Automated Deduction

Laura Kovács
TU Wien

Decision Procedures for Propositional Satisfiability

More sophisticated decision procedures:

- Splitting;
- DPLL;
- (BDDs – binary decision diagrams);
- (resolution).

Decision Procedures for Propositional Satisfiability

More sophisticated decision procedures:

- Splitting;
- DPLL;
- (BDDs – binary decision diagrams);
- (resolution).

We start first with the splitting method.

Outline

Splitting: idea

$A_p^{\perp}$ and $A_p^{\top}$: the formulas obtained by replacing in A all occurrences of p by $\perp$ and $\top$, respectively.

Splitting: idea

$A_p^{\perp}$ and $A_p^{\top}$: the formulas obtained by replacing in A all occurrences of p by $\perp$ and $\top$, respectively.

Lemma

Let p be an atom, A be a formula, and I be an interpretation.

Splitting: idea

$A_p^\bot$ and $A_p^\top$: the formulas obtained by replacing in A all occurrences of p by $\bot$ and $\top$, respectively.

Lemma

Let p be an atom, A be a formula, and I be an interpretation.

1. *If $I \not\models p$, then A is equivalent to $A_p^\bot$ in I.*
2. *If $I \models p$, then A is equivalent to $A_p^\top$ in I.*

Splitting: idea

$A_p^{\perp}$ and $A_p^{\top}$: the formulas obtained by replacing in A all occurrences of p by $\perp$ and $\top$, respectively.

Lemma

Let p be an atom, A be a formula, and I be an interpretation.

1. *If $I \not\models p$, then A is equivalent to $A_p^{\perp}$ in I.*
2. *If $I \models p$, then A is equivalent to $A_p^{\top}$ in I.*

Theorem

Let A be a formula and p an atom.
Then A is satisfiable iff at least one of the formulas $A_p^{\top}$ and $A_p^{\perp}$ is satisfiable.

Splitting: idea

$A_p^{\perp}$ and $A_p^{\top}$: the formulas obtained by replacing in A all occurrences of p by $\perp$ and $\top$, respectively.

Lemma

Let p be an atom, A be a formula, and I be an interpretation.

1. *If $I \not\models p$, then A is equivalent to $A_p^{\perp}$ in I.*
2. *If $I \models p$, then A is equivalent to $A_p^{\top}$ in I.*

Theorem

Let A be a formula and p an atom.
Then A is satisfiable iff at least one of the formulas $A_p^{\top}$ and $A_p^{\perp}$ is satisfiable.

- Pick a variable p and perform case analysis on this variable:
  - If p is false, replace p by $\perp$;
  - If p is true, replace p by $\top$.

Splitting: idea

$A_p^\perp$ and $A_p^\top$: the formulas obtained by replacing in A all occurrences of p by $\perp$ and $\top$, respectively.

Lemma

Let p be an atom, A be a formula, and I be an interpretation.

1. *If $I \not\models p$, then A is equivalent to $A_p^\perp$ in I.*
2. *If $I \models p$, then A is equivalent to $A_p^\top$ in I.*

Theorem

Let A be a formula and p an atom.
Then A is satisfiable iff at least one of the formulas $A_p^\top$ and $A_p^\perp$ is satisfiable.

- Pick a variable p and perform case analysis on this variable:
  - If p is false, replace p by $\perp$;
  - If p is true, replace p by $\top$.
- When a formula contains occurrences of $\top$ or $\perp$, simplify it.

Simplification rules for $\top$ and $\bot$

Note: we need new simplification rules since formulas we simplify may contain propositional variables.

Simplification rules for $\top$:

$$
\begin{array}{c}
\neg\top \Rightarrow \bot \\
\top \wedge A_1 \wedge \ldots \wedge A_n \Rightarrow A_1 \wedge \ldots \wedge A_n \\
\top \vee A_1 \vee \ldots \vee A_n \Rightarrow \top \\
A \rightarrow \top \Rightarrow \top \qquad \top \rightarrow A \Rightarrow A \\
A \leftrightarrow \top \Rightarrow A \qquad \top \leftrightarrow A \Rightarrow A
\end{array}
$$

Simplification rules for $\bot$:

$$
\begin{array}{c}
\neg\bot \Rightarrow \top \\
\bot \wedge A_1 \wedge \ldots \wedge A_n \Rightarrow \bot \\
\bot \vee A_1 \vee \ldots \vee A_n \Rightarrow A_1 \vee \ldots \vee A_n \\
A \rightarrow \bot \Rightarrow \neg A \qquad \bot \rightarrow A \Rightarrow \top \\
A \leftrightarrow \bot \Rightarrow \neg A \qquad \bot \leftrightarrow A \Rightarrow \neg A
\end{array}
$$

Simplification rules for $\top$ and $\bot$

Note: we need new simplification rules since formulas we simplify may contain propositional variables.

Simplification rules for $\top$:

$$\neg\top \Rightarrow \bot$$
$$\top \wedge A_1 \wedge \ldots \wedge A_n \Rightarrow A_1 \wedge \ldots \wedge A_n$$
$$\top \vee A_1 \vee \ldots \vee A_n \Rightarrow \top$$
$$A \rightarrow \top \Rightarrow \top \qquad \top \rightarrow A \Rightarrow A$$
$$A \leftrightarrow \top \Rightarrow A \qquad \top \leftrightarrow A \Rightarrow A$$

Simplification rules for $\bot$:

$$\neg\bot \Rightarrow \top$$
$$\bot \wedge A_1 \wedge \ldots \wedge A_n \Rightarrow \bot$$
$$\bot \vee A_1 \vee \ldots \vee A_n \Rightarrow A_1 \vee \ldots \vee A_n$$
$$A \rightarrow \bot \Rightarrow \neg A \qquad \bot \rightarrow A \Rightarrow \top$$
$$A \leftrightarrow \bot \Rightarrow \neg A \qquad \bot \leftrightarrow A \Rightarrow \neg A$$

Note that they cover all cases when $\bot$ or $\top$ occurs in the formula apart from the trivial ones.

Simplification rules for $\top$ and $\bot$

Note: we need new simplification rules since formulas we simplify may contain propositional variables.

