Probabilistic analysis of satisfiability

Next:

- What is quantitative relationship between satisfiable and unsatisfiable problems? In other words if we pick a set of clauses at random with what probability it will be satisfiable?
- How can we randomly generate hard problems?
- Randomized algorithms for showing satisfiability.

SAT and k-SAT

SAT is the problem of satisfiability checking for sets of clauses.

A k-clause is a cluase with k literals. k-SAT is the problem of satisfiability checking for sets of k-clauses.

- SAT is NP-complete;
- 3-SAT is NP-complete.
- 2-SAT is decidable in linear time;

SAT and k-SAT

SAT is the problem of satisfiability checking for sets of clauses.

A k-clause is a cluase with k literals. k-SAT is the problem of satisfiability checking for sets of k-clauses.

- SAT is NP-complete;
- 3-SAT is NP-complete.
- 2-SAT is decidable in linear time;

SAT and k-SAT

SAT is the problem of satisfiability checking for sets of clauses.

A k-clause is a cluase with k literals. k-SAT is the problem of satisfiability checking for sets of k-clauses.

- SAT is NP-complete;
- 3-SAT is NP-complete.
- 2-SAT is decidable in linear time;

SAT and k-SAT

SAT is the problem of satisfiability checking for sets of clauses.

A k-clause is a cluase with k literals. k-SAT is the problem of satisfiability checking for sets of k-clauses.

- SAT is NP-complete;
- 3-SAT is NP-complete.
- 2-SAT is decidable in linear time;

SAT and k-SAT

SAT is the problem of satisfiability checking for sets of clauses.

A k-clause is a cluase with k literals. k-SAT is the problem of satisfiability checking for sets of k-clauses.

- SAT is NP-complete;
- 3-SAT is NP-complete.
- 2-SAT is decidable in linear time;

There is a simple reduction of SAT to 3-SAT based on the same ideas as used for generating short clausal forms (naming). Take a clause having more than 3 literals:

$$L_1 \vee L_2 \vee L_3 \vee L_4 \ldots$$

And replace it by two clauses:

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

where n is a new variable.

SAT and k-SAT

SAT is the problem of satisfiability checking for sets of clauses.

A k-clause is a cluase with k literals. k-SAT is the problem of satisfiability checking for sets of k-clauses.

- SAT is NP-complete;
- 3-SAT is NP-complete.
- 2-SAT is decidable in linear time;

There is a simple reduction of SAT to 3-SAT based on the same ideas as used for generating short clausal forms (naming). Take a clause having more than 3 literals:

$$L_1 \vee L_2 \vee L_3 \vee L_4 \ldots$$

And replace it by two clauses:

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

where n is a new variable.
We will consider k-SAT for a fixed k.

Random Clause Generation

How can one generate a random k-clause?

Random Clause Generation

How can one generate a random k-clause?

A random clause is a collection of random literals.

Random Clause Generation

How can one generate a random k-clause?

A random clause is a collection of random literals.

Let's first generate a random literal.

Random Clause Generation

How can one generate a random k-clause?

A random clause is a collection of random literals.

Let's first generate a random literal.

- Fix a number n of boolean variables;

Random Clause Generation

How can one generate a random k-clause?

A random clause is a collection of random literals.

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 k-clause?

A random clause is a collection of random literals.

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 k-clause: generate k random literals

Random Clause Generation

How can one generate a random k-clause?

A random clause is a collection of random literals.

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 k-clause: generate k random literals

Suppose we generate random clauses one after one. How does the set of models of this set change?

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: 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: 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: 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: 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: 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: 20

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

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

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: 20

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

p_1	p_2	p_3	p_4	p_5

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

p_1	p_2	p_3	p_4	p_5
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: 12

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

p_1	p_2	p_3	p_4	p_5

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

p_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: 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: 12

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

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

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: 9

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_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: 9

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_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: 7

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

p_1	p_2	p_3	p_4	p_5

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

p_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: 7

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_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: 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	0
1	1	0	1	1

Number of models: 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	1

Number of models: 3

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

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

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	1

Number of models: 3

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
1	0	0	0	1

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

Number of models: 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_1	p_2	p_3	p_4	p_5
1	0	0	0	1

Number of models: 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
1	0	0	0	1

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

Number of models: 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
1	0	0	0	1

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

Number of models: 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
$\neg p_2 \lor \neg p_3$										
$\neg p_2 \lor p_1$										
$\neg p_2 \lor p_2$										
$p_1 \lor p_1$										
$\neg p_5 \lor p_5$										
$p_4 \lor p_5$										
$\neg p_5 \lor \neg p_3$										
$p_2 \lor \neg p_4$										
$p_5 \lor \neg p_2$										
$p_5 \lor p_2$										
$\neg p_1 \lor \neg p_4$										
$p_5 \lor p_2$										
$\neg p_1 \lor \neg p_5$										

Number of models: 0

This set of 13 clauses is unsatisfiable.

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

Number of models: 0
This set of 13 clauses is unsatisfiable.

Increasing number of generated cluases we can observe transition from satisfiable to unsatisfiable.

Random Clause Generation

We are interested in the probability π that a set of 3-clauses is unsatisfiable.

Random Clause Generation

We are interested in the probability π that a set of 3-clauses is unsatisfiable.

Fix:

- Number n of boolean variables;
- Number m of the clauses.
- Randomly generate m clauses with an equal probability.

Random Clause Generation

We are interested in the probability π that a set of 3-clauses is unsatisfiable.

Fix:

- Number n of boolean variables;
- Number m of the clauses.
- Randomly generate m clauses with an equal probability.

Random Clause Generation

We are interested in the probability π that a set of 3-clauses is unsatisfiable.

Fix:

- Number n of boolean variables;
- Number m of the clauses.
- Randomly generate m clauses with an equal probability.

Random Clause Generation

We are interested in the probability π that a set of 3-clauses is unsatisfiable.

Fix:

- Number n of boolean variables;
- Number m of the clauses.
- Randomly generate m clauses with an equal probability.

Random Clause Generation

We are interested in the probability π that a set of 3-clauses is unsatisfiable.

Fix:

- Number n of boolean variables;
- Number m of the clauses.
- Randomly generate m clauses with an equal probability.

Important parameter: ratio of clauses per variable $r = m/n$.
We will investigate dependence of π with respect to the ratio r and the number of varibales n.

Random Clause Generation

We are interested in the probability π that a set of 3-clauses is unsatisfiable.

Fix:

- Number n of boolean variables;
- Number m of the clauses.
- Randomly generate m clauses with an equal probability.

Important parameter: ratio of clauses per variable $r = m/n$.
We will investigate dependence of π with respect to the ratio r and the number of varibales n.

Note that the robability $\pi(r, n)$ is a monotone function: the more clauses we generate, the higher chance we have that the set is unsatisfiable.

Roulette

We will generate random instances of 3-SAT with 10-variables.

- 5 clauses?
- 30 clauses?
- 60 clauses?
- 100 clauses?
- 1000 clauses?

Roulette

We will generate random instances of 3-SAT with 10-variables.

You will bet on whether the resuting set of clauses is satisfiable or not.

- 5 clauses?
- 30 clauses?
- 60 clauses?
- 100 clauses?
- 1000 clauses?

SAT Roulette

We will generate random instances of 3-SAT with 10-variables.

You will bet on whether the resuting set of clauses is satisfiable or not.
What will you bet on if we generate

- 5 clauses?
- 30 clauses?
- 60 clauses?
- 100 clauses?
- 1000 clauses?

SAT Roulette

We will generate random instances of 3-SAT with 10-variables.

You will bet on whether the resuting set of clauses is satisfiable or not.
What will you bet on if we generate

- 5 clauses?
- 30 clauses?
- 60 clauses?
- 100 clauses?
- 1000 clauses?

SAT Roulette

We will generate random instances of 3-SAT with 10-variables.

You will bet on whether the resuting set of clauses is satisfiable or not.
What will you bet on if we generate

- 5 clauses?
- 30 clauses?
- 60 clauses?
- 100 clauses?
- 1000 clauses?

SAT Roulette

We will generate random instances of 3-SAT with 10-variables.

You will bet on whether the resuting set of clauses is satisfiable or not.
What will you bet on if we generate

- 5 clauses?
- 30 clauses?
- 60 clauses?
- 100 clauses?
- 1000 clauses?

SAT Roulette

We will generate random instances of 3-SAT with 10-variables.

You will bet on whether the resuting set of clauses is satisfiable or not.
What will you bet on if we generate

- 5 clauses?
- 30 clauses?
- 60 clauses?
- 100 clauses?
- 1000 clauses?

SAT Roulette

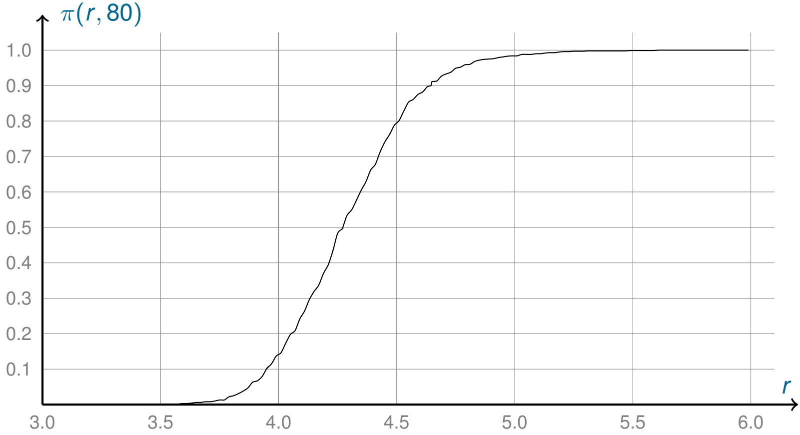
We will generate random instances of 3-SAT with 10-variables.

You will bet on whether the resuting set of clauses is satisfiable or not.
What will you bet on if we generate

- 5 clauses?
- 30 clauses?
- 60 clauses?
- 100 clauses?
- 1000 clauses?

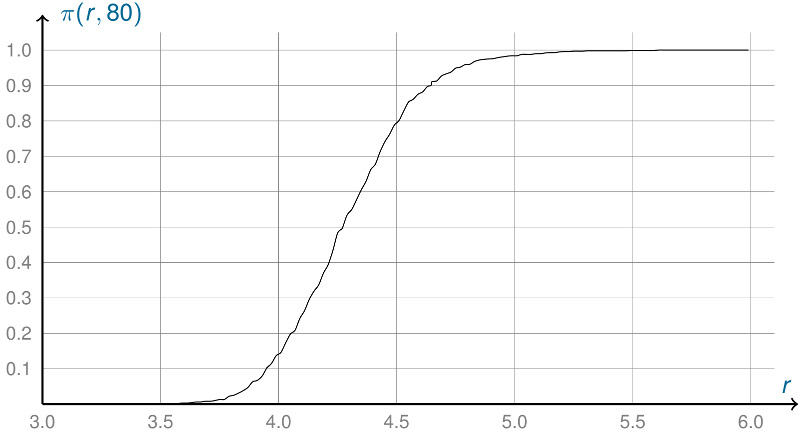
What would be your betting ratio?

