CS:4350 Logic in Computer Science

Propositional Satisfiability

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

Propositional Satisfiability

In many real-world problems, we are interested in whether a set of constraints is solvable.

When these constraints are expressible in Propositional Logic, the problem reduces to checking the satisfiability of a set of formulas.

Satisfiability in PL is a very general problem

Propositional Satisfiability

In many real-world problems, we are interested in whether a set of constraints is solvable.

When these constraints are expressible in Propositional Logic, the problem reduces to checking the satisfiability of a set of formulas.

Satisfiability in PL is a very general problem

Propositional Satisfiability

In many real-world problems, we are interested in whether a set of constraints is solvable.

When these constraints are expressible in Propositional Logic, the problem reduces to checking the satisfiability of a set of formulas.

Satisfiability in PL is a very general problem

Propositional Satisfiability

In fact, even entailment in PL can be reduced to satisfiability. Recall:

$$A_1, \ldots, A_n \models B \quad \text{iff} \quad \{A_1, \ldots, A_n, \neg B\} \text{ is unsatisfiable}$$

Upshot: we do not really need a derivation system to prove PL formulas if we have a satisfiability procedure!

Great news: satisfiability in PL, aka the *SAT problem*, is decidable

Bad news: no fast (polynomial-time) and general algorithms for SAT in general are known

Reality: there are automated reasoning techniques that work extremely well for SAT in practice

Propositional Satisfiability

In fact, even entailment in PL can be reduced to satisfiability. Recall:

$$A_1, \ldots, A_n \models B \quad \text{iff} \quad \{A_1, \ldots, A_n, \neg B\} \text{ is unsatisfiable}$$

Upshot: we do not really need a derivation system to prove PL formulas if we have a satisfiability procedure!

Great news: satisfiability in PL, aka the *SAT problem*, is decidable

Bad news: no fast (polynomial-time) and general algorithms for SAT in general are known

Reality: there are automated reasoning techniques that work extremely well for SAT in practice

Propositional Satisfiability

In fact, even entailment in PL can be reduced to satisfiability. Recall:

$$A_1, \ldots, A_n \models B \quad \text{iff} \quad \{A_1, \ldots, A_n, \neg B\} \text{ is unsatisfiable}$$

Upshot: we do not really need a derivation system to prove PL formulas if we have a satisfiability procedure!

Great news: satisfiability in PL, aka the *SAT problem*, is decidable

Bad news: no fast (polynomial-time) and general algorithms for SAT in general are known

Reality: there are automated reasoning techniques that work extremely well for SAT in practice

Propositional Satisfiability

In fact, even entailment in PL can be reduced to satisfiability. Recall:

$$A_1, \ldots, A_n \models B \quad \text{iff} \quad \{A_1, \ldots, A_n, \neg B\} \text{ is unsatisfiable}$$

Upshot: we do not really need a derivation system to prove PL formulas if we have a satisfiability procedure!

Great news: satisfiability in PL, aka the *SAT problem*, is decidable

Bad news: no fast (polynomial-time) and general algorithms for SAT in general are known

Reality: there are automated reasoning techniques that work extremely well for SAT in practice

Propositional Satisfiability

In fact, even entailment in PL can be reduced to satisfiability. Recall:

$$A_1, \ldots, A_n \models B \quad \text{iff} \quad \{A_1, \ldots, A_n, \neg B\} \text{ is unsatisfiable}$$

Upshot: we do not really need a derivation system to prove PL formulas if we have a satisfiability procedure!

Great news: satisfiability in PL, aka the *SAT problem*, is decidable

Bad news: no fast (polynomial-time) and general algorithms for SAT in general are known

Reality: there are automated reasoning techniques that work extremely well for SAT in practice

A Puzzle

Isaac and Albert were excitedly describing the result of the Third Annual International Science Fair Extravaganza in Sweden.

There were three contestants: Louis, Rene, and Johannes.

Isaac reported that Louis won the fair, while Rene came in second. Albert, on the other hand, reported that Johannes won the fair, while Louis came in second.

In fact, neither Isaac nor Albert had given a correct report of the results of the science fair. Each of them had given one true statement and one false statement.

What was the actual placing of the three contestants?

How can we solve this kind of puzzle?

A Puzzle

Isaac and Albert were excitedly describing the result of the Third Annual International Science Fair Extravaganza in Sweden.

There were three contestants: Louis, Rene, and Johannes.

Isaac reported that Louis won the fair, while Rene came in second. Albert, on the other hand, reported that Johannes won the fair, while Louis came in second.

In fact, neither Isaac nor Albert had given a correct report of the results of the science fair. Each of them had given one true statement and one false statement.

What was the actual placing of the three contestants?

How can we solve this kind of puzzle?

A Puzzle

Isaac and Albert were excitedly describing the result of the Third Annual International Science Fair Extravaganza in Sweden.

There were three contestants: Louis, Rene, and Johannes.

Isaac reported that Louis won the fair, while Rene came in second. Albert, on the other hand, reported that Johannes won the fair, while Louis came in second.

In fact, neither Isaac nor Albert had given a correct report of the results of the science fair. Each of them had given one true statement and one false statement.

What was the actual placing of the three contestants?

How can we solve this kind of puzzle?

A Puzzle

Isaac and Albert were excitedly describing the result of the Third Annual International Science Fair Extravaganza in Sweden.

There were three contestants: Louis, Rene, and Johannes.

Isaac reported that Louis won the fair, while Rene came in second. Albert, on the other hand, reported that Johannes won the fair, while Louis came in second.

In fact, neither Isaac nor Albert had given a correct report of the results of the science fair. Each of them had given one true statement and one false statement.

What was the actual placing of the three contestants?

How can we solve this kind of puzzle?

A Puzzle

Isaac and Albert were excitedly describing the result of the Third Annual International Science Fair Extravaganza in Sweden.

There were three contestants: Louis, Rene, and Johannes.

Isaac reported that Louis won the fair, while Rene came in second. Albert, on the other hand, reported that Johannes won the fair, while Louis came in second.

In fact, neither Isaac nor Albert had given a correct report of the results of the science fair. Each of them had given one true statement and one false statement.

What was the actual placing of the three contestants?

How can we solve this kind of puzzle?

Propositional Satisfiability Problem

Given a propositional formula A, check if it is satisfiable or not.

If it is, also find a *satisfying assignment* for A (a model of A).

One of the most famous combinatorial problems in CS

It is a very hard problem computationally, with a surprisingly large number of practical applications.

It was also the first ever problem to be proved NP-complete.

Propositional Satisfiability Problem

Given a propositional formula A, check if it is satisfiable or not.

If it is, also find a *satisfying assignment* for A (a model of A).

One of the most famous combinatorial problems in CS

It is a very hard problem computationally, with a surprisingly large number of practical applications.

It was also the first ever problem to be proved NP-complete.

Propositional Satisfiability Problem

Given a propositional formula A, check if it is satisfiable or not.

If it is, also find a *satisfying assignment* for A (a model of A).

One of the most famous combinatorial problems in CS

It is a very hard problem computationally, with a surprisingly large number of practical applications.

It was also the first ever problem to be proved NP-complete.

Propositional Satisfiability Problem

Given a propositional formula A, check if it is satisfiable or not.

If it is, also find a *satisfying assignment* for A (a model of A).

One of the most famous combinatorial problems in CS

It is a very hard problem computationally, with a surprisingly large number of practical applications.

It was also the first ever problem to be proved NP-complete.

Propositional Satisfiability Problem

Given a propositional formula A, check if it is satisfiable or not.

If it is, also find a *satisfying assignment* for A (a model of A).

One of the most famous combinatorial problems in CS

It is a very hard problem computationally, with a surprisingly large number of practical applications.

It was also the first ever problem to be proved NP-complete.

Russian spy puzzle

There are three people: Stirlitz, Müller, and Eismann. It is known that exactly one of them is Russian, while the other two are German. Is is also know that every Russian is a spy.

When Stirlitz meets Müller in a hallway, he makes the following joke: "you know, Müller, you are as German as I am Russian". It is known that Stirlitz always tells the truth when he is joking.

We have to show that Eismann is not a Russian spy.

How can we solve problems of this kind?

Russian spy puzzle

There are three people: Stirlitz, Müller, and Eismann. It is known that exactly one of them is Russian, while the other two are German. Is is also know that every Russian is a spy.

When Stirlitz meets Müller in a hallway, he makes the following joke: “you know, Müller, you are as German as I am Russian”. It is known that Stirlitz always tells the truth when he is joking.

We have to show that Eismann is not a Russian spy.

How can we solve problems of this kind?

Russian spy puzzle

There are three people: Stirlitz, Müller, and Eismann. It is known that exactly one of them is Russian, while the other two are German. Is is also know that every Russian is a spy.

When Stirlitz meets Müller in a hallway, he makes the following joke: “you know, Müller, you are as German as I am Russian”. It is known that Stirlitz always tells the truth when he is joking.

We have to show that Eismann is not a Russian spy.

How can we solve problems of this kind?

Russian spy puzzle

There are three people: Stirlitz, Müller, and Eismann. It is known that exactly one of them is Russian, while the other two are German. Is is also know that every Russian is a spy.

When Stirlitz meets Müller in a hallway, he makes the following joke: “you know, Müller, you are as German as I am Russian”. It is known that Stirlitz always tells the truth when he is joking.

We have to show that Eismann is not a Russian spy.

How can we solve problems of this kind?

