CS:4350 Logic in Computer Science

Conjunctive Normal Form and DPLL

Cesare Tinelli

Spring 2022

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 of clauses

The efficiency of splitting algorithms for satisfiability

- can be massively improved in practice
- by first putting the input formula in normal form

A popular satisfiability procedure called DPLL requires formulas in conjunctive normal form

We will see this next

Satisfiability of clauses

The efficiency of splitting algorithms for satisfiability

- can be massively improved in practice
- by first putting the input formula in normal form

A popular satisfiability procedure called DPLL requires formulas in conjunctive normal form

We will see this next

Satisfiability of clauses

The efficiency of splitting algorithms for satisfiability

- can be massively improved in practice
- by first putting the input formula in normal form

A popular satisfiability procedure called DPLL requires formulas in conjunctive normal form

We will see this next

Literals

Literal: an atom p (*positive literal*) or
a negated atom $\neg p$ (*negative literal*)

The *complement* $\overline{L}$ of a literal L:

$$\overline{L} \stackrel{\text{def}}{=} \begin{cases} \neg L & \text{if } L \text{ is positive} \\ p & \text{if } L \text{ has the form } \neg p \end{cases}$$

Note: p and $\neg p$ are each other's complement

Literals

Literal: an atom p (*positive literal*) or
a negated atom $\neg p$ (*negative literal*)

The *complement* $\overline{L}$ of a literal L:

$$\overline{L} \stackrel{\text{def}}{=} \begin{cases} \neg L & \text{if } L \text{ is positive} \\ p & \text{if } L \text{ has the form } \neg p \end{cases}$$

Note: p and $\neg p$ are each other's complement

Literals

Literal: an atom p (*positive literal*) or
a negated atom $\neg p$ (*negative literal*)

The *complement* $\overline{L}$ of a literal L:

$$\overline{L} \stackrel{\text{def}}{=} \begin{cases} \neg L & \text{if } L \text{ is positive} \\ p & \text{if } L \text{ has the form } \neg p \end{cases}$$

Note: p and $\neg p$ are each other's complement

Clauses

Clause: a disjunction $L_1 \vee \cdots \vee L_n$ of literals with $n \geq 0$

- *Empty clause* ($\square$) : when $n = 0$
- *Unit clause*: when $n = 1$
- *Horn clause*: when it has at most one positive literal

Note: $\square$ is false in every interpretation

Clauses

Clause: a disjunction $L_1 \vee \cdots \vee L_n$ of literals with $n \geq 0$

- *Empty clause* ($\square$) : when $n = 0$
- *Unit clause*: when $n = 1$
- *Horn clause*: when it has at most one positive literal

Note: $\square$ is false in every interpretation

Clauses

Clause: a disjunction $L_1 \vee \cdots \vee L_n$ of literals with $n \geq 0$

- *Empty clause* ($\square$) : when $n = 0$
- *Unit clause*: when $n = 1$
- *Horn clause*: when it has at most one positive literal

Note: $\square$ is false in every interpretation

Conjunctive Normal Form

A formula A is in *conjunctive normal form*, or simply CNF, if it is either $\top$ or a conjunction of clauses:

$$A = \bigwedge_i \bigvee_j L_{i,j}$$

Example

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

A formula B is *a conjunctive normal form of* a formula A if $B \equiv A$ and B is in CNF

Conjunctive Normal Form

A formula A is in *conjunctive normal form*, or simply CNF, if it is either $\top$ or a conjunction of clauses:

$$A = \bigwedge_i \bigvee_j L_{i,j}$$

Example

$$(\neg p \lor q) \land (\neg p \lor \neg q \lor r) \land p \land (\neg q \lor \neg r)$$

A formula B is *a conjunctive normal form of* a formula A if $B \equiv A$ and B is in CNF

Satisfiability on CNF

Note: An interpretation $\mathcal{I}$

1. satisfies a formula in CNF

$$A = \bigwedge_i \bigvee_j L_{i,j}$$

   iff it satisfies every clause $C_i = \bigvee_j L_{i,j}$ in A

2. satisfies a clause

$$C = L_1 \vee \cdots \vee L_n$$

   iff it satisfies some (i.e., at least one) literal L_k in C

Satisfiability on CNF

Note: An interpretation $\mathcal{I}$

1. satisfies a formula in CNF

$$A = \bigwedge_i \bigvee_j L_{i,j}$$

   iff it satisfies every clause $C_i = \bigvee_j L_{i,j}$ in A

2. satisfies a clause

$$C = L_1 \vee \cdots \vee L_n$$

   iff it satisfies some (i.e., at least one) literal L_k in C

CNF transformation

Any propositional formula can be converted to CNF by the repeated applications of these rewrite rules:

$$
\begin{array}{lrcl}
1. & A \leftrightarrow B & \Rightarrow & (\neg A \vee B) \wedge (\neg B \vee A) \\
2. & A \rightarrow B & \Rightarrow & \neg A \vee B \\
3. & \neg(A \wedge B) & \Rightarrow & \neg A \vee \neg B \\
4. & \neg(A \vee B) & \Rightarrow & \neg A \wedge \neg B \\
5. & \neg\neg A & \Rightarrow & A \\
6. & (A_1 \wedge \cdots \wedge A_m) \vee B_1 \vee \cdots \vee B_n & \Rightarrow & (A_1 \vee B_1 \vee \cdots \vee B_n) \quad \wedge \\
 & & & \cdots \quad \wedge \\
 & & & (A_m \vee B_1 \vee \cdots \vee B_n)
\end{array}
$$

A formula to which no rewrite rule is applicable

- contains no $\leftrightarrow$
- contains no $\rightarrow$
- may contain $\neg$ but only applied to atoms
- does not contain $\wedge$ in the scope of $\vee$
- (hence) is in CNF

CNF transformation

Any propositional formula can be converted to CNF by the repeated applications of these rewrite rules:

$$
\begin{array}{lrcl}
1. & A \leftrightarrow B & \Rightarrow & (\neg A \lor B) \land (\neg B \lor A) \\
2. & A \rightarrow B & \Rightarrow & \neg A \lor B \\
3. & \neg(A \land B) & \Rightarrow & \neg A \lor \neg B \\
4. & \neg(A \lor B) & \Rightarrow & \neg A \land \neg B \\
5. & \neg\neg A & \Rightarrow & A \\
6. & (A_1 \land \cdots \land A_m) \lor B_1 \lor \cdots \lor B_n & \Rightarrow & (A_1 \lor B_1 \lor \cdots \lor B_n) \quad \land \\
 & & & \cdots \quad \land \\
 & & & (A_m \lor B_1 \lor \cdots \lor B_n)
\end{array}
$$

A formula to which no rewrite rule is applicable

- contains no $\leftrightarrow$
- contains no $\rightarrow$
- may contain $\neg$ but only applied to atoms
- does not contain $\land$ in the scope of $\lor$
- (hence) is in CNF

CNF transformation

Any propositional formula can be converted to CNF by the repeated applications of these rewrite rules:

$$
\begin{array}{lrcl}
1. & A \leftrightarrow B & \Rightarrow & (\neg A \vee B) \wedge (\neg B \vee A) \\
2. & A \rightarrow B & \Rightarrow & \neg A \vee B \\
3. & \neg(A \wedge B) & \Rightarrow & \neg A \vee \neg B \\
4. & \neg(A \vee B) & \Rightarrow & \neg A \wedge \neg B \\
5. & \neg\neg A & \Rightarrow & A \\
6. & (A_1 \wedge \cdots \wedge A_m) \vee B_1 \vee \cdots \vee B_n & \Rightarrow & (A_1 \vee B_1 \vee \cdots \vee B_n) \quad \wedge \\
 & & & \cdots \quad \wedge \\
 & & & (A_m \vee B_1 \vee \cdots \vee B_n)
\end{array}
$$

A formula to which no rewrite rule is applicable

- contains no $\leftrightarrow$
- contains no $\rightarrow$
- may contain $\neg$ but only applied to atoms
- does not contain $\wedge$ in the scope of $\vee$
- (hence) is in CNF

CNF transformation

Any propositional formula can be converted to CNF by the repeated applications of these rewrite rules:

$$
\begin{array}{lrcl}
1. & A \leftrightarrow B & \Rightarrow & (\neg A \vee B) \wedge (\neg B \vee A) \\
2. & A \rightarrow B & \Rightarrow & \neg A \vee B \\
3. & \neg(A \wedge B) & \Rightarrow & \neg A \vee \neg B \\
4. & \neg(A \vee B) & \Rightarrow & \neg A \wedge \neg B \\
5. & \neg\neg A & \Rightarrow & A \\
6. & (A_1 \wedge \cdots \wedge A_m) \vee B_1 \vee \cdots \vee B_n & \Rightarrow & (A_1 \vee B_1 \vee \cdots \vee B_n) \quad \wedge \\
 & & & \cdots \quad \wedge \\
 & & & (A_m \vee B_1 \vee \cdots \vee B_n)
\end{array}
$$

A formula to which no rewrite rule is applicable

- contains no $\leftrightarrow$
- contains no $\rightarrow$
- may contain $\neg$ but only applied to atoms
- does not contain $\wedge$ in the scope of $\vee$
- (hence) is in CNF

CNF transformation

Any propositional formula can be converted to CNF by the repeated applications of these rewrite rules:

$$
\begin{array}{lrcl}
1. & A \leftrightarrow B & \Rightarrow & (\neg A \vee B) \wedge (\neg B \vee A) \\
2. & A \rightarrow B & \Rightarrow & \neg A \vee B \\
3. & \neg(A \wedge B) & \Rightarrow & \neg A \vee \neg B \\
4. & \neg(A \vee B) & \Rightarrow & \neg A \wedge \neg B \\
5. & \neg\neg A & \Rightarrow & A \\
6. & (A_1 \wedge \cdots \wedge A_m) \vee B_1 \vee \cdots \vee B_n & \Rightarrow & (A_1 \vee B_1 \vee \cdots \vee B_n) \quad \wedge \\
 & & & \cdots \quad \wedge \\
 & & & (A_m \vee B_1 \vee \cdots \vee B_n)
\end{array}
$$

A formula to which no rewrite rule is applicable

- contains no $\leftrightarrow$
- contains no $\rightarrow$
- may contain $\neg$ but only applied to atoms
- does not contain $\wedge$ in the scope of $\vee$
- (hence) is in CNF

CNF transformation

Any propositional formula can be converted to CNF by the repeated applications of these rewrite rules:

$$
\begin{array}{lrcl}
1. & A \leftrightarrow B & \Rightarrow & (\neg A \vee B) \wedge (\neg B \vee A) \\
2. & A \rightarrow B & \Rightarrow & \neg A \vee B \\
3. & \neg(A \wedge B) & \Rightarrow & \neg A \vee \neg B \\
4. & \neg(A \vee B) & \Rightarrow & \neg A \wedge \neg B \\
5. & \neg\neg A & \Rightarrow & A \\
6. & (A_1 \wedge \cdots \wedge A_m) \vee B_1 \vee \cdots \vee B_n & \Rightarrow & (A_1 \vee B_1 \vee \cdots \vee B_n) \quad \wedge \\
 & & & \cdots \quad \wedge \\
 & & & (A_m \vee B_1 \vee \cdots \vee B_n)
\end{array}
$$

A formula to which no rewrite rule is applicable

- contains no $\leftrightarrow$
- contains no $\rightarrow$
- may contain $\neg$ but only applied to atoms
- does not contain $\wedge$ in the scope of $\vee$
- (hence) is in CNF

CNF, example

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

$$\neg(\neg((p \to q) \land (p \land q \to r)) \lor (p \to r)) \Rightarrow$$

$$\neg\neg((p \to q) \land (p \land q \to r)) \land \neg(p \to r) \Rightarrow$$

$$(p \to q) \land (p \land q \to r) \land \neg(p \to r) \Rightarrow$$

$$(p \to q) \land (p \land q \to r) \land \neg(\neg p \lor r) \Rightarrow$$

$$(p \to q) \land (p \land q \to r) \land \neg\neg p \land \neg r \Rightarrow$$

$$(p \to q) \land (p \land q \to r) \land p \land \neg r \Rightarrow$$

$$(p \to q) \land (\neg(p \land q) \lor r) \land p \land \neg r \Rightarrow$$

$$(p \to q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$

$$(\neg p \lor q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$

$$\begin{array}{rcl}
A \leftrightarrow B & \Rightarrow & (\neg A \lor B) \land (\neg B \lor A) \\
A \to B & \Rightarrow & \neg A \lor B \\
\neg(A \land B) & \Rightarrow & \neg A \lor \neg B \\
\neg(A \lor B) & \Rightarrow & \neg A \land \neg B \\
\neg\neg A & \Rightarrow & A \\
(A_1 \land \cdots \land A_m) & & (A_1 \lor B_1 \lor \cdots \lor B_n) \land \\
\lor B_1 \lor \cdots \lor B_n & \Rightarrow & \cdots \land \\
 & & (A_m \lor B_1 \lor \cdots \lor B_n)
\end{array}$$

CNF, example

$$\neg((p \to q) \land (p \land q \to r) \to (p \to r)) \Rightarrow$$
$$\neg(\neg((p \to q) \land (p \land q \to r)) \lor (p \to r)) \Rightarrow$$
$$\neg\neg((p \to q) \land (p \land q \to r)) \land \neg(p \to r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg(p \to r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg(\neg p \lor r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg\neg p \land \neg r \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land p \land \neg r \Rightarrow$$
$$(p \to q) \land (\neg(p \land q) \lor r) \land p \land \neg r \Rightarrow$$
$$(p \to q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$
$$(\neg p \lor q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$

$$\begin{aligned}
A \leftrightarrow B &\Rightarrow (\neg A \lor B) \land (\neg B \lor A) \\
A \to B &\Rightarrow \neg A \lor B \\
\neg(A \land B) &\Rightarrow \neg A \lor \neg B \\
\neg(A \lor B) &\Rightarrow \neg A \land \neg B \\
\neg\neg A &\Rightarrow A \\
(A_1 \land \cdots \land A_m) \lor B_1 \lor \cdots \lor B_n &\Rightarrow (A_1 \lor B_1 \lor \cdots \lor B_n) \land \cdots \land (A_m \lor B_1 \lor \cdots \lor B_n)
\end{aligned}$$