Probability of obtaining an unsatisfiable set



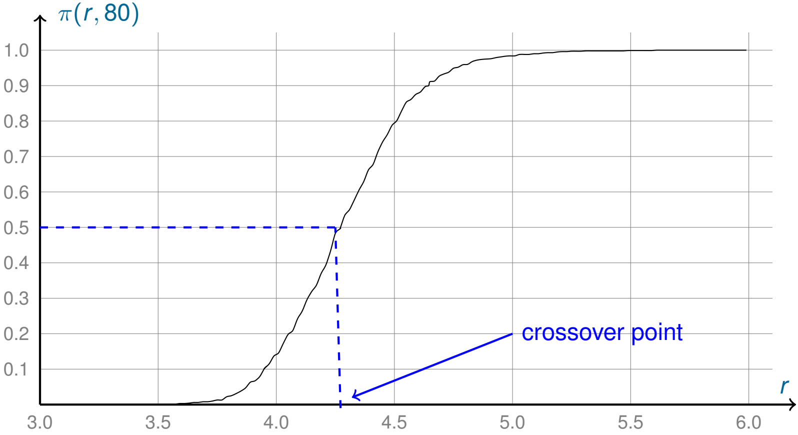
Probability of obtaining an unsatisfiable set

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



Probability of obtaining an unsatisfiable set

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



Experimentally: for large n crossover point is close to 4.25.

ϵ-window

Take a (small) number $\epsilon > 0$. ϵ-window is the interval of values of r where the probability is between ϵ and $1 - \epsilon$.

For example, take $\epsilon = 0.1$.

ϵ-window

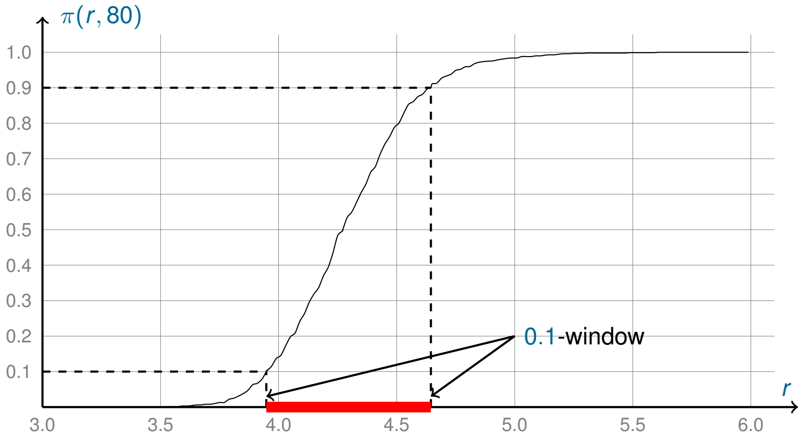
Take a (small) number $\epsilon > 0$. ϵ-window is the interval of values of r where the probability is between ϵ and $1 - \epsilon$.

For example, take $\epsilon = 0.1$.

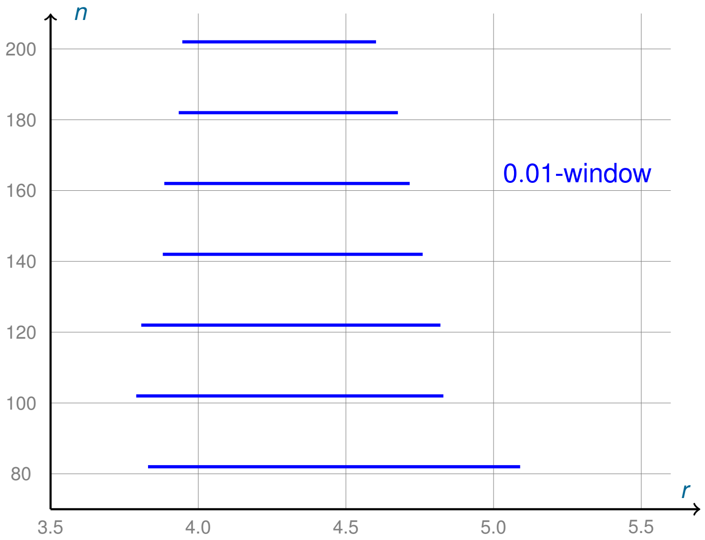
ϵ-window

Take a (small) number $\epsilon > 0$. ϵ-window is the interval of values of r where the probability is between ϵ and $1 - \epsilon$.

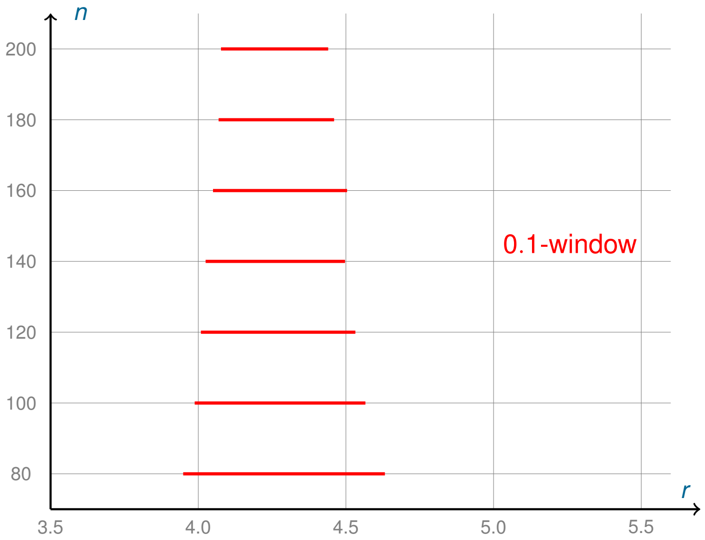
For example, take $\epsilon = 0.1$.



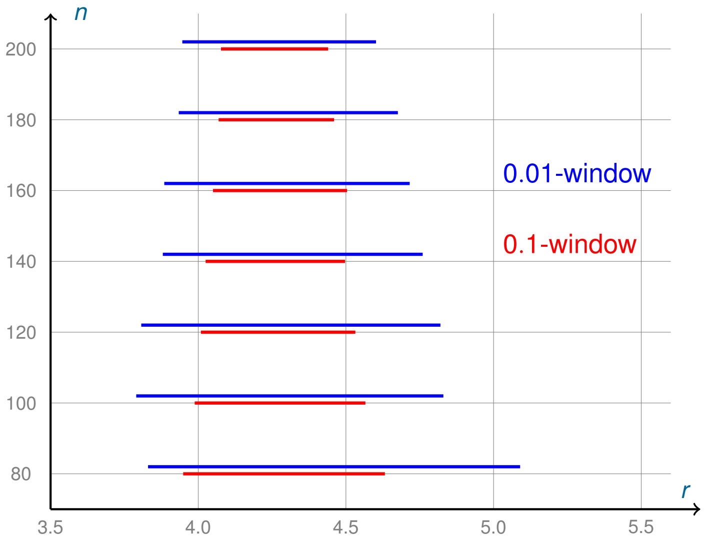
Scaling Window Effect



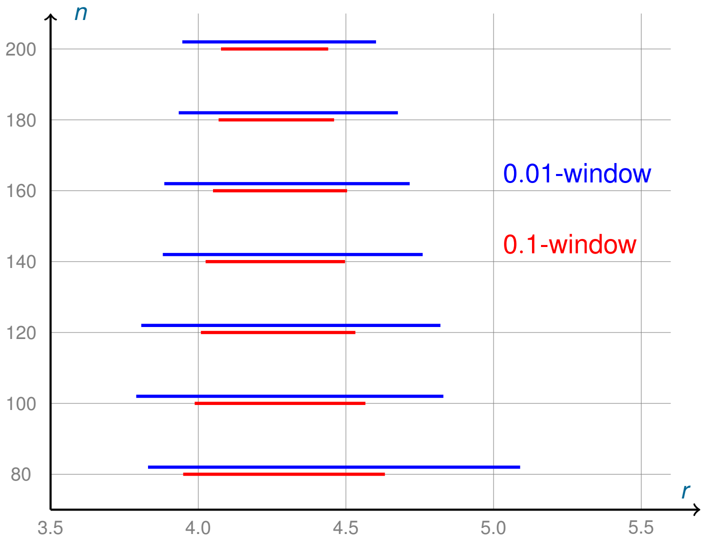
Scaling Window Effect



Scaling Window Effect

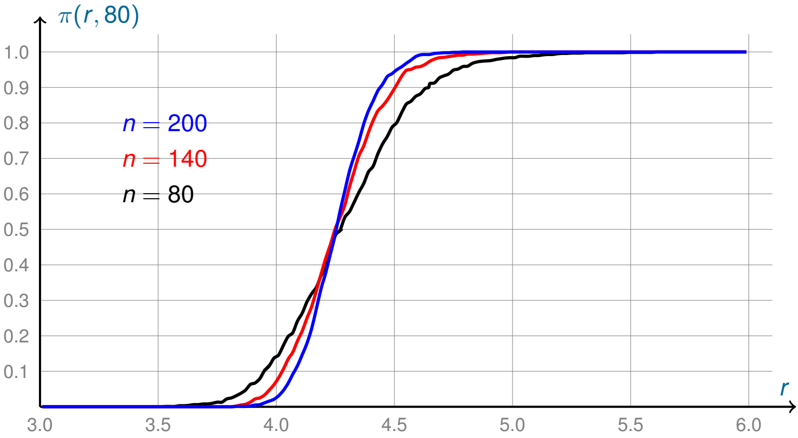


Scaling Window Effect

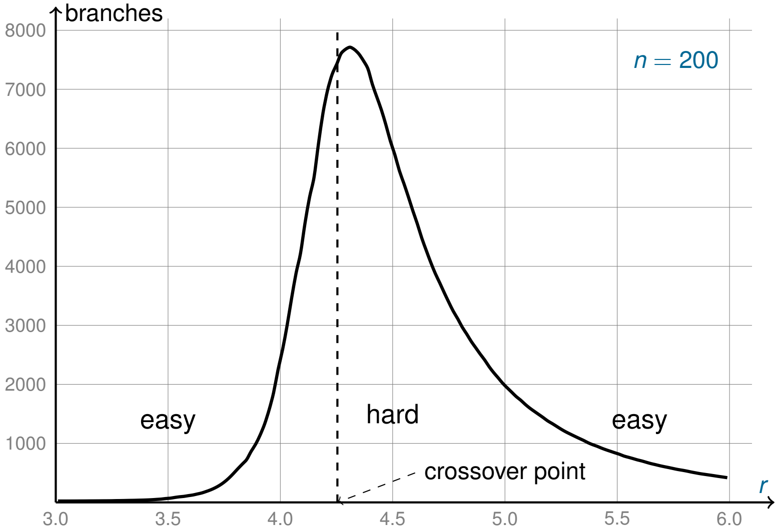


Conjecture: for $n \to \infty$ every ϵ-window “degenerates into a point”.

Sharp Phase Transition



Easy-Hard-Easy Pattern



Next

Next: Randomized satisfiability algorithms.

SAT as a Decision Problem

Decision problem: any collection of problems that have a yes-no answer. Each element of this collection is called an instance of this problem.

Example: solvability of systems of linear inequalities over integers.

- an instance in a system of linear inequalities;
- an answer is yes if it has a solution.

SAT is a decision problem:

- an instance is a finite set of clauses.
- it has a yes-no answer: yes (satisfiable) or no (unsatisfiable)

Witness for a instance I: any data D such that, given D, one can check in polynomial time (in D) that I has a yes-answer.

Satisfiability has short witnesses: interpretations.

Unsatisfiability has no polynomial-size witnesses, unless $NP = coNP$.

SAT as a Decision Problem

Decision problem: any collection of problems that have a yes-no answer. Each element of this collection is called an instance of this problem.

Example: solvability of systems of linear inequalities over integers.

- an instance in a system of linear inequalities;
- an answer is yes if it has a solution.