Formalization in propositional logic

Introduce nine propositional variables as in the following table:

	Stirlitz	Müller	Eismann
Russian	RS	RM	RE
German	GS	GM	GE
Spy	SS	SM	SE

Example

SE : Eismann is a Spy
RS : Stirlitz is Russian

Formalization in propositional logic

Introduce nine propositional variables as in the following table:

	Stirlitz	Müller	Eismann
Russian	RS	RM	RE
German	GS	GM	GE
Spy	SS	SM	SE

Example

SE : Eismann is a Spy
RS : Stirlitz is Russian

Formalization in propositional logic

There are three people: Stirlitz, Müller, and Eismann. It is known that exactly one of them is Russian, while the other two are German.

$$(RS \land GM \land GE) \lor (GS \land RM \land GE) \lor (GS \land GM \land RE)$$

It is also known that every Russian is a spy.

$$(RS \to SS) \land (RM \to SM) \land (RE \to SE)$$

When Stirlitz meets Müller in a hallway, he makes the following joke: “you know, Müller, you are as German as I am Russian.”

$$RS \leftrightarrow GM$$

Formalization in propositional logic

There are three people: Stirlitz, Müller, and Eismann. It is known that exactly one of them is Russian, while the other two are German.

$$(RS \wedge GM \wedge GE) \vee (GS \wedge RM \wedge GE) \vee (GS \wedge GM \wedge RE)$$

It is also known that every Russian is a spy.

$$(RS \to SS) \wedge (RM \to SM) \wedge (RE \to SE)$$

When Stirlitz meets Müller in a hallway, he makes the following joke: “you know, Müller, you are as German as I am Russian.”

$$RS \leftrightarrow GM$$

Formalization in propositional logic

There are three people: Stirlitz, Müller, and Eismann. It is known that exactly one of them is Russian, while the other two are German.

$$(RS \wedge GM \wedge GE) \vee (GS \wedge RM \wedge GE) \vee (GS \wedge GM \wedge RE)$$

It is also known that every Russian is a spy.

$$(RS \rightarrow SS) \wedge (RM \rightarrow SM) \wedge (RE \rightarrow SE)$$

When Stirlitz meets Müller in a hallway, he makes the following joke: “you know, Müller, you are as German as I am Russian.”

$$RS \leftrightarrow GM$$

Formalization in propositional logic

There are three people: Stirlitz, Müller, and Eismann. It is known that exactly one of them is Russian, while the other two are German.

$$(RS \wedge GM \wedge GE) \vee (GS \wedge RM \wedge GE) \vee (GS \wedge GM \wedge RE)$$

It is also known that every Russian is a spy.

$$(RS \rightarrow SS) \wedge (RM \rightarrow SM) \wedge (RE \rightarrow SE)$$

When Stirlitz meets Müller in a hallway, he makes the following joke: “you know, Müller, you are as German as I am Russian.”

$$RS \leftrightarrow GM$$

Formalization in propositional logic

Implicit knowledge: Russians are not Germans.

$$(RS \leftrightarrow \neg GS) \land (RM \leftrightarrow \neg GM) \land (RE \leftrightarrow \neg GE)$$

We want to prove that Eismann is not a Russian spy.

Formalization in propositional logic

Implicit knowledge: Russians are not Germans.

$$(RS \leftrightarrow \neg GS) \wedge (RM \leftrightarrow \neg GM) \wedge (RE \leftrightarrow \neg GE)$$

We want to prove that Eismann is not a Russian spy.

Formalization in propositional logic

Implicit knowledge: Russians are not Germans.

$$(RS \leftrightarrow \neg GS) \land (RM \leftrightarrow \neg GM) \land (RE \leftrightarrow \neg GE)$$

We want to prove that Eismann is not a Russian spy.

Formalization in propositional logic

Implicit knowledge: Russians are not Germans.

$$(RS \leftrightarrow \neg GS) \wedge (RM \leftrightarrow \neg GM) \wedge (RE \leftrightarrow \neg GE)$$

We want to prove that Eismann is not a Russian spy.

To this end, we add the following constraint, stating the opposite.

$$RE \wedge SE$$

Formalization in propositional logic

Implicit knowledge: Russians are not Germans.

$$(RS \leftrightarrow \neg GS) \wedge (RM \leftrightarrow \neg GM) \wedge (RE \leftrightarrow \neg GE)$$

We want to prove that Eismann is not a Russian spy.

To this end, we add the following constraint, stating the opposite.

$$RE \wedge SE$$

Then we verify that the full set of constraints is unsatisfiable.

Formalization in propositional logic

Implicit knowledge: Russians are not Germans.

$$(RS \leftrightarrow \neg GS) \wedge (RM \leftrightarrow \neg GM) \wedge (RE \leftrightarrow \neg GE)$$

We want to prove that Eismann is not a Russian spy.
To this end, we add the following constraint, stating the opposite.

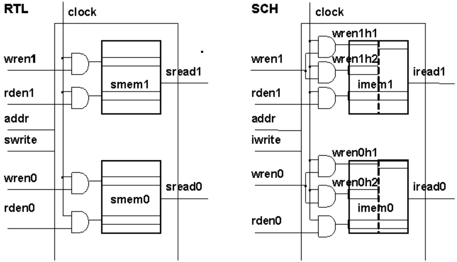
$$RE \wedge SE$$

Then we verify that the full set of constraints is unsatisfiable.

If the set is unsatisfiable, then Eismann cannot be a Russian spy

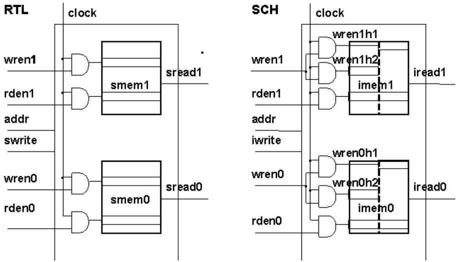
Circuit Equivalence

Given two circuits, check if they are equivalent. For example:



Circuit Equivalence

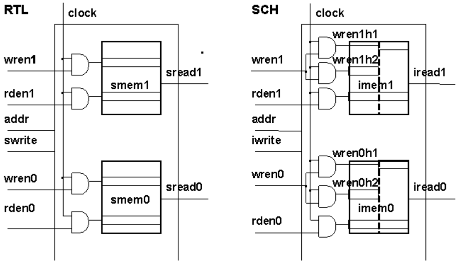
Given two circuits, check if they are equivalent. For example:



Every circuit is, in fact, a propositional formula and …

Circuit Equivalence

Given two circuits, check if they are equivalent. For example:

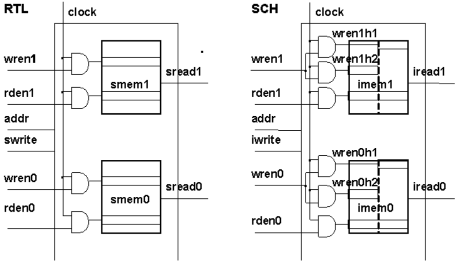


Every circuit is, in fact, a propositional formula and …

> equivalence checking for propositional formulas can be reduced to unsatisfiability checking

Circuit Equivalence

Given two circuits, check if they are equivalent. For example:



C_1 C_2

$C_1 \equiv C_2$ iff $\neg(C_1 \leftrightarrow C_2)$ is unsatisfiable

Idea for SAT: use formula evaluation methods

$$A \;=\; \neg((p \to q) \land (p \land q \to r) \to (p \to r))$$

We can evaluate A in any interpretation, e.g., $\mathcal{I}_1 = \{\, p \mapsto 0, q \mapsto 0, r \mapsto 0 \,\}$:

	subformula	$\mathcal{I}_1$
1	$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$	0
2	$(p \to q) \land (p \land q \to r) \to (p \to r)$	1
3	$p \to r$	1
4	$(p \to q) \land (p \land q \to r)$	1
5	$p \land q \to r$	1
6	$p \to q$	1
7	$p \land q$	0
8	$p \quad p \quad p$	0
9	$q \quad q$	0
10	$r \quad r$	0

Truth tables

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

Similarly, we can evaluate A in all interpretations:

	subformula	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_5$	$\mathcal{I}_6$	$\mathcal{I}_7$	$\mathcal{I}_8$
1	$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$	0	0	0	0	0	0	0	0
2	$(p \to q) \land (p \land q \to r) \to (p \to r)$	1	1	1	1	1	1	1	1
3	$p \to r$	1	1	1	1	0	1	0	1
4	$(p \to q) \land (p \land q \to r)$	1	1	1	1	0	0	0	1
5	$p \land q \to r$	1	1	1	1	1	1	0	1
6	$p \to q$	1	1	1	1	0	0	1	1
7	$p \land q$	0	0	0	0	0	0	1	1
8	p	0	0	0	0	1	1	1	1
9	q	0	0	1	1	0	0	1	1
10	r	0	1	0	1	0	1	0	1

Truth tables

$$A \;=\; \neg((p \to q) \land (p \land q \to r) \to (p \to r))$$

Formula A is unsatisfiable since it is false in every interpretation

So we have a fully automated method to check the satisfiability propositional formulas

Problem: A propositional formula with n variables has 2^n different interpretations!

Generating and checking each interpretation in $1\ ms$ for a formula with 50 variables would take $2^{50}\ ms \approx 257\ centuries \ldots$

With current automated reasoning technology, we can check formulas with 10K variables in seconds

Truth tables

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

Formula A is unsatisfiable since it is false in every interpretation

So we have a fully automated method to check the satisfiability propositional formulas

Problem: A propositional formula with n variables has 2^n different interpretations!

Generating and checking each interpretation in $1\ ms$ for a formula with 50 variables would take $2^{50}\ ms \approx 257\ centuries \ldots$

With current automated reasoning technology, we can check formulas with 10K variables in seconds

Truth tables

So we have a fully automated method to check the satisfiability propositional formulas

Problem: A propositional formula with n variables has 2^n different interpretations!

Generating and checking each interpretation in $1\ ms$ for a formula with 50 variables would take $2^{50}\ ms \approx 257\ centuries \ldots$

With current automated reasoning technology, we can check formulas with 10K variables in seconds

Truth tables

So we have a fully automated method to check the satisfiability propositional formulas

Problem: A propositional formula with n variables has 2^n different interpretations!

Generating and checking each interpretation in $1\ ms$ for a formula with 50 variables would take $2^{50}\ ms \approx 257\ centuries \ldots$

With current automated reasoning technology, we can check formulas with 10K variables in seconds

Truth tables

So we have a fully automated method to check the satisfiability propositional formulas

Problem: A propositional formula with n variables has 2^n different interpretations!