CNF, example

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

$$\begin{array}{rcl}
A \leftrightarrow B & \Rightarrow & (\neg A \vee B) \wedge (\neg B \vee A) \\
A \to B & \Rightarrow & \neg A \vee B \\
\neg(A \wedge B) & \Rightarrow & \neg A \vee \neg B \\
\neg(A \vee B) & \Rightarrow & \neg A \wedge \neg B \\
\neg\neg A & \Rightarrow & A \\
(A_1 \wedge \cdots \wedge A_m) & & (A_1 \vee B_1 \vee \cdots \vee B_n) \wedge \\
\vee B_1 \vee \cdots \vee B_n & \Rightarrow & \cdots \wedge \\
 & & (A_m \vee B_1 \vee \cdots \vee B_n)
\end{array}$$

CNF, example

$$\neg((p \to q) \land (p \land q \to r) \to (p \to r)) \Rightarrow$$

$$\neg(\neg((p \to q) \land (p \land q \to r)) \lor (p \to r)) \Rightarrow$$

$$\neg\neg((p \to q) \land (p \land q \to r)) \land \neg(p \to r) \Rightarrow$$

$$(p \to q) \land (p \land q \to r) \land \neg(p \to r) \Rightarrow$$

$$(p \to q) \land (p \land q \to r) \land \neg(\neg p \lor r) \Rightarrow$$

$$(p \to q) \land (p \land q \to r) \land \neg\neg p \land \neg r \Rightarrow$$

$$(p \to q) \land (p \land q \to r) \land p \land \neg r \Rightarrow$$

$$(p \to q) \land (\neg(p \land q) \lor r) \land p \land \neg r \Rightarrow$$

$$(p \to q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$

$$(\neg p \lor q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$

$$\begin{array}{rcl}
A \leftrightarrow B & \Rightarrow & (\neg A \lor B) \land (\neg B \lor A) \\
A \to B & \Rightarrow & \neg A \lor B \\
\neg(A \land B) & \Rightarrow & \neg A \lor \neg B \\
\neg(A \lor B) & \Rightarrow & \neg A \land \neg B \\
\neg\neg A & \Rightarrow & A \\
(A_1 \land \cdots \land A_m) & & (A_1 \lor B_1 \lor \cdots \lor B_n) \land \\
\lor B_1 \lor \cdots \lor B_n & \Rightarrow & \cdots \land \\
 & & (A_m \lor B_1 \lor \cdots \lor B_n)
\end{array}$$

CNF, example

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

$$\begin{aligned}
A \leftrightarrow B &\Rightarrow (\neg A \vee B) \wedge (\neg B \vee A) \\
A \to B &\Rightarrow \neg A \vee B \\
\neg(A \wedge B) &\Rightarrow \neg A \vee \neg B \\
\neg(A \vee B) &\Rightarrow \neg A \wedge \neg B \\
\neg\neg A &\Rightarrow A \\
(A_1 \wedge \cdots \wedge A_m) \vee B_1 \vee \cdots \vee B_n &\Rightarrow (A_1 \vee B_1 \vee \cdots \vee B_n) \wedge \cdots \wedge (A_m \vee B_1 \vee \cdots \vee B_n)
\end{aligned}$$

CNF, example

$$\neg((p \to q) \land (p \land q \to r) \to (p \to r)) \Rightarrow$$
$$\neg(\neg((p \to q) \land (p \land q \to r)) \lor (p \to r)) \Rightarrow$$
$$\neg\neg((p \to q) \land (p \land q \to r)) \land \neg(p \to r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg(p \to r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg(\neg p \lor r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg\neg p \land \neg r \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land p \land \neg r \Rightarrow$$
$$(p \to q) \land (\neg(p \land q) \lor r) \land p \land \neg r \Rightarrow$$
$$(p \to q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$
$$(\neg p \lor q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$

$$\begin{aligned}
A \leftrightarrow B &\Rightarrow (\neg A \lor B) \land (\neg B \lor A) \\
A \to B &\Rightarrow \neg A \lor B \\
\neg(A \land B) &\Rightarrow \neg A \lor \neg B \\
\neg(A \lor B) &\Rightarrow \neg A \land \neg B \\
\neg\neg A &\Rightarrow A \\
(A_1 \land \dots \land A_m) & \quad (A_1 \lor B_1 \lor \dots \lor B_n) \land \\
\lor B_1 \lor \dots \lor B_n &\Rightarrow \quad \dots \land \\
& \quad (A_m \lor B_1 \lor \dots \lor B_n)
\end{aligned}$$

CNF, example

$$
\begin{array}{l}
\neg((p \to q) \land (p \land q \to r) \to (p \to r)) \Rightarrow \\
\neg(\neg((p \to q) \land (p \land q \to r)) \lor (p \to r)) \Rightarrow \\
\neg\neg((p \to q) \land (p \land q \to r)) \land \neg(p \to r) \Rightarrow \\
(p \to q) \land (p \land q \to r) \land \neg(p \to r) \Rightarrow \\
(p \to q) \land (p \land q \to r) \land \neg(\neg p \lor r) \Rightarrow \\
(p \to q) \land (p \land q \to r) \land \neg\neg p \land \neg r \Rightarrow \\
(p \to q) \land (p \land q \to r) \land p \land \neg r \Rightarrow \\
(p \to q) \land (\neg(p \land q) \lor r) \land p \land \neg r \Rightarrow \\
(p \to q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r \\
(\neg p \lor q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r
\end{array}
$$

$$
\begin{array}{rcl}
A \leftrightarrow B & \Rightarrow & (\neg A \lor B) \land (\neg B \lor A) \\
A \to B & \Rightarrow & \neg A \lor B \\
\neg(A \land B) & \Rightarrow & \neg A \lor \neg B \\
\neg(A \lor B) & \Rightarrow & \neg A \land \neg B \\
\neg\neg A & \Rightarrow & A \\
(A_1 \land \cdots \land A_m) & & (A_1 \lor B_1 \lor \cdots \lor B_n) \land \\
\lor B_1 \lor \cdots \lor B_n & \Rightarrow & \quad \cdots \quad \land \\
& & (A_m \lor B_1 \lor \cdots \lor B_n)
\end{array}
$$

CNF, example

$$\neg((p \to q) \land (p \land q \to r) \to (p \to r)) \Rightarrow$$
$$\neg(\neg((p \to q) \land (p \land q \to r)) \lor (p \to r)) \Rightarrow$$
$$\neg\neg((p \to q) \land (p \land q \to r)) \land \neg(p \to r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg(p \to r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg(\neg p \lor r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg\neg p \land \neg r \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land p \land \neg r \Rightarrow$$
$$(p \to q) \land (\neg(p \land q) \lor r) \land p \land \neg r \Rightarrow$$
$$(p \to q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$
$$(\neg p \lor q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$

$$\begin{array}{rcl}
A \leftrightarrow B & \Rightarrow & (\neg A \lor B) \land (\neg B \lor A) \\
A \to B & \Rightarrow & \neg A \lor B \\
\neg(A \land B) & \Rightarrow & \neg A \lor \neg B \\
\neg(A \lor B) & \Rightarrow & \neg A \land \neg B \\
\neg\neg A & \Rightarrow & A \\
(A_1 \land \cdots \land A_m) & & (A_1 \lor B_1 \lor \cdots \lor B_n) \land \\
\lor B_1 \lor \cdots \lor B_n & \Rightarrow & \cdots \land \\
 & & (A_m \lor B_1 \lor \cdots \lor B_n)
\end{array}$$

CNF, example

$$\begin{aligned}
A \leftrightarrow B &\Rightarrow (\neg A \vee B) \wedge (\neg B \vee A) \\
A \rightarrow B &\Rightarrow \neg A \vee B \\
\neg(A \wedge B) &\Rightarrow \neg A \vee \neg B \\
\neg(A \vee B) &\Rightarrow \neg A \wedge \neg B \\
\neg\neg A &\Rightarrow A \\
(A_1 \wedge \cdots \wedge A_m) \vee B_1 \vee \cdots \vee B_n &\Rightarrow (A_1 \vee B_1 \vee \cdots \vee B_n) \wedge \cdots \wedge (A_m \vee B_1 \vee \cdots \vee B_n)
\end{aligned}$$

$$\begin{aligned}
&\neg((p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)) \Rightarrow \\
&\neg(\neg((p \rightarrow q) \wedge (p \wedge q \rightarrow r)) \vee (p \rightarrow r)) \Rightarrow \\
&\neg\neg((p \rightarrow q) \wedge (p \wedge q \rightarrow r)) \wedge \neg(p \rightarrow r) \Rightarrow \\
&(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \wedge \neg(p \rightarrow r) \Rightarrow \\
&(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \wedge \neg(\neg p \vee r) \Rightarrow \\
&(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \wedge \neg\neg p \wedge \neg r \Rightarrow \\
&(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \wedge p \wedge \neg r \Rightarrow \\
&(p \rightarrow q) \wedge (\neg(p \wedge q) \vee r) \wedge p \wedge \neg r \Rightarrow \\
&(p \rightarrow q) \wedge (\neg p \vee \neg q \vee r) \wedge p \wedge \neg r \\
&(\neg p \vee q) \wedge (\neg p \vee \neg q \vee r) \wedge p \wedge \neg r
\end{aligned}$$

CNF, example

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

$$\begin{array}{rcl}
A \leftrightarrow B & \Rightarrow & (\neg A \vee B) \wedge (\neg B \vee A) \\
A \rightarrow B & \Rightarrow & \neg A \vee B \\
\neg(A \wedge B) & \Rightarrow & \neg A \vee \neg B \\
\neg(A \vee B) & \Rightarrow & \neg A \wedge \neg B \\
\neg\neg A & \Rightarrow & A \\
(A_1 \wedge \cdots \wedge A_m) & & (A_1 \vee B_1 \vee \cdots \vee B_n) \wedge \\
\vee B_1 \vee \cdots \vee B_n & \Rightarrow & \cdots \wedge \\
& & (A_m \vee B_1 \vee \cdots \vee B_n)
\end{array}$$

CNF, example

$$\begin{aligned}
A \leftrightarrow B &\Rightarrow (\neg A \vee B) \wedge (\neg B \vee A) \\
A \rightarrow B &\Rightarrow \neg A \vee B \\
\neg(A \wedge B) &\Rightarrow \neg A \vee \neg B \\
\neg(A \vee B) &\Rightarrow \neg A \wedge \neg B \\
\neg\neg A &\Rightarrow A \\
(A_1 \wedge \cdots \wedge A_m) \vee B_1 \vee \cdots \vee B_n &\Rightarrow (A_1 \vee B_1 \vee \cdots \vee B_n) \wedge \cdots \wedge (A_m \vee B_1 \vee \cdots \vee B_n)
\end{aligned}$$

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

CNF, example

$$\neg((p \to q) \land (p \land q \to r) \to (p \to r)) \Rightarrow$$
$$\neg(\neg((p \to q) \land (p \land q \to r)) \lor (p \to r)) \Rightarrow$$
$$\neg\neg((p \to q) \land (p \land q \to r)) \land \neg(p \to r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg(p \to r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg(\neg p \lor r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg\neg p \land \neg r \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land p \land \neg r \Rightarrow$$
$$(p \to q) \land (\neg(p \land q) \lor r) \land p \land \neg r \Rightarrow$$
$$(p \to q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$
$$(\neg p \lor q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$

$$\begin{array}{rcl}
A \leftrightarrow B & \Rightarrow & (\neg A \lor B) \land (\neg B \lor A) \\
A \to B & \Rightarrow & \neg A \lor B \\
\neg(A \land B) & \Rightarrow & \neg A \lor \neg B \\
\neg(A \lor B) & \Rightarrow & \neg A \land \neg B \\
\neg\neg A & \Rightarrow & A \\
(A_1 \land \cdots \land A_m) & & (A_1 \lor B_1 \lor \cdots \lor B_n) \land \\
\lor B_1 \lor \cdots \lor B_n & \Rightarrow & \cdots \land \\
 & & (A_m \lor B_1 \lor \cdots \lor B_n)
\end{array}$$

CNF, example

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

$$\begin{array}{rcl}
A \leftrightarrow B & \Rightarrow & (\neg A \vee B) \wedge (\neg B \vee A) \\
A \to B & \Rightarrow & \neg A \vee B \\
\neg(A \wedge B) & \Rightarrow & \neg A \vee \neg B \\
\neg(A \vee B) & \Rightarrow & \neg A \wedge \neg B \\
\neg\neg A & \Rightarrow & A \\
(A_1 \wedge \cdots \wedge A_m) & & (A_1 \vee B_1 \vee \cdots \vee B_n) \wedge \\
\vee B_1 \vee \cdots \vee B_n & \Rightarrow & \cdots \wedge \\
 & & (A_m \vee B_1 \vee \cdots \vee B_n)
\end{array}$$

CNF, example

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

$$\begin{array}{rcl}
A \leftrightarrow B & \Rightarrow & (\neg A \vee B) \wedge (\neg B \vee A) \\
A \to B & \Rightarrow & \neg A \vee B \\
\neg(A \wedge B) & \Rightarrow & \neg A \vee \neg B \\
\neg(A \vee B) & \Rightarrow & \neg A \wedge \neg B \\
\neg\neg A & \Rightarrow & A \\
(A_1 \wedge \cdots \wedge A_m) & & (A_1 \vee B_1 \vee \cdots \vee B_n) \wedge \\
\vee B_1 \vee \cdots \vee B_n & \Rightarrow & \cdots \wedge \\
 & & (A_m \vee B_1 \vee \cdots \vee B_n)
\end{array}$$

CNF, example

$$
\begin{aligned}
&\neg((p \to q) \land (p \land q \to r) \to (p \to r)) \Rightarrow \\
&\neg(\neg((p \to q) \land (p \land q \to r)) \lor (p \to r)) \Rightarrow \\
&\neg\neg((p \to q) \land (p \land q \to r)) \land \neg(p \to r) \Rightarrow \\
&(p \to q) \land (p \land q \to r) \land \neg(p \to r) \Rightarrow \\
&(p \to q) \land (p \land q \to r) \land \neg(\neg p \lor r) \Rightarrow \\
&(p \to q) \land (p \land q \to r) \land \neg\neg p \land \neg r \Rightarrow \\
&(p \to q) \land (p \land q \to r) \land p \land \neg r \Rightarrow \\
&(p \to q) \land (\neg(p \land q) \lor r) \land p \land \neg r \Rightarrow \\
&(p \to q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r \\
&(\neg p \lor q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r
\end{aligned}
$$

$$
\begin{aligned}
A \leftrightarrow B &\Rightarrow (\neg A \lor B) \land (\neg B \lor A) \\
A \to B &\Rightarrow \neg A \lor B \\
\neg(A \land B) &\Rightarrow \neg A \lor \neg B \\
\neg(A \lor B) &\Rightarrow \neg A \land \neg B \\
\neg\neg A &\Rightarrow A \\
(A_1 \land \cdots \land A_m) \lor B_1 \lor \cdots \lor B_n &\Rightarrow (A_1 \lor B_1 \lor \cdots \lor B_n) \land \cdots \land (A_m \lor B_1 \lor \cdots \lor B_n)
\end{aligned}
$$