Simplification rules for $\top$:

$$
\begin{array}{rcl}
\neg\top & \Rightarrow & \bot \\
\top \wedge A_1 \wedge \ldots \wedge A_n & \Rightarrow & A_1 \wedge \ldots \wedge A_n \\
\top \vee A_1 \vee \ldots \vee A_n & \Rightarrow & \top
\end{array}
$$

$$
\begin{array}{rclcrcl}
A \rightarrow \top & \Rightarrow & \top & \quad & \top \rightarrow A & \Rightarrow & A \\
A \leftrightarrow \top & \Rightarrow & A & \quad & \top \leftrightarrow A & \Rightarrow & A
\end{array}
$$

Simplification rules for $\bot$:

$$
\begin{array}{rcl}
\neg\bot & \Rightarrow & \top \\
\bot \wedge A_1 \wedge \ldots \wedge A_n & \Rightarrow & \bot \\
\bot \vee A_1 \vee \ldots \vee A_n & \Rightarrow & A_1 \vee \ldots \vee A_n
\end{array}
$$

$$
\begin{array}{rclcrcl}
A \rightarrow \bot & \Rightarrow & \neg A & \quad & \bot \rightarrow A & \Rightarrow & \top \\
A \leftrightarrow \bot & \Rightarrow & \neg A & \quad & \bot \leftrightarrow A & \Rightarrow & \neg A
\end{array}
$$

Note that they cover all cases when $\bot$ or $\top$ occurs in the formula apart from the trivial ones.
Thus, if we apply these rules until they are no more applicable we obtain either $\bot$, or $\top$, or a formula containing neither $\bot$ nor $\top$.

Splitting algorithm

procedure *split*(G)
parameters: function *select*
input: formula G
output: "satisfiable" or "unsatisfiable"
begin
  $G \;:=\; simplify(G)$
  if $G = \top$ **then** **return** "satisfiable"
  if $G = \bot$ **then** **return** "unsatisfiable"
  $(p, b) \;:=\; select(G)$
  case b **of**
  $1 \Rightarrow$
   if $split(G_p^{\top}) =$"satisfiable"
    then **return** "satisfiable"
    else **return** $split(G_p^{\bot})$
  $0 \Rightarrow$
   if $split(G_p^{\bot}) =$"satisfiable"
    then **return** "satisfiable"
    else **return** $split(G_p^{\top})$
end

Splitting algorithm, example, splitting tree

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

Splitting algorithm, example, splitting tree

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

$q \rightarrow 0$

$$\neg((p \rightarrow \bot) \wedge (p \wedge \bot \rightarrow r) \rightarrow (p \rightarrow r))$$

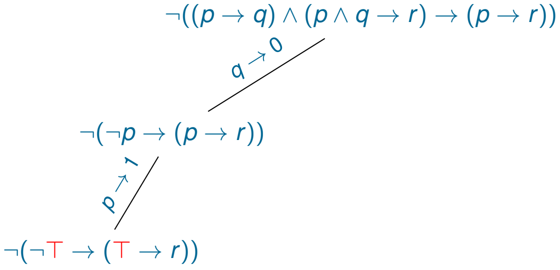
Splitting algorithm, example, splitting tree

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

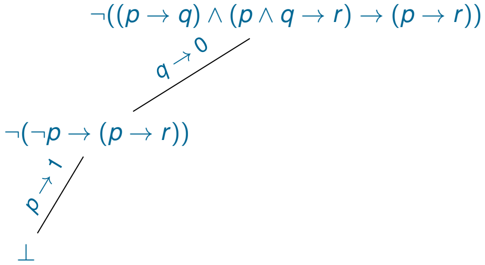
$q \to 0$

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

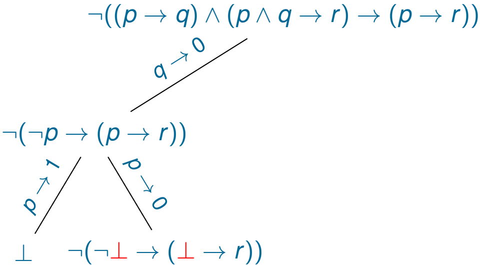
Splitting algorithm, example, splitting tree



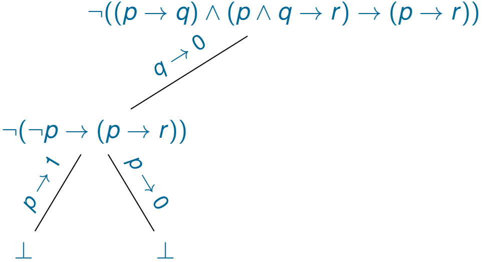
Splitting algorithm, example, splitting tree



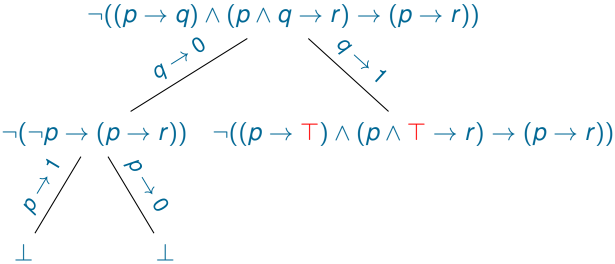
Splitting algorithm, example, splitting tree



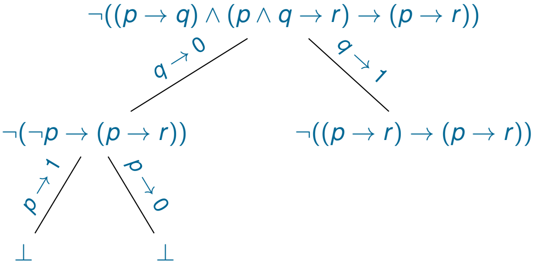
Splitting algorithm, example, splitting tree



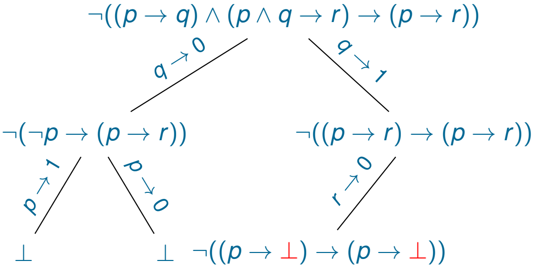
Splitting algorithm, example, splitting tree



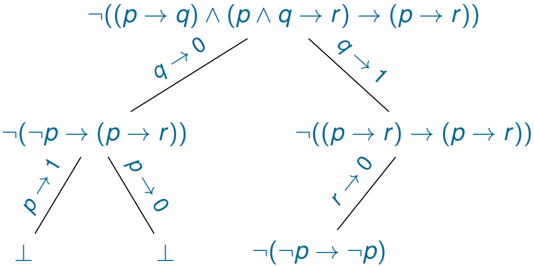
Splitting algorithm, example, splitting tree