Generating and checking each interpretation in $1\ ms$ for a formula with 50 variables would take $2^{50}\ ms \approx 257\ centuries \ldots$

With current automated reasoning technology, we can check formulas with 10K variables in seconds

Compact truth table

Idea: Sometimes we can evaluate a formula based only on *partial interpretations*

subformula	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_1$
$\neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r))$	0	0	0	0
$(p \to q) \wedge (p \wedge q \to r) \to (p \to r)$	1	1	1	1
$p \to r$	1	0	0	1
$(p \to q) \wedge (p \wedge q \to r)$		0	0	
$p \wedge q \to r$		1	0	1
$p \to q$		0	1	
$p \wedge q$		0	1	
p	0	1	1	
q		0	1	
r	0	0	0	1

Compact truth table

Idea: Sometimes we can evaluate a formula based only on *partial interpretations*

subformula	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_1$
$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$	0	0	0	0
$(p \to q) \land (p \land q \to r) \to (p \to r)$	1	1	1	1
$p \to r$	1	0	0	1
$(p \to q) \land (p \land q \to r)$		0	0	
$p \land q \to r$		1	0	1
$p \to q$		0	1	
$p \land q$		0	1	
p	0	1	1	
q		0	1	
r	0	0	0	1

Compact truth table

Idea: Sometimes we can evaluate a formula based only on *partial interpretations*

subformula	$\mathcal{J}_2$	$\mathcal{J}_3$	$\mathcal{J}_4$	$\mathcal{J}_1$
$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$	0	0	0	0
$(p \to q) \land (p \land q \to r) \to (p \to r)$	1	1	1	1
$p \to r$	1	0	0	1
$(p \to q) \land (p \land q \to r)$		0	0	
$p \land q \to r$		1	0	1
$p \to q$		0	1	
$p \land q$		0	1	
$p \quad p \quad p$	0	1	1	
$q \quad q$		0	1	
$r \quad r$	0	0	0	1

Compact truth table

Idea: Sometimes we can evaluate a formula based only on *partial interpretations*

subformula	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_1$
$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$	0	0	0	0
$(p \to q) \land (p \land q \to r) \to (p \to r)$	1	1	1	1
$p \to r$	1	0	0	1
$(p \to q) \land (p \land q \to r)$		0	0	
$p \land q \to r$		1	0	1
$p \to q$		0	1	
$p \land q$		0	1	
p $\quad p$ $\quad p$	0	1	1	
q $\quad q$		0	1	
r $\quad r$	0	0	0	1

Compact truth table

Idea: Sometimes we can evaluate a formula based only on *partial interpretations*

subformula	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_1$
$\neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r))$	0	0	0	0
$(p \to q) \wedge (p \wedge q \to r) \to (p \to r)$	1	1	1	1
$p \to r$	1	0	0	1
$(p \to q) \wedge (p \wedge q \to r)$		0	0	
$p \wedge q \to r$		1	0	1
$p \to q$		0	1	
$p \wedge q$		0	1	
$p \quad p \quad p$	0	1	1	
$q \quad q$		0	1	
$r \quad r$	0	0	0	1

Compact truth table

Idea: Sometimes we can evaluate a formula based only on *partial interpretations*

subformula	$\mathcal{J}_2$	$\mathcal{J}_3$	$\mathcal{J}_4$	$\mathcal{J}_1$
$\neg((p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r))$	0	0	0	0
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (p \rightarrow r)$	1	1	1	1
$p \rightarrow r$	1	0	0	1
$(p \rightarrow q) \wedge (p \wedge q \rightarrow r)$		0	0	
$p \wedge q \rightarrow r$		1	0	1
$p \rightarrow q$		0	1	
$p \wedge q$		0	1	
p p p	0	1	1	
q q		0	1	
r r	0	0	0	1

Compact truth table

Idea: Sometimes we can evaluate a formula based only on *partial interpretations*

subformula	$\mathcal{J}_2$	$\mathcal{J}_3$	$\mathcal{J}_4$	$\mathcal{J}_1$
$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$	0	0	0	0
$(p \to q) \land (p \land q \to r) \to (p \to r)$	1	1	1	1
$p \to r$	1	0	0	1
$(p \to q) \land (p \land q \to r)$		0	0	
$p \land q \to r$		1	0	1
$p \to q$		0	1	
$p \land q$		0	1	
p $\quad$ p $\quad$ p	0	1	1	
q $\quad$ q		0	1	
r $\quad$ r	0	0	0	1

Compact truth table

Idea: Sometimes we can evaluate a formula based only on *partial interpretations*

subformula	$\mathcal{J}_2$	$\mathcal{J}_3$	$\mathcal{J}_4$	$\mathcal{J}_1$
$\neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r))$	0	0	0	0
$(p \to q) \wedge (p \wedge q \to r) \to (p \to r)$	1	1	1	1
$p \to r$	1	0	0	1
$(p \to q) \wedge (p \wedge q \to r)$		0	0	
$p \wedge q \to r$		1	0	1
$p \to q$		0	1	
$p \wedge q$		0	1	
p	0	1	1	
q		0	1	
r	0	0	0	1

Compact truth table

Idea: Sometimes we can evaluate a formula based only on *partial interpretations*

subformula	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_1$
$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$	0	0	0	0
$(p \to q) \land (p \land q \to r) \to (p \to r)$	1	1	1	1
$p \to r$	1	0	0	1
$(p \to q) \land (p \land q \to r)$		0	0	
$p \land q \to r$		1	0	1
$p \to q$		0	1	
$p \land q$		0	1	
p p p	0	1	1	
q q		0	1	
r r	0	0	0	1

Compact truth table

Idea: Sometimes we can evaluate a formula based only on *partial interpretations*

subformula	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_1$
$\neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r))$	0	0	0	0
$(p \to q) \wedge (p \wedge q \to r) \to (p \to r)$	1	1	1	1
$p \to r$	1	0	0	1
$(p \to q) \wedge (p \wedge q \to r)$		0	0	
$p \wedge q \to r$		1	0	1
$p \to q$		0	1	
$p \wedge q$		0	1	
p p p	0	1	1	
q q		0	1	
r r	0	0	0	1

Compact truth table

Idea: Sometimes we can evaluate a formula based only on *partial interpretations*

subformula	$\mathcal{J}_2$	$\mathcal{J}_3$	$\mathcal{J}_4$	$\mathcal{J}_1$
$\neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r))$	0	0	0	0
$(p \to q) \wedge (p \wedge q \to r) \to (p \to r)$	1	1	1	1
$p \to r$	1	0	0	1
$(p \to q) \wedge (p \wedge q \to r)$		0	0	
$p \wedge q \to r$		1	0	1
$p \to q$		0	1	
$p \wedge q$		0	1	
p	0	1	1	
q		0	1	
r	0	0	0	1

Compact truth table

Idea: Sometimes we can evaluate a formula based only on *partial interpretations*

subformula	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_1$
$\neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r))$	0	0	0	0
$(p \to q) \wedge (p \wedge q \to r) \to (p \to r)$	1	1	1	1
$p \to r$	1	0	0	1
$(p \to q) \wedge (p \wedge q \to r)$		0	0	
$p \wedge q \to r$		1	0	1
$p \to q$		0	1	
$p \wedge q$		0	1	
p p p	0	1	1	
q q		0	1	
r r	0	0	0	1

Compact truth table

Idea: Sometimes we can evaluate a formula based only on *partial interpretations*

subformula	$\mathcal{J}_2$	$\mathcal{J}_3$	$\mathcal{J}_4$	$\mathcal{J}_1$
$\neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r))$	0	0	0	0
$(p \to q) \wedge (p \wedge q \to r) \to (p \to r)$	1	1	1	1
$p \to r$	1	0	0	1
$(p \to q) \wedge (p \wedge q \to r)$		0	0	
$p \wedge q \to r$		1	0	1
$p \to q$		0	1	
$p \wedge q$		0	1	
$p \quad p \quad p$	0	1	1	
$q \quad q$		0	1	
$r \quad r$	0	0	0	1

$\mathcal{J}_2$ stands for 2 (total) interpretations

$\mathcal{J}_1$ stands for 4 interpretations

Compact truth table

Idea: Sometimes we can evaluate a formula based only on *partial interpretations*

subformula	$\mathcal{J}_2$	$\mathcal{J}_3$	$\mathcal{J}_4$	$\mathcal{J}_1$
$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$	0	0	0	0
$(p \to q) \land (p \land q \to r) \to (p \to r)$	1	1	1	1
$p \to r$	1	0	0	1
$(p \to q) \land (p \land q \to r)$		0	0	
$p \land q \to r$		1	0	1
$p \to q$		0	1	
$p \land q$		0	1	
p p p	0	1	1	
q q		0	1	
r r	0	0	0	1

Note: The size of the compact table (but not the result) depends on the order of variables!

Compact truth table

Idea: Sometimes we can evaluate a formula based only on *partial interpretations*

subformula	$\mathcal{J}_2$	$\mathcal{J}_3$	$\mathcal{J}_4$	$\mathcal{J}_1$
$\neg((p \to q) \wedge (p \wedge q \to r) \to (p \to r))$	0	0	0	0
$(p \to q) \wedge (p \wedge q \to r) \to (p \to r)$	1	1	1	1
$p \to r$	1	0	0	1
$(p \to q) \wedge (p \wedge q \to r)$		0	0	
$p \wedge q \to r$		1	0	1
r r	0	0	0	1

Guessing variable values (i.e., case analysis) and propagation are the key ideas in nearly all propositional satisfiability algorithms

Note: The size of the compact table (but not the result) depends on the order of variables!

Case splitting: idea

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

Lemma 1

Let p be an atom, A be a formula, and $\mathcal{I}$ be an interpretation.