CNF, example

$$
\begin{aligned}
&\neg((p \to q) \land (p \land q \to r) \to (p \to r)) \Rightarrow \\
&\neg(\neg((p \to q) \land (p \land q \to r)) \lor (p \to r)) \Rightarrow \\
&\neg\neg((p \to q) \land (p \land q \to r)) \land \neg(p \to r) \Rightarrow \\
&(p \to q) \land (p \land q \to r) \land \neg(p \to r) \Rightarrow \\
&(p \to q) \land (p \land q \to r) \land \neg(\neg p \lor r) \Rightarrow \\
&(p \to q) \land (p \land q \to r) \land \neg\neg p \land \neg r \Rightarrow \\
&(p \to q) \land (p \land q \to r) \land p \land \neg r \Rightarrow \\
&(p \to q) \land (\neg(p \land q) \lor r) \land p \land \neg r \Rightarrow \\
&(p \to q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r \\
&(\neg p \lor q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r
\end{aligned}
$$

$$
\begin{aligned}
A \leftrightarrow B &\Rightarrow (\neg A \lor B) \land (\neg B \lor A) \\
A \to B &\Rightarrow \neg A \lor B \\
\neg(A \land B) &\Rightarrow \neg A \lor \neg B \\
\neg(A \lor B) &\Rightarrow \neg A \land \neg B \\
\neg\neg A &\Rightarrow A \\
(A_1 \land \cdots \land A_m) \lor B_1 \lor \cdots \lor B_n &\Rightarrow (A_1 \lor B_1 \lor \cdots \lor B_n) \land \cdots \land (A_m \lor B_1 \lor \cdots \lor B_n)
\end{aligned}
$$

CNF, example

$$\neg((p \to q) \land (p \land q \to r) \to (p \to r)) \Rightarrow$$
$$\neg(\neg((p \to q) \land (p \land q \to r)) \lor (p \to r)) \Rightarrow$$
$$\neg\neg((p \to q) \land (p \land q \to r)) \land \neg(p \to r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg(p \to r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg(\neg p \lor r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg\neg p \land \neg r \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land p \land \neg r \Rightarrow$$
$$(p \to q) \land (\neg(p \land q) \lor r) \land p \land \neg r \Rightarrow$$
$$(p \to q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$
$$(\neg p \lor q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$

$$\begin{aligned}
A \leftrightarrow B &\Rightarrow (\neg A \lor B) \land (\neg B \lor A) \\
A \to B &\Rightarrow \neg A \lor B \\
\neg(A \land B) &\Rightarrow \neg A \lor \neg B \\
\neg(A \lor B) &\Rightarrow \neg A \land \neg B \\
\neg\neg A &\Rightarrow A \\
(A_1 \land \cdots \land A_m) \lor B_1 \lor \cdots \lor B_n &\Rightarrow (A_1 \lor B_1 \lor \cdots \lor B_n) \land \cdots \land (A_m \lor B_1 \lor \cdots \lor B_n)
\end{aligned}$$

CNF, example

$$
\begin{aligned}
&\neg((p \to q) \land (p \land q \to r) \to (p \to r)) \Rightarrow \\
&\neg(\neg((p \to q) \land (p \land q \to r)) \lor (p \to r)) \Rightarrow \\
&\neg\neg((p \to q) \land (p \land q \to r)) \land \neg(p \to r) \Rightarrow \\
&(p \to q) \land (p \land q \to r) \land \neg(p \to r) \Rightarrow \\
&(p \to q) \land (p \land q \to r) \land \neg(\neg p \lor r) \Rightarrow \\
&(p \to q) \land (p \land q \to r) \land \neg\neg p \land \neg r \Rightarrow \\
&(p \to q) \land (p \land q \to r) \land p \land \neg r \Rightarrow \\
&(p \to q) \land (\neg(p \land q) \lor r) \land p \land \neg r \Rightarrow \\
&(p \to q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r \\
&(\neg p \lor q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r
\end{aligned}
$$

$$
\begin{aligned}
A \leftrightarrow B &\Rightarrow (\neg A \lor B) \land (\neg B \lor A) \\
A \to B &\Rightarrow \neg A \lor B \\
\neg(A \land B) &\Rightarrow \neg A \lor \neg B \\
\neg(A \lor B) &\Rightarrow \neg A \land \neg B \\
\neg\neg A &\Rightarrow A \\
(A_1 \land \cdots \land A_m) &\quad (A_1 \lor B_1 \lor \cdots \lor B_n) \land \\
\lor B_1 \lor \cdots \lor B_n &\Rightarrow \quad \cdots \quad \land \\
&\quad (A_m \lor B_1 \lor \cdots \lor B_n)
\end{aligned}
$$

CNF, example

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

$$\begin{array}{rcl}
A \leftrightarrow B & \Rightarrow & (\neg A \vee B) \wedge (\neg B \vee A) \\
A \to B & \Rightarrow & \neg A \vee B \\
\neg(A \wedge B) & \Rightarrow & \neg A \vee \neg B \\
\neg(A \vee B) & \Rightarrow & \neg A \wedge \neg B \\
\neg\neg A & \Rightarrow & A \\
(A_1 \wedge \cdots \wedge A_m) & & (A_1 \vee B_1 \vee \cdots \vee B_n) \wedge \\
\vee B_1 \vee \cdots \vee B_n & \Rightarrow & \cdots \wedge \\
 & & (A_m \vee B_1 \vee \cdots \vee B_n)
\end{array}$$

CNF, example

$$\neg((p \to q) \land (p \land q \to r) \to (p \to r)) \Rightarrow$$
$$\neg(\neg((p \to q) \land (p \land q \to r)) \lor (p \to r)) \Rightarrow$$
$$\neg\neg((p \to q) \land (p \land q \to r)) \land \neg(p \to r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg(p \to r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg(\neg p \lor r) \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land \neg\neg p \land \neg r \Rightarrow$$
$$(p \to q) \land (p \land q \to r) \land p \land \neg r \Rightarrow$$
$$(p \to q) \land (\neg(p \land q) \lor r) \land p \land \neg r \Rightarrow$$
$$(p \to q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$
$$(\neg p \lor q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$

$$\begin{aligned}
A \leftrightarrow B &\Rightarrow (\neg A \lor B) \land (\neg B \lor A) \\
A \to B &\Rightarrow \neg A \lor B \\
\neg(A \land B) &\Rightarrow \neg A \lor \neg B \\
\neg(A \lor B) &\Rightarrow \neg A \land \neg B \\
\neg\neg A &\Rightarrow A \\
(A_1 \land \cdots \land A_m) \lor B_1 \lor \cdots \lor B_n &\Rightarrow (A_1 \lor B_1 \lor \cdots \lor B_n) \land \cdots \land (A_m \lor B_1 \lor \cdots \lor B_n)
\end{aligned}$$

Theorem 1

If A' is obtained from A by one or more applications of the CNF conversion rules, then $A' \equiv A$.

CNF and satisfiability

$$A = \neg((p \to q) \land (p \land q \to r) \to (p \to r)) \Rightarrow$$
$$\cdots$$
$$(\neg p \lor q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$

Note: Formula A above has the same models as the set consisting of these four clauses

$$\neg p \lor q$$
$$\neg p \lor \neg q \lor r$$
$$p$$
$$\neg r$$

(An interpretation *satisfies, or is a model of, a set S* of formulas if it satisfies every formula in S)

CNF and satisfiability

$$A = \neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r)) \Rightarrow$$
$$\ldots$$
$$(\neg p \vee q) \wedge (\neg p \vee \neg q \vee r) \wedge p \wedge \neg r$$

Note: Formula A above has the same models as the set consisting of these four clauses

$$\neg p \vee q$$
$$\neg p \vee \neg q \vee r$$
$$p$$
$$\neg r$$

(An interpretation *satisfies, or is a model of, a set S* of formulas if it satisfies every formula in S)

CNF and satisfiability

$$A = \neg((p \to q) \land (p \land q \to r) \to (p \to r)) \Rightarrow$$
$$\ldots$$
$$(\neg p \lor q) \land (\neg p \lor \neg q \lor r) \land p \land \neg r$$

Note: Formula A above has the same models as the set consisting of these four clauses

$$\neg p \lor q$$
$$\neg p \lor \neg q \lor r$$
$$p$$
$$\neg r$$

(An interpretation *satisfies, or is a model of, a set* S of formulas if it satisfies every formula in S)

CNF and satisfiability

$$A = \neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r)) \Rightarrow$$
$$\ldots$$
$$(\neg p \vee q) \wedge (\neg p \vee \neg q \vee r) \wedge p \wedge \neg r$$

Note: ng of

these

The CNF transformation reduces the sat problem for formulas to the sat problem for sets of clauses

p

$\neg r$

(An interpretation *satisfies, or is a model of, a set* S of formulas if it satisfies every formula in S)

Problem

Let's compute the CNF of

$$F = p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6))))$$

$$p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6))))$$

Problem

Let's compute the CNF of

$$F = p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6))))$$

$$\begin{aligned}
& p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6)))) \Rightarrow \\
& (\neg p_1 \lor (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6))))) \land \\
& (p_1 \lor \neg(p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6))))) \\
& (\neg p_1 \lor \quad ((\neg p_2 \lor (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6)))) \land \\
& \qquad\qquad (p_2 \lor \neg(p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6)))))) \land \\
& (p_1 \lor \neg(p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6)))))
\end{aligned}$$

If we continue, the formula will grow exponentially large!

Problem

Let's compute the CNF of

$$F = p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6))))$$

$$
\begin{aligned}
& p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6)))) \Rightarrow \\
& (\neg p_1 \vee (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6))))) \wedge \\
& (p_1 \vee \neg(p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6))))) \Rightarrow \\
& (\neg p_1 \vee \quad ((\neg p_2 \vee (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6)))) \wedge \\
& \qquad\qquad (p_2 \vee \neg(p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6)))))) \wedge \\
& (p_1 \vee \neg(p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6)))))
\end{aligned}
$$

If we continue, the formula will grow exponentially large!

Problem

Let's compute the CNF of

$$F = p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6))))$$

$$\begin{aligned}
& p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6)))) \Rightarrow \\
& (\neg p_1 \vee (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6))))) \wedge \\
& (p_1 \vee \neg(p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6))))) \Rightarrow \\
& (\neg p_1 \vee \quad ((\neg p_2 \vee (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6)))) \wedge \\
& \qquad\qquad (p_2 \vee \neg(p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6)))))) \wedge \\
& (p_1 \vee \neg(p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6)))))
\end{aligned}$$

If we continue, the formula will grow exponentially large!

Problem

Let's compute the CNF of

$$F = p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6))))$$

$$\begin{aligned}
& p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6)))) \Rightarrow \\
& (\neg p_1 \lor (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6))))) \land \\
& (p_1 \lor \neg(p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6))))) \Rightarrow \\
& (\neg p_1 \lor \quad ((\neg p_2 \lor (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6)))) \land \\
& \qquad\qquad (p_2 \lor \neg(p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6)))))) \land \\
& (p_1 \lor \neg(p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6)))))
\end{aligned}$$

If we continue, the formula will grow exponentially large!

CNF transformation can be exponential

There are formulas whose shortest CNF has an exponential size

Is there any way to avoid exponential blowup? Yes!

CNF transformation can be exponential

There are formulas whose shortest CNF has an exponential size

Is there any way to avoid exponential blowup? Yes!

CNF transformation can be exponential

There are formulas whose shortest CNF has an exponential size

Is there any way to avoid exponential blowup? Yes!