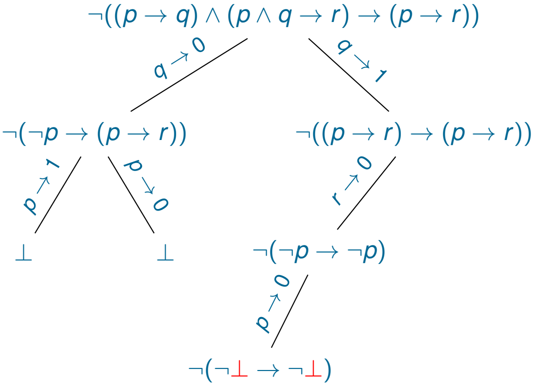
Splitting algorithm, example, splitting tree



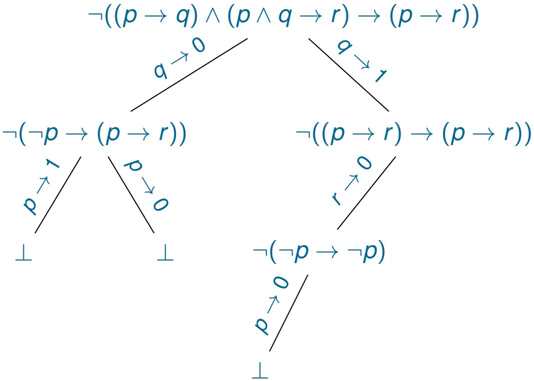
Splitting algorithm, example, splitting tree



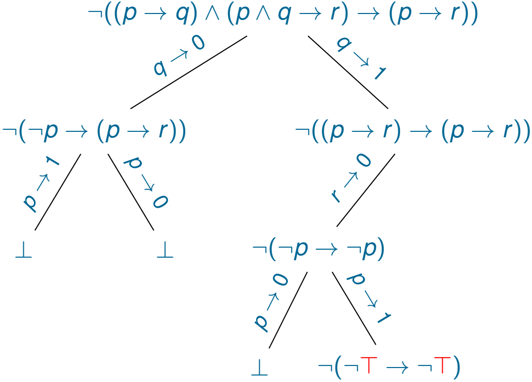
Splitting algorithm, example, splitting tree



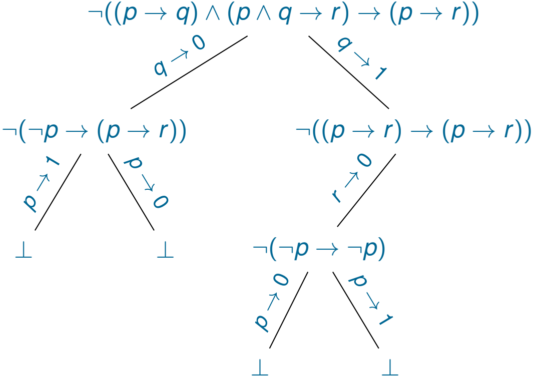
Splitting algorithm, example, splitting tree



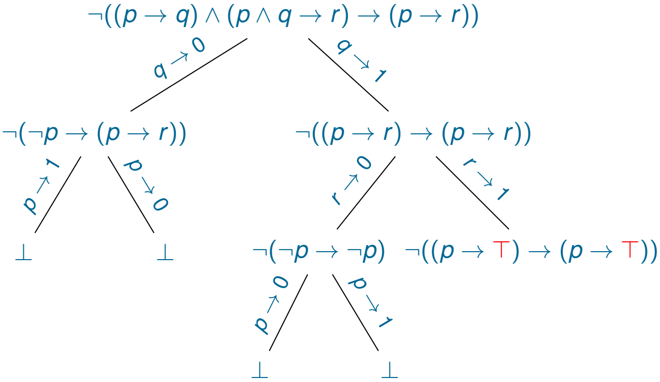
Splitting algorithm, example, splitting tree



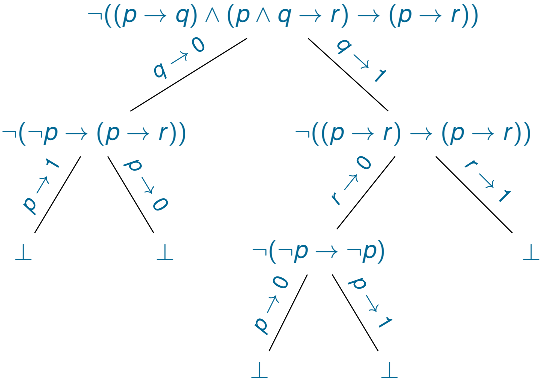
Splitting algorithm, example, splitting tree



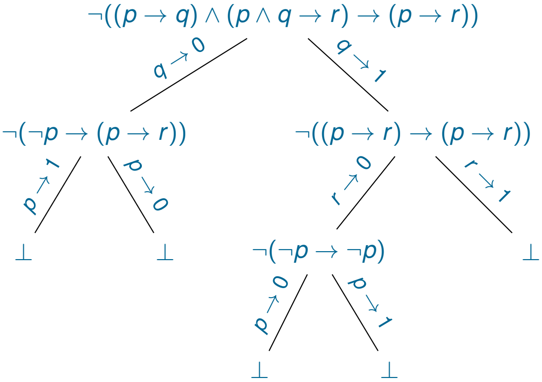
Splitting algorithm, example, splitting tree



Splitting algorithm, example, splitting tree

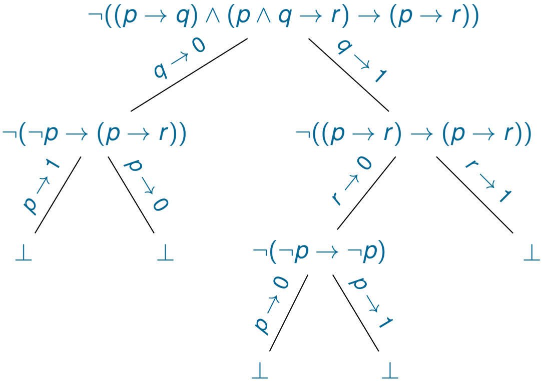


Splitting algorithm, example, splitting tree



The formula is unsatisfiable.

Splitting algorithm, example, splitting tree



The formula is unsatisfiable.

What is going on here is very similar to using compact truth tables, but on the syntactic level.