1. *If $\mathcal{I} \models p$, then A has the same value as $A_p^\top$ in $\mathcal{I}$.*
2. *If $\mathcal{I} \not\models p$, then A has the same value as $A_p^\bot$ in $\mathcal{I}$.*

Satisfiability checking by case analysis

1. Pick a variable p of A and perform case analysis on it:
   Case 1) replace p by $\bot$ (for false)
   Case 2) replace p by $\top$ (for true)
2. Simplify formula as much as possible
3. Repeat until A is $\top$ or $\bot$

Case splitting: idea

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

Lemma 1

Let p be an atom, A be a formula, and $\mathcal{I}$ be an interpretation.

1. *If $\mathcal{I} \models p$, then A has the same value as $A_p^\top$ in $\mathcal{I}$.*
2. *If $\mathcal{I} \not\models p$, then A has the same value as $A_p^\bot$ in $\mathcal{I}$.*

Satisfiability checking by case analysis

1. Pick a variable p of A and perform case analysis on it:
   Case 1) replace p by $\bot$ (for false)
   Case 2) replace p by $\top$ (for true)
2. Simplify formula as much as possible
3. Repeat until A is $\top$ or $\bot$

Case splitting: idea

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

Lemma 1

Let p be an atom, A be a formula, and $\mathcal{I}$ be an interpretation.

1. *If $\mathcal{I} \models p$, then A has the same value as $A_p^{\top}$ in $\mathcal{I}$.*
2. *If $\mathcal{I} \not\models p$, then A has the same value as $A_p^{\perp}$ in $\mathcal{I}$.*

Satisfiability checking by case analysis

1. Pick a variable p of A and perform case analysis on it:
   Case 1) replace p by $\perp$ (for false)
   Case 2) replace p by $\top$ (for true)
2. Simplify formula as much as possible
3. Repeat until A is $\top$ or $\perp$

Case splitting: idea

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

Lemma 1

Let p be an atom, A be a formula, and $\mathcal{I}$ be an interpretation.

1. *If $\mathcal{I} \models p$, then A has the same value as $A_p^\top$ in $\mathcal{I}$.*
2. *If $\mathcal{I} \not\models p$, then A has the same value as $A_p^\bot$ in $\mathcal{I}$.*

Satisfiability checking by case analysis

1. Pick a variable p of A and perform case analysis on it:
   Case 1) replace p by $\bot$ (for false)
   Case 2) replace p by $\top$ (for true)
2. Simplify formula as much as possible
3. Repeat until A is $\top$ or $\bot$

Case splitting: idea

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

Lemma 1

Let p be an atom, A be a formula, and $\mathcal{I}$ be an interpretation.

1. *If $\mathcal{I} \models p$, then A has the same value as $A_p^\top$ in $\mathcal{I}$.*
2. *If $\mathcal{I} \not\models p$, then A has the same value as $A_p^\bot$ in $\mathcal{I}$.*

Satisfiability checking by case analysis

1. Pick a variable p of A and perform case analysis on it:
   Case 1) replace p by $\bot$ (for false)
   Case 2) replace p by $\top$ (for true)
2. Simplify formula as much as possible
3. Repeat until A is $\top$ or $\bot$

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

Simplification rules for $\top$
$\neg\top \Rightarrow \bot$
$A_1 \wedge \cdots \wedge \top \wedge \cdots \wedge A_n \Rightarrow A_1 \wedge \cdots \wedge A_n$
$A_1 \vee \cdots \vee \top \vee \cdots \vee A_n \Rightarrow \top$
$A \to \top \Rightarrow \top$ $\qquad$ $\top \to A \Rightarrow A$
$A \leftrightarrow \top \Rightarrow A$ $\qquad$ $\top \leftrightarrow A \Rightarrow A$

Simplification rules for $\bot$
$\neg\bot \Rightarrow \top$
$A_1 \wedge \cdots \wedge \bot \wedge \cdots \wedge A_n \Rightarrow \bot$
$A_1 \vee \cdots \vee \bot \vee \cdots \vee A_n \Rightarrow A_1 \vee \cdots \vee A_n$
$A \to \bot \Rightarrow \neg A$ $\qquad$ $\bot \to A \Rightarrow \top$
$A \leftrightarrow \bot \Rightarrow \neg A$ $\qquad$ $\bot \leftrightarrow A \Rightarrow \neg A$

Claim: If we apply these rules to a formula to completion (i.e., until no more rules apply), we get either

- $\bot$,
- $\top$, or
- a formula with no occurrences of $\bot$ and $\top$

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

Note: we need new simplification rules to account for propositional variables

Simplification rules for $\top$	
$\neg\top \Rightarrow \bot$	
$A_1 \wedge \cdots \wedge \top \wedge \cdots \wedge A_n \Rightarrow A_1 \wedge \cdots \wedge A_n$	
$A_1 \vee \cdots \vee \top \vee \cdots \vee A_n \Rightarrow \top$	
$A \rightarrow \top \Rightarrow \top$	$\top \rightarrow A \Rightarrow A$
$A \leftrightarrow \top \Rightarrow A$	$\top \leftrightarrow A \Rightarrow A$

Simplification rules for $\bot$	
$\neg\bot \Rightarrow \top$	
$A_1 \wedge \cdots \wedge \bot \wedge \cdots \wedge A_n \Rightarrow \bot$	
$A_1 \vee \cdots \vee \bot \vee \cdots \vee A_n \Rightarrow A_1 \vee \cdots \vee A_n$	
$A \rightarrow \bot \Rightarrow \neg A$	$\bot \rightarrow A \Rightarrow \top$
$A \leftrightarrow \bot \Rightarrow \neg A$	$\bot \leftrightarrow A \Rightarrow \neg A$

Claim: If we apply these rules to a formula to completion (i.e., until no more rules apply), we get either

- $\bot$,
- $\top$, or
- a formula with no occurrences of $\bot$ and $\top$

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

Simplification rules for $\top$	
$\neg\top \;\Rightarrow\; \bot$	
$A_1 \wedge \cdots \wedge \top \wedge \cdots \wedge A_n \;\Rightarrow\; A_1 \wedge \cdots \wedge A_n$	
$A_1 \vee \cdots \vee \top \vee \cdots \vee A_n \;\Rightarrow\; \top$	
$A \rightarrow \top \;\Rightarrow\; \top$	$\top \rightarrow A \;\Rightarrow\; A$
$A \leftrightarrow \top \;\Rightarrow\; A$	$\top \leftrightarrow A \;\Rightarrow\; A$

Simplification rules for $\bot$	
$\neg\bot \;\Rightarrow\; \top$	
$A_1 \wedge \cdots \wedge \bot \wedge \cdots \wedge A_n \;\Rightarrow\; \bot$	
$A_1 \vee \cdots \vee \bot \vee \cdots \vee A_n \;\Rightarrow\; A_1 \vee \cdots \vee A_n$	
$A \rightarrow \bot \;\Rightarrow\; \neg A$	$\bot \rightarrow A \;\Rightarrow\; \top$
$A \leftrightarrow \bot \;\Rightarrow\; \neg A$	$\bot \leftrightarrow A \;\Rightarrow\; \neg A$

Claim: If we apply these rules to a formula to completion (i.e., until no more rules apply), we get either

- $\bot$,
- $\top$, or
- a formula with no occurrences of $\bot$ and $\top$

Splitting algorithm

procedure *split*(G)
parameters: function *select*
input: formula G
output: "satisfiable" or "unsatisfiable"
begin
  $G := \mathit{simplify}(G)$ // apply simplification rules to completion
  if $G = \top$ **then return** "satisfiable"
  if $G = \bot$ **then return** "unsatisfiable"
  $(p, b) := \mathit{select}(G)$ // pick a variable p of G and a value b for it
  case b **of**
  $1 \Rightarrow$
    if $\mathit{split}(G_p^\top) =$ "satisfiable"
      then return "satisfiable"
      else return $\mathit{split}(G_p^\bot)$
  $0 \Rightarrow$
    if $\mathit{split}(G_p^\bot) =$ "satisfiable"
      then return "satisfiable"
      else return $\mathit{split}(G_p^\top)$
end

Splitting algorithm, example

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

$\neg\top$	$\Rightarrow$	$\bot$
$\top \wedge A$	$\Rightarrow$	A
$\top \vee A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \wedge A$	$\Rightarrow$	$\bot$
$\bot \vee A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

Splitting algorithm, example

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

$q = 0$

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

$\neg\top \Rightarrow \bot$	$\neg\bot \Rightarrow \top$
$\top \land A \Rightarrow A$	$\bot \land A \Rightarrow \bot$
$\top \lor A \Rightarrow \top$	$\bot \lor A \Rightarrow A$
$A \to \top \Rightarrow \top$	$A \to \bot \Rightarrow \neg A$
$\top \to A \Rightarrow A$	$\bot \to A \Rightarrow \top$
$A \leftrightarrow \top \Rightarrow A$	$A \leftrightarrow \bot \Rightarrow \neg A$
$\top \leftrightarrow A \Rightarrow A$	$\bot \leftrightarrow A \Rightarrow \neg A$

Splitting algorithm, example

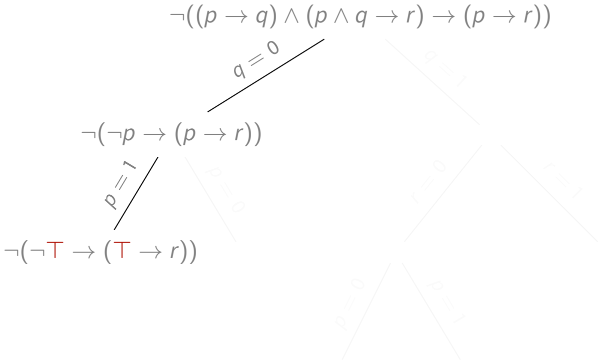
$$\neg((p \to q) \land (p \land q \to r) \to (p \to r))$$

$q = 0$

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

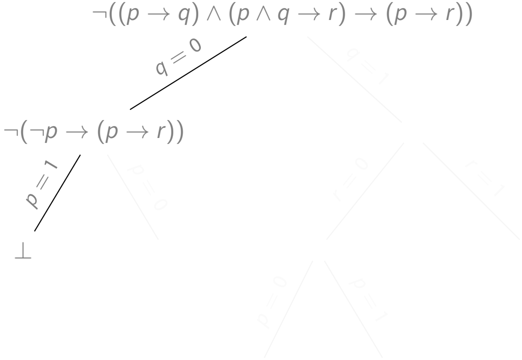
$\neg\top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