A space-efficient CNF transformation

Using so-called *naming* or *definition introduction*

1. Take a non-literal subformula A of formula F
2. Introduce a new *name* n for it, i.e., a fresh propositional variable
3. Add a *definition for* n, i.e., a formula stating that n is equivalent to A

$$\begin{array}{rcl} F & = & p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (\overbrace{p_5 \leftrightarrow p_6}^{A})))) \\ & & n \leftrightarrow (p_5 \leftrightarrow p_6) \end{array}$$

$$S = \left\{ \begin{array}{l} p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow n))) \\ n \leftrightarrow (p_5 \leftrightarrow p_6) \end{array} \right\}$$

A space-efficient CNF transformation

Using so-called *naming* or *definition introduction*

1. Take a non-literal subformula A of formula F
2. Introduce a new *name* n for it, i.e., a fresh propositional variable
3. Add a *definition for* n, i.e., a formula stating that n is equivalent to A

$$F \quad = \quad p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (\overbrace{p_5 \leftrightarrow p_6}^{A}))))$$

$$n \leftrightarrow (p_5 \leftrightarrow p_6)$$

$$S = \left\{ \begin{array}{l} p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow n))) \\ n \leftrightarrow (p_5 \leftrightarrow p_6) \end{array} \right\}$$

A space-efficient CNF transformation

Using so-called *naming* or *definition introduction*

1. Take a non-literal subformula A of formula F
2. Introduce a new *name* n for it, i.e., a fresh propositional variable
3. Add a *definition for n*, i.e., a formula stating that n is equivalent to A

$$F \quad = \quad p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (\overbrace{p_5 \leftrightarrow p_6}^{A}))))$$

$$n \leftrightarrow (p_5 \leftrightarrow p_6)$$

$$S = \left\{ \begin{array}{l} p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow n))) \\ n \leftrightarrow (p_5 \leftrightarrow p_6) \end{array} \right\}$$

A space-efficient CNF transformation

Using so-called *naming* or *definition introduction*

1. Take a non-literal subformula A of formula F
2. Introduce a new *name* n for it, i.e., a fresh propositional variable
3. Add a *definition for* n, i.e., a formula stating that n is equivalent to A

$$F = p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (\overbrace{p_5 \leftrightarrow p_6}^{A}))))$$

$$n \leftrightarrow (p_5 \leftrightarrow p_6)$$

$$S = \left\{ \begin{array}{l} p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow n))) \\ n \leftrightarrow (p_5 \leftrightarrow p_6) \end{array} \right\}$$

A space-efficient CNF transformation

Using so-called *naming* or *definition introduction*

1. Take a non-literal subformula A of formula F
2. Introduce a new *name* n for it, i.e., a fresh propositional variable
3. Add a *definition for* n, i.e., a formula stating that n is equivalent to A

$$
\begin{aligned}
F \quad = \quad & p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (\overbrace{p_5 \leftrightarrow p_6}^{A})))) \\
& n \leftrightarrow (p_5 \leftrightarrow p_6)
\end{aligned}
$$

4. Replace A in F by its name n:

$$
S = \left\{ \begin{array}{l} p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow n))) \\ n \leftrightarrow (p_5 \leftrightarrow p_6) \end{array} \right\}
$$

A space-efficient CNF transformation

The new set S of formulas and the original formula F are not equivalent but they are *equisatisfiable*:

1. every model of S is a model of F and
2. every model of F can be extended to a model of S (by assigning to n the value of $p_5 \leftrightarrow p_6$)

$$F \quad = \quad p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (\overbrace{p_5 \leftrightarrow p_6}^{A}))))$$
$$n \leftrightarrow (p_5 \leftrightarrow p_6)$$

$$S = \left\{ \begin{array}{l} p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow n))) \\ n \leftrightarrow (p_5 \leftrightarrow p_6) \end{array} \right\}$$

A space-efficient CNF transformation

The new set S of formulas and the original formula F are not equivalent but they are *equisatisfiable*:

1. every model of S is a model of F and
2. every model of F can be extended to a model of S (by assigning to n the value of $p_5 \leftrightarrow p_6$)

$$F = p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (\overbrace{p_5 \leftrightarrow p_6}^{A}))))$$

$$n \leftrightarrow (p_5 \leftrightarrow p_6)$$

$$S = \left\{ \begin{array}{l} p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow n))) \\ n \leftrightarrow (p_5 \leftrightarrow p_6) \end{array} \right\}$$

After several steps

$$p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6))))$$

$$\begin{aligned} p_1 &\leftrightarrow (p_2 \leftrightarrow n_3) \\ n_3 &\leftrightarrow (p_3 \leftrightarrow n_4) \\ n_4 &\leftrightarrow (p_4 \leftrightarrow n_5) \\ n_5 &\leftrightarrow (p_5 \leftrightarrow p_6) \end{aligned}$$

The conversion of the original formula to CNF introduces 32 copies of p_6

The conversion of the new set of formulas to CNF introduces 4 copies of p_6

After several steps

$$p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6))))$$

$$\begin{array}{l} p_1 \leftrightarrow (p_2 \leftrightarrow n_3) \\ n_3 \leftrightarrow (p_3 \leftrightarrow n_4) \\ n_4 \leftrightarrow (p_4 \leftrightarrow n_5) \\ n_5 \leftrightarrow (p_5 \leftrightarrow p_6) \end{array}$$

The conversion of the original formula to CNF introduces 32 copies of p_6

The conversion of the new set of formulas to CNF introduces 4 copies of p_6

After several steps

$$p_1 \leftrightarrow (p_2 \leftrightarrow (p_3 \leftrightarrow (p_4 \leftrightarrow (p_5 \leftrightarrow p_6))))$$

$$\begin{gathered} p_1 \leftrightarrow (p_2 \leftrightarrow n_3) \\ n_3 \leftrightarrow (p_3 \leftrightarrow n_4) \\ n_4 \leftrightarrow (p_4 \leftrightarrow n_5) \\ n_5 \leftrightarrow (p_5 \leftrightarrow p_6) \end{gathered}$$

The conversion of the original formula to CNF introduces 32 copies of p_6

The conversion of the new set of formulas to CNF introduces 4 copies of p_6

Clausal Form

Clausal form of a formula A: a set S_A of clauses which is satisfiable iff A is satisfiable

Clausal form of a set S of formulas: a set S' of clauses which is satisfiable iff so is S

In fact, we can require something stronger:

1. A and S_A have the same models in the language of A
2. S and S' have the same models in the language of S

Big advantage of clausal normal form over CNF:
we can convert any formula to a set of clauses in almost linear time

Clausal Form

Clausal form of a formula A: a set S_A of clauses which is satisfiable iff A is satisfiable

Clausal form of a set S *of formulas:* a set S' of clauses which is satisfiable iff so is S

In fact, we can require something stronger:

1. A and S_A have the same models in the language of A
2. S and S' have the same models in the language of S

Big advantage of clausal normal form over CNF:
we can convert any formula to a set of clauses in almost linear time

Clausal Form

Clausal form of a formula A: a set S_A of clauses which is satisfiable iff A is satisfiable

Clausal form of a set S *of formulas:* a set S' of clauses which is satisfiable iff so is S

In fact, we can require something stronger:

1. A and S_A have the same models in the language of A
2. S and S' have the same models in the language of S

Big advantage of clausal normal form over CNF:
we can convert any formula to a set of clauses in almost linear time

Clausal Form

Clausal form of a formula A: a set S_A of clauses which is satisfiable iff A is satisfiable

Clausal form of a set S *of formulas:* a set S' of clauses which is satisfiable iff so is S

In fact, we can require something stronger:

1. A and S_A have the same models in the language of A
2. S and S' have the same models in the language of S

Big advantage of clausal normal form over CNF:
we can convert any formula to a set of clauses in almost linear time

Definitional Clause Form Transformation

How to convert a formula A into a set S of clauses that is clausal normal form of A:

1. If A has the form $C_1 \wedge \cdots \wedge C_n$, where $n \geq 1$ and each C_i is a clause, then $S \stackrel{\text{def}}{=} \{ C_1, \ldots, C_n \}$.
2. Otherwise, introduce a name for each subformula B of A such that B is not a literal and use this name instead of the formula.

Converting a formula to clausal form, Example

	subformula	definition	clauses
	$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$		n_1
n_1	$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$	$n_1 \leftrightarrow \neg n_2$	$\neg n_1 \lor \neg n_2$ $n_1 \lor n_2$
n_2	$(p \to q) \land (p \land q \to r) \to (p \to r)$	$n_2 \leftrightarrow (n_3 \to n_7)$	$\neg n_2 \lor \neg n_3 \lor n_7$ $n_3 \lor n_2$ $\neg n_7 \lor n_2$
n_3	$(p \to q) \land (p \land q \to r)$	$n_3 \leftrightarrow (n_4 \land n_5)$	$\neg n_3 \lor n_4$ $\neg n_3 \lor n_5$ $\neg n_4 \lor \neg n_5 \lor n_3$
n_4	$p \to q$	$n_4 \leftrightarrow (p \to q)$	$\neg n_4 \lor \neg p \lor q$ $p \lor n_4$ $\neg q \lor n_4$
n_5	$p \land q \to r$	$n_5 \leftrightarrow (n_6 \to r)$	$\neg n_5 \lor \neg n_6 \lor r$ $n_6 \lor n_5$ $\neg r \lor n_5$
n_6	$p \land q$	$n_6 \leftrightarrow (p \land q)$	$\neg n_6 \lor p$ $\neg n_6 \lor q$ $\neg p \lor \neg q \lor n_6$
n_7	$p \to r$	$n_7 \leftrightarrow (p \to r)$	$\neg n_7 \lor \neg p \lor r$ $p \lor n_7$ $\neg r \lor n_7$

Converting a formula to clausal form, Example

	subformula	definition	clauses
	$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$		n_1
n_1	$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$	$n_1 \leftrightarrow \neg n_2$	$\neg n_1 \lor \neg n_2$ $n_1 \lor n_2$
n_2	$(p \to q) \land (p \land q \to r) \to (p \to r)$	$n_2 \leftrightarrow (n_3 \to n_7)$	$\neg n_2 \lor \neg n_3 \lor n_7$ $n_3 \lor n_2$ $\neg n_7 \lor n_2$
n_3	$(p \to q) \land (p \land q \to r)$	$n_3 \leftrightarrow (n_4 \land n_5)$	$\neg n_3 \lor n_4$ $\neg n_3 \lor n_5$ $\neg n_4 \lor \neg n_5 \lor n_3$
n_4	$p \to q$	$n_4 \leftrightarrow (p \to q)$	$\neg n_4 \lor \neg p \lor q$ $p \lor n_4$ $\neg q \lor n_4$
n_5	$p \land q \to r$	$n_5 \leftrightarrow (n_6 \to r)$	$\neg n_5 \lor \neg n_6 \lor r$ $n_6 \lor n_5$ $\neg r \lor n_5$
n_6	$p \land q$	$n_6 \leftrightarrow (p \land q)$	$\neg n_6 \lor p$ $\neg n_6 \lor q$ $\neg p \lor \neg q \lor n_6$
n_7	$p \to r$	$n_7 \leftrightarrow (p \to r)$	$\neg n_7 \lor \neg p \lor r$ $p \lor n_7$ $\neg r \lor n_7$

Consider all subformulas that are not literals

Converting a formula to clausal form, Example

	subformula	definition	clauses
	$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$		n_1
n_1	$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$	$n_1 \leftrightarrow \neg n_2$	$\neg n_1 \lor \neg n_2$ $n_1 \lor n_2$
n_2	$(p \to q) \land (p \land q \to r) \to (p \to r)$	$n_2 \leftrightarrow (n_3 \to n_7)$	$\neg n_2 \lor \neg n_3 \lor n_7$ $n_3 \lor n_2$ $\neg n_7 \lor n_2$
n_3	$(p \to q) \land (p \land q \to r)$	$n_3 \leftrightarrow (n_4 \land n_5)$	$\neg n_3 \lor n_4$ $\neg n_3 \lor n_5$ $\neg n_4 \lor \neg n_5 \lor n_3$
n_4	$p \to q$	$n_4 \leftrightarrow (p \to q)$	$\neg n_4 \lor \neg p \lor q$ $p \lor n_4$ $\neg q \lor n_4$
n_5	$p \land q \to r$	$n_5 \leftrightarrow (n_6 \to r)$	$\neg n_5 \lor \neg n_6 \lor r$ $n_6 \lor n_5$ $\neg r \lor n_5$
n_6	$p \land q$	$n_6 \leftrightarrow (p \land q)$	$\neg n_6 \lor p$ $\neg n_6 \lor q$ $\neg p \lor \neg q \lor n_6$
n_7	$p \to r$	$n_7 \leftrightarrow (p \to r)$	$\neg n_7 \lor \neg p \lor r$ $p \lor n_7$ $\neg r \lor n_7$

Introduce names for these formulas

Converting a formula to clausal form, Example

	subformula	definition	clauses
	$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$		n_1
n_1	$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$	$n_1 \leftrightarrow \neg n_2$	$\neg n_1 \lor \neg n_2$ $n_1 \lor n_2$
n_2	$(p \to q) \land (p \land q \to r) \to (p \to r)$	$n_2 \leftrightarrow (n_3 \to n_7)$	$\neg n_2 \lor \neg n_3 \lor n_7$ $n_3 \lor n_2$ $\neg n_7 \lor n_2$
n_3	$(p \to q) \land (p \land q \to r)$	$n_3 \leftrightarrow (n_4 \land n_5)$	$\neg n_3 \lor n_4$ $\neg n_3 \lor n_5$ $\neg n_4 \lor \neg n_5 \lor n_3$
n_4	$p \to q$	$n_4 \leftrightarrow (p \to q)$	$\neg n_4 \lor \neg p \lor q$ $p \lor n_4$ $\neg q \lor n_4$
n_5	$p \land q \to r$	$n_5 \leftrightarrow (n_6 \to r)$	$\neg n_5 \lor \neg n_6 \lor r$ $n_6 \lor n_5$ $\neg r \lor n_5$
n_6	$p \land q$	$n_6 \leftrightarrow (p \land q)$	$\neg n_6 \lor p$ $\neg n_6 \lor q$ $\neg p \lor \neg q \lor n_6$
n_7	$p \to r$	$n_7 \leftrightarrow (p \to r)$	$\neg n_7 \lor \neg p \lor r$ $p \lor n_7$ $\neg r \lor n_7$

Introduce definitions

Converting a formula to clausal form, Example

	subformula	definition	clauses
	$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$		n_1
n_1	$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$	$n_1 \leftrightarrow \neg n_2$	$\neg n_1 \lor \neg n_2$ $n_1 \lor n_2$
n_2	$(p \to q) \land (p \land q \to r) \to (p \to r)$	$n_2 \leftrightarrow (n_3 \to n_7)$	$\neg n_2 \lor \neg n_3 \lor n_7$ $n_3 \lor n_2$ $\neg n_7 \lor n_2$
n_3	$(p \to q) \land (p \land q \to r)$	$n_3 \leftrightarrow (n_4 \land n_5)$	$\neg n_3 \lor n_4$ $\neg n_3 \lor n_5$ $\neg n_4 \lor \neg n_5 \lor n_3$
n_4	$p \to q$	$n_4 \leftrightarrow (p \to q)$	$\neg n_4 \lor \neg p \lor q$ $p \lor n_4$ $\neg q \lor n_4$
n_5	$p \land q \to r$	$n_5 \leftrightarrow (n_6 \to r)$	$\neg n_5 \lor \neg n_6 \lor r$ $n_6 \lor n_5$ $\neg r \lor n_5$
n_6	$p \land q$	$n_6 \leftrightarrow (p \land q)$	$\neg n_6 \lor p$ $\neg n_6 \lor q$ $\neg p \lor \neg q \lor n_6$
n_7	$p \to r$	$n_7 \leftrightarrow (p \to r)$	$\neg n_7 \lor \neg p \lor r$ $p \lor n_7$ $\neg r \lor n_7$