Splitting algorithm, example 2

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

Splitting algorithm, example 2

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

$p \to 0$

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

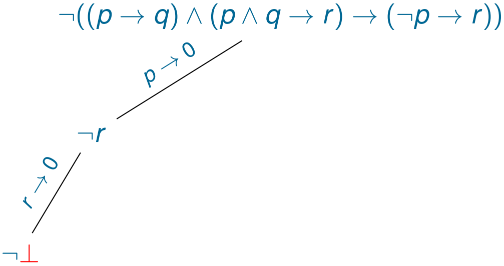
Splitting algorithm, example 2

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

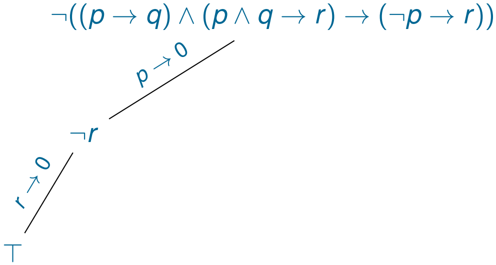
$p \to 0$

$\neg r$

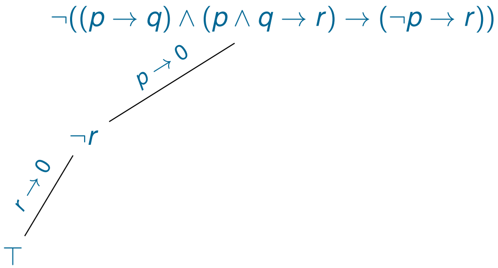
Splitting algorithm, example 2



Splitting algorithm, example 2

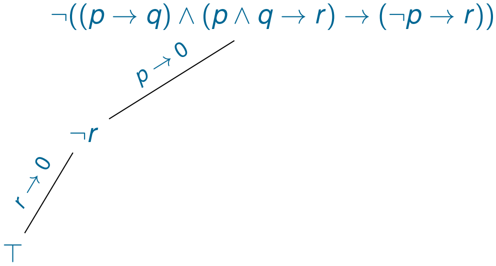


Splitting algorithm, example 2



The formula is satisfiable.

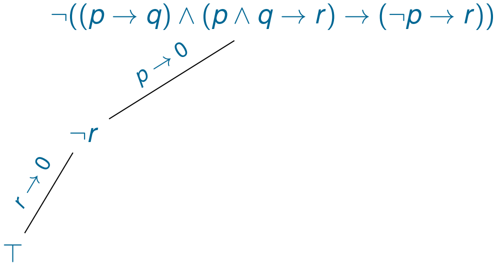
Splitting algorithm, example 2



The formula is satisfiable.

To find a model of this formula, we should simply collect choices made on the branch terminating at $\top$.

Splitting algorithm, example 2



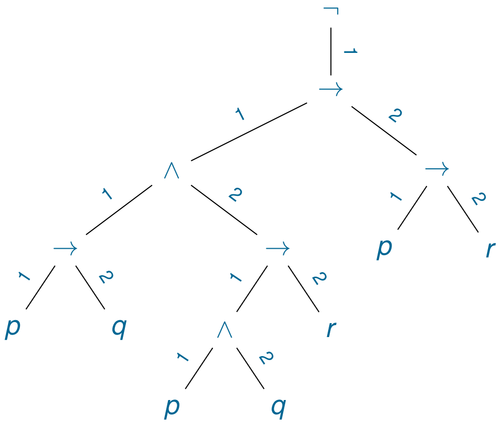
The formula is satisfiable.

To find a model of this formula, we should simply collect choices made on the branch terminating at $\top$.

Any interpretation I such that $I(p) = I(r) = 0$ satisfies the formula, for example the interpretation $\{p \mapsto 0, q \mapsto 0, r \mapsto 0\}$.

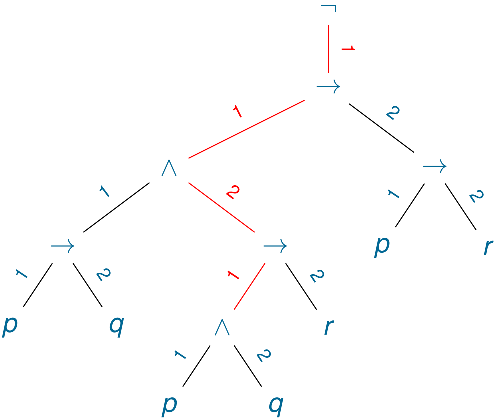
Parse tree

$A \stackrel{\text{def}}{=} \neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r))$.



Parse tree

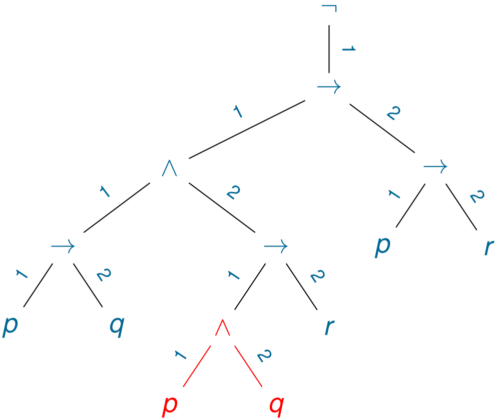
$A \stackrel{\text{def}}{=} \neg((p \to q) \land (p \land q \to r) \to (p \to r))$.



- Position in the formula: 1.1.2.1;

Parse tree

$A \stackrel{\text{def}}{=} \neg((p \to q) \land (p \land q \to r) \to (p \to r))$.



- Position in the formula: 1.1.2.1;
- Subformula at this position: $p \land q$.

Positions and Subformulas

- Position is any sequence of positive integers $a_1, \ldots, a_n$, where $n \geq 0$, written as $a_1.a_2.\ \cdots\ .a_n$.
- Empty position, denoted by ϵ: when $n = 0$.
- Position π in a formula A, subformula at a position, denoted $A|_\pi$.

Positions and Subformulas

- Position is any sequence of positive integers $a_1, \ldots, a_n$, where $n \geq 0$, written as $a_1.a_2.\ \cdots\ .a_n$.
- Empty position, denoted by ϵ: when $n = 0$.
- Position π in a formula A, subformula at a position, denoted $A|_\pi$.