Splitting algorithm, example



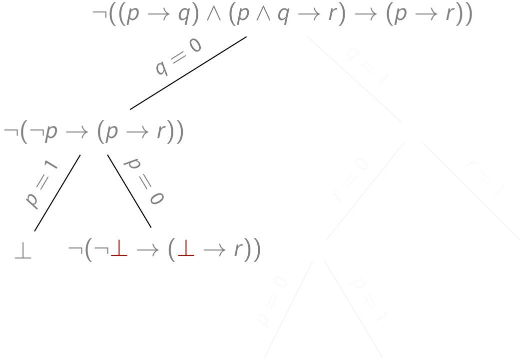
$\neg\top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

Splitting algorithm, example



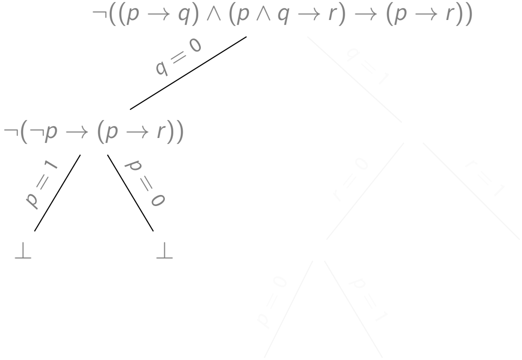
$\neg\top$	$\Rightarrow \bot$
$\top \land A$	$\Rightarrow A$
$\top \lor A$	$\Rightarrow \top$
$A \to \top$	$\Rightarrow \top$
$\top \to A$	$\Rightarrow A$
$A \leftrightarrow \top$	$\Rightarrow A$
$\top \leftrightarrow A$	$\Rightarrow A$
$\neg\bot$	$\Rightarrow \top$
$\bot \land A$	$\Rightarrow \bot$
$\bot \lor A$	$\Rightarrow A$
$A \to \bot$	$\Rightarrow \neg A$
$\bot \to A$	$\Rightarrow \top$
$A \leftrightarrow \bot$	$\Rightarrow \neg A$
$\bot \leftrightarrow A$	$\Rightarrow \neg A$

Splitting algorithm, example



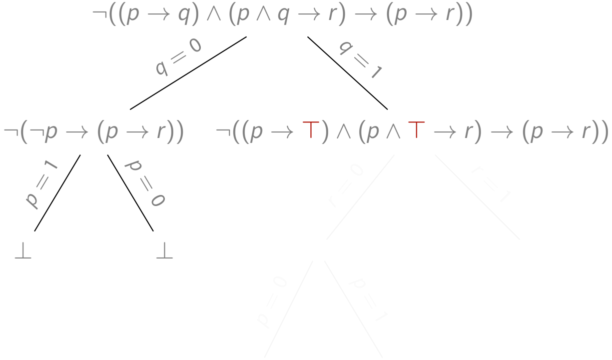
$\neg\top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

Splitting algorithm, example



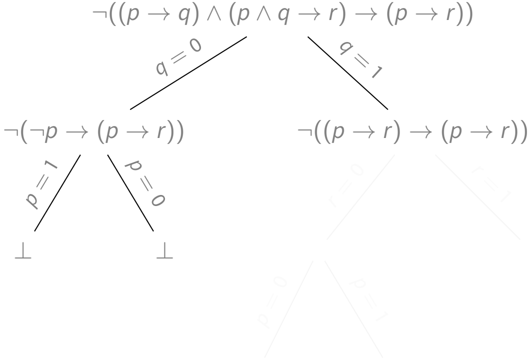
$\neg\top$	$\Rightarrow$	$\bot$
$\top \wedge A$	$\Rightarrow$	A
$\top \vee A$	$\Rightarrow$	$\top$
$A \rightarrow \top$	$\Rightarrow$	$\top$
$\top \rightarrow A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \wedge A$	$\Rightarrow$	$\bot$
$\bot \vee A$	$\Rightarrow$	A
$A \rightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \rightarrow A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

Splitting algorithm, example



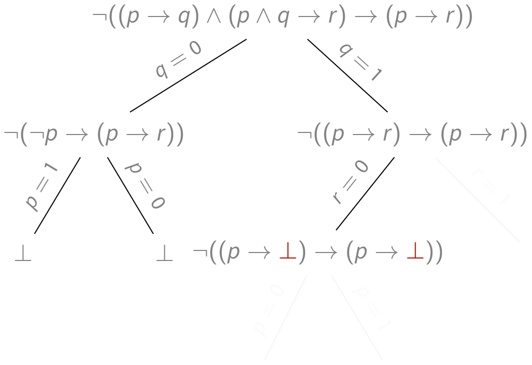
$$
\begin{array}{rcl}
\neg\top & \Rightarrow & \bot \\
\top \land A & \Rightarrow & A \\
\top \lor A & \Rightarrow & \top \\
A \to \top & \Rightarrow & \top \\
\top \to A & \Rightarrow & A \\
A \leftrightarrow \top & \Rightarrow & A \\
\top \leftrightarrow A & \Rightarrow & A \\
\hline
\neg\bot & \Rightarrow & \top \\
\bot \land A & \Rightarrow & \bot \\
\bot \lor A & \Rightarrow & A \\
A \to \bot & \Rightarrow & \neg A \\
\bot \to A & \Rightarrow & \top \\
A \leftrightarrow \bot & \Rightarrow & \neg A \\
\bot \leftrightarrow A & \Rightarrow & \neg A
\end{array}
$$

Splitting algorithm, example



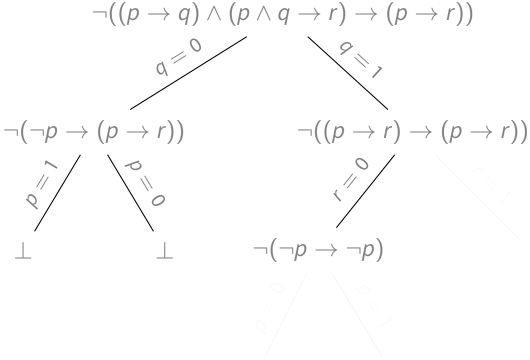
$\neg\top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

Splitting algorithm, example



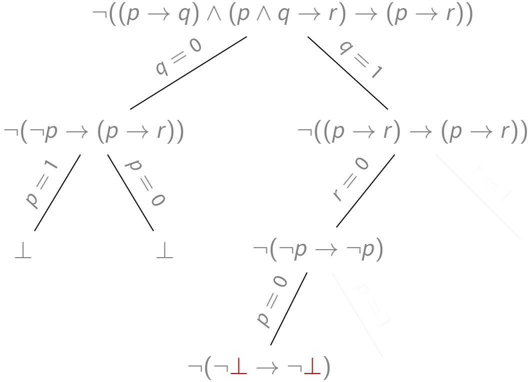
$\neg\top$	$\Rightarrow$	$\bot$
$\top \wedge A$	$\Rightarrow$	A
$\top \vee A$	$\Rightarrow$	$\top$
$A \rightarrow \top$	$\Rightarrow$	$\top$
$\top \rightarrow A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \wedge A$	$\Rightarrow$	$\bot$
$\bot \vee A$	$\Rightarrow$	A
$A \rightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \rightarrow A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

Splitting algorithm, example



$\neg\top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

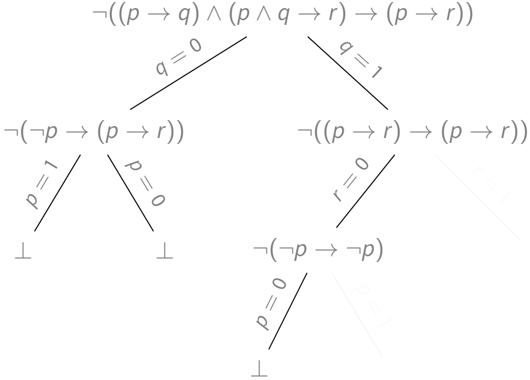
Splitting algorithm, example



$$\neg\top \Rightarrow \bot$$
$$\top \wedge A \Rightarrow A$$
$$\top \vee A \Rightarrow \top$$
$$A \rightarrow \top \Rightarrow \top$$
$$\top \rightarrow A \Rightarrow A$$
$$A \leftrightarrow \top \Rightarrow A$$
$$\top \leftrightarrow A \Rightarrow A$$

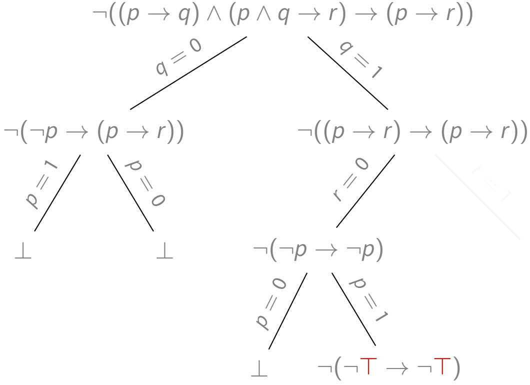
$$\neg\bot \Rightarrow \top$$
$$\bot \wedge A \Rightarrow \bot$$
$$\bot \vee A \Rightarrow A$$
$$A \rightarrow \bot \Rightarrow \neg A$$
$$\bot \rightarrow A \Rightarrow \top$$
$$A \leftrightarrow \bot \Rightarrow \neg A$$
$$\bot \leftrightarrow A \Rightarrow \neg A$$