Convert the definition formulas to CNF in the standard way

Optimized Definitional Clause Form Transformation

If

- we introduce a name for a subformula and
- the occurence of the subformula is positive or negative (not 0)

then an implication can be used instead of equivalence

See Chapter 7 of LRCS for a precise description

Optimized Definitional Clause Form Transformation

If

- we introduce a name for a subformula and
- the occurence of the subformula is positive or negative (not 0)

then an implication can be used instead of equivalence

See Chapter 7 of LRCS for a precise description

Example

	subformula	definition	clauses
	$\neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r))$		n_1
n_1	$\neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r))$	$n_1 \to \neg n_2$	$\neg n_1 \vee \neg n_2$ $n_1 \vee n_2$
n_2	$(p \to q) \wedge (p \wedge q \to r) \to (p \to r)$	$(n_3 \to n_7) \to n_2$	$\neg n_2 \vee \neg n_3 \vee n_7$ $n_3 \vee n_2$ $\neg n_7 \vee n_2$
n_3	$(p \to q) \wedge (p \wedge q \to r)$	$n_3 \to (n_4 \wedge n_5)$	$\neg n_3 \vee n_4$ $\neg n_3 \vee n_5$ $\neg n_4 \vee \neg n_5 \vee n_3$
n_4	$p \to q$	$n_4 \to (p \to q)$	$\neg n_4 \vee \neg p \vee q$ $p \vee n_4$ $\neg q \vee n_4$
n_5	$p \wedge q \to r$	$n_5 \to (n_6 \to r)$	$\neg n_5 \vee \neg n_6 \vee r$ $n_6 \vee n_5$ $\neg r \vee n_5$
n_6	$p \wedge q$	$n_6 \to (p \wedge q)$	$\neg n_6 \vee p$ $\neg n_6 \vee q$ $\neg p \vee \neg q \vee n_6$
n_7	$p \to r$	$(p \to r) \to n_7$	$\neg n_7 \vee \neg p \vee r$ $p \vee n_7$ $\neg r \vee n_7$

Example

	subformula	definition	clauses
	$\neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r))$		n_1
n_1	$\neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r))$	$n_1 \to \neg n_2$	$\neg n_1 \vee \neg n_2$ $n_1 \vee n_2$
n_2	$(p \to q) \wedge (p \wedge q \to r) \to (p \to r)$	$(n_3 \to n_7) \to n_2$	$\neg n_2 \vee \neg n_3 \vee n_7$ $n_3 \vee n_2$ $\neg n_7 \vee n_2$
n_3	$(p \to q) \wedge (p \wedge q \to r)$	$n_3 \to (n_4 \wedge n_5)$	$\neg n_3 \vee n_4$ $\neg n_3 \vee n_5$ $\neg n_4 \vee \neg n_5 \vee n_3$
n_4	$p \to q$	$n_4 \to (p \to q)$	$\neg n_4 \vee \neg p \vee q$ $p \vee n_4$ $\neg q \vee n_4$
n_5	$p \wedge q \to r$	$n_5 \to (n_6 \to r)$	$\neg n_5 \vee \neg n_6 \vee r$ $n_6 \vee n_5$ $\neg r \vee n_5$
n_6	$p \wedge q$	$n_6 \to (p \wedge q)$	$\neg n_6 \vee p$ $\neg n_6 \vee q$ $\neg p \vee \neg q \vee n_6$
n_7	$p \to r$	$(p \to r) \to n_7$	$\neg n_7 \vee \neg p \vee r$ $p \vee n_7$ $\neg r \vee n_7$

The clauses in red are omitted by optimized transformation

Example

	subformula	definition	clauses
	$\neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r))$		n_1
n_1	$\neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r))$	$n_1 \to \neg n_2$	$\neg n_1 \vee \neg n_2$
n_2	$(p \to q) \wedge (p \wedge q \to r) \to (p \to r)$	$(n_3 \to n_7) \to n_2$	$n_3 \vee n_2$ $\neg n_7 \vee n_2$
n_3	$(p \to q) \wedge (p \wedge q \to r)$	$n_3 \to (n_4 \wedge n_5)$	$\neg n_3 \vee n_4$ $\neg n_3 \vee n_5$
n_4	$p \to q$	$n_4 \to (p \to q)$	$\neg n_4 \vee \neg p \vee q$
n_5	$p \wedge q \to r$	$n_5 \to (n_6 \to r)$	$\neg n_5 \vee \neg n_6 \vee r$
n_6	$p \wedge q$	$n_6 \to (p \wedge q)$	$\neg p \vee \neg q \vee n_6$
n_7	$p \to r$	$(p \to r) \to n_7$	$p \vee n_7$ $\neg r \vee n_7$

The result is fewer clauses

Satisfiability-checking for sets of clauses

The CNF transformation of

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

gives the set of four clauses:

$$\left\{\begin{array}{l} \neg p \lor q, \\ \neg p \lor \neg q \lor r, \\ p, \\ \neg r, \end{array}\right\}$$

To satisfy all these clauses we must assign 1 to p and 0 to r, so we do not have to guess values for them

In this case, we can do even better and establish unsatisfiability with no guessing

Satisfiability-checking for sets of clauses

The CNF transformation of

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

gives the set of four clauses:

$$\left\{ \begin{array}{l} \neg p \lor q, \\ \neg p \lor \neg q \lor r, \\ p, \\ \neg r, \end{array} \right\}$$

To satisfy all these clauses we must assign 1 to p and 0 to r, so we do not have to guess values for them

In this case, we can do even better and establish unsatisfiability with no guessing

Satisfiability-checking for sets of clauses

The CNF transformation of

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

gives the set of four clauses:

$$\left\{ \begin{array}{l} \neg p \lor q, \\ \neg p \lor \neg q \lor r, \\ p, \\ \neg r, \end{array} \right\}$$

To satisfy all these clauses we must assign 1 to p and 0 to r, so we do not have to guess values for them

In this case, we can do even better and establish unsatisfiability with no guessing

Smart search for satisfiability

$$\{p \mapsto 1, r \mapsto 0, q \mapsto 1\}$$

$$\left\{ \begin{array}{l} \neg p \lor q, \\ \neg p \lor \neg q \lor r, \\ p, \\ \neg r \end{array} \right\}$$

This set of clauses is unsatisfiable

Smart search for satisfiability

$$\{p \mapsto 1, r \mapsto 0, q \mapsto 1\}$$

$$\left\{ \begin{array}{l} \neg p \vee q, \\ \neg p \vee \neg q \vee r, \\ p, \\ \neg r \end{array} \right\}$$

This set of clauses is unsatisfiable

Smart search for satisfiability

$$\{p \mapsto 1, r \mapsto 0, q \mapsto 1\}$$

$$\left\{ \begin{array}{l} \neg p \vee q, \\ \neg p \vee \neg q \vee r, \\ p, \\ \neg r \end{array} \right\}$$

This set of clauses is unsatisfiable

Smart search for satisfiability

$$\{p \mapsto 1, r \mapsto 0, q \mapsto 1\}$$

$$\left\{ \begin{array}{l} \neg p \vee q, \\ \neg p \vee \neg q \vee r, \\ p, \\ \neg r \end{array} \right\}$$

This set of clauses is unsatisfiable

Smart search for satisfiability

$$\{p \mapsto 1, r \mapsto 0, q \mapsto 1\}$$

$$\left\{ \begin{array}{l} \neg p \vee q, \\ \neg p \vee \neg q \vee r, \\ p, \\ \neg r \end{array} \right\}$$

This set of clauses is unsatisfiable

Smart search for satisfiability

$$\{p \mapsto 1, r \mapsto 0, q \mapsto 1\}$$

$$\left\{ \begin{array}{l} \neg p \lor q, \\ \neg p \lor \neg q \lor r, \\ p, \\ \neg r \end{array} \right\}$$

This set of clauses is unsatisfiable

Smart search for satisfiability

$$\{p \mapsto 1, r \mapsto 0, q \mapsto 1\}$$

$$\left\{ \begin{array}{l} \neg p \vee q, \\ \neg p \vee \neg q \vee r, \\ p, \\ \neg r \end{array} \right\}$$

This set of clauses is unsatisfiable

Smart search for satisfiability

$$\{p \mapsto 1, r \mapsto 0, q \mapsto 1\}$$

$$\left\{ \begin{array}{l} \neg p \vee q, \\ \neg p \vee \neg q \vee r, \\ p, \\ \neg r \end{array} \right\}$$

This set of clauses is unsatisfiable

Smart search for satisfiability

$$\{p \mapsto 1, r \mapsto 0, q \mapsto 1\}$$

$$\left\{ \begin{array}{l} \neg p \vee q, \\ \neg p \vee \neg q \\ p, \\ \neg r \end{array} \right\}$$

This set of clauses is unsatisfiable

Smart search for satisfiability

$$\{p \mapsto 1, r \mapsto 0, q \mapsto 1\}$$

$$\left\{ \begin{array}{l} \neg p \vee q, \\ \neg p \vee \neg q \\ p, \\ \neg r \end{array} \right\}$$

This set of clauses is unsatisfiable

Smart search for satisfiability

$$\{p \mapsto 1, r \mapsto 0, q \mapsto 1\}$$

$$\left\{ \begin{array}{l} \neg p \vee q, \\ \neg p \vee \neg q \\ p, \\ \neg r \end{array} \right\}$$

This set of clauses is unsatisfiable

Smart search for satisfiability

$$\{p \mapsto 1, r \mapsto 0, q \mapsto 1\}$$

$$\left\{ \begin{array}{ll} \neg p \vee & q, \\ \neg p \vee & \neg q \\ p, & \\ \neg r & \end{array} \right\}$$

This set of clauses is unsatisfiable

Smart search for satisfiability

$$\{p \mapsto 1, r \mapsto 0, q \mapsto 1\}$$

$$\left\{ \begin{array}{l} \neg p \vee q, \\ \neg p \vee \neg q \; \square \\ p, \\ \neg r \end{array} \right\}$$

This set of clauses is unsatisfiable

Unit propagation

Let S be a set of clauses.

Unit propagation. Repeatedly apply the following transformation:

if S contains a unit clause, i.e. a clause consisting of one literal L, then

1. remove from S every clause of the form $L \vee C'$
2. replace in S every clause of the form $\overline{L} \vee C'$ by the clause C'

Unit Propagation, Example

$$\left\{\begin{array}{ll}
n_1, & \neg q \lor n_4, \\
\neg n_1 \lor \neg n_2, & \neg n_5 \lor \neg n_6 \lor r, \\
n_1 \lor n_2, & n_6 \lor n_5, \\
\neg n_2 \lor \neg n_3 \lor n_7, & \neg r \lor n_5, \\
n_3 \lor n_2, & \neg n_6 \lor p, \\
\neg n_7 \lor n_2, & \neg n_6 \lor q, \\
\neg n_3 \lor n_4, & \neg p \lor, \neg q \lor n_6, \\
\neg n_3 \lor n_5, & \neg n_7 \lor \neg p \lor r, \\
\neg n_4 \lor \neg n_5 \lor n_3, & p \lor n_7, \\
\neg n_4 \lor \neg p \lor q, & \neg r \lor n_7 \\
p \lor n_4, &
\end{array}\right\}$$

Unit Propagation, Example

$$\left\{\begin{array}{ll} n_1, & \neg q \vee n_4, \\ \neg n_1 \vee \neg n_2, & \neg n_5 \vee \neg n_6 \vee r, \\ n_1 \vee n_2, & n_6 \vee n_5, \\ \neg n_2 \vee \neg n_3 \vee n_7, & \neg r \vee n_5, \\ n_3 \vee n_2, & \neg n_6 \vee p, \\ \neg n_7 \vee n_2, & \neg n_6 \vee q, \\ \neg n_3 \vee n_4, & \neg p \vee, \neg q \vee n_6, \\ \neg n_3 \vee n_5, & \neg n_7 \vee \neg p \vee r, \\ \neg n_4 \vee \neg n_5 \vee n_3, & p \vee n_7, \\ \neg n_4 \vee \neg p \vee q, & \neg r \vee n_7 \\ p \vee n_4, & \end{array}\right\}$$

Unit Propagation, Example

$$\left\{ \begin{array}{ll} & \neg q \vee n_4, \\ \neg n_2, & \neg n_5 \vee \neg n_6 \vee r, \\ & n_6 \vee n_5, \\ \neg n_2 \vee \neg n_3 \vee n_7, & \neg r \vee n_5, \\ n_3 \vee n_2, & \neg n_6 \vee p, \\ \neg n_7 \vee n_2, & \neg n_6 \vee q, \\ \neg n_3 \vee n_4, & \neg p \vee, \neg q \vee n_6, \\ \neg n_3 \vee n_5, & \neg n_7 \vee \neg p \vee r, \\ \neg n_4 \vee \neg n_5 \vee n_3, & p \vee n_7, \\ \neg n_4 \vee \neg p \vee q, & \neg r \vee n_7 \\ p \vee n_4, & \end{array} \right\}$$

Unit Propagation, Example

$$\left\{\begin{array}{ll} & \neg q \vee n_4, \\ \neg n_2, & \neg n_5 \vee \neg n_6 \vee r, \\ & n_6 \vee n_5, \\ \neg n_2 \vee \neg n_3 \vee n_7, & \neg r \vee n_5, \\ n_3 \vee n_2, & \neg n_6 \vee p, \\ \neg n_7 \vee n_2, & \neg n_6 \vee q, \\ \neg n_3 \vee n_4, & \neg p \vee, \neg q \vee n_6, \\ \neg n_3 \vee n_5, & \neg n_7 \vee \neg p \vee r, \\ \neg n_4 \vee \neg n_5 \vee n_3, & p \vee n_7, \\ \neg n_4 \vee \neg p \vee q, & \neg r \vee n_7 \\ p \vee n_4, & \end{array}\right\}$$

Unit Propagation, Example

$$
\left\{
\begin{array}{ll}
 & \neg q \vee n_4, \\
 & \neg n_5 \vee \neg n_6 \vee r, \\
 & n_6 \vee n_5, \\
 & \neg r \vee n_5, \\
n_3 \qquad , & \neg n_6 \vee p, \\
\neg n_7 \qquad , & \neg n_6 \vee q, \\
\neg n_3 \vee n_4, & \neg p \vee, \neg q \vee n_6, \\
\neg n_3 \vee n_5, & \neg n_7 \vee \neg p \vee r, \\
\neg n_4 \vee \neg n_5 \vee n_3, & p \vee n_7, \\
\neg n_4 \vee \neg p \vee q, & \neg r \vee n_7 \\
p \vee n_4, &
\end{array}
\right\}
$$