1. For every formula A, ϵ is a position in A and $A|_\epsilon \stackrel{\text{def}}{=} A$.
2. Let $A|_\pi = B$.
   2.1 If B has the form $B_1 \wedge \ldots \wedge B_n$ or $B_1 \vee \ldots \vee B_n$, then for all $i \in \{1, \ldots, n\}$ the position $\pi.i$ is a position in A, $A|_{\pi.i} \stackrel{\text{def}}{=} B_i$.
   2.2 If B has the form $\neg B_1$, then $\pi.1$ is a position in A, $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$.
   2.3 If B has the form $B_1 \rightarrow B_2$, then $\pi.1$ and $\pi.2$ are positions in A and we have $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$, $A|_{\pi.2} \stackrel{\text{def}}{=} B_2$;
   2.4 If B has the form $B_1 \leftrightarrow B_2$, then $\pi.1$ and $\pi.2$ are positions in A and $A|_{\pi.i} \stackrel{\text{def}}{=} B_i$.

If $A|_\pi = B$, we also say that B occurs in A at the position π.

Polarity

1. For every formula A, ϵ is a position in A, $A|_{\epsilon} \stackrel{\text{def}}{=} A$

2. Let $A|_{\pi} = B$.

   2.1 If B has the form $B_1 \wedge \ldots \wedge B_n$ or $B_1 \vee \ldots \vee B_n$, then for all $i \in \{1, \ldots, n\}$ the position $\pi.i$ is a position in A, $A|_{\pi.i} \stackrel{\text{def}}{=} B_i$

   2.2 If B has the form $\neg B_1$, then $\pi.1$ is a position in A, $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$

   2.3 If B has the form $B_1 \rightarrow B_2$, then $\pi.1$ and $\pi.2$ are positions in A and we have $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$, $A|_{\pi.2} \stackrel{\text{def}}{=} B_2$

   2.4 If B has the form $B_1 \leftrightarrow B_2$, then $\pi.1$ and $\pi.2$ are positions in A and $A|_{\pi.i} \stackrel{\text{def}}{=} B_i$ for $i = 1, 2$.

Polarity

Polarity of subformula at a position. Notation: $pol(A, \pi)$.

1. For every formula A, ϵ is a position in A, $A|_\epsilon \stackrel{\text{def}}{=} A$

2. Let $A|_\pi = B$.

   2.1 If B has the form $B_1 \wedge \ldots \wedge B_n$ or $B_1 \vee \ldots \vee B_n$, then for all $i \in \{1, \ldots, n\}$ the position $\pi.i$ is a position in A, $A|_{\pi.i} \stackrel{\text{def}}{=} B_i$

   2.2 If B has the form $\neg B_1$, then $\pi.1$ is a position in A, $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$

   2.3 If B has the form $B_1 \rightarrow B_2$, then $\pi.1$ and $\pi.2$ are positions in A and we have $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$, $A|_{\pi.2} \stackrel{\text{def}}{=} B_2$

   2.4 If B has the form $B_1 \leftrightarrow B_2$, then $\pi.1$ and $\pi.2$ are positions in A and $A|_{\pi.i} \stackrel{\text{def}}{=} B_i$ for $i = 1, 2$.

Polarity

Polarity of subformula at a position. Notation: $pol(A, \pi)$.

1. For every formula A, ϵ is a position in A, $A|_\epsilon \stackrel{\text{def}}{=} A$ and $pol(A, \epsilon) \stackrel{\text{def}}{=} 1$.
2. Let $A|_\pi = B$.
   - 2.1 If B has the form $B_1 \land \ldots \land B_n$ or $B_1 \lor \ldots \lor B_n$, then for all $i \in \{1, \ldots, n\}$ the position $\pi.i$ is a position in A, $A|_{\pi.i} \stackrel{\text{def}}{=} B_i$, and $pol(A, \pi.i) \stackrel{\text{def}}{=} pol(A, \pi)$.
   - 2.2 If B has the form $\neg B_1$, then $\pi.1$ is a position in A, $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$ and $pol(A, \pi.1) \stackrel{\text{def}}{=} -pol(A, \pi)$.
   - 2.3 If B has the form $B_1 \to B_2$, then $\pi.1$ and $\pi.2$ are positions in A and we have $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$, $A|_{\pi.2} \stackrel{\text{def}}{=} B_2$, $pol(A, \pi.1) \stackrel{\text{def}}{=} -pol(A, \pi)$, $pol(A, \pi.2) \stackrel{\text{def}}{=} pol(A, \pi)$.
   - 2.4 If B has the form $B_1 \leftrightarrow B_2$, then $\pi.1$ and $\pi.2$ are positions in A and $A|_{\pi.i} \stackrel{\text{def}}{=} B_i$ and $pol(A, \pi.i) \stackrel{\text{def}}{=} 0$ for $i = 1, 2$.

Polarity

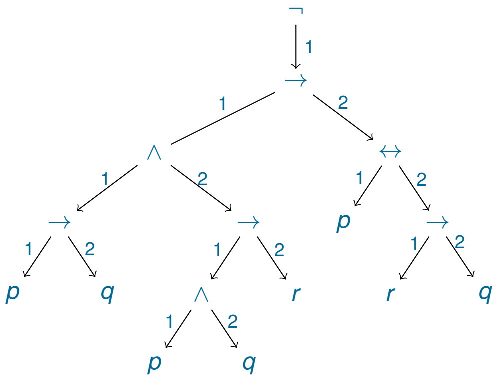
Polarity of subformula at a position. Notation: $pol(A, \pi)$.

1. For every formula A, ϵ is a position in A, $A|_\epsilon \stackrel{\text{def}}{=} A$ and $pol(A, \epsilon) \stackrel{\text{def}}{=} 1$.
2. Let $A|_\pi = B$.
   2.1 If B has the form $B_1 \wedge \ldots \wedge B_n$ or $B_1 \vee \ldots \vee B_n$, then for all $i \in \{1, \ldots, n\}$ the position $\pi.i$ is a position in A, $A|_{\pi.i} \stackrel{\text{def}}{=} B_i$, and $pol(A, \pi.i) \stackrel{\text{def}}{=} pol(A, \pi)$.
   2.2 If B has the form $\neg B_1$, then $\pi.1$ is a position in A, $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$ and $pol(A, \pi.1) \stackrel{\text{def}}{=} -pol(A, \pi)$.
   2.3 If B has the form $B_1 \rightarrow B_2$, then $\pi.1$ and $\pi.2$ are positions in A and we have $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$, $A|_{\pi.2} \stackrel{\text{def}}{=} B_2$, $pol(A, \pi.1) \stackrel{\text{def}}{=} -pol(A, \pi)$, $pol(A, \pi.2) \stackrel{\text{def}}{=} pol(A, \pi)$.
   2.4 If B has the form $B_1 \leftrightarrow B_2$, then $\pi.1$ and $\pi.2$ are positions in A and $A|_{\pi.i} \stackrel{\text{def}}{=} B_i$ and $pol(A, \pi.i) \stackrel{\text{def}}{=} 0$ for $i = 1, 2$.