Splitting algorithm, example



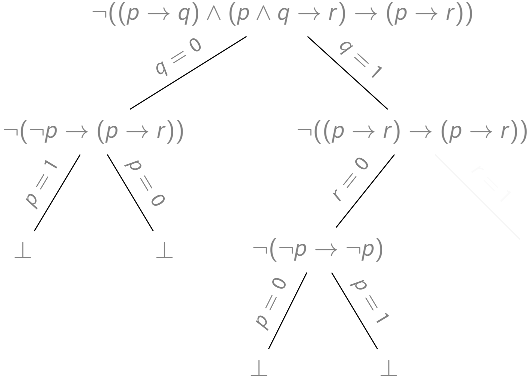
$\neg\top$	$\Rightarrow \bot$
$\top \wedge A$	$\Rightarrow A$
$\top \vee A$	$\Rightarrow \top$
$A \to \top$	$\Rightarrow \top$
$\top \to A$	$\Rightarrow A$
$A \leftrightarrow \top$	$\Rightarrow A$
$\top \leftrightarrow A$	$\Rightarrow A$
$\neg\bot$	$\Rightarrow \top$
$\bot \wedge A$	$\Rightarrow \bot$
$\bot \vee A$	$\Rightarrow A$
$A \to \bot$	$\Rightarrow \neg A$
$\bot \to A$	$\Rightarrow \top$
$A \leftrightarrow \bot$	$\Rightarrow \neg A$
$\bot \leftrightarrow A$	$\Rightarrow \neg A$

Splitting algorithm, example



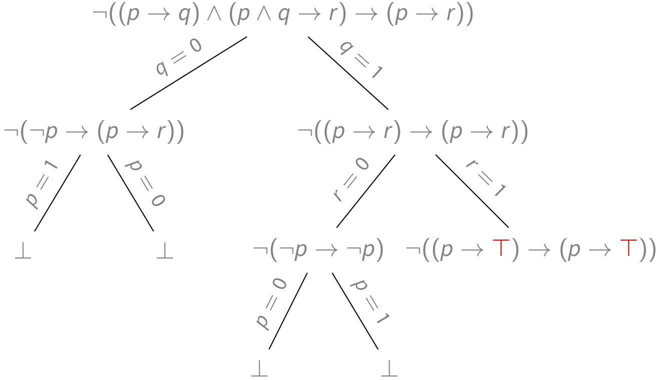
$\neg\top$	$\Rightarrow$	$\bot$
$\top \wedge A$	$\Rightarrow$	A
$\top \vee A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \wedge A$	$\Rightarrow$	$\bot$
$\bot \vee A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

Splitting algorithm, example



$\neg\top$	$\Rightarrow \bot$
$\top \land A$	$\Rightarrow A$
$\top \lor A$	$\Rightarrow \top$
$A \to \top$	$\Rightarrow \top$
$\top \to A$	$\Rightarrow A$
$A \leftrightarrow \top$	$\Rightarrow A$
$\top \leftrightarrow A$	$\Rightarrow A$
$\neg\bot$	$\Rightarrow \top$
$\bot \land A$	$\Rightarrow \bot$
$\bot \lor A$	$\Rightarrow A$
$A \to \bot$	$\Rightarrow \neg A$
$\bot \to A$	$\Rightarrow \top$
$A \leftrightarrow \bot$	$\Rightarrow \neg A$
$\bot \leftrightarrow A$	$\Rightarrow \neg A$

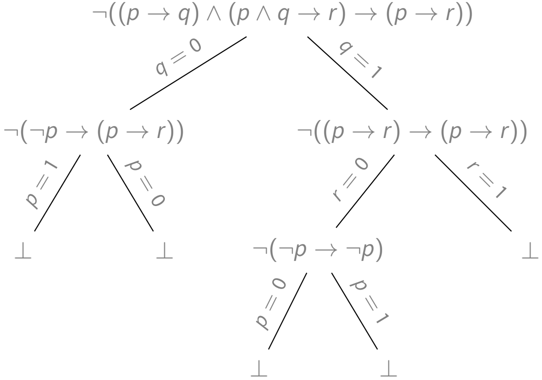
Splitting algorithm, example



$$\begin{array}{rcl}
\neg\top & \Rightarrow & \bot \\
\top \land A & \Rightarrow & A \\
\top \lor A & \Rightarrow & \top \\
A \to \top & \Rightarrow & \top \\
\top \to A & \Rightarrow & A \\
A \leftrightarrow \top & \Rightarrow & A \\
\top \leftrightarrow A & \Rightarrow & A
\end{array}$$

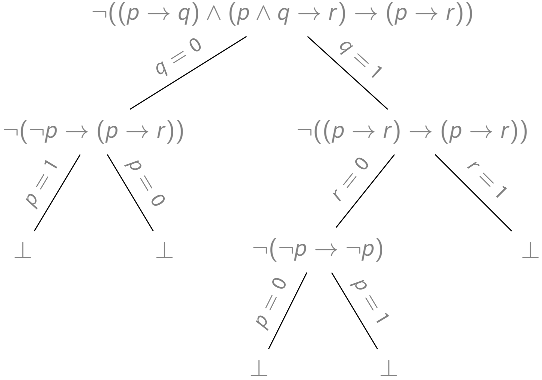
$$\begin{array}{rcl}
\neg\bot & \Rightarrow & \top \\
\bot \land A & \Rightarrow & \bot \\
\bot \lor A & \Rightarrow & A \\
A \to \bot & \Rightarrow & \neg A \\
\bot \to A & \Rightarrow & \top \\
A \leftrightarrow \bot & \Rightarrow & \neg A \\
\bot \leftrightarrow A & \Rightarrow & \neg A
\end{array}$$

Splitting algorithm, example



$$
\begin{array}{rcl}
\neg\top & \Rightarrow & \bot \\
\top \land A & \Rightarrow & A \\
\top \lor A & \Rightarrow & \top \\
A \rightarrow \top & \Rightarrow & \top \\
\top \rightarrow A & \Rightarrow & A \\
A \leftrightarrow \top & \Rightarrow & A \\
\top \leftrightarrow A & \Rightarrow & A \\
\hline
\neg\bot & \Rightarrow & \top \\
\bot \land A & \Rightarrow & \bot \\
\bot \lor A & \Rightarrow & A \\
A \rightarrow \bot & \Rightarrow & \neg A \\
\bot \rightarrow A & \Rightarrow & \top \\
A \leftrightarrow \bot & \Rightarrow & \neg A \\
\bot \leftrightarrow A & \Rightarrow & \neg A
\end{array}
$$

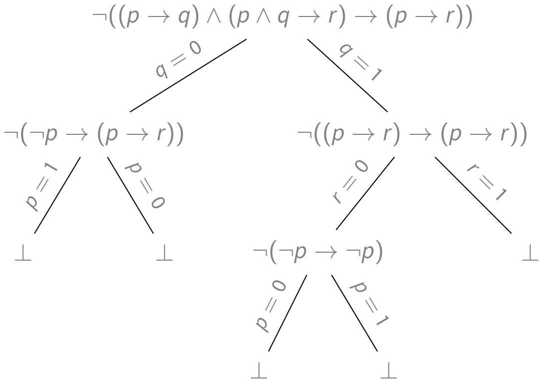
Splitting algorithm, example



$\neg\top$	$\Rightarrow$	$\bot$
$\top \wedge A$	$\Rightarrow$	A
$\top \vee A$	$\Rightarrow$	$\top$
$A \rightarrow \top$	$\Rightarrow$	$\top$
$\top \rightarrow A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \wedge A$	$\Rightarrow$	$\bot$
$\bot \vee A$	$\Rightarrow$	A
$A \rightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \rightarrow A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

The formula is unsatisfiable

Splitting algorithm, example



$$
\begin{array}{rcl}
\neg\top & \Rightarrow & \bot \\
\top \wedge A & \Rightarrow & A \\
\top \vee A & \Rightarrow & \top \\
A \rightarrow \top & \Rightarrow & \top \\
\top \rightarrow A & \Rightarrow & A \\
A \leftrightarrow \top & \Rightarrow & A \\
\top \leftrightarrow A & \Rightarrow & A \\
\hline
\neg\bot & \Rightarrow & \top \\
\bot \wedge A & \Rightarrow & \bot \\
\bot \vee A & \Rightarrow & A \\
A \rightarrow \bot & \Rightarrow & \neg A \\
\bot \rightarrow A & \Rightarrow & \top \\
A \leftrightarrow \bot & \Rightarrow & \neg A \\
\bot \leftrightarrow A & \Rightarrow & \neg A
\end{array}
$$

The formula is unsatisfiable

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

Exercise

1. For each unsimplified node of the tree in the previous slide, simplify the formula one step at a time by applying in each step one of the simplification rules in the slide.

   Apply the rules modulo commutativity of $\wedge$, $\vee$ and $\leftrightarrow$. For instance, consider the rule $\top \wedge A \;\Rightarrow\; A$ as also standing for the rule $A \wedge \top \;\Rightarrow\; A$.