SAT is a decision problem:

- an instance is a finite set of clauses.
- it has a yes-no answer: yes (satisfiable) or no (unsatisfiable)

Witness for a instance I: any data D such that, given D, one can check in polynomial time (in D) that I has a yes-answer.

Satisfiability has short witnesses: interpretations.

Unsatisfiability has no polynomial-size witnesses, unless $NP = coNP$.

SAT as a Decision Problem

Decision problem: any collection of problems that have a yes-no answer. Each element of this collection is called an instance of this problem.

Example: solvability of systems of linear inequalities over integers.

- an instance in a system of linear inequalities;
- an answer is yes if it has a solution.

SAT is a decision problem:

- an instance is a finite set of clauses.
- it has a yes-no answer: yes (satisfiable) or no (unsatisfiable)

Witness for a instance I: any data D such that, given D, one can check in polynomial time (in D) that I has a yes-answer.

Satisfiability has short witnesses: interpretations.

Unsatisfiability has no polynomial-size witnesses, unless $NP = coNP$.

SAT as a Decision Problem

Decision problem: any collection of problems that have a yes-no answer. Each element of this collection is called an instance of this problem.

Example: solvability of systems of linear inequalities over integers.

- an instance in a system of linear inequalities;
- an answer is yes if it has a solution.

SAT is a decision problem:

- an instance is a finite set of clauses.
- it has a yes-no answer: yes (satisfiable) or no (unsatisfiable)

Witness for a instance I: any data D such that, given D, one can check in polynomial time (in D) that I has a yes-answer.

Satisfiability has short witnesses: interpretations.

Unsatisfiability has no polynomial-size witnesses, unless $NP = coNP$.

SAT as a Decision Problem

Decision problem: any collection of problems that have a yes-no answer. Each element of this collection is called an instance of this problem.

Example: solvability of systems of linear inequalities over integers.

- an instance in a system of linear inequalities;
- an answer is yes if it has a solution.

SAT is a decision problem:

- an instance is a finite set of clauses.
- it has a yes-no answer: yes (satisfiable) or no (unsatisfiable)

Witness for a instance I: any data D such that, given D, one can check in polynomial time (in D) that I has a yes-answer.

Satisfiability has short witnesses: interpretations.

Unsatisfiability has no polynomial-size witnesses, unless $NP = coNP$.

SAT as a Decision Problem

Decision problem: any collection of problems that have a yes-no answer. Each element of this collection is called an instance of this problem.

Example: solvability of systems of linear inequalities over integers.

- an instance in a system of linear inequalities;
- an answer is yes if it has a solution.

SAT is a decision problem:

- an instance is a finite set of clauses.
- it has a yes-no answer: yes (satisfiable) or no (unsatisfiable)

Witness for a instance I: any data D such that, given D, one can check in polynomial time (in D) that I has a yes-answer.

Satisfiability has short witnesses: interpretations.

Unsatisfiability has no polynomial-size witnesses, unless $NP = coNP$.

Satisfiability Algorithm that Cannot Establish Unsatisfiability

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

Satisfiability Algorithm that Cannot Establish Unsatisfiability

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

Satisfiability Algorithm that Cannot Establish Unsatisfiability

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

Randomized satisfiability algorithms:

- random search for a satisfying assignment;
- cannot establish unsatisfiability;
- may return "don't know"

Randomised Algorithms for SAT

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

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

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

In other words, the interpretation $flip(I,p)$ is obtained from I by changing its value on p.

Randomised Algorithms for SAT

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

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

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

In other words, the interpretation $flip(I, p)$ is obtained from I by changing its value on p.

Randomised Algorithms for SAT

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

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

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

In other words, the interpretation $\mathit{flip}(I, p)$ is obtained from I by changing its value on p.

Randomised Algorithms for SAT

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

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

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

In other words, the interpretation $\mathit{flip}(I,p)$ is obtained from I by changing its value on p.

Randomised Algorithms for SAT

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

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

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

In other words, the interpretation $flip(I, p)$ is obtained from I by changing its value on p.

GSAT

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

GSAT

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

GSAT

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

end

GSAT

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

GSAT

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

GSAT is a local search algorithm, it tries to maximise the number of satisfied clauses by local changes.

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	$\lor$	$\neg p_2$	$\lor$	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			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	4	4		

GSAT example

$$
\begin{array}{ccccc}
0 & & \color{red}{1} & & 1 \\
\hline
p_1 & \vee & \neg p_2 & \vee & p_3 \\
 & & \neg p_2 & \vee & \neg p_3 \\
\neg p_1 & & & \vee & \neg p_3 \\
\neg p_1 & \vee & p_2 & & \\
p_1 & \vee & p_2 & &
\end{array}
$$

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	$\color{red}{p_2}$
2	0	1	1						

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
2	0	1	1	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	p_2, p_3	p_2
2	0	1	1	4	3	4	4		

GSAT example

$$
\begin{array}{ccccc}
0 & & 1 & & 0 \\
\hline
p_1 & \vee & \neg p_2 & \vee & p_3 \\
 & & \neg p_2 & \vee & \neg p_3 \\
\neg p_1 & & & \vee & \neg p_3 \\
\neg p_1 & \vee & p_2 & & \\
p_1 & \vee & p_2 & &
\end{array}
$$

flip	interpretation			satisfied clauses				candidates	flipped
no.	p_1	p_2	p_3		p_1	p_2	p_3	for flipping	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 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		

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			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	p_1	p_1
	1	1	0						

GSAT example

$$
\begin{array}{ccccc}
1 & & 1 & & 0 \\
\hline
p_1 & \vee & \neg p_2 & \vee & p_3 \\
 & & \neg p_2 & \vee & \neg p_3 \\
\neg p_1 & & & \vee & \neg p_3 \\
\neg p_1 & \vee & p_2 & & \\
p_1 & \vee & p_2 & &
\end{array}
$$

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	p_1	p_1
	1	1	0	5					

GSAT example

$$
\begin{array}{ccccc}
1 & & 1 & & 0 \\
\hline
p_1 & \vee & \neg p_2 & \vee & p_3 \\
 & & \neg p_2 & \vee & \neg p_3 \\
\neg p_1 & & & \vee & \neg p_3 \\
\neg p_1 & \vee & p_2 & & \\
p_1 & \vee & p_2 & &
\end{array}
$$

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	p_1	p_1
	1	1	0	5					

Advantages: Can quickly find a satisfying assignment in large problems.

Issues: during the inner loop GSAT can get stuck in a ”plateau” optimum point, where further flips do not change the number of satisfied clauses.

GSAT with random walks

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

GSAT with random walks

procedure *GSATwithWalks*(S)
input: set of clauses S
output: interpretation I such that $I \models S$ 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 I such that $I \models S$ 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
I := random interpretation ;
if $I \models S$ **then** **return** I

end

GSAT with random walks

```
procedure GSATwithWalks(S)
input: set of clauses S
output: interpretation I such that I ⊨ S or don't know
parameters: integers MAX-TRIES, MAX-FLIPS
            real number 0 ≤ π ≤ 1 (probability of a sideways move),
begin
 repeat MAX-TRIES times
  I  :=  random interpretation ;
  if I ⊨ S then return I
  repeat MAX-FLIPS times
   with probability π
    p  :=  a variable such that flip(I, p) satisfies
           the maximal number of clauses in S
   with probability 1 − π
    randomly select p among all variables occurring in clauses false in I
   I = flip(I, p) ;
   if I ⊨ S then return I
 return don't know
end
```

Walk SAT (WSAT)

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

Walk SAT (WSAT)

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

end

Walk SAT (WSAT)

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

Walk SAT 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		

Walk SAT 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 p_1	p_2	p_3	unsatisfied clauses	candidates for flipping	flipped variable
1	0	0	1			

Walk SAT example

$$\begin{array}{rcrcr} 0 & & 0 & & 1 \\ \hline p_1 & \vee & \neg p_2 & \vee & p_3 \\ & & \neg p_2 & \vee & \neg p_3 \\ \neg p_1 & & & \vee & \neg p_3 \\ \neg p_1 & \vee & p_2 & & \\ p_1 & \vee & p_2 & & \end{array}$$

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	

Walk SAT 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			

Walk SAT 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 p_1	p_2	p_3	unsatisfied clauses	candidates for flipping	flipped variable
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	

Walk SAT example

1		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			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			

Walk SAT example

$$\begin{array}{rcrcr}
1 & & 1 & & 1 \\
\hline
p_1 & \vee & \neg p_2 & \vee & p_3 \\
 & & \neg p_2 & \vee & \neg p_3 \\
\neg p_1 & & & \vee & \neg p_3 \\
\neg p_1 & \vee & p_2 & & \\
p_1 & \vee & p_2 & &
\end{array}$$

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	

Walk SAT 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			

Walk SAT 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			

Satisfiability of formulas: Semantic Tableaux

Next: Satisfiability of general (signed) formulas.

Algorithm: Semantic tableaux

Signed Formula

- Signed formula: an expression $A = b$, where A is a formula and b a boolean value.
- A signed formula $A = b$ is true in an interpretation I, denoted by $I \models A = b$, if $I(A) = b$.
- If $A = b$ is true in I, we also say that I is a model of $A = b$, or that I satisfies $A = b$.
- A signed formula is satisfiable if it has a model.

Note:

1. For every formula A and interpretation I exactly one of the signed formulas $A = 1$ and $A = 0$ is true in I.
2. A formula A is satisfiable if and only if so is the signed formula $A = 1$.

Signed Formula

- Signed formula: an expression $A = b$, where A is a formula and b a boolean value.
- A signed formula $A = b$ is true in an interpretation I, denoted by $I \models A = b$, if $I(A) = b$.
- If $A = b$ is true in I, we also say that I is a model of $A = b$, or that I satisfies $A = b$.
- A signed formula is satisfiable if it has a model.

Note:

1. For every formula A and interpretation I exactly one of the signed formulas $A = 1$ and $A = 0$ is true in I.
2. A formula A is satisfiable if and only if so is the signed formula $A = 1$.

Signed Formula

- Signed formula: an expression $A = b$, where A is a formula and b a boolean value.
- A signed formula $A = b$ is true in an interpretation I, denoted by $I \models A = b$, if $I(A) = b$.
- If $A = b$ is true in I, we also say that I is a model of $A = b$, or that I satisfies $A = b$.
- A signed formula is satisfiable if it has a model.

Note:

1. For every formula A and interpretation I exactly one of the signed formulas $A = 1$ and $A = 0$ is true in I.
2. A formula A is satisfiable if and only if so is the signed formula $A = 1$.

Signed Formula

- Signed formula: an expression $A = b$, where A is a formula and b a boolean value.
- A signed formula $A = b$ is true in an interpretation I, denoted by $I \models A = b$, if $I(A) = b$.
- If $A = b$ is true in I, we also say that I is a model of $A = b$, or that I satisfies $A = b$.
- A signed formula is satisfiable if it has a model.

Note:

1. For every formula A and interpretation I exactly one of the signed formulas $A = 1$ and $A = 0$ is true in I.
2. A formula A is satisfiable if and only if so is the signed formula $A = 1$.

Signed Formula

- Signed formula: an expression $A = b$, where A is a formula and b a boolean value.
- A signed formula $A = b$ is true in an interpretation I, denoted by $I \models A = b$, if $I(A) = b$.
- If $A = b$ is true in I, we also say that I is a model of $A = b$, or that I satisfies $A = b$.
- A signed formula is satisfiable if it has a model.

Note:

1. For every formula A and interpretation I exactly one of the signed formulas $A = 1$ and $A = 0$ is true in I.
2. A formula A is satisfiable if and only if so is the signed formula $A = 1$.