- If $pol(A, \pi) = 1$ and $A|_\pi = B$, then we call the occurrence of B at the position π in A positive.
- If $pol(A, \pi) = -1$ and $A|_\pi = B$, then we call the occurrence of B at the position π in A negative.

The coloring algorithm for determining polarity

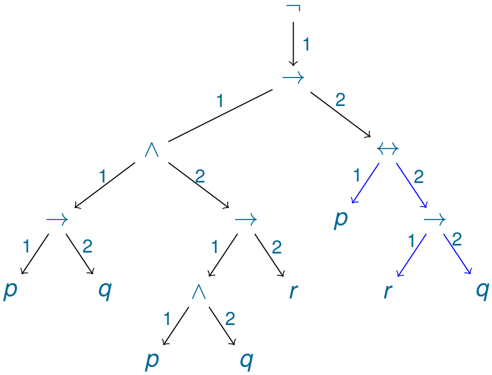
$\neg((p \to q) \land (p \land q \to r) \to (p \leftrightarrow (r \to q)))$.



The coloring algorithm for determining polarity

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

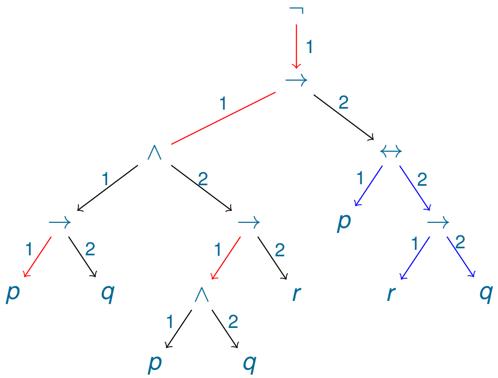
- Color in blue all arcs below an equivalence.



The coloring algorithm for determining polarity

$\neg((p \to q) \wedge (p \wedge q \to r) \to (p \leftrightarrow (r \to q)))$.

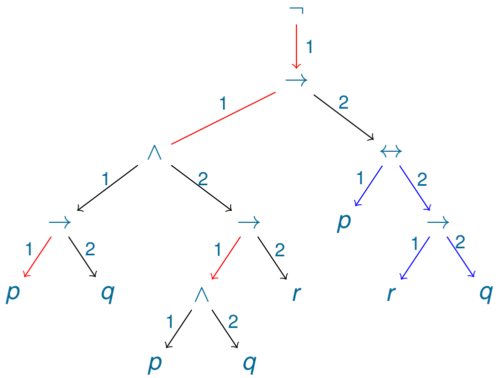
- Color in blue all arcs below an equivalence.
- Color in red all uncolored arcs going down from a negation or left-hand side of an implication.



The coloring algorithm for determining polarity

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

- Color in blue all arcs below an equivalence.
- Color in red all uncolored arcs going down from a negation or left-hand side of an implication.

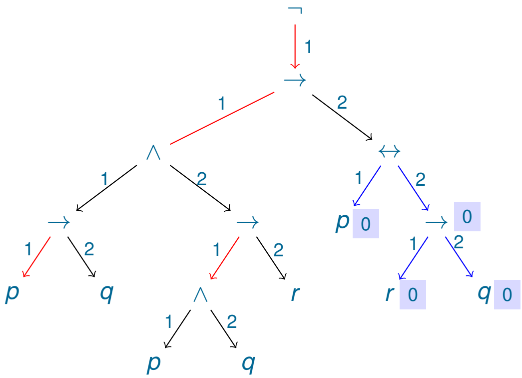


- If a position has at least one blue arc above it, its polarity is 0.

The coloring algorithm for determining polarity

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

- Color in blue all arcs below an equivalence.
- Color in red all uncolored arcs going down from a negation or left-hand side of an implication.

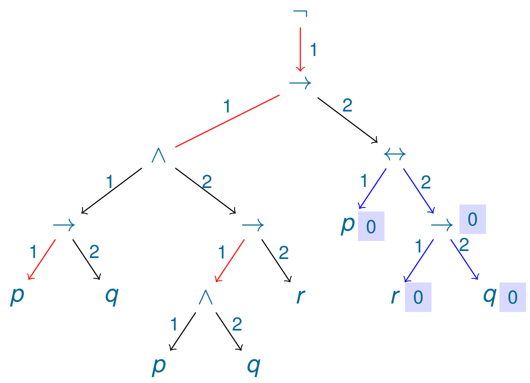


- If a position has at least one blue arc above it, its polarity is 0.

The coloring algorithm for determining polarity

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

- Color in blue all arcs below an equivalence.
- Color in red all uncolored arcs going down from a negation or left-hand side of an implication.

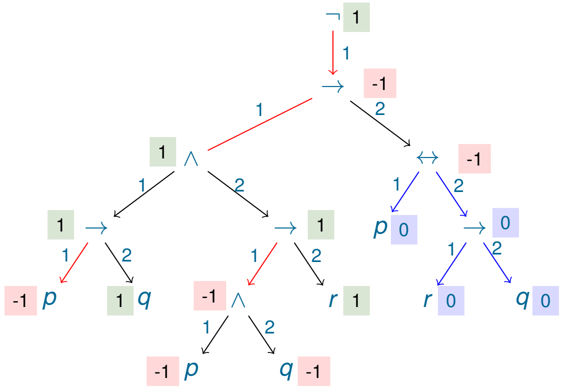


- If a position has at least one blue arc above it, its polarity is 0.
- Otherwise, its polarity is -1 if it has an odd number of red arcs above it.

The coloring algorithm for determining polarity

$\neg((p \to q) \wedge (p \wedge q \to r) \to (p \leftrightarrow (r \to q)))$.

- Color in blue all arcs below an equivalence.
- Color in red all uncolored arcs going down from a negation or left-hand side of an implication.



- If a position has at least one blue arc above it, its polarity is 0.
- Otherwise, its polarity is -1 if it has an odd number of red arcs above it.