2. Verify that the formula you obtain in each case corresponds to the simplified formula provided in the previous slide.

Splitting algorithm, example 2

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

$\neg\top \Rightarrow \bot$	$\neg\bot \Rightarrow \top$
$\top \land A \Rightarrow A$	$\bot \land A \Rightarrow \bot$
$\top \lor A \Rightarrow \top$	$\bot \lor A \Rightarrow A$
$A \to \top \Rightarrow \top$	$A \to \bot \Rightarrow \neg A$
$\top \to A \Rightarrow A$	$\bot \to A \Rightarrow \top$
$A \leftrightarrow \top \Rightarrow A$	$A \leftrightarrow \bot \Rightarrow \neg A$
$\top \leftrightarrow A \Rightarrow A$	$\bot \leftrightarrow A \Rightarrow \neg A$

The formula is satisfiable

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

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

Splitting algorithm, example 2

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

$p = 0$

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

$r = 0$

$\neg\top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \rightarrow \top$	$\Rightarrow$	$\top$
$\top \rightarrow A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \rightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \rightarrow A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

The formula is satisfiable

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

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

Splitting algorithm, example 2

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

$p = 0$

$$\neg r$$

$r = 0$

$\neg\top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

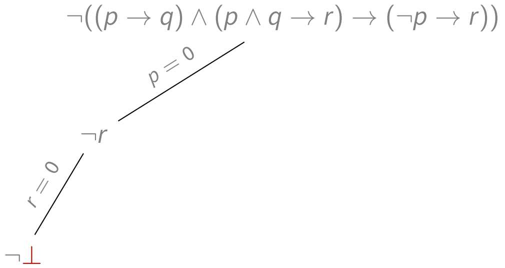
The formula is satisfiable

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

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

Splitting algorithm, example 2

$\neg\top$	$\Rightarrow$	$\bot$
$\top \wedge A$	$\Rightarrow$	A
$\top \vee A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \wedge A$	$\Rightarrow$	$\bot$
$\bot \vee A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$



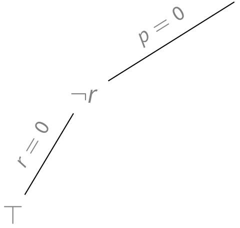
The formula is satisfiable

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

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

Splitting algorithm, example 2

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



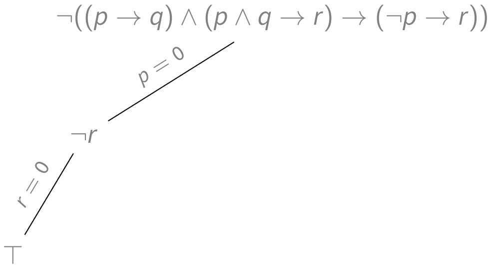
$\neg\top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

The formula is satisfiable

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

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

Splitting algorithm, example 2



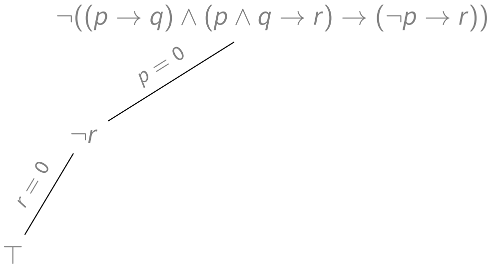
$\neg\top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

The formula is satisfiable

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

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

Splitting algorithm, example 2



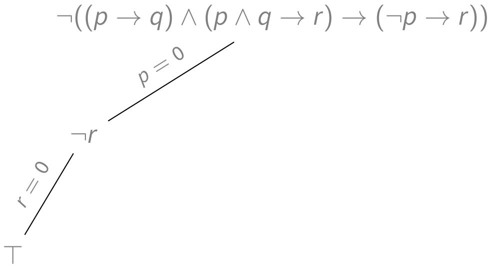
$\neg\top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

The formula is satisfiable

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

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

Splitting algorithm, example 2



$\neg\top$	$\Rightarrow \bot$
$\top \wedge A$	$\Rightarrow A$
$\top \vee A$	$\Rightarrow \top$
$A \rightarrow \top$	$\Rightarrow \top$
$\top \rightarrow A$	$\Rightarrow A$
$A \leftrightarrow \top$	$\Rightarrow A$
$\top \leftrightarrow A$	$\Rightarrow A$
$\neg\bot$	$\Rightarrow \top$
$\bot \wedge A$	$\Rightarrow \bot$
$\bot \vee A$	$\Rightarrow A$
$A \rightarrow \bot$	$\Rightarrow \neg A$
$\bot \rightarrow A$	$\Rightarrow \top$
$A \leftrightarrow \bot$	$\Rightarrow \neg A$
$\bot \leftrightarrow A$	$\Rightarrow \neg A$

The formula is satisfiable

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

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

Improving the search for satisfying assignments

The order in which one chooses

1. the variable to replace and
2. the truth value for the chosen variable

is essential for the efficiency of the splitting algorithm

In certain cases, Choice (2) can be done *deterministically* (without having to try the other alternative)

We will see the case of pure literals

Improving the search for satisfying assignments

The order in which one chooses

1. the variable to replace and
2. the truth value for the chosen variable

is essential for the efficiency of the splitting algorithm

In certain cases, Choice (2) can be done *deterministically* (without having to try the other alternative)

We will see the case of pure literals

Improving the search for satisfying assignments

The order in which one chooses

1. the variable to replace and
2. the truth value for the chosen variable

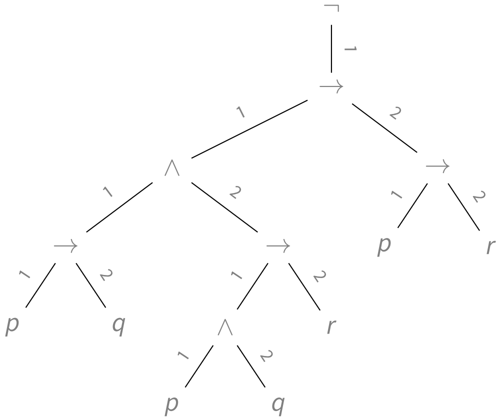
is essential for the efficiency of the splitting algorithm

In certain cases, Choice (2) can be done *deterministically* (without having to try the other alternative)

We will see the case of pure literals

Parse tree

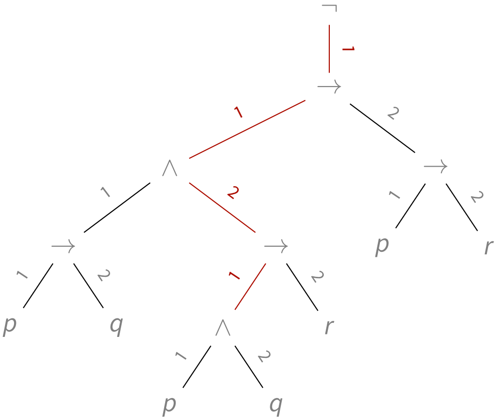
$$A = \neg((p \to q) \land (p \land q \to r) \to (p \to r))$$



Position in formula A: 1.1.2.1 Subformula of A at this position: $p \land q$

Parse tree

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

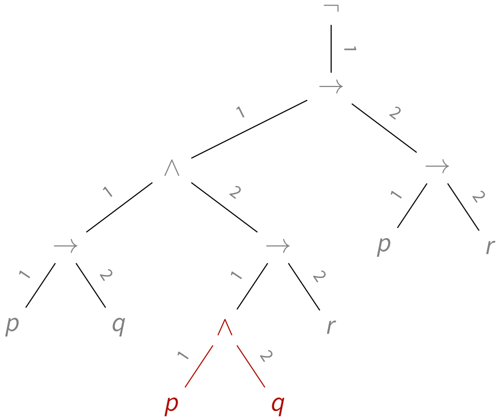


Position in formula A: 1.1.2.1

Subformula of A at this position: $p \land q$

Parse tree

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



Position in formula A: 1.1.2.1 Subformula of A 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 by $A|_\pi$

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 $B_1 \wedge \cdots \wedge B_n$ or $B_1 \vee \cdots \vee B_n$, then for all $i \in \{1, \ldots, n\}$ the position $\pi.i$ is a position in A and $A|_{\pi.i} \stackrel{\text{def}}{=} B_i$
   - 2.2 If B has the form $\neg B_1$, then $\pi.1$ is a position in A and $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$
   - 2.3 If B has the form $B_1 \rightarrow B_2$, then $\pi.1$ and $\pi.2$ are positions in A and $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$ and $A|_{\pi.2} \stackrel{\text{def}}{=} B_2$
   - 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 position π*

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 by $A|_\pi$

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 $B_1 \wedge \cdots \wedge B_n$ or $B_1 \vee \cdots \vee B_n$, then for all $i \in \{1, \ldots, n\}$ the position $\pi.i$ is a position in A and $A|_{\pi.i} \stackrel{\text{def}}{=} B_i$
   - 2.2 If B has the form $\neg B_1$, then $\pi.1$ is a position in A and $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$
   - 2.3 If B has the form $B_1 \rightarrow B_2$, then $\pi.1$ and $\pi.2$ are positions in A and $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$ and $A|_{\pi.2} \stackrel{\text{def}}{=} B_2$
   - 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 position* π

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 by $A|_\pi$

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 $B_1 \wedge \cdots \wedge B_n$ or $B_1 \vee \cdots \vee B_n$, then for all $i \in \{1, \ldots, n\}$ the position $\pi.i$ is a position in A and $A|_{\pi.i} \stackrel{\text{def}}{=} B_i$
   - 2.2 If B has the form $\neg B_1$, then $\pi.1$ is a position in A and $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$
   - 2.3 If B has the form $B_1 \rightarrow B_2$, then $\pi.1$ and $\pi.2$ are positions in A and $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$ and $A|_{\pi.2} \stackrel{\text{def}}{=} B_2$
   - 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 position* π

Polarity

Polarity of subformula at a position **Notation:** $pol(A, \pi)$ **Values:** $\{-1, 0, 1\}$

1. For every formula A, ϵ is a position in A and $A|_\epsilon \stackrel{\text{def}}{=} A$ and $pol(A, \epsilon) \stackrel{\text{def}}{=} 1$
2. Let $A|_\pi = B$

    2.1 If B has the form $B_1 \wedge \cdots \wedge B_n$ or $B_1 \vee \cdots \vee B_n$, then for all $i \in \{1, \ldots, n\}$ the position $\pi.i$ is a position in A and $A|_{\pi.i} \stackrel{\text{def}}{=} B_i$, 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 and $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$ 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 $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$ and $A|_{\pi.2} \stackrel{\text{def}}{=} B_2$, $pol(A, \pi.1) \stackrel{\text{def}}{=} -pol(A, \pi)$, $pol(A, \pi.2) \stackrel{\text{def}}{=} pol(A, \pi)$

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

Polarity

Polarity of subformula at a position **Notation:** $pol(A, \pi)$ **Values:** $\{-1, 0, 1\}$

