	1	1	1	$\neg p_2 \lor \neg p_3$ $\neg p_1 \lor \neg p_3$	p_1, p_2, p_3	p_3
	1	1	0			

WSAT example

1		1		0	
p_1	$\vee$	$\neg p_2$	$\vee$	p_3	
		$\neg p_2$	$\vee$	$\neg p_3$	✓
$\neg p_1$			$\vee$	$\neg p_3$	✓
$\neg p_1$	$\vee$	p_2			✓
p_1	$\vee$	p_2			✓

flip no.	interpretation			unsatisfied clauses	candidates for flipping	flipped variable
	p_1	p_2	p_3			
1	0	0	1	$p_1 \vee p_2$	p_1, p_2	p_1
2	1	0	1	$\neg p_1 \vee \neg p_3$ $\neg p_1 \vee p_2$	p_1, p_2, p_3	p_2
3	1	1	1	$\neg p_2 \vee \neg p_3$ $\neg p_1 \vee \neg p_3$	p_1, p_2, p_3	p_3
	1	1	0			