Unit Propagation, Example

$$\left\{\begin{array}{ll} & \neg q \lor n_4, \\ & \neg n_5 \lor \neg n_6 \lor r, \\ & n_6 \lor n_5, \\ & \neg r \lor n_5, \\ n_3 \qquad , & \neg n_6 \lor p, \\ \neg n_7 \qquad , & \neg n_6 \lor q, \\ \neg n_3 \lor n_4, & \neg p \lor, \neg q \lor n_6, \\ \neg n_3 \lor n_5, & \neg n_7 \lor \neg p \lor r, \\ \neg n_4 \lor \neg n_5 \lor n_3, & p \lor n_7, \\ \neg n_4 \lor \neg p \lor q, & \neg r \lor n_7 \\ p \lor n_4, & \end{array}\right\}$$

Unit Propagation, Example

$$\left\{\begin{array}{ll} & \neg q \vee n_4, \\ & \neg n_5 \vee \neg n_6 \vee r, \\ & n_6 \vee n_5, \\ & \neg r \vee n_5, \\ & \neg n_6 \vee p, \\ & \neg n_6 \vee q, \\ \quad n_4, & \neg p \vee, \neg q \vee n_6, \\ \quad n_5, & \\ & p \qquad , \\ \neg n_4 \vee \neg p \vee q, & \neg r \\ p \vee n_4, & \end{array}\right\}$$

Unit Propagation, Example

$$
\left\{
\begin{array}{ll}
 & \neg q \vee n_4, \\
 & \neg n_5 \vee \neg n_6 \vee r, \\
 & n_6 \vee n_5, \\
 & \neg r \vee n_5, \\
 & \neg n_6 \vee p, \\
 & \neg n_6 \vee q, \\
n_4, & \neg p \vee, \neg q \vee n_6, \\
n_5, & \\
 & p \quad , \\
\neg n_4 \vee \neg p \vee q, & \neg r \\
p \vee n_4, &
\end{array}
\right\}
$$

Unit Propagation, Example

$$\left\{ \begin{array}{r} \neg n_6 \quad , \\ \\ \\ \\ \neg n_6 \lor q, \\ \neg q \lor n_6, \\ \\ \\ q, \end{array} \right\}$$

Unit Propagation, Example

$$\left\{ \begin{array}{c} \neg n_6 \quad , \\ \\ \\ \neg n_6 \lor q, \\ \neg q \lor n_6, \\ \\ q, \end{array} \right\}$$

Unit Propagation, Example

$$\left\{ \quad \square \quad \right\}$$

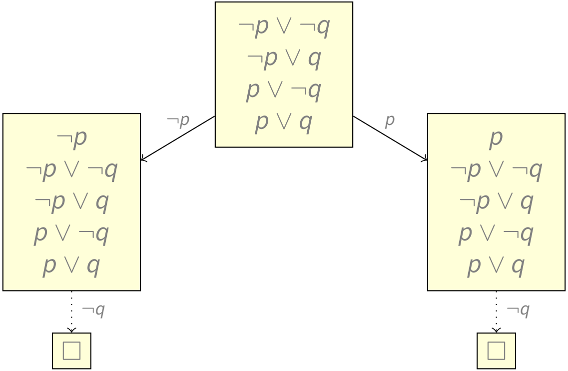
We established unsatisfiability of this set of clauses in a completely deterministic way, by unit propagation.

DPLL = splitting + unit propagation

procedure *DPLL*(S)
input: set of clauses S
output: *satisfiable* or *unsatisfiable*
parameters: function *select_literal*
begin
 $S := propagate(S)$
 if S is empty **then return** *satisfiable*
 if S contains $\square$ **then return** *unsatisfiable*
 $L := select\_literal(S)$
 if $DPLL(S \cup \{ L \}) = satisfiable$
  then return *satisfiable*
  else return $DPLL(S \cup \{ \overline{L} \})$
end

DPLL, Example 1

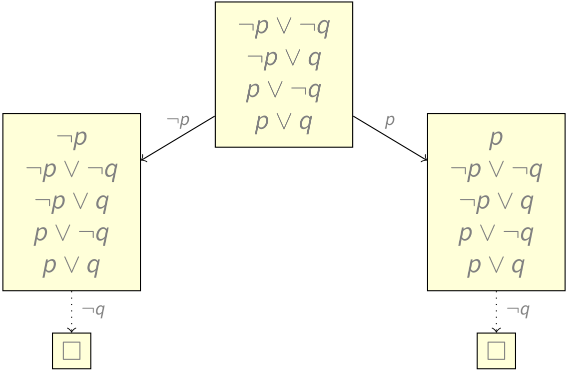
Can be illustrated using DPLL trees (similar to splitting trees)



Since all branches end up in a set contaning the empty clause, the initial set of clauses is unsatisfiable.

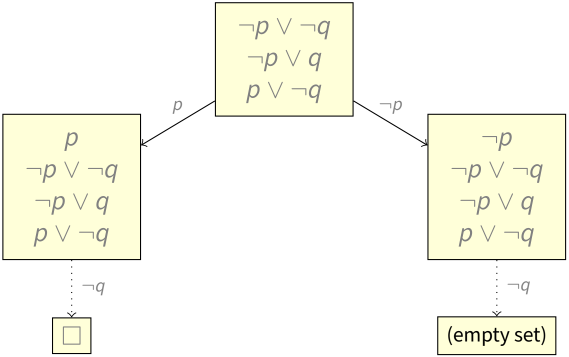
DPLL, Example 1

Can be illustrated using DPLL trees (similar to splitting trees)



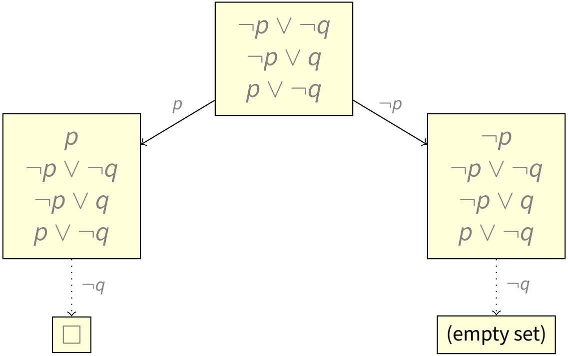
Since all branches end up in a set contaning the empty clause, the initial set of clauses is unsatisfiable.

DPLL, Example 2



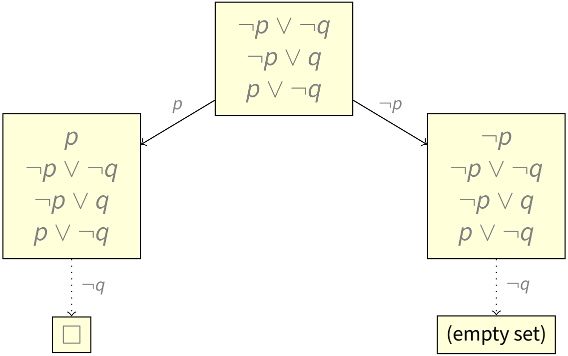
The set of clauses is satisfiable

DPLL, Example 2



A model is described by all selected literals and unit-propagated literals on the branch ending in the empty set

DPLL, Example 2



This DPLL tree gives us the model $\{ p \mapsto 0, q \mapsto 0 \}$

Two optimizations

1. Any clause of the form $p \vee \neg p \vee C$ is a tautology

   Tautologies can be removed from a set without affecting its satisfiability

2. A literal L in S is called pure if S contains no clauses of the form $\overline{L} \vee C$

   All clauses containing a pure literal can be satisfied by making that literal true

   Hence, clauses containing pure literals can be removed, too

Two optimizations

1. Any clause of the form $p \vee \neg p \vee C$ is a tautology

   Tautologies can be removed from a set without affecting its satisfiability

2. A literal L in S is called pure if S contains no clauses of the form $\overline{L} \vee C$

   All clauses containing a pure literal can be satisfied by making that literal true

   Hence, clauses containing pure literals can be removed, too

Two optimizations

1. Any clause of the form $p \vee \neg p \vee C$ is a tautology

   Tautologies can be removed from a set without affecting its satisfiability

2. A literal L in S is called pure if S contains no clauses of the form $\overline{L} \vee C$

   All clauses containing a pure literal can be satisfied by making that literal true

   Hence, clauses containing pure literals can be removed, too

Pure literals: example

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

Pure literals: example

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

Literal $\neg p_3$ is pure in this clause set: we can remove all clauses containing it (by assigning 0 to p_3)

Pure literals: example

$$p_1 \lor \neg p_2$$

$$p_1 \lor p_2$$

Literal $\neg p_3$ is pure in this clause set: we can remove all clauses containing it (by assigning 0 to p_3)

Pure literals: example

$$p_1 \lor \neg p_2$$

$$p_1 \lor p_2$$

Literal p_1 is pure in the resulting set: we can remove all clauses containing it (by assigning 1 to p_1)

Pure literals: example

Literal p_1 is pure in the resulting set: we can remove all clauses containing it (by assigning 1 to p_1)

We obtain the empty set of clauses

Pure literals: example

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

Since r remained unconstrained, this gives us two models:

$$\{ p_1 \mapsto 1, p_2 \mapsto 0, p_3 \mapsto 0 \}$$
$$\{ p_1 \mapsto 1, p_2 \mapsto 1, p_3 \mapsto 0 \}$$

Horn clauses

A clause is called Horn if it contains at most one positive literal

Examples

Horn:

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

Non-Horn:

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

Horn clauses

A clause is called Horn if it contains at most one positive literal

Examples

Horn:

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

Non-Horn:

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

Satisfiability of Horn clauses

Can be decided by unit propagation

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

Satisfiability of Horn clauses

Can be decided by unit propagation

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

Satisfiability of Horn clauses

Can be decided by unit propagation

$$p_2$$
$$\neg p_2 \vee p_3$$
$$\neg p_3 \vee \neg p_4$$

Satisfiability of Horn clauses

Can be decided by unit propagation

$$p_3$$

$$\neg p_3 \vee \neg p_4$$

Satisfiability of Horn clauses

Can be decided by unit propagation

$$\neg p_4$$

Satisfiability of Horn clauses

Can be decided by unit propagation

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

Model: $\{ p_1 \mapsto 1, p_2 \mapsto 1, p_3 \mapsto 1, p_4 \mapsto 0 \}$

Satisfiability of Horn clauses

Can be decided by unit propagation

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

Model: $\{ p_1 \mapsto 1, p_2 \mapsto 1, p_3 \mapsto 1, p_4 \mapsto 0 \}$

Note: deleting a literal from a Horn clause gives a Horn clause.

Satisfiability of Horn clauses

Can be decided by unit propagation

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

Model: $\{ p_1 \mapsto 1, p_2 \mapsto 1, p_3 \mapsto 1, p_4 \mapsto 0 \}$

Note: deleting a literal from a Horn clause gives a Horn clause.
Therefore, unit propagation applied to a set C of Horn clauses gives a set C' of Horn clauses.

Satisfiability of Horn clauses

Can be decided by unit propagation

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

Model: $\{ p_1 \mapsto 1, p_2 \mapsto 1, p_3 \mapsto 1, p_4 \mapsto 0 \}$

Note: deleting a literal from a Horn clause gives a Horn clause.
Therefore, unit propagation applied to a set C of Horn clauses gives a set C' of Horn clauses.

Two cases:

1. C' contains $\square$. Then, C' (and hence C) is unsatisfiable.
2. C' does not contain $\square$.

Satisfiability of Horn clauses

Can be decided by unit propagation

$$
\begin{gathered}
p_1 \\
\neg p_1 \vee p_2 \\
\neg p_1 \vee \neg p_2 \vee p_3 \\
\neg p_3 \vee \neg p_4
\end{gathered}
$$

Model: $\{ p_1 \mapsto 1, p_2 \mapsto 1, p_3 \mapsto 1, p_4 \mapsto 0 \}$

Note: deleting a literal from a Horn clause gives a Horn clause.
Therefore, unit propagation applied to a set C of Horn clauses gives a set C' of Horn clauses.

Two cases:

1. C' contains $\square$. Then, C' (and hence C) is unsatisfiable.
2. C' does not contain $\square$.
   - Each clause in C' has at least two literals.
   - Hence each clause in C' contains at least one negative literal;
   - Hence setting all variables in C' to 0 satisfies C'.

Satisfiability of Horn clauses

Can be decided by unit propagation

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

Model: $\{ p_1 \mapsto 1, p_2 \mapsto 1, p_3 \mapsto 1, p_4 \mapsto 0 \}$

Note: deleting a literal from a Horn clause gives a Horn clause.
Therefore, unit propagation applied to a set C of Horn clauses gives a set C' of Horn clauses.

Two cases:

1. C' contains $\square$. Then, C' (and hence C) is unsatisfiable.
2. C' does not contain $\square$. Then:
   - Each clause in C' has at least two literals.
   - Hence each clause in C' contains at least one negative literal;
   - Hence setting all variables in C' to 0 satisfies C'.

Satisfiability of Horn clauses

Can be decided by unit propagation

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

Model: $\{\, p_1 \mapsto 1, p_2 \mapsto 1, p_3 \mapsto 1, p_4 \mapsto 0 \,\}$

Note: deleting a literal from a Horn clause gives a Horn clause.
Therefore, unit propagation applied to a set C of Horn clauses gives a set C' of Horn clauses.

Two cases:

1. C' contains $\square$. Then, C' (and hence C) is unsatisfiable.
2. C' does not contain $\square$. Then:
   - Each clause in C' has at least two literals.
   - Hence each clause in C' contains at least one negative literal;
   - Hence setting all variables in C' to 0 satisfies C'.

Satisfiability of Horn clauses

Can be decided by unit propagation

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

Model: $\{ p_1 \mapsto 1, p_2 \mapsto 1, p_3 \mapsto 1, p_4 \mapsto 0 \}$

Note: deleting a literal from a Horn clause gives a Horn clause.
Therefore, unit propagation applied to a set C of Horn clauses gives a set C' of Horn clauses.

Two cases:

1. C' contains $\square$. Then, C' (and hence C) is unsatisfiable.
2. C' does not contain $\square$. Then:
   - Each clause in C' has at least two literals.
   - Hence each clause in C' contains at least one negative literal;
   - Hence setting all variables in C' to 0 satisfies C'.