How to find a model of a signed formula?

Operation table for $\to$:

$\to$	$B = 1$	$B = 0$
$A = 1$	1	0
$A = 0$	1	1

$\to$	$B = 1$	$B = 0$
$A = 1$	1	0
$A = 0$	1	1

Example: $(A \to B) = 1$.

So $(A \to B) = 1$ if and only if $A = 0$ OR $B = 1$.

Likewise, $(A \to B) = 0$ if and only if $A = 1$ AND $B = 0$.

So we can use AND-OR trees to carry out case analysis.

How to find a model of a signed formula?

Operation table for $\rightarrow$:

$\rightarrow$	$B = 1$	$B = 0$
$A = 1$	1	0
$A = 0$	1	1

$\rightarrow$	$B = 1$	$B = 0$
$A = 1$	1	0
$A = 0$	1	1

Example: $(A \rightarrow B) = 1$.

So $(A \rightarrow B) = 1$ if and only if $A = 0$ OR $B = 1$.

Likewise, $(A \rightarrow B) = 0$ if and only if $A = 1$ AND $B = 0$.

So we can use AND-OR trees to carry out case analysis.

How to find a model of a signed formula?

Operation table for $\rightarrow$:

$\rightarrow$	$B = 1$	$B = 0$
$A = 1$	1	0
$A = 0$	1	1

$\rightarrow$	$B = 1$	$B = 0$
$A = 1$	1	0
$A = 0$	1	1

Example: $(A \rightarrow B) = 1$.

So $(A \rightarrow B) = 1$ if and only if $A = 0$ OR $B = 1$.

Likewise, $(A \rightarrow B) = 0$ if and only if $A = 1$ AND $B = 0$.

So we can use AND-OR trees to carry out case analysis.

How to find a model of a signed formula?

Operation table for $\rightarrow$:

$\rightarrow$	$B = 1$	$B = 0$
$A = 1$	1	0
$A = 0$	1	1

$\rightarrow$	$B = 1$	$B = 0$
$A = 1$	1	0
$A = 0$	1	1

Example: $(A \rightarrow B) = 1$.

So $(A \rightarrow B) = 1$ if and only if $A = 0$ OR $B = 1$.

Likewise, $(A \rightarrow B) = 0$ if and only if $A = 1$ AND $B = 0$.

So we can use AND-OR trees to carry out case analysis.

How to find a model of a signed formula?

Operation table for $\rightarrow$:

$\rightarrow$	$B = 1$	$B = 0$
$A = 1$	1	0
$A = 0$	1	1

$\rightarrow$	$B = 1$	$B = 0$
$A = 1$	1	0
$A = 0$	1	1

Example: $(A \rightarrow B) = 1$.

So $(A \rightarrow B) = 1$ if and only if $A = 0$ OR $B = 1$.

Likewise, $(A \rightarrow B) = 0$ if and only if $A = 1$ AND $B = 0$.

So we can use AND-OR trees to carry out case analysis.

Tableau

Tableau: a tree having signed formulas at nodes.

Tableau for a signed formula $A = b$ has $A = b$ as a root.

Alternatively, we can regard a tableau as a collection of branches; each branch is a set of signed formulas.

Notation for branches: $A_1 = b_1 \mid \ldots \mid A_n = b_n$.

Tableau

Tableau: a tree having signed formulas at nodes.

Tableau for a signed formula $A = b$ has $A = b$ as a root.

Alternatively, we can regard a tableau as a collection of branches; each branch is a set of signed formulas.

Notation for branches: $A_1 = b_1 \mid \ldots \mid A_n = b_n$.

Tableau

Tableau: a tree having signed formulas at nodes.

Tableau for a signed formula $A = b$ has $A = b$ as a root.

Alternatively, we can regard a tableau as a collection of branches; each branch is a set of signed formulas.

Notation for branches: $A_1 = b_1 \mid \ldots \mid A_n = b_n$.

Tableau

Tableau: a tree having signed formulas at nodes.

Tableau for a signed formula $A = b$ has $A = b$ as a root.

Alternatively, we can regard a tableau as a collection of branches; each branch is a set of signed formulas.

Notation for branches: $A_1 = b_1 \mid \ldots \mid A_n = b_n$.

Branch Expansion Rules

$$
\begin{aligned}
(A_1 \wedge \ldots \wedge A_n) = 0 &\quad\rightsquigarrow\quad A_1 = 0 \mid \ldots \mid A_n = 0 \\
(A_1 \wedge \ldots \wedge A_n) = 1 &\quad\rightsquigarrow\quad A_1 = 1, \ldots, A_n = 1 \\[1em]
(A_1 \vee \ldots \vee A_n) = 0 &\quad\rightsquigarrow\quad A_1 = 0, \ldots, A_n = 0 \\
(A_1 \vee \ldots \vee A_n) = 1 &\quad\rightsquigarrow\quad A_1 = 1 \mid \ldots \mid A_n = 1 \\[1em]
(A_1 \rightarrow A_2) = 0 &\quad\rightsquigarrow\quad A_1 = 1, A_2 = 0 \\
(A_1 \rightarrow A_2) = 1 &\quad\rightsquigarrow\quad A_1 = 0 \mid A_2 = 1 \\[1em]
(\neg A_1) = 0 &\quad\rightsquigarrow\quad A_1 = 1 \\
(\neg A_1) = 1 &\quad\rightsquigarrow\quad A_1 = 0 \\[1em]
(A_1 \leftrightarrow A_2) = 0 &\quad\rightsquigarrow\quad A_1 = 0, A_2 = 1 \mid A_1 = 1, A_2 = 0 \\
(A_1 \leftrightarrow A_2) = 1 &\quad\rightsquigarrow\quad A_1 = 0, A_2 = 0 \mid A_1 = 1, A_2 = 1
\end{aligned}
$$

Branch Closure Rules

These rules are introduced to mark when the set of signed formulas on a branch is unsatisfiable.

A branch is marked closed in any of the following cases:

- it contains both $p = 0$ and $p = 1$ for some atom p
- it contains $\top = 0$;
- it contains $\bot = 1$.

Branch Closure Rules

These rules are introduced to mark when the set of signed formulas on a branch is unsatisfiable.

A branch is marked closed in any of the following cases:

- it contains both $p = 0$ and $p = 1$ for some atom p
- it contains $\top = 0$;
- it contains $\bot = 1$.

Branch Closure Rules

These rules are introduced to mark when the set of signed formulas on a branch is unsatisfiable.

A branch is marked closed in any of the following cases:

- it contains both $p = 0$ and $p = 1$ for some atom p
- it contains $\top = 0$;
- it contains $\bot = 1$.

A Semantic Tableau

(a) $(\neg(q \vee p \to p \vee q)) = 1$

(a)

(b) $(q \vee p \to p \vee q) = 0$

(b)

(c) $(q \vee p) = 1$

(d) $(p \vee q) = 0$

(d)

$p = 0$

$q = 0$

(c) (c)

$q = 1$ closed

$p = 1$ closed

$$
\begin{array}{rcl}
(A_1 \vee A_2) = 0 & \rightsquigarrow & A_1 = 0, A_2 = 0 \\
(A_1 \vee A_2) = 1 & \rightsquigarrow & A_1 = 1 \mid A_2 = 1 \\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

A Semantic Tableau

(a) $(\neg(q \vee p \to p \vee q)) = 1$

(a)

(b) $(q \vee p \to p \vee q) = 0$

(b)

(c) $(q \vee p) = 1$
(d) $(p \vee q) = 0$

(d)

$p = 0$
$q = 0$

(c) (c)

$q = 1$ closed | $p = 1$ closed

$$
\begin{array}{rcl}
(A_1 \vee A_2) = 0 & \rightsquigarrow & A_1 = 0, A_2 = 0 \\
(A_1 \vee A_2) = 1 & \rightsquigarrow & A_1 = 1 \mid A_2 = 1 \\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

A Semantic Tableau

(a) $(\neg(q \vee p \to p \vee q)) = 1$

(a)

(b) $(q \vee p \to p \vee q) = 0$

(b)

(c) $(q \vee p) = 1$

(d) $(p \vee q) = 0$

(d)

$p = 0$

$q = 0$

(c) (c)

$q = 1$ closed

$p = 1$ closed

$$(A_1 \vee A_2) = 0 \quad \rightsquigarrow \quad A_1 = 0, A_2 = 0$$

$$(A_1 \vee A_2) = 1 \quad \rightsquigarrow \quad A_1 = 1 \mid A_2 = 1$$

$$(A_1 \to A_2) = 0 \quad \rightsquigarrow \quad A_1 = 1, A_2 = 0$$

$$(\neg A_1) = 1 \quad \rightsquigarrow \quad A_1 = 0$$

A Semantic Tableau

(a) $(\neg(q \vee p \to p \vee q)) = 1$

(a)

(b) $(q \vee p \to p \vee q) = 0$

(b)

(c) $(q \vee p) = 1$

(d) $(p \vee q) = 0$

(d)

$p = 0$

$q = 0$

(c) (c)

$q = 1$ $p = 1$

closed closed

$(A_1 \vee A_2) = 0$	$\rightsquigarrow$	$A_1 = 0, A_2 = 0$
$(A_1 \vee A_2) = 1$	$\rightsquigarrow$	$A_1 = 1 \mid A_2 = 1$
$(A_1 \to A_2) = 0$	$\rightsquigarrow$	$A_1 = 1, A_2 = 0$
$(\neg A_1) = 1$	$\rightsquigarrow$	$A_1 = 0$

A Semantic Tableau

(a) $(\neg(q \vee p \to p \vee q)) = 1$

(a)

(b) $(q \vee p \to p \vee q) = 0$

(b)

(c) $(q \vee p) = 1$

(d) $(p \vee q) = 0$

(d)

$p = 0$

$q = 0$

(c) (c)

$q = 1$ $p = 1$

closed closed

$$(A_1 \vee A_2) = 0 \quad \rightsquigarrow \quad A_1 = 0, A_2 = 0$$
$$(A_1 \vee A_2) = 1 \quad \rightsquigarrow \quad A_1 = 1 \mid A_2 = 1$$
$$(A_1 \to A_2) = 0 \quad \rightsquigarrow \quad A_1 = 1, A_2 = 0$$
$$(\neg A_1) = 1 \quad \rightsquigarrow \quad A_1 = 0$$

A Semantic Tableau

(a) $(\neg(q \vee p \to p \vee q)) = 1$

(a)

(b) $(q \vee p \to p \vee q) = 0$

(b)

(c) $(q \vee p) = 1$

(d) $(p \vee q) = 0$

(d)

$p = 0$

$q = 0$

(c) (c)

$q = 1$ $p = 1$

closed closed

$$(A_1 \vee A_2) = 0 \quad \rightsquigarrow \quad A_1 = 0, A_2 = 0$$

$$(A_1 \vee A_2) = 1 \quad \rightsquigarrow \quad A_1 = 1 \mid A_2 = 1$$

$$(A_1 \to A_2) = 0 \quad \rightsquigarrow \quad A_1 = 1, A_2 = 0$$

$$(\neg A_1) = 1 \quad \rightsquigarrow \quad A_1 = 0$$

A Semantic Tableau

(a) $(\neg(q \vee p \rightarrow p \vee q)) = 1$

(a)

(b) $(q \vee p \rightarrow p \vee q) = 0$

(b)

(c) $(q \vee p) = 1$

(d) $(p \vee q) = 0$

(d)