Position and polarity, again

position	subformula	polarity
ϵ	$\neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r))$	1
1	$(p \to q) \wedge (p \wedge q \to r) \to (p \to r)$	-1
1.1	$(p \to q) \wedge (p \wedge q \to r)$	1
1.1.1	$p \to q$	1
1.1.1.1	p	-1
1.1.1.2	q	1
1.1.2	$p \wedge q \to r$	1
1.1.2.1	$p \wedge q$	-1
1.1.2.1.1	p	-1
1.1.2.1.2	q	-1
1.1.2.2	r	1
1.2	$p \to r$	-1
1.2.1	p	1
1.2.2	r	-1

Monotonic replacement

Notation: $A[B]_\pi$:

- formula A with the subformula B at the position π;
- formula A with the subformula at the position π replaced by B.

Monotonic replacement

Notation: $A[B]_\pi$:

- formula A with the subformula B at the position π;
- formula A with the subformula at the position π replaced by B.

Lemma (Monotonic Replacement)

Let A, B, B' be formulas, I be an interpretation, and $I \models B \rightarrow B'$. If $pol(A, \pi) = 1$, then $I \models A[B]_\pi \rightarrow A[B']_\pi$. Likewise, if $pol(A, \pi) = -1$, then $I \models A[B']_\pi \rightarrow A[B]_\pi$.

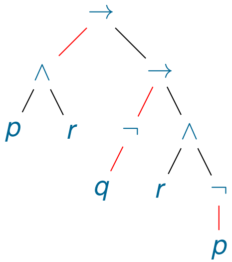
Pure Atom

Atom p is pure in a formula A, if either all occurrences of p in A are positive or all occurrences of p in A are negative.

Pure Atom

Atom p is pure in a formula A, if either all occurrences of p in A are positive or all occurrences of p in A are negative.

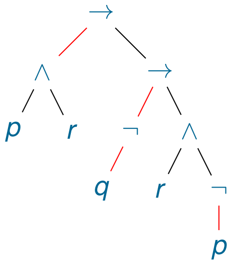
$$p \wedge r \rightarrow (\neg q \rightarrow (r \wedge \neg p))$$



Pure Atom

Atom p is pure in a formula A, if either all occurrences of p in A are positive or all occurrences of p in A are negative.

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

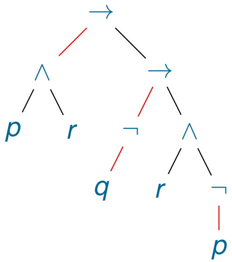


- Both occurrences of p are negative, so p is pure.

Pure Atom

Atom p is pure in a formula A, if either all occurrences of p in A are positive or all occurrences of p in A are negative.

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

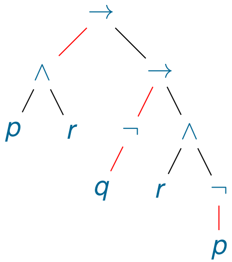


- Both occurrences of p are negative, so p is pure.
- The only occurrence of q is positive, so q is pure.

Pure Atom

Atom p is pure in a formula A, if either all occurrences of p in A are positive or all occurrences of p in A are negative.

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



- Both occurrences of p are negative, so p is pure.
- The only occurrence of q is positive, so q is pure.
- r is not pure, since it has both negative and positive occurrences.

Properties of Pure Atoms

Lemma (Pure Atom)

Let p has only positive occurrences in A and $I \models A$. Define

$$I' \stackrel{\text{def}}{=} I + (p \mapsto 1)$$

Then $I' \models A$.

Properties of Pure Atoms

Lemma (Pure Atom)

Let p has only positive occurrences in A and $I \models A$. Define

$$I' \stackrel{\text{def}}{=} I + (p \mapsto 1)$$

Then $I' \models A$.
Likewise, let p has only negative occurrences in A and $I \models A$. Define

$$I' \stackrel{\text{def}}{=} I + (p \mapsto 0)$$

Then $I' \models A$.

Properties of Pure Atoms

Lemma (Pure Atom)

Let p has only positive occurrences in A and $I \models A$. Define

$$I' \stackrel{\text{def}}{=} I + (p \mapsto 1)$$

Then $I' \models A$.
Likewise, let p has only negative occurrences in A and $I \models A$. Define

$$I' \stackrel{\text{def}}{=} I + (p \mapsto 0)$$

Then $I' \models A$.

Theorem (Pure Atom)

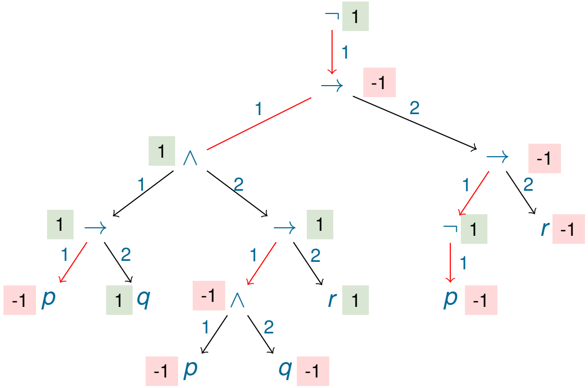
Let an atom p has only positive (respectively, only negative) occurrences in A. Then A is satisfiable if and only if so is $A_p^\top$ (respectively, $A_p^\perp$).

Pure atom, example

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

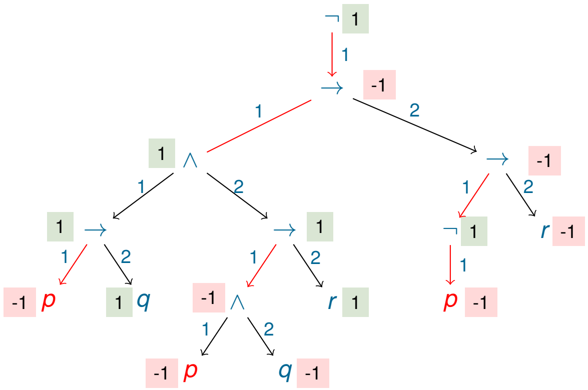
Pure atom, example

Consider $\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r))$.



Pure atom, example

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



All occurrences of p are negative, so, for the purpose of checking satisfiability we can replace p by $\bot$.

Example, continued

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

All occurrences of p are negative

Example, continued

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

All occurrences of p are negative, so, for the purpose of checking satisfiability we can replace p by $\bot$.