Running a SAT solver

Very simple but efficient SAT solver: MiniSat, `http://minisat.se/`

Running a SAT solver

Very simple but efficient SAT solver: MiniSat, http://minisat.se/

$$
\begin{gathered}
p_1 \\
\neg p_1 \vee p_2 \\
\neg p_1 \vee \neg p_2 \vee p_3 \\
\neg p_2 \vee \neg p_3
\end{gathered}
$$

Running a SAT solver

Very simple but efficient SAT solver: MiniSat, `http://minisat.se/`

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

DIMACS input format:

```
p cnf 3 4
1 0
-1 2 0
-1 -2 3 0
-2 -3 0
```

Running a SAT solver

Very simple but efficient SAT solver: MiniSat, `http://minisat.se/`

$$
\begin{gathered}
p_1 \\
\neg p_1 \lor p_2 \\
\neg p_1 \lor \neg p_2 \lor p_3 \\
\neg p_2 \lor \neg p_3
\end{gathered}
$$

DIMACS input format:

```
p cnf 3 4
1 0
-1 2 0
-1 -2 3 0
-2 -3 0
```

3 variables, 4 clauses.

Running a SAT solver

Very simple but efficient SAT solver: MiniSat, `http://minisat.se/`

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

DIMACS input format:

```
p cnf 3 4
1 0
-1 2 0
-1 -2 3 0
-2 -3 0
```

3 variables, 4 clauses.

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

Expressing Properties "k out of n variables are true"

Suppose we have variables $v_1, \ldots, v_n$ and
we want to express that exactly k of them are true

We will write this property as a formula $T_k(v_1, \ldots, v_n)$

Such formulas are very useful for encoding various problems in SAT

Expressing Properties "k out of n variables are true"

Suppose we have variables $v_1, \ldots, v_n$ and
we want to express that exactly k of them are true

We will write this property as a formula $T_k(v_1, \ldots, v_n)$

Such formulas are very useful for encoding various problems in SAT

First, let us express some simple special cases:

$$\begin{aligned} T_0(v_1, \ldots, v_n) &\stackrel{\text{def}}{=} \neg v_1 \wedge \cdots \wedge \neg v_n \\ T_1(v_1, \ldots, v_n) &\stackrel{\text{def}}{=} (v_1 \vee \cdots \vee v_n) \wedge \bigwedge_{i<j} (\neg v_i \vee \neg v_j) \end{aligned}$$

Expressing Properties “k out of n variables are true”

Suppose we have variables $v_1, \ldots, v_n$ and
we want to express that exactly k of them are true

We will write this property as a formula $T_k(v_1, \ldots, v_n)$

Such formulas are very useful for encoding various problems in SAT

First, let us express some simple special cases:

$$T_0(v_1, \ldots, v_n) \stackrel{\text{def}}{=} \neg v_1 \wedge \cdots \wedge \neg v_n$$
$$T_1(v_1, \ldots, v_n) \stackrel{\text{def}}{=} (v_1 \vee \cdots \vee v_n) \wedge \bigwedge_{i<j}(\neg v_i \vee \neg v_j)$$

$$T_{n-1}(v_1, \ldots, v_n) \stackrel{\text{def}}{=} (\neg v_1 \vee \cdots \vee \neg v_n) \wedge \bigwedge_{i<j}(v_i \vee v_j)$$
$$T_n(v_1, \ldots, v_n) \stackrel{\text{def}}{=} v_1 \wedge \cdots \wedge v_n$$

Expressing Properties "k out of n variables are true"

To define T_k for $0 < k < n$, introduce two formulas:

- $T_{\leq k}(v_1, \ldots, v_n)$: at most k variables among $v_1, \ldots, v_n$ are true, where $k = 0 \ldots n-1$
- $T_{\geq k}(v_1, \ldots, v_n)$: at least k variables among $v_1, \ldots, v_n$ are true, where $k = 1 \ldots n$

Expressing Properties "k out of n variables are true"

To define T_k for $0 < k < n$, introduce two formulas:

$T_{\leq k}(v_1, \ldots, v_n)$: at most k variables among $v_1, \ldots, v_n$ are true, where $k = 0 \ldots n-1$

$T_{\geq k}(v_1, \ldots, v_n)$: at least k variables among $v_1, \ldots, v_n$ are true, where $k = 1 \ldots n$

Expressing Properties "k out of n variables are true"

To define T_k for $0 < k < n$, introduce two formulas:

$T_{\leq k}(v_1, \ldots, v_n)$: at most k variables among $v_1, \ldots, v_n$ are true, where $k = 0 \ldots n-1$

$T_{\geq k}(v_1, \ldots, v_n)$: at least k variables among $v_1, \ldots, v_n$ are true, where $k = 1 \ldots n$

$$T_{\leq k}(v_1, \ldots, v_n) \stackrel{\text{def}}{=} \bigwedge_{\substack{x_1, \ldots, x_{k+1} \in \{ v_1, \ldots, v_n \} \\ x_1, \ldots, x_{k+1} \text{ are distinct}}} \neg x_1 \lor \cdots \lor \neg x_{k+1}$$

Expressing Properties "k out of n variables are true"

To define T_k for $0 < k < n$, introduce two formulas:

$T_{\leq k}(v_1, \ldots, v_n)$: at most k variables among $v_1, \ldots, v_n$ are true, where $k = 0 \ldots n-1$

$T_{\geq k}(v_1, \ldots, v_n)$: at least k variables among $v_1, \ldots, v_n$ are true, where $k = 1 \ldots n$

$$T_{\leq k}(v_1, \ldots, v_n) \stackrel{\text{def}}{=} \bigwedge_{\substack{x_1, \ldots, x_{k+1} \in \{ v_1, \ldots, v_n \} \\ x_1, \ldots, x_{k+1} \text{ are distinct}}} \neg x_1 \vee \cdots \vee \neg x_{k+1}$$

$$T_{\geq k}(v_1, \ldots, v_n) \stackrel{\text{def}}{=} \bigwedge_{\substack{x_1, \ldots, x_{n-k+1} \in \{ v_1, \ldots, v_n \} \\ x_1, \ldots, x_{n-k+1} \text{ are distinct}}} x_1 \vee \cdots \vee x_{n-k+1}$$

Sudoku

4		8	7	9			3	
		9	8	2		5		
	2							
9		2			6		1	7
		6	5	8	7	9		
7	8		2			4		6
							4	
		5		4	8	2		
	9			7	2	3		5

Enter digits from 1 to 9 into the blank spaces

Every row must contain one of each digit

So must every column

as must every 3x3 square

This instance has exactly one solution

Sudoku

4		8	7	9			3	
		9	8	2		5		
	2							
9		2			6		1	7
		6	5	8	7	9		
7	8		2			4		6
							4	
		5		4	8	2		
	9			7	2	3		5

Enter digits from 1 to 9 into the blank spaces

Every row must contain one of each digit

So must every column

as must every 3x3 square

This instance has exactly one solution

Sudoku

4		8	7	9			3	
		9	8	2		5		
	2							
9		2			6		1	7
		6	5	8	7	9		
7	8		2			4		6
							4	
		5		4	8	2		
	9			7	2	3		5

Enter digits from 1 to 9 into the blank spaces

Every row must contain one of each digit

So must every column

as must every 3x3 square

This instance has exactly one solution

Sudoku

4		8	7	9			3	
		9	8	2		5		
	2							
9		2			6		1	7
		6	5	8	7	9		
7	8		2			4		6
							4	
		5		4	8	2		
	9			7	2	3		5

Enter digits from 1 to 9 into the blank spaces

Every row must contain one of each digit

So must every column

as must every 3x3 square

This instance has exactly one solution

Sudoku

4	1	8	7	9	5	6	3	2
6	3	9	8	2	1	5	7	4
5	2	7	3	6	4	1	8	9
9	5	2	4	3	6	8	1	7
1	4	6	5	8	7	9	2	3
7	8	3	2	1	9	4	5	6
2	6	1	9	5	3	7	4	8
3	7	5	6	4	8	2	9	1
8	9	4	1	7	2	3	6	5

Enter digits from 1 to 9 into the blank spaces

Every row must contain one of each digit

So must every column

as must every 3x3 square

This instance has exactly one solution

Sudoku

4	1	8	7	9	5	6	3	2
6	3	9	8	2	1	5	7	4
5	2	7	3	6	4	1	8	9
9	5	2	4	3	6	8	1	7
1	4	6	5	8	7	9	2	3
7	8	3	2	1	9	4	5	6
2	6	1	9	5	3	7	4	8
3	7	5	6	4	8	2	9	1
8	9	4	1	7	2	3	6	5

Enter digits from 1 to 9 into the blank spaces

Every row must contain one of each digit

So must every column

as must every 3x3 square

This instance has exactly one solution

Sudoku

4	1	8	7	9	5	6	3	2
6	3	9	8	2	1	5	7	4
5	2	7	3	6	4	1	8	9
9	5	2	4	3	6	8	1	7
1	4	6	5	8	7	9	2	3
7	8	3	2	1	9	4	5	6
2	6	1	9	5	3	7	4	8
3	7	5	6	4	8	2	9	1
8	9	4	1	7	2	3	6	5

Enter digits from 1 to 9 into the blank spaces

Every row must contain one of each digit

So must every column

as must every 3x3 square

This instance has exactly one solution

Sudoku

4	1	8	7	9	5	6	3	2
6	3	9	8	2	1	5	7	4
5	2	7	3	6	4	1	8	9
9	5	2	4	3	6	8	1	7
1	4	6	5	8	7	9	2	3
7	8	3	2	1	9	4	5	6
2	6	1	9	5	3	7	4	8
3	7	5	6	4	8	2	9	1
8	9	4	1	7	2	3	6	5

Enter digits from 1 to 9 into the blank spaces

Every row must contain one of each digit

So must every column

as must every 3x3 square

This instance has exactly one solution

Sudoku as an instance of SAT

9	4		8	7	9			3	
8			9	8	2		5		
7		2							
6	9		2			6		1	7
5			6	5	8	7	9		
4	7	8		2			4		6
3								4	
2			5		4	8	2		
1		9			7	2	3		5
	1	2	3	4	5	6	7	8	9

Sudoku as an instance of SAT

9	4		8	7	9			3	
8			9	8	2		5		
7		2							
6	9		2			6		1	7
5			6	5	8	7	9		
4	7	8		2			4		6
3								4	
2			5		4	8	2		
1		9			7	2	3		5
	1	2	3	4	5	6	7	8	9

Introduce 729 propositional variables v_{rcd}, where $r, c, d \in \{ 1, \ldots, 9 \}$

The variable v_{rcd} denotes that the cell in the row number r and column number c contains the digit d

Sudoku as an instance of SAT

9	4		8	7	9			3	
8			9	8	2		5		
7		2							
6	9		2			6		1	7
5			6	5	8	7	9		
4	7	8		2			4		6
3								4	
2			5		4	8	2		
1		9			7	2	3		5
	1	2	3	4	5	6	7	8	9

Introduce 729 propositional variables v_{rcd}, where $r, c, d \in \{ 1, \ldots, 9 \}$

The variable v_{rcd} denotes that the cell in the row number r and column number c contains the digit d

For example, this configuration satisfies the formula $v_{129} \wedge v_{268} \wedge \neg v_{691}$

Sudoku as an instance of SAT

9	4		8	7	9			3	
8			9	8	2		5		
7		2							
6	9		2			6		1	7
5			6	5	8	7	9		
4	7	8		2			4		6
3								4	
2			5		4	8	2		
1		9			7	2	3		5
	1	2	3	4	5	6	7	8	9

Introduce 729 propositional variables v_{rcd}, where $r, c, d \in \{1, \ldots, 9\}$

The variable v_{rcd} denotes that the cell in the row number r and column number c contains the digit d

For example, this configuration satisfies the formula $v_{129} \wedge v_{268} \wedge \neg v_{691}$

We should express all rules of Sudoku using the variables v_{rcd}

Encoding Sudoku in SAT

We have to write down that each cell contains exactly one digit

Encoding Sudoku in SAT

We have to write down that each cell contains exactly one digit

$$\{\, v_{rc1} \vee v_{rc2} \vee \cdots \vee v_{rc8} \vee v_{rc9} \mid r, c \in \{\, 1, \ldots, 9 \,\}\,\}$$
$$\{\, \neg v_{rc1} \vee \neg v_{rc2} \mid r, c \in \{\, 1, \ldots, 9 \,\}\,\}$$
$$\{\, \neg v_{rc1} \vee \neg v_{rc3} \mid r, c \in \{\, 1, \ldots, 9 \,\}\,\}$$
$$\vdots$$
$$\{\, \neg v_{rc8} \vee \neg v_{rc9} \mid r, c \in \{\, 1, \ldots, 9 \,\}\,\}$$

Encoding Sudoku in SAT

We have to write down that each cell contains exactly one digit

$$\{\, v_{rc1} \vee v_{rc2} \vee \cdots \vee v_{rc8} \vee v_{rc9} \mid r, c \in \{\, 1, \ldots, 9 \,\} \,\}$$
$$\{\, \neg v_{rc1} \vee \neg v_{rc2} \mid r, c \in \{\, 1, \ldots, 9 \,\} \,\}$$
$$\{\, \neg v_{rc1} \vee \neg v_{rc3} \mid r, c \in \{\, 1, \ldots, 9 \,\} \,\}$$
$$\vdots$$
$$\{\, \neg v_{rc8} \vee \neg v_{rc9} \mid r, c \in \{\, 1, \ldots, 9 \,\} \,\}$$

Every row must contain one of each digit:

Encoding Sudoku in SAT

We have to write down that each cell contains exactly one digit

$$\{\, v_{rc1} \vee v_{rc2} \vee \cdots \vee v_{rc8} \vee v_{rc9} \mid r, c \in \{\, 1, \ldots, 9 \,\}\,\}$$
$$\{\, \neg v_{rc1} \vee \neg v_{rc2} \mid r, c \in \{\, 1, \ldots, 9 \,\}\,\}$$
$$\{\, \neg v_{rc1} \vee \neg v_{rc3} \mid r, c \in \{\, 1, \ldots, 9 \,\}\,\}$$
$$\vdots$$
$$\{\, \neg v_{rc8} \vee \neg v_{rc9} \mid r, c \in \{\, 1, \ldots, 9 \,\}\,\}$$

Every row must contain one of each digit:

$$\{\, \neg v_{r,c,d} \vee \neg v_{r,c',d} \mid r, c, c', d \in \{\, 1, \ldots, 9 \,\}, c < c' \,\}$$