$p = 0$

$q = 0$

(c)	(c)
$q = 1$	$p = 1$
closed	closed

$$
\begin{aligned}
(A_1 \vee A_2) = 0 &\quad \rightsquigarrow \quad A_1 = 0, A_2 = 0 \\
(A_1 \vee A_2) = 1 &\quad \rightsquigarrow \quad A_1 = 1 \mid A_2 = 1 \\
(A_1 \rightarrow A_2) = 0 &\quad \rightsquigarrow \quad A_1 = 1, A_2 = 0 \\
(\neg A_1) = 1 &\quad \rightsquigarrow \quad A_1 = 0
\end{aligned}
$$

A Semantic Tableau

(a) $(\neg(q \vee p \to p \vee q)) = 1$

(a)

(b) $(q \vee p \to p \vee q) = 0$

(b)

(c) $(q \vee p) = 1$

(d) $(p \vee q) = 0$

(d)

$p = 0$

$q = 0$

(c) (c)

$q = 1$ $p = 1$

closed closed

$$(A_1 \vee A_2) = 0 \quad \rightsquigarrow \quad A_1 = 0, A_2 = 0$$

$$(A_1 \vee A_2) = 1 \quad \rightsquigarrow \quad A_1 = 1 \mid A_2 = 1$$

$$(A_1 \to A_2) = 0 \quad \rightsquigarrow \quad A_1 = 1, A_2 = 0$$

$$(\neg A_1) = 1 \quad \rightsquigarrow \quad A_1 = 0$$

A Semantic Tableau

(a) $(\neg(q \vee p \to p \vee q)) = 1$

(a)

(b) $(q \vee p \to p \vee q) = 0$

(b)

(c) $(q \vee p) = 1$

(d) $(p \vee q) = 0$

(d)

$p = 0$

$q = 0$

(c) (c)

$q = 1$ $p = 1$

closed closed

$$(A_1 \vee A_2) = 0 \quad \rightsquigarrow \quad A_1 = 0, A_2 = 0$$

$$(A_1 \vee A_2) = 1 \quad \rightsquigarrow \quad A_1 = 1 \mid A_2 = 1$$

$$(A_1 \to A_2) = 0 \quad \rightsquigarrow \quad A_1 = 1, A_2 = 0$$

$$(\neg A_1) = 1 \quad \rightsquigarrow \quad A_1 = 0$$

A Semantic Tableau

(a) $(\neg(q \vee p \rightarrow p \vee q)) = 1$

(a)

(b) $(q \vee p \rightarrow p \vee q) = 0$

(b)

(c) $(q \vee p) = 1$

(d) $(p \vee q) = 0$

(d)

$p = 0$

$q = 0$

(c) (c)

$q = 1$ $p = 1$

closed closed

$$(A_1 \vee A_2) = 0 \quad \rightsquigarrow \quad A_1 = 0, A_2 = 0$$

$$(A_1 \vee A_2) = 1 \quad \rightsquigarrow \quad A_1 = 1 \mid A_2 = 1$$

$$(A_1 \rightarrow A_2) = 0 \quad \rightsquigarrow \quad A_1 = 1, A_2 = 0$$

$$(\neg A_1) = 1 \quad \rightsquigarrow \quad A_1 = 0$$

A Semantic Tableau

(a) $(\neg(q \vee p \to p \vee q)) = 1$

(a)

(b) $(q \vee p \to p \vee q) = 0$

(b)

(c) $(q \vee p) = 1$

(d) $(p \vee q) = 0$

(d)

$p = 0$

$q = 0$

(c)	(c)
$q = 1$	$p = 1$
closed	closed

$$
\begin{array}{rcl}
(A_1 \vee A_2) = 0 & \rightsquigarrow & A_1 = 0, A_2 = 0 \\
(A_1 \vee A_2) = 1 & \rightsquigarrow & A_1 = 1 \mid A_2 = 1 \\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

Example 2

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

$$
\begin{aligned}
(A_1 \wedge A_2) = 0 &\quad \rightsquigarrow \quad A_1 = 0 \mid A_2 = 0 \\
(A_1 \wedge A_2) = 1 &\quad \rightsquigarrow \quad A_1 = 1, A_2 = 1 \\[1ex]
(A_1 \to A_2) = 0 &\quad \rightsquigarrow \quad A_1 = 1, A_2 = 0 \\
(A_1 \to A_2) = 1 &\quad \rightsquigarrow \quad A_1 = 0 \mid A_2 = 1 \\[1ex]
(\neg A_1) = 1 &\quad \rightsquigarrow \quad A_1 = 0
\end{aligned}
$$

Example 2

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

$$
\begin{aligned}
(A_1 \wedge A_2) = 0 \quad &\rightsquigarrow \quad A_1 = 0 \mid A_2 = 0 \\
(A_1 \wedge A_2) = 1 \quad &\rightsquigarrow \quad A_1 = 1, A_2 = 1 \\[1ex]
(A_1 \to A_2) = 0 \quad &\rightsquigarrow \quad A_1 = 1, A_2 = 0 \\
(A_1 \to A_2) = 1 \quad &\rightsquigarrow \quad A_1 = 0 \mid A_2 = 1 \\[1ex]
(\neg A_1) = 1 \quad &\rightsquigarrow \quad A_1 = 0
\end{aligned}
$$

Example 2

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

(a) |

(b) $((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0$

$$
\begin{array}{rcl}
(A_1 \wedge A_2) = 0 & \rightsquigarrow & A_1 = 0 \mid A_2 = 0 \\
(A_1 \wedge A_2) = 1 & \rightsquigarrow & A_1 = 1, A_2 = 1 \\
\\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(A_1 \to A_2) = 1 & \rightsquigarrow & A_1 = 0 \mid A_2 = 1 \\
\\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

Example 2

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

(a) $|$

(b) $((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0$

$$
\begin{array}{rcl}
(A_1 \wedge A_2) = 0 & \rightsquigarrow & A_1 = 0 \mid A_2 = 0 \\
(A_1 \wedge A_2) = 1 & \rightsquigarrow & A_1 = 1, A_2 = 1 \\
\\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(A_1 \to A_2) = 1 & \rightsquigarrow & A_1 = 0 \mid A_2 = 1 \\
\\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

Example 2

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

(a) |

(b) $((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0$

(b) |

(c) $((p \to q) \wedge (p \wedge q \to r)) = 1$

(d) $(\neg p \to r) = 0$

$$
\begin{array}{rcl}
(A_1 \wedge A_2) = 0 & \rightsquigarrow & A_1 = 0 \mid A_2 = 0 \\
(A_1 \wedge A_2) = 1 & \rightsquigarrow & A_1 = 1, A_2 = 1 \\
\\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(A_1 \to A_2) = 1 & \rightsquigarrow & A_1 = 0 \mid A_2 = 1 \\
\\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

Example 2

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

(a) $|$

(b) $((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0$

(b) $|$

(c) $((p \to q) \wedge (p \wedge q \to r)) = 1$

(d) $(\neg p \to r) = 0$

$$
\begin{array}{rcl}
(A_1 \wedge A_2) = 0 & \rightsquigarrow & A_1 = 0 \mid A_2 = 0 \\
(A_1 \wedge A_2) = 1 & \rightsquigarrow & A_1 = 1, A_2 = 1 \\
\\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(A_1 \to A_2) = 1 & \rightsquigarrow & A_1 = 0 \mid A_2 = 1 \\
\\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

Example 2

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

(a) $|$

(b) $((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0$

(b) $|$

(c) $((p \to q) \wedge (p \wedge q \to r)) = 1$

(d) $(\neg p \to r) = 0$

(c) $|$

(e) $(p \to q) = 1$

(f) $(p \wedge q \to r) = 1$

$$
\begin{aligned}
(A_1 \wedge A_2) = 0 &\quad \rightsquigarrow \quad A_1 = 0 \mid A_2 = 0 \\
(A_1 \wedge A_2) = 1 &\quad \rightsquigarrow \quad A_1 = 1, A_2 = 1 \\[1ex]
(A_1 \to A_2) = 0 &\quad \rightsquigarrow \quad A_1 = 1, A_2 = 0 \\
(A_1 \to A_2) = 1 &\quad \rightsquigarrow \quad A_1 = 0 \mid A_2 = 1 \\[1ex]
(\neg A_1) = 1 &\quad \rightsquigarrow \quad A_1 = 0
\end{aligned}
$$

Example 2

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

$$\text{(a)} \mid$$

(b) $((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0$

$$\text{(b)} \mid$$

(c) $((p \to q) \wedge (p \wedge q \to r)) = 1$

(d) $(\neg p \to r) = 0$

$$\text{(c)} \mid$$

(e) $(p \to q) = 1$

(f) $(p \wedge q \to r) = 1$

$$
\begin{array}{rcl}
(A_1 \wedge A_2) = 0 & \rightsquigarrow & A_1 = 0 \mid A_2 = 0 \\
(A_1 \wedge A_2) = 1 & \rightsquigarrow & A_1 = 1, A_2 = 1 \\
& & \\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(A_1 \to A_2) = 1 & \rightsquigarrow & A_1 = 0 \mid A_2 = 1 \\
& & \\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

Example 2

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

(a) $|$

(b) $((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0$

(b) $|$

(c) $((p \to q) \wedge (p \wedge q \to r)) = 1$
(d) $(\neg p \to r) = 0$

(c) $|$

(e) $(p \to q) = 1$
(f) $(p \wedge q \to r) = 1$

(d) $|$

(g) $(\neg p) = 1$
$r = 0$

$$
\begin{array}{rcl}
(A_1 \wedge A_2) = 0 & \rightsquigarrow & A_1 = 0 \mid A_2 = 0 \\
(A_1 \wedge A_2) = 1 & \rightsquigarrow & A_1 = 1, A_2 = 1 \\
\\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(A_1 \to A_2) = 1 & \rightsquigarrow & A_1 = 0 \mid A_2 = 1 \\
\\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

Example 2

(a) $(\neg((p \to q) \land (p \land q \to r) \to (\neg p \to r))) = 1$

(a) $|$

(b) $((p \to q) \land (p \land q \to r) \to (\neg p \to r)) = 0$

(b) $|$

(c) $((p \to q) \land (p \land q \to r)) = 1$

(d) $(\neg p \to r) = 0$

(c) $|$

(e) $(p \to q) = 1$

(f) $(p \land q \to r) = 1$

(d) $|$

(g) $(\neg p) = 1$

$r = 0$

$$
\begin{array}{rcl}
(A_1 \land A_2) = 0 & \rightsquigarrow & A_1 = 0 \mid A_2 = 0 \\
(A_1 \land A_2) = 1 & \rightsquigarrow & A_1 = 1, A_2 = 1 \\
\\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(A_1 \to A_2) = 1 & \rightsquigarrow & A_1 = 0 \mid A_2 = 1 \\
\\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

Example 2

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

(a) |

(b) $((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0$

(b) |

(c) $((p \to q) \wedge (p \wedge q \to r)) = 1$

(d) $(\neg p \to r) = 0$

(c) |

(e) $(p \to q) = 1$

(f) $(p \wedge q \to r) = 1$

(d) |

(g) $(\neg p) = 1$

$r = 0$

(e) / (e) \

$p = 0$ $\quad$ $q = 1$

$$
\begin{array}{rcl}
(A_1 \wedge A_2) = 0 & \rightsquigarrow & A_1 = 0 \mid A_2 = 0 \\
(A_1 \wedge A_2) = 1 & \rightsquigarrow & A_1 = 1, A_2 = 1 \\
\\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(A_1 \to A_2) = 1 & \rightsquigarrow & A_1 = 0 \mid A_2 = 1 \\
\\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