Example, continued

$$
\begin{array}{rl}
\neg((p \to q) \land (p \land q \to r) \to (\neg p \to r)) & \Rightarrow \\
\neg((\bot \to q) \land (\bot \land q \to r) \to (\neg \bot \to r)) & \Rightarrow \\
\neg(\top \land (\bot \land q \to r) \to (\neg \bot \to r)) &
\end{array}
$$

Example, continued

$$\neg((p \to q) \land (p \land q \to r) \to (\neg p \to r)) \quad \Rightarrow$$
$$\neg((\bot \to q) \land (\bot \land q \to r) \to (\neg \bot \to r)) \quad \Rightarrow$$
$$\neg(\top \land (\bot \land q \to r) \to (\neg \bot \to r)) \quad \Rightarrow$$
$$\neg((\bot \land q \to r) \to (\neg \bot \to r))$$

Example, continued

$$\begin{array}{lr}
\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) & \Rightarrow \\
\neg((\bot \to q) \wedge (\bot \wedge q \to r) \to (\neg\bot \to r)) & \Rightarrow \\
\neg(\top \wedge (\bot \wedge q \to r) \to (\neg\bot \to r)) & \Rightarrow \\
\neg((\bot \wedge q \to r) \to (\neg\bot \to r)) & \Rightarrow \\
\neg((\bot \to r) \to (\neg\bot \to r)) &
\end{array}$$

Example, continued

$$\neg((p \to q) \land (p \land q \to r) \to (\neg p \to r)) \quad \Rightarrow$$
$$\neg((\bot \to q) \land (\bot \land q \to r) \to (\neg \bot \to r)) \quad \Rightarrow$$
$$\neg(\top \land (\bot \land q \to r) \to (\neg \bot \to r)) \quad \Rightarrow$$
$$\neg((\bot \land q \to r) \to (\neg \bot \to r)) \quad \Rightarrow$$
$$\neg((\bot \to r) \to (\neg \bot \to r)) \quad \Rightarrow$$
$$\neg(\top \to (\neg \bot \to r))$$

Example, continued

$$
\begin{array}{cl}
\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) & \Rightarrow \\
\neg((\bot \to q) \wedge (\bot \wedge q \to r) \to (\neg \bot \to r)) & \Rightarrow \\
\neg(\top \wedge (\bot \wedge q \to r) \to (\neg \bot \to r)) & \Rightarrow \\
\neg((\bot \wedge q \to r) \to (\neg \bot \to r)) & \Rightarrow \\
\neg((\bot \to r) \to (\neg \bot \to r)) & \Rightarrow \\
\neg(\top \to (\neg \bot \to r)) & \Rightarrow \\
\neg(\neg \bot \to r) &
\end{array}
$$

Example, continued

$$\neg((p \to q) \land (p \land q \to r) \to (\neg p \to r)) \quad \Rightarrow$$
$$\neg((\bot \to q) \land (\bot \land q \to r) \to (\neg \bot \to r)) \quad \Rightarrow$$
$$\neg(\top \land (\bot \land q \to r) \to (\neg \bot \to r)) \quad \Rightarrow$$
$$\neg((\bot \land q \to r) \to (\neg \bot \to r)) \quad \Rightarrow$$
$$\neg((\bot \to r) \to (\neg \bot \to r)) \quad \Rightarrow$$
$$\neg(\top \to (\neg \bot \to r)) \quad \Rightarrow$$
$$\neg(\neg \bot \to r) \quad \Rightarrow$$
$$\neg(\top \to r)$$

Example, continued

$$\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) \quad \Rightarrow$$
$$\neg((\bot \to q) \wedge (\bot \wedge q \to r) \to (\neg \bot \to r)) \quad \Rightarrow$$
$$\neg(\top \wedge (\bot \wedge q \to r) \to (\neg \bot \to r)) \quad \Rightarrow$$
$$\neg((\bot \wedge q \to r) \to (\neg \bot \to r)) \quad \Rightarrow$$
$$\neg((\bot \to r) \to (\neg \bot \to r)) \quad \Rightarrow$$
$$\neg(\top \to (\neg \bot \to r)) \quad \Rightarrow$$
$$\neg(\neg \bot \to r) \quad \Rightarrow$$
$$\neg(\top \to r) \quad \Rightarrow$$
$$\neg r$$

All occurrences of r are negative

Example, continued

$$\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) \quad \Rightarrow$$
$$\neg((\bot \to q) \wedge (\bot \wedge q \to r) \to (\neg\bot \to r)) \quad \Rightarrow$$
$$\neg(\top \wedge (\bot \wedge q \to r) \to (\neg\bot \to r)) \quad \Rightarrow$$
$$\neg((\bot \wedge q \to r) \to (\neg\bot \to r)) \quad \Rightarrow$$
$$\neg((\bot \to r) \to (\neg\bot \to r)) \quad \Rightarrow$$
$$\neg(\top \to (\neg\bot \to r)) \quad \Rightarrow$$
$$\neg(\neg\bot \to r) \quad \Rightarrow$$
$$\neg(\top \to r) \quad \Rightarrow$$
$$\neg r \quad \Rightarrow$$
$$\neg\bot$$

All occurrences of r are negative, so, for the purpose of checking satisfiability we can replace r by $\bot$.

Example, continued

$$\neg((p \to q) \wedge (p \wedge q \to r) \to (\neg p \to r)) \quad \Rightarrow$$
$$\neg((\bot \to q) \wedge (\bot \wedge q \to r) \to (\neg\bot \to r)) \quad \Rightarrow$$
$$\neg(\top \wedge (\bot \wedge q \to r) \to (\neg\bot \to r)) \quad \Rightarrow$$
$$\neg((\bot \wedge q \to r) \to (\neg\bot \to r)) \quad \Rightarrow$$
$$\neg((\bot \to r) \to (\neg\bot \to r)) \quad \Rightarrow$$
$$\neg(\top \to (\neg\bot \to r)) \quad \Rightarrow$$
$$\neg(\neg\bot \to r) \quad \Rightarrow$$
$$\neg(\top \to r) \quad \Rightarrow$$
$$\neg r \quad \Rightarrow$$
$$\neg\bot \quad \Rightarrow$$
$$\top$$

We have shown satisfiability of this formula deterministically, using only the pure atom rule.

Splitting algorithm with pure atom optimizaton

```
procedure split(G)
parameters: function select
input: formula G
output: "satisfiable" or "unsatisfiable"
begin
  G := simplify_with_pure_atoms(G)
  if G = ⊤ then return "satisfiable"
  if G = ⊥ then return "unsatisfiable"
  (p, b) := select(G)
  case b of
  1 ⇒
    if split(G_p^⊤) ="satisfiable"
      then return "satisfiable"
      else return split(G_p^⊥)
  0 ⇒
    if split(G_p^⊥) ="satisfiable"
      then return "satisfiable"
      else return split(G_p^⊤)
end
```