Encoding Sudoku in SAT

We have to write down that each cell contains exactly one digit

$$\{ v_{rc1} \lor v_{rc2} \lor \cdots \lor v_{rc8} \lor v_{rc9} \mid r, c \in \{ 1, \ldots, 9 \} \}$$
$$\{ \neg v_{rc1} \lor \neg v_{rc2} \mid r, c \in \{ 1, \ldots, 9 \} \}$$
$$\{ \neg v_{rc1} \lor \neg v_{rc3} \mid r, c \in \{ 1, \ldots, 9 \} \}$$
$$\vdots$$
$$\{ \neg v_{rc8} \lor \neg v_{rc9} \mid r, c \in \{ 1, \ldots, 9 \} \}$$

Every row must contain one of each digit:

$$\{ \neg v_{r,c,d} \lor \neg v_{r,c',d} \mid r, c, c', d \in \{ 1, \ldots, 9 \}, c < c' \}$$

Every column must contain one of each digit: similar

Every 3x3 square must contain one of each digit: similar

Encoding Sudoku in SAT

We have to write down that each cell contains exactly one digit

$$\{ v_{rc1} \vee v_{rc2} \vee \cdots \vee v_{rc8} \vee v_{rc9} \mid r, c \in \{ 1, \ldots, 9 \} \}$$
$$\{ \neg v_{rc1} \vee \neg v_{rc2} \mid r, c \in \{ 1, \ldots, 9 \} \}$$
$$\{ \neg v_{rc1} \vee \neg v_{rc3} \mid r, c \in \{ 1, \ldots, 9 \} \}$$
$$\vdots$$
$$\{ \neg v_{rc8} \vee \neg v_{rc9} \mid r, c \in \{ 1, \ldots, 9 \} \}$$

2,997 clauses
6,561 literals

Every row must contain one of each digit:

$$\{ \neg v_{r,c,d} \vee \neg v_{r,c',d} \mid r, c, c', d \in \{ 1, \ldots, 9 \}, c < c' \}$$

Every column must contain one of each digit: similar

Every 3x3 square must contain one of each digit: similar

Encoding Sudoku in SAT

We have to write down that each cell contains exactly one digit

$$\{\, v_{rc1} \vee v_{rc2} \vee \cdots \vee v_{rc8} \vee v_{rc9} \mid r, c \in \{\, 1, \ldots, 9 \,\} \,\}$$
$$\{\, \neg v_{rc1} \vee \neg v_{rc2} \mid r, c \in \{\, 1, \ldots, 9 \,\} \,\}$$
$$\{\, \neg v_{rc1} \vee \neg v_{rc3} \mid r, c \in \{\, 1, \ldots, 9 \,\} \,\}$$
$$\vdots$$
$$\{\, \neg v_{rc8} \vee \neg v_{rc9} \mid r, c \in \{\, 1, \ldots, 9 \,\} \,\}$$

2,997 clauses
6,561 literals

Every row must contain one of each digit:

$$\{\, \neg v_{r,c,d} \vee \neg v_{r,c',d} \mid r, c, c', d \in \{\, 1, \ldots, 9 \,\}, c < c' \,\}$$

2,916 clauses
5,832 literals

Every column must contain one of each digit: similar

2,916 clauses
5,832 literals

Every 3x3 square must contain one of each digit: similar

2,916 clauses
5,832 literals

Encoding Sudoku in SAT

We have to write down that each cell contains exactly one digit

$$\{ v_{rc1} \vee v_{rc2} \vee \cdots \vee v_{rc8} \vee v_{rc9} \mid r, c \in \{1, \ldots, 9\} \}$$
$$\{ \neg v_{rc1} \vee \neg v_{rc2} \mid r, c \in \{1, \ldots, 9\} \}$$
$$\{ \neg v_{rc1} \vee \neg v_{rc3} \mid r, c \in \{1, \ldots, 9\} \}$$
$$\vdots$$
$$\{ \neg v_{rc8} \vee \neg v_{rc9} \mid r, c \in \{1, \ldots, 9\} \}$$

2,997 clauses
6,561 literals

Every row must contain one of each digit:

$$\{ \neg v_{r,c,d} \vee \neg v_{r,c',d} \mid r, c, c', d \in \{1, \ldots, 9\}, c < c' \}$$

2,916 clauses
5,832 literals

Every column must contain one of each digit: similar

2,916 clauses
5,832 literals

Every 3x3 square must contain one of each digit: similar

2,916 clauses
5,832 literals

729 variables, 11,745 clauses, 24,057 literals, nearly all clauses are binary

Encoding Sudoku in SAT

We have to write down that each cell contains exactly one digit

$$\{ v_{rc1} \lor v_{rc2} \lor \cdots \lor v_{rc8} \lor v_{rc9} \mid r, c \in \{ 1, \ldots, 9 \} \}$$
$$\{ \neg v_{rc1} \lor \neg v_{rc2} \mid r, c \in \{ 1, \ldots, 9 \} \}$$
$$\{ \neg v_{rc1} \lor \neg v_{rc3} \mid r, c \in \{ 1, \ldots, 9 \} \}$$
$$\vdots$$
$$\{ \neg v_{rc8} \lor \neg v_{rc9} \mid r, c \in \{ 1, \ldots, 9 \} \}$$

2,997 clauses
6,561 literals

Every row must contain one of each digit:

$$\{ \neg v_{r,c,d} \lor \neg v_{r,c',d} \mid r, c, c', d \in \{ 1, \ldots, 9 \}, c < c' \}$$

2,916 clauses
5,832 literals

Every column must contain one of each digit: similar

2,916 clauses
5,832 literals

Every 3x3 square must contain one of each digit: similar

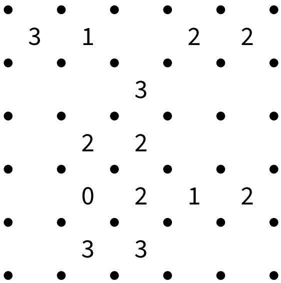
2,916 clauses
5,832 literals

Finally, we add unit clauses (e.g., v_{129}) corresponding to the initial configuration

Loop the Loop

3	1		2	2
		3		
	2	2		
	0	2	1	2
	3	3		

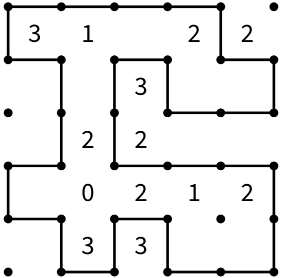
Loop the Loop



You have to draw lines between the dots to form a single loop without crossings or branches.

The numbers indicate how many lines surround it

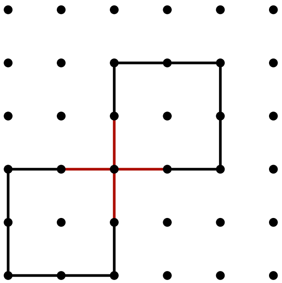
Loop the Loop



You have to draw lines between the dots to form a single loop without crossings or branches.

The numbers indicate how many lines surround it

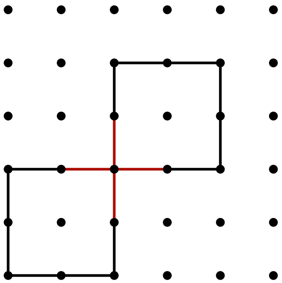
Loop the Loop



You have to draw lines between the dots to form a single loop without crossings or branches.

The numbers indicate how many lines surround it

Loop the Loop

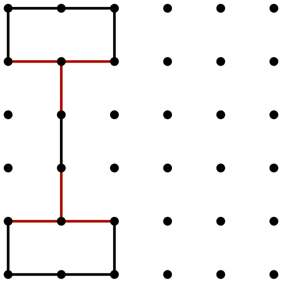


You have to draw lines between the dots to form a single loop without crossings or branches.

The numbers indicate how many lines surround it

A crossing is a node with four arcs attached to it

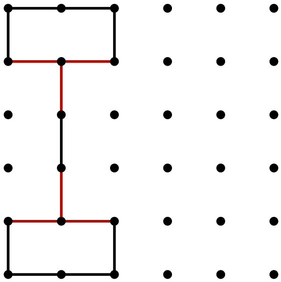
Loop the Loop



You have to draw lines between the dots to form a single loop without crossings or branches.

The numbers indicate how many lines surround it

Loop the Loop

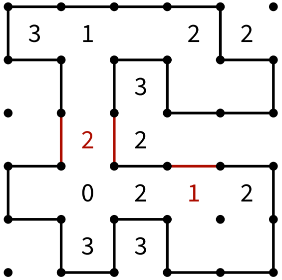


You have to draw lines between the dots to form a single loop without crossings or branches.

The numbers indicate how many lines surround it

A branch is a node with three arcs attached to it

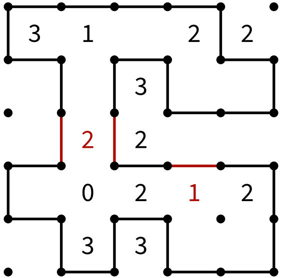
Loop the Loop



You have to draw lines between the dots to form a single loop without crossings or branches.

The numbers indicate how many lines surround it

Loop the Loop

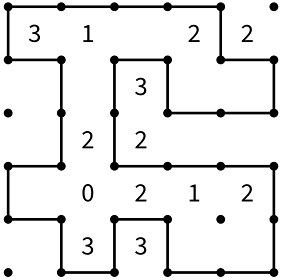


You have to draw lines between the dots to form a single loop without crossings or branches.

The numbers indicate how many lines surround it

If a cell contains a number m, then there should be m arcs around this number

Loop the Loop



You have to draw lines between the dots to form a single loop without crossings or branches.

The numbers indicate how many lines surround it

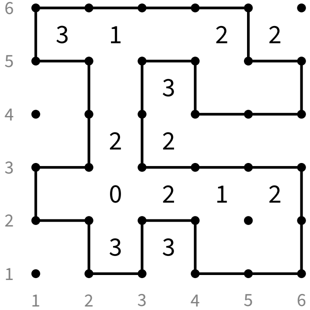
A crossing is a node with four arcs attached to it

A branch is a node with three arcs attached to it

If a cell contains a number *m*, then there should be *m* arcs around this number

All these properties are formulated in terms of (a number of) arcs

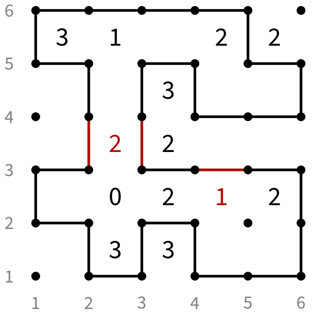
Formalization



Introduce variables denoting arcs:

- v_{ij}: there is a vertical arc between the nodes (i, j) and $(i, j+1)$
- h_{ij}: there is a horizontal arc between the nodes (i, j) and $(i+1, j)$

Formalization

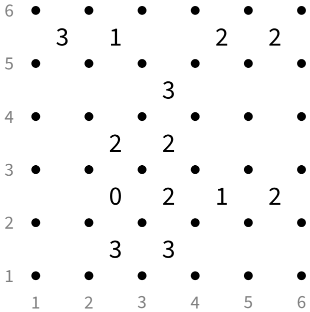


Introduce variables denoting arcs:

- v_{ij}: there is a vertical arc between the nodes (i, j) and $(i, j+1)$
- h_{ij}: there is a horizontal arc between the nodes (i, j) and $(i+1, j)$

Example: $v_{23} \wedge v_{33} \wedge h_{43}$

Formalization

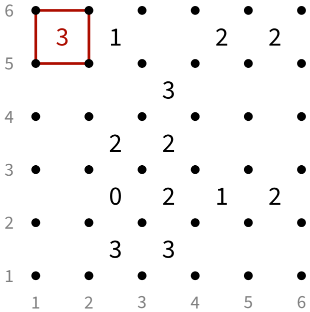


Introduce variables denoting arcs:

- v_{ij}: there is a vertical arc between the nodes (i, j) and $(i, j + 1)$
- h_{ij}: there is a horizontal arc between the nodes (i, j) and $(i + 1, j)$

Then almost all properties are formulated using the formulas T_k and these variables

Formalization



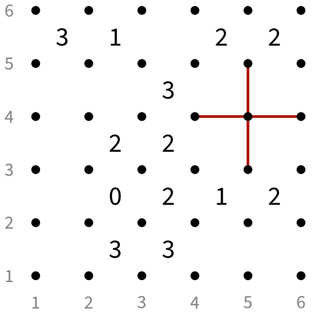
Introduce variables denoting arcs:

- v_{ij}: there is a vertical arc between the nodes (i, j) and $(i, j+1)$
- h_{ij}: there is a horizontal arc between the nodes (i, j) and $(i+1, j)$

Then almost all properties are formulated using the formulas T_k and these variables For example,

$$T_3(v_{15}, v_{25}, h_{15}, h_{16})$$

Formalization



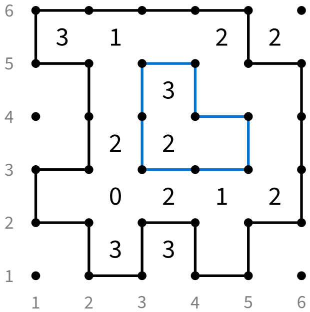
Introduce variables denoting arcs:

- v_{ij}: there is a vertical arc between the nodes (i, j) and $(i, j+1)$
- h_{ij}: there is a horizontal arc between the nodes (i, j) and $(i+1, j)$

Then almost all properties are formulated using the formulas T_k and these variables For example,

$$T_3(v_{15}, v_{25}, h_{15}, h_{16})$$
$$T_0(v_{53}, v_{54}, h_{44}, h_{45}) \vee T_2(v_{53}, v_{54}, h_{44}, h_{45})$$

Formalization



Introduce variables denoting arcs:

- v_{ij}: there is a vertical arc between the nodes (i, j) and $(i, j+1)$
- h_{ij}: there is a horizontal arc between the nodes (i, j) and $(i+1, j)$

Then almost all properties are formulated using the formulas T_k and these variables For example,

$$T_3(v_{15}, v_{25}, h_{15}, h_{16})$$
$$T_0(v_{53}, v_{54}, h_{44}, h_{45}) \vee T_2(v_{53}, v_{54}, h_{44}, h_{45})$$

What we cannot express is the property to have a single loop

There is no simple way of expressing this in PL