Example 2

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

(a) |

(b) $((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0$

(b) |

(c) $((p \to q) \wedge (p \wedge q \to r)) = 1$
(d) $(\neg p \to r) = 0$

(c) |

(e) $(p \to q) = 1$
(f) $(p \wedge q \to r) = 1$

(d) |

(g) $(\neg p) = 1$
$r = 0$

(e) (e)

$p = 0$ $\quad$ $q = 1$

$$(A_1 \wedge A_2) = 0 \quad \rightsquigarrow \quad A_1 = 0 \mid A_2 = 0$$
$$(A_1 \wedge A_2) = 1 \quad \rightsquigarrow \quad A_1 = 1, A_2 = 1$$

$$(A_1 \to A_2) = 0 \quad \rightsquigarrow \quad A_1 = 1, A_2 = 0$$
$$(A_1 \to A_2) = 1 \quad \rightsquigarrow \quad A_1 = 0 \mid A_2 = 1$$

$$(\neg A_1) = 1 \quad \rightsquigarrow \quad A_1 = 0$$

Example 2

(a) $(\neg((p \to q) \land (p \land q \to r) \to (\neg p \to r))) = 1$

(a) $|$

(b) $((p \to q) \land (p \land q \to r) \to (\neg p \to r)) = 0$

(b) $|$

(c) $((p \to q) \land (p \land q \to r)) = 1$
(d) $(\neg p \to r) = 0$

(c) $|$

(e) $(p \to q) = 1$
(f) $(p \land q \to r) = 1$

(d) $|$

(g) $(\neg p) = 1$
$r = 0$

(e) (e)

$p = 0$ $\qquad q = 1$

(g) $|$

$p = 0$

$$(A_1 \land A_2) = 0 \quad \rightsquigarrow \quad A_1 = 0 \mid A_2 = 0$$
$$(A_1 \land A_2) = 1 \quad \rightsquigarrow \quad A_1 = 1, A_2 = 1$$

$$(A_1 \to A_2) = 0 \quad \rightsquigarrow \quad A_1 = 1, A_2 = 0$$
$$(A_1 \to A_2) = 1 \quad \rightsquigarrow \quad A_1 = 0 \mid A_2 = 1$$

$$(\neg A_1) = 1 \quad \rightsquigarrow \quad A_1 = 0$$

Example 2

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

(a) |

(b) $((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0$

(b) |

(c) $((p \to q) \wedge (p \wedge q \to r)) = 1$

(d) $(\neg p \to r) = 0$

(c) |

(e) $(p \to q) = 1$

(f) $(p \wedge q \to r) = 1$

(d) |

(g) $(\neg p) = 1$

$r = 0$

(e) (e)

$p = 0$ $\qquad$ $q = 1$

(g) |

$p = 0$

$$
\begin{aligned}
(A_1 \wedge A_2) = 0 &\quad \rightsquigarrow \quad A_1 = 0 \mid A_2 = 0 \\
(A_1 \wedge A_2) = 1 &\quad \rightsquigarrow \quad A_1 = 1, A_2 = 1 \\
(A_1 \to A_2) = 0 &\quad \rightsquigarrow \quad A_1 = 1, A_2 = 0 \\
(A_1 \to A_2) = 1 &\quad \rightsquigarrow \quad A_1 = 0 \mid A_2 = 1 \\
(\neg A_1) = 1 &\quad \rightsquigarrow \quad A_1 = 0
\end{aligned}
$$

Example 2

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

(a) |

(b) $((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0$

(b) |

(c) $((p \to q) \wedge (p \wedge q \to r)) = 1$

(d) $(\neg p \to r) = 0$

(c) |

(e) $(p \to q) = 1$

(f) $(p \wedge q \to r) = 1$

(d) |

(g) $(\neg p) = 1$

$r = 0$

(e) $p = 0$ | (e) $q = 1$

(g) |

$p = 0$

(f) (h) $(p \wedge q) = 0$ | (f) $r = 1$

$$(A_1 \wedge A_2) = 0 \quad \rightsquigarrow \quad A_1 = 0 \mid A_2 = 0$$
$$(A_1 \wedge A_2) = 1 \quad \rightsquigarrow \quad A_1 = 1, A_2 = 1$$

$$(A_1 \to A_2) = 0 \quad \rightsquigarrow \quad A_1 = 1, A_2 = 0$$
$$(A_1 \to A_2) = 1 \quad \rightsquigarrow \quad A_1 = 0 \mid A_2 = 1$$

$$(\neg A_1) = 1 \quad \rightsquigarrow \quad A_1 = 0$$

Example 2

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

(a) |

(b) $((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0$

(b) |

(c) $((p \to q) \wedge (p \wedge q \to r)) = 1$

(d) $(\neg p \to r) = 0$

(c) |

(e) $(p \to q) = 1$

(f) $(p \wedge q \to r) = 1$

(d) |

(g) $(\neg p) = 1$

$r = 0$

(e) / (e) \

$p = 0$ $\qquad$ $q = 1$

(g) |

$p = 0$

(f) / (f) \

(h) $(p \wedge q) = 0$ $\qquad$ $r = 1$

$$(A_1 \wedge A_2) = 0 \quad \rightsquigarrow \quad A_1 = 0 \mid A_2 = 0$$
$$(A_1 \wedge A_2) = 1 \quad \rightsquigarrow \quad A_1 = 1, A_2 = 1$$

$$(A_1 \to A_2) = 0 \quad \rightsquigarrow \quad A_1 = 1, A_2 = 0$$
$$(A_1 \to A_2) = 1 \quad \rightsquigarrow \quad A_1 = 0 \mid A_2 = 1$$

$$(\neg A_1) = 1 \quad \rightsquigarrow \quad A_1 = 0$$

Example 2

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

(a) $|$

(b) $((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0$

(b) $|$

(c) $((p \to q) \wedge (p \wedge q \to r)) = 1$

(d) $(\neg p \to r) = 0$

(c) $|$

(e) $(p \to q) = 1$

(f) $(p \wedge q \to r) = 1$

(d) $|$

(g) $(\neg p) = 1$

$r = 0$

(e) (e)

$p = 0$ $\qquad$ $q = 1$

(g) $|$

$p = 0$

(f) (f)

(h) $(p \wedge q) = 0$ $\qquad$ $r = 1$

(h) (h)

$p = 0$ $\qquad$ $q = 0$

$$\begin{array}{rcl}
(A_1 \wedge A_2) = 0 & \rightsquigarrow & A_1 = 0 \mid A_2 = 0 \\
(A_1 \wedge A_2) = 1 & \rightsquigarrow & A_1 = 1, A_2 = 1 \\
\\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(A_1 \to A_2) = 1 & \rightsquigarrow & A_1 = 0 \mid A_2 = 1 \\
\\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}$$

Example 2

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

(a) |

(b) $((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0$

(b) |

(c) $((p \to q) \wedge (p \wedge q \to r)) = 1$

(d) $(\neg p \to r) = 0$

(c) |

(e) $(p \to q) = 1$

(f) $(p \wedge q \to r) = 1$

(d) |

(g) $(\neg p) = 1$

$r = 0$

(e) / (e)

$p = 0$ $\qquad$ $q = 1$

(g) |

$p = 0$

(f) / (f)

(h) $(p \wedge q) = 0$ $\qquad$ $r = 1$

(h) / (h)

$p = 0$ $\qquad$ $q = 0$

$$(A_1 \wedge A_2) = 0 \quad \rightsquigarrow \quad A_1 = 0 \mid A_2 = 0$$
$$(A_1 \wedge A_2) = 1 \quad \rightsquigarrow \quad A_1 = 1, A_2 = 1$$

$$(A_1 \to A_2) = 0 \quad \rightsquigarrow \quad A_1 = 1, A_2 = 0$$
$$(A_1 \to A_2) = 1 \quad \rightsquigarrow \quad A_1 = 0 \mid A_2 = 1$$

$$(\neg A_1) = 1 \quad \rightsquigarrow \quad A_1 = 0$$

All rules on this branch have been applied, so the formula is satisfiable.

Finding Models Using Tableaux

(a) $(\neg((p \to q) \land (p \land q \to r) \to (\neg p \to r))) = 1$

(a)

(b) $((p \to q) \land (p \land q \to r) \to (\neg p \to r)) = 0$

(b)

(c) $((p \to q) \land (p \land q \to r)) = 1$

(d) $(\neg p \to r) = 0$

(c)

(e) $(p \to q) = 1$

(f) $(p \land q \to r) = 1$

(d)

(g) $(\neg p) = 1$

$r = 0$

(e) $p = 0$ (e) $q = 1$

(g)

$p = 0$

(f) (f)

(h) $(p \land q) = 0$ $r = 1$

(h) $p = 0$ (h) $q = 0$

Build an open branch on which all rules have been applied: a complete open branch

Select signed atoms on this branch

This gives us a partial assignemnt any extension of which is a model

$\{r \mapsto 0, p \mapsto 0, q \mapsto \cdots\}$

Finding Models Using Tableaux

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

(a)

(b) $((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0$

(b)

(c) $((p \to q) \wedge (p \wedge q \to r)) = 1$

(d) $(\neg p \to r) = 0$

(c)

(e) $(p \to q) = 1$

(f) $(p \wedge q \to r) = 1$

(d)

(g) $(\neg p) = 1$

$r = 0$

(e) (e)

$p = 0$ $\quad$ $q = 1$

(g)

$p = 0$

(f) (f)

(h) $(p \wedge q) = 0$ $\quad$ $r = 1$

(h) (h)

$p = 0$ $\quad$ $q = 0$

Build an open branch on which all rules have been applied: a complete open branch

Select signed atoms on this branch

This gives us a partial assignemnt any extension of which is a model

$\{r \mapsto 0, p \mapsto 0, q \mapsto \cdots\}$

Finding Models Using Tableaux

(a) $(\neg((p \to q) \land (p \land q \to r) \to (\neg p \to r))) = 1$

(a)

(b) $((p \to q) \land (p \land q \to r) \to (\neg p \to r)) = 0$

(b)

(c) $((p \to q) \land (p \land q \to r)) = 1$
(d) $(\neg p \to r) = 0$

(c)

(e) $(p \to q) = 1$
(f) $(p \land q \to r) = 1$

(d)

(g) $(\neg p) = 1$
$r = 0$

(e) $p = 0$ | (e) $q = 1$

(g)

$p = 0$

(f) $(h)\ (p \land q) = 0$ | (f) $r = 1$

(h) $p = 0$ | (h) $q = 0$

Build an open branch on which all rules have been applied: a complete open branch

Select signed atoms on this branch

This gives us a partial assignemnt any extension of which is a model

$\{r \mapsto 0, p \mapsto 0, q \mapsto \cdots\}$

Finding Models Using Tableaux

(a) $(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$

(a) |

(b) $((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0$

(b) |

(c) $((p \to q) \wedge (p \wedge q \to r)) = 1$

(d) $(\neg p \to r) = 0$

(c) |

(e) $(p \to q) = 1$

(f) $(p \wedge q \to r) = 1$

(d) |

(g) $(\neg p) = 1$

$r = 0$

(e) / (e)

$p = 0$ $\qquad$ $q = 1$

(g) |

$p = 0$

(f) / (f)

(h) $(p \wedge q) = 0$ $\qquad$ $r = 1$

(h) / (h)

$p = 0$ $\qquad$ $q = 0$

Build an open branch on which all rules have been applied: a complete open branch

Select signed atoms on this branch

This gives us a partial assignemnt any extension of which is a model

$$\{r \mapsto 0, p \mapsto 0, q \mapsto \cdots\}$$

Checking Other Properties with Tableaux

A formula A is satisfiable iff a tableau for $A = 1$ contains a complete open branch (and iff every tableau for $A = 1$ contains a complete open branch).

A formula A is valid iff there is a closed a tableau for $A = 0$ (and iff every tableau for $A = 0$ is closed).

Formulas A and B are equivalent iff there is a closed tableau for $(A \leftrightarrow B) = 0$ (and iff every tableau for $(A \leftrightarrow B) = 0$ is closed).

A fully expanded tableau for $A = 1$ gives us all models of A.

Checking Other Properties with Tableaux

A formula A is satisfiable iff a tableau for $A = 1$ contains a complete open branch (and iff every tableau for $A = 1$ contains a complete open branch).

A formula A is valid iff there is a closed a tableau for $A = 0$ (and iff every tableau for $A = 0$ is closed).

Formulas A and B are equivalent iff there is a closed tableau for $(A \leftrightarrow B) = 0$ (and iff every tableau for $(A \leftrightarrow B) = 0$ is closed).

A fully expanded tableau for $A = 1$ gives us all models of A.

Checking Other Properties with Tableaux

A formula A is satisfiable iff a tableau for $A = 1$ contains a complete open branch (and iff every tableau for $A = 1$ contains a complete open branch).

A formula A is valid iff there is a closed a tableau for $A = 0$ (and iff every tableau for $A = 0$ is closed).

Formulas A and B are equivalent iff there is a closed tableau for $(A \leftrightarrow B) = 0$ (and iff every tableau for $(A \leftrightarrow B) = 0$ is closed).

A fully expanded tableau for $A = 1$ gives us all models of A.

Checking Other Properties with Tableaux

A formula A is satisfiable iff a tableau for $A = 1$ contains a complete open branch (and iff every tableau for $A = 1$ contains a complete open branch).

A formula A is valid iff there is a closed a tableau for $A = 0$ (and iff every tableau for $A = 0$ is closed).

Formulas A and B are equivalent iff there is a closed tableau for $(A \leftrightarrow B) = 0$ (and iff every tableau for $(A \leftrightarrow B) = 0$ is closed).

A fully expanded tableau for $A = 1$ gives us all models of A.

Extras: Flat View of Tableaux

We will make the following changes:

1. show a tableau using the $B_1 \mid \cdots \mid B_n$ notation;
2. remove closed branches;
3. if we apply a table expansion rule to a signed formula on a branch, we will remove the formula from the branch.

$$
\begin{array}{rcl}
(A_1 \vee A_2) = 0 & \rightsquigarrow & A_1 = 0, A_2 = 0 \\
(A_1 \vee A_2) = 1 & \rightsquigarrow & A_1 = 1 \mid A_2 = 1 \\[1ex]
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\[1ex]
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

Extras: Flat View of Tableaux

We will make the following changes:

1. show a tableau using the $B_1 \mid \cdots \mid B_n$ notation;
2. remove closed branches;
3. if we apply a table expansion rule to a signed formula on a branch, we will remove the formula from the branch.

Consider Example 1 again.

$$(\neg(q \lor p \to p \lor q)) = 1$$

$$\begin{array}{rcl}
(A_1 \lor A_2) = 0 & \rightsquigarrow & A_1 = 0, A_2 = 0 \\
(A_1 \lor A_2) = 1 & \rightsquigarrow & A_1 = 1 \mid A_2 = 1 \\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}$$

Extras: Flat View of Tableaux

We will make the following changes:

1. show a tableau using the $B_1 \mid \cdots \mid B_n$ notation;
2. remove closed branches;
3. if we apply a table expansion rule to a signed formula on a branch, we will remove the formula from the branch.

Consider Example 1 again.

$$(\neg(q \vee p \to p \vee q)) = 1$$

$$\begin{array}{rcl}
(A_1 \vee A_2) = 0 & \rightsquigarrow & A_1 = 0, A_2 = 0 \\
(A_1 \vee A_2) = 1 & \rightsquigarrow & A_1 = 1 \mid A_2 = 1 \\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}$$

Extras: Flat View of Tableaux

We will make the following changes:

1. show a tableau using the $B_1 \mid \cdots \mid B_n$ notation;
2. remove closed branches;
3. if we apply a table expansion rule to a signed formula on a branch, we will remove the formula from the branch.

Consider Example 1 again.

$$(\neg(q \vee p \rightarrow p \vee q)) = 1 \rightsquigarrow$$
$$(q \vee p \rightarrow p \vee q) = 0$$

$$\begin{array}{rcl}
(A_1 \vee A_2) = 0 & \rightsquigarrow & A_1 = 0, A_2 = 0 \\
(A_1 \vee A_2) = 1 & \rightsquigarrow & A_1 = 1 \mid A_2 = 1 \\
(A_1 \rightarrow A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}$$

Extras: Flat View of Tableaux

We will make the following changes:

1. show a tableau using the $B_1 \mid \cdots \mid B_n$ notation;
2. remove closed branches;
3. if we apply a table expansion rule to a signed formula on a branch, we will remove the formula from the branch.

Consider Example 1 again.

$$(\neg(q \lor p \to p \lor q)) = 1 \rightsquigarrow$$
$$(q \lor p \to p \lor q) = 0$$

$$\begin{array}{rcl}
(A_1 \lor A_2) = 0 & \rightsquigarrow & A_1 = 0, A_2 = 0 \\
(A_1 \lor A_2) = 1 & \rightsquigarrow & A_1 = 1 \mid A_2 = 1 \\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}$$

Extras: Flat View of Tableaux

We will make the following changes:

1. show a tableau using the $B_1 \mid \cdots \mid B_n$ notation;
2. remove closed branches;
3. if we apply a table expansion rule to a signed formula on a branch, we will remove the formula from the branch.

Consider Example 1 again.

$$
\begin{array}{l}
(\neg(q \lor p \to p \lor q)) = 1 \rightsquigarrow \\
(q \lor p \to p \lor q) = 0 \rightsquigarrow \\
(q \lor p) = 1, (p \lor q) = 0
\end{array}
$$

$$
\begin{array}{rcl}
(A_1 \lor A_2) = 0 & \rightsquigarrow & A_1 = 0, A_2 = 0 \\
(A_1 \lor A_2) = 1 & \rightsquigarrow & A_1 = 1 \mid A_2 = 1 \\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

Extras: Flat View of Tableaux

We will make the following changes:

1. show a tableau using the $B_1 \mid \cdots \mid B_n$ notation;
2. remove closed branches;
3. if we apply a table expansion rule to a signed formula on a branch, we will remove the formula from the branch.

Consider Example 1 again.

$$
\begin{array}{rcl}
(A_1 \lor A_2) = 0 & \rightsquigarrow & A_1 = 0, A_2 = 0 \\
(A_1 \lor A_2) = 1 & \rightsquigarrow & A_1 = 1 \mid A_2 = 1 \\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

$(\neg(q \lor p \to p \lor q)) = 1 \rightsquigarrow$
$(q \lor p \to p \lor q) = 0 \rightsquigarrow$
$(q \lor p) = 1, (p \lor q) = 0$

Extras: Flat View of Tableaux

We will make the following changes:

1. show a tableau using the $B_1 \mid \cdots \mid B_n$ notation;
2. remove closed branches;
3. if we apply a table expansion rule to a signed formula on a branch, we will remove the formula from the branch.

Consider Example 1 again.

$$
\begin{array}{l}
(\neg(q \vee p \to p \vee q)) = 1 \leadsto \\
(q \vee p \to p \vee q) = 0 \leadsto \\
(q \vee p) = 1, (p \vee q) = 0 \leadsto \\
(q \vee p) = 1, p = 0, q = 0
\end{array}
$$

$$
\begin{array}{rcl}
(A_1 \vee A_2) = 0 & \leadsto & A_1 = 0, A_2 = 0 \\
(A_1 \vee A_2) = 1 & \leadsto & A_1 = 1 \mid A_2 = 1 \\
(A_1 \to A_2) = 0 & \leadsto & A_1 = 1, A_2 = 0 \\
(\neg A_1) = 1 & \leadsto & A_1 = 0
\end{array}
$$

Extras: Flat View of Tableaux

We will make the following changes:

1. show a tableau using the $B_1 \mid \cdots \mid B_n$ notation;
2. remove closed branches;
3. if we apply a table expansion rule to a signed formula on a branch, we will remove the formula from the branch.

Consider Example 1 again.

$$
\begin{array}{rcl}
(A_1 \lor A_2) = 0 & \leadsto & A_1 = 0, A_2 = 0 \\
(A_1 \lor A_2) = 1 & \leadsto & A_1 = 1 \mid A_2 = 1 \\
(A_1 \to A_2) = 0 & \leadsto & A_1 = 1, A_2 = 0 \\
(\neg A_1) = 1 & \leadsto & A_1 = 0
\end{array}
$$

$$
\begin{array}{l}
(\neg(q \lor p \to p \lor q)) = 1 \leadsto \\
(q \lor p \to p \lor q) = 0 \leadsto \\
(q \lor p) = 1, (p \lor q) = 0 \leadsto \\
(q \lor p) = 1, p = 0, q = 0
\end{array}
$$

Extras: Flat View of Tableaux

We will make the following changes:

1. show a tableau using the $B_1 \mid \cdots \mid B_n$ notation;
2. remove closed branches;
3. if we apply a table expansion rule to a signed formula on a branch, we will remove the formula from the branch.

Consider Example 1 again.

$$
\begin{array}{rcl}
(A_1 \vee A_2) = 0 & \rightsquigarrow & A_1 = 0, A_2 = 0 \\
(A_1 \vee A_2) = 1 & \rightsquigarrow & A_1 = 1 \mid A_2 = 1 \\
(A_1 \rightarrow A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

$(\neg(q \vee p \rightarrow p \vee q)) = 1 \rightsquigarrow$
$(q \vee p \rightarrow p \vee q) = 0 \rightsquigarrow$
$(q \vee p) = 1, (p \vee q) = 0 \rightsquigarrow$
$(q \vee p) = 1, p = 0, q = 0 \rightsquigarrow$
$q = 1, p = 0, q = 0 \mid p = 1, p = 0, q = 0$

Extras: Flat View of Tableaux

We will make the following changes:

1. show a tableau using the $B_1 \mid \cdots \mid B_n$ notation;
2. remove closed branches;
3. if we apply a table expansion rule to a signed formula on a branch, we will remove the formula from the branch.

Consider Example 1 again.

$$
\begin{array}{rcl}
(A_1 \lor A_2) = 0 & \rightsquigarrow & A_1 = 0, A_2 = 0 \\
(A_1 \lor A_2) = 1 & \rightsquigarrow & A_1 = 1 \mid A_2 = 1 \\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

$$
\begin{array}{l}
(\neg(q \lor p \to p \lor q)) = 1 \rightsquigarrow \\
(q \lor p \to p \lor q) = 0 \rightsquigarrow \\
(q \lor p) = 1, (p \lor q) = 0 \rightsquigarrow \\
(q \lor p) = 1, p = 0, q = 0 \rightsquigarrow \\
q = 1, p = 0, q = 0 \mid p = 1, p = 0, q = 0
\end{array}
$$

Extras: Flat View of Tableaux

We will make the following changes:

1. show a tableau using the $B_1 \mid \cdots \mid B_n$ notation;
2. remove closed branches;
3. if we apply a table expansion rule to a signed formula on a branch, we will remove the formula from the branch.

Consider Example 1 again.

$$
\begin{array}{rcl}
(A_1 \vee A_2) = 0 & \rightsquigarrow & A_1 = 0, A_2 = 0 \\
(A_1 \vee A_2) = 1 & \rightsquigarrow & A_1 = 1 \mid A_2 = 1 \\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

$$
\begin{array}{l}
(\neg(q \vee p \to p \vee q)) = 1 \rightsquigarrow \\
(q \vee p \to p \vee q) = 0 \rightsquigarrow \\
(q \vee p) = 1, (p \vee q) = 0 \rightsquigarrow \\
(q \vee p) = 1, p = 0, q = 0 \rightsquigarrow \\
q = 1, p = 0, q = 0 \mid p = 1, p = 0, q = 0 \rightsquigarrow \\
p = 1, p = 0, q = 0
\end{array}
$$

Extras: Flat View of Tableaux

We will make the following changes:

1. show a tableau using the $B_1 \mid \cdots \mid B_n$ notation;
2. remove closed branches;
3. if we apply a table expansion rule to a signed formula on a branch, we will remove the formula from the branch.

Consider Example 1 again.

$$\begin{array}{rcl} (A_1 \vee A_2) = 0 & \rightsquigarrow & A_1 = 0, A_2 = 0 \\ (A_1 \vee A_2) = 1 & \rightsquigarrow & A_1 = 1 \mid A_2 = 1 \\ (A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\ (\neg A_1) = 1 & \rightsquigarrow & A_1 = 0 \end{array}$$

$(\neg(q \vee p \to p \vee q)) = 1 \rightsquigarrow$
$(q \vee p \to p \vee q) = 0 \rightsquigarrow$
$(q \vee p) = 1, (p \vee q) = 0 \rightsquigarrow$
$(q \vee p) = 1, p = 0, q = 0 \rightsquigarrow$
$q = 1, p = 0, q = 0 \mid p = 1, p = 0, q = 0 \rightsquigarrow$
$p = 1, p = 0, q = 0$

Extras: Flat View of Tableaux

We will make the following changes:

1. show a tableau using the $B_1 \mid \cdots \mid B_n$ notation;
2. remove closed branches;
3. if we apply a table expansion rule to a signed formula on a branch, we will remove the formula from the branch.

Consider Example 1 again.

$$
\begin{array}{rcl}
(A_1 \vee A_2) = 0 & \rightsquigarrow & A_1 = 0, A_2 = 0 \\
(A_1 \vee A_2) = 1 & \rightsquigarrow & A_1 = 1 \mid A_2 = 1 \\
(A_1 \to A_2) = 0 & \rightsquigarrow & A_1 = 1, A_2 = 0 \\
(\neg A_1) = 1 & \rightsquigarrow & A_1 = 0
\end{array}
$$

$(\neg(q \vee p \to p \vee q)) = 1 \rightsquigarrow$
$(q \vee p \to p \vee q) = 0 \rightsquigarrow$
$(q \vee p) = 1, (p \vee q) = 0 \rightsquigarrow$
$(q \vee p) = 1, p = 0, q = 0 \rightsquigarrow$
$q = 1, p = 0, q = 0 \mid p = 1, p = 0, q = 0 \rightsquigarrow$
$p = 1, p = 0, q = 0$

All branches are closed, so the signed formula $(\neg(q \vee p \to p \vee q)) = 1$ is unsatisfiable.

Extras: Flat View of Tableaux: Example 2

$$(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1$$

Extras: Flat View of Tableaux: Example 2

$$(\neg((p \to q) \land (p \land q \to r) \to (\neg p \to r))) = 1$$

Extras: Flat View of Tableaux: Example 2

$(\neg((p \to q) \land (p \land q \to r) \to (\neg p \to r))) = 1 \rightsquigarrow$
$((p \to q) \land (p \land q \to r) \to (\neg p \to r)) = 0$

Extras: Flat View of Tableaux: Example 2

$$(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1 \rightsquigarrow$$
$$((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0$$

Extras: Flat View of Tableaux: Example 2

$(\neg((p \to q) \land (p \land q \to r) \to (\neg p \to r))) = 1 \rightsquigarrow$
$((p \to q) \land (p \land q \to r) \to (\neg p \to r)) = 0 \rightsquigarrow$
$((p \to q) \land (p \land q \to r)) = 1, (\neg p \to r) = 0$

Extras: Flat View of Tableaux: Example 2

$(\neg((p \to q) \land (p \land q \to r) \to (\neg p \to r))) = 1 \rightsquigarrow$
$((p \to q) \land (p \land q \to r) \to (\neg p \to r)) = 0 \rightsquigarrow$
$((p \to q) \land (p \land q \to r)) = 1, (\neg p \to r) = 0$

Extras: Flat View of Tableaux: Example 2

$(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1 \leadsto$
$((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0 \leadsto$
$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p \to r) = 0 \leadsto$
$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p) = 1, r = 0$

Extras: Flat View of Tableaux: Example 2

$$(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1 \rightsquigarrow$$
$$((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0 \rightsquigarrow$$
$$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p \to r) = 0 \rightsquigarrow$$
$$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p) = 1, r = 0$$

Extras: Flat View of Tableaux: Example 2

$(\neg((p \to q) \land (p \land q \to r) \to (\neg p \to r))) = 1 \rightsquigarrow$
$((p \to q) \land (p \land q \to r) \to (\neg p \to r)) = 0 \rightsquigarrow$
$((p \to q) \land (p \land q \to r)) = 1, (\neg p \to r) = 0 \rightsquigarrow$
$((p \to q) \land (p \land q \to r)) = 1, (\neg p) = 1, r = 0 \rightsquigarrow$
$((p \to q) \land (p \land q \to r)) = 1, p = 0, r = 0$

Extras: Flat View of Tableaux: Example 2

$(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p \to r) = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p) = 1, r = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, p = 0, r = 0$

Extras: Flat View of Tableaux: Example 2

$(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p \to r) = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p) = 1, r = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, p = 0, r = 0 \rightsquigarrow$
$(p \to q) = 1, (p \wedge q \to r) = 1, p = 0, r = 0$

Extras: Flat View of Tableaux: Example 2

$(\neg((p \to q) \land (p \land q \to r) \to (\neg p \to r))) = 1 \rightsquigarrow$
$((p \to q) \land (p \land q \to r) \to (\neg p \to r)) = 0 \rightsquigarrow$
$((p \to q) \land (p \land q \to r)) = 1, (\neg p \to r) = 0 \rightsquigarrow$
$((p \to q) \land (p \land q \to r)) = 1, (\neg p) = 1, r = 0 \rightsquigarrow$
$((p \to q) \land (p \land q \to r)) = 1, p = 0, r = 0 \rightsquigarrow$
$(p \to q) = 1, (p \land q \to r) = 1, p = 0, r = 0$

Extras: Flat View of Tableaux: Example 2

$(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p \to r) = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p) = 1, r = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, p = 0, r = 0 \rightsquigarrow$
$(p \to q) = 1, (p \wedge q \to r) = 1, p = 0, r = 0 \rightsquigarrow$
$p = 0, (p \wedge q \to r) = 1, r = 0 \mid$
$q = 1, (p \wedge q \to r) = 1, p = 0, r = 0$

Extras: Flat View of Tableaux: Example 2

$(\neg((p \to q) \land (p \land q \to r) \to (\neg p \to r))) = 1 \rightsquigarrow$
$((p \to q) \land (p \land q \to r) \to (\neg p \to r)) = 0 \rightsquigarrow$
$((p \to q) \land (p \land q \to r)) = 1, (\neg p \to r) = 0 \rightsquigarrow$
$((p \to q) \land (p \land q \to r)) = 1, (\neg p) = 1, r = 0 \rightsquigarrow$
$((p \to q) \land (p \land q \to r)) = 1, p = 0, r = 0 \rightsquigarrow$
$(p \to q) = 1, (p \land q \to r) = 1, p = 0, r = 0 \rightsquigarrow$
$p = 0, (p \land q \to r) = 1, r = 0 \mid$
$q = 1, (p \land q \to r) = 1, p = 0, r = 0$

Extras: Flat View of Tableaux: Example 2

$(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p \to r) = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p) = 1, r = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, p = 0, r = 0 \rightsquigarrow$
$(p \to q) = 1, (p \wedge q \to r) = 1, p = 0, r = 0 \rightsquigarrow$
$p = 0, (p \wedge q \to r) = 1, r = 0 \mid$
$q = 1, (p \wedge q \to r) = 1, p = 0, r = 0 \rightsquigarrow$
$p = 0, (p \wedge q) = 0, r = 0 \mid$
$p = 0, r = 1, r = 0 \mid$
$q = 1, (p \wedge q \to r) = 1, p = 0, r = 0$

Extras: Flat View of Tableaux: Example 2

$(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p \to r) = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p) = 1, r = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, p = 0, r = 0 \rightsquigarrow$
$(p \to q) = 1, (p \wedge q \to r) = 1, p = 0, r = 0 \rightsquigarrow$
$p = 0, (p \wedge q \to r) = 1, r = 0 \mid$
$q = 1, (p \wedge q \to r) = 1, p = 0, r = 0 \rightsquigarrow$
$p = 0, (p \wedge q) = 0, r = 0 \mid$
$p = 0, r = 1, r = 0 \mid$
$q = 1, (p \wedge q \to r) = 1, p = 0, r = 0$

Extras: Flat View of Tableaux: Example 2

$(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p \to r) = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p) = 1, r = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, p = 0, r = 0 \rightsquigarrow$
$(p \to q) = 1, (p \wedge q \to r) = 1, p = 0, r = 0 \rightsquigarrow$
$p = 0, (p \wedge q \to r) = 1, r = 0 \mid$
$q = 1, (p \wedge q \to r) = 1, p = 0, r = 0 \rightsquigarrow$
$p = 0, (p \wedge q) = 0, r = 0 \mid$
$p = 0, r = 1, r = 0 \mid$
$q = 1, (p \wedge q \to r) = 1, p = 0, r = 0 \rightsquigarrow$
$p = 0, r = 0 \mid$
$p = 0, q = 0, r = 0 \mid$
$p = 0, r = 1, r = 0 \mid$
$q = 1, (p \wedge q \to r) = 1, p = 0, r = 0$

Extras: Flat View of Tableaux: Example 2

$(\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))) = 1 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p \to r) = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, (\neg p) = 1, r = 0 \rightsquigarrow$
$((p \to q) \wedge (p \wedge q \to r)) = 1, p = 0, r = 0 \rightsquigarrow$
$(p \to q) = 1, (p \wedge q \to r) = 1, p = 0, r = 0 \rightsquigarrow$
$p = 0, (p \wedge q \to r) = 1, r = 0 \mid$
$q = 1, (p \wedge q \to r) = 1, p = 0, r = 0 \rightsquigarrow$
$p = 0, (p \wedge q) = 0, r = 0 \mid$
$p = 0, r = 1, r = 0 \mid$
$q = 1, (p \wedge q \to r) = 1, p = 0, r = 0 \rightsquigarrow$
$p = 0, r = 0 \mid$
$p = 0, q = 0, r = 0 \mid$
$p = 0, r = 1, r = 0 \mid$
$q = 1, (p \wedge q \to r) = 1, p = 0, r = 0$

The branch containing $p = 0, r = 0$ can no more be expanded or closed so it gives us a model (in fact, two models)

Summary

We were studying various algorithms for satisfiability:

- for general formulas:
  - Splitting algorithm
  - Semantic Tableaux algorithm
- for sets of clauses:
  - DPLL
  - Randomized algorithms