1. For every formula A, ϵ is a position in A and $A|_{\epsilon} \stackrel{\text{def}}{=} A$ and $pol(A, \epsilon) \stackrel{\text{def}}{=} 1$
2. Let $A|_{\pi} = B$
   - 2.1 If B has the form $B_1 \wedge \cdots \wedge B_n$ or $B_1 \vee \cdots \vee B_n$, then for all $i \in \{1, \ldots, n\}$ the position $\pi.i$ is a position in A and $A|_{\pi.i} \stackrel{\text{def}}{=} B_i$, 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 and $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$ 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 $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$ and $A|_{\pi.2} \stackrel{\text{def}}{=} B_2$, $pol(A, \pi.1) \stackrel{\text{def}}{=} -pol(A, \pi)$, $pol(A, \pi.2) \stackrel{\text{def}}{=} pol(A, \pi)$
   - 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 $A|_{\pi} = B$, the occurrence of B at position π in A is *positive*
- If $pol(A, \pi) = -1$ and $A|_{\pi} = B$, the occurrence of B at position π in A is *negative*

Polarity

Polarity of subformula at a position **Notation:** $pol(A, \pi)$ **Values:** $\{-1, 0, 1\}$

1. For every formula A, ϵ is a position in A and $A|_\epsilon \stackrel{\text{def}}{=} A$ and $pol(A, \epsilon) \stackrel{\text{def}}{=} 1$
2. Let $A|_\pi = B$
   2.1 If B has the form $B_1 \wedge \cdots \wedge B_n$ or $B_1 \vee \cdots \vee B_n$, then for all $i \in \{1, \ldots, n\}$ the position $\pi.i$ is a position in A and $A|_{\pi.i} \stackrel{\text{def}}{=} B_i$, 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 and $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$ 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 $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$ and $A|_{\pi.2} \stackrel{\text{def}}{=} B_2$, $pol(A, \pi.1) \stackrel{\text{def}}{=} -pol(A, \pi)$, $pol(A, \pi.2) \stackrel{\text{def}}{=} pol(A, \pi)$
   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 $A|_\pi = B$, the occurrence of B at position π in A is *positive*
- If $pol(A, \pi) = -1$ and $A|_\pi = B$, the occurrence of B at position π in A is *negative*

Polarity

Polarity of subformula at a position **Notation:** $pol(A, \pi)$ **Values:** $\{-1, 0, 1\}$

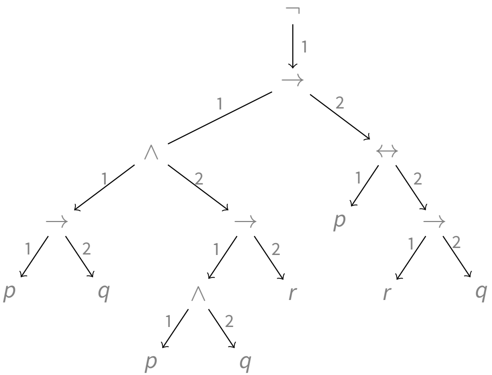
1. For every formula A, ϵ is a position in A and $A|_\epsilon \stackrel{\text{def}}{=} A$ and $pol(A, \epsilon) \stackrel{\text{def}}{=} 1$
2. Let $A|_\pi = B$
   - 2.1 If B has the form $B_1 \wedge \cdots \wedge B_n$ or $B_1 \vee \cdots \vee B_n$, then for all $i \in \{1, \ldots, n\}$ the position $\pi.i$ is a position in A and $A|_{\pi.i} \stackrel{\text{def}}{=} B_i$, 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 and $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$ 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 $A|_{\pi.1} \stackrel{\text{def}}{=} B_1$ and $A|_{\pi.2} \stackrel{\text{def}}{=} B_2$, $pol(A, \pi.1) \stackrel{\text{def}}{=} -pol(A, \pi)$, $pol(A, \pi.2) \stackrel{\text{def}}{=} pol(A, \pi)$
   - 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 $A|_\pi = B$, the occurrence of B at position π in A is *positive*
- If $pol(A, \pi) = -1$ and $A|_\pi = B$, the occurrence of B at position π in A is *negative*

The coloring algorithm for determining polarity

$A = \neg((p \to q) \land (p \land q \to r) \to (p \leftrightarrow (r \to q)))$

- Color in blue all arcs below an equivalence
- Color in red all uncolored arcs exiting a negation or the left-hand side of an implication

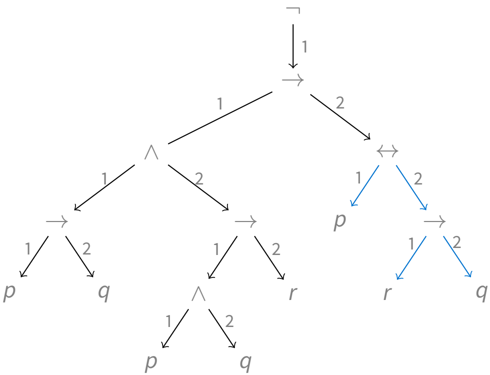


- 0 if it has at least one blue arc above it
- -1 if it has no blue arc and an odd number of red arcs above it
- 1 otherwise

The coloring algorithm for determining polarity

$A = \neg((p \to q) \wedge (p \wedge q \to r) \to (p \leftrightarrow (r \to q)))$

- Color in blue all arcs below an equivalence
- Color in red all uncolored arcs exiting a negation or the left-hand side of an implication

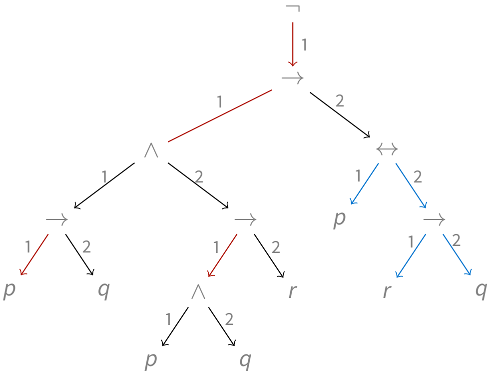


- 0 if it has at least one blue arc above it
- −1 if it has no blue arc and an odd number of red arcs above it
- 1 otherwise

The coloring algorithm for determining polarity

$A = \neg((p \to q) \wedge (p \wedge q \to r) \to (p \leftrightarrow (r \to q)))$

- Color in blue all arcs below an equivalence
- Color in red all uncolored arcs exiting a negation or the left-hand side of an implication

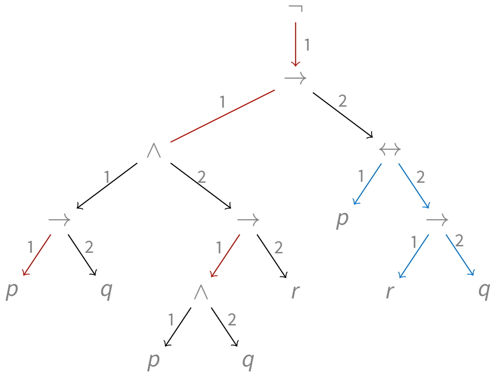


- 0 if it has at least one blue arc above it
- -1 if it has no blue arc and an odd number of red arcs above it
- 1 otherwise

The coloring algorithm for determining polarity

$A = \neg((p \to q) \land (p \land q \to r) \to (p \leftrightarrow (r \to q)))$

- Color in blue all arcs below an equivalence
- Color in red all uncolored arcs exiting a negation or the left-hand side of an implication



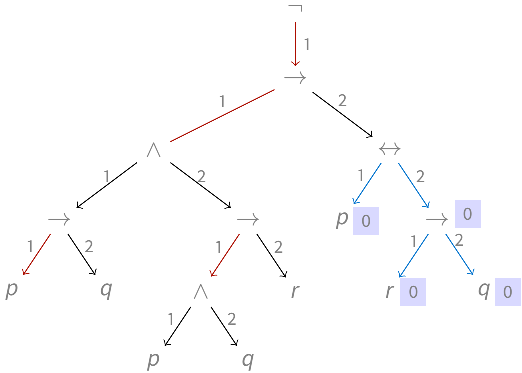
The polarity of a position is

- 0 if it has at least one blue arc above it
- -1 if it has no blue arc and an odd number of red arcs above it
- 1 otherwise

The coloring algorithm for determining polarity

$A = \neg((p \to q) \land (p \land q \to r) \to (p \leftrightarrow (r \to q)))$

- Color in blue all arcs below an equivalence
- Color in red all uncolored arcs exiting a negation or the left-hand side of an implication



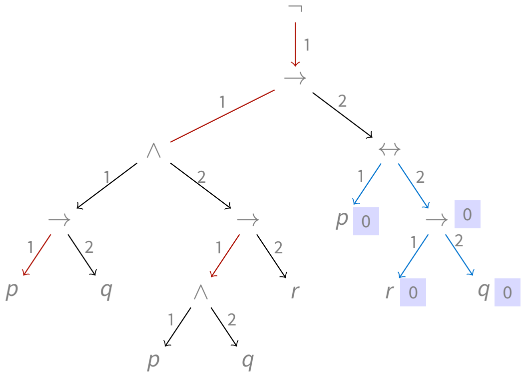
The polarity of a position is

- 0 if it has at least one blue arc above it
- −1 if it has no blue arc and an odd number of red arcs above it
- 1 otherwise

The coloring algorithm for determining polarity

$A = \neg((p \to q) \wedge (p \wedge q \to r) \to (p \leftrightarrow (r \to q)))$

- Color in blue all arcs below an equivalence
- Color in red all uncolored arcs exiting a negation or the left-hand side of an implication



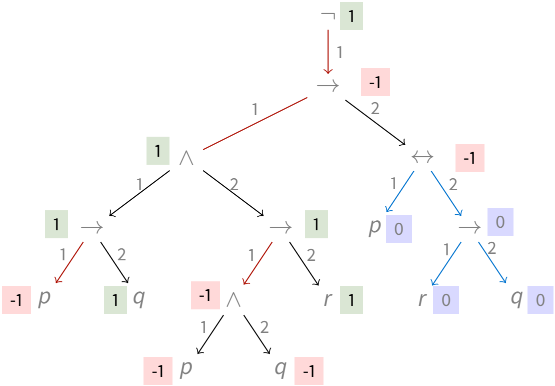
The polarity of a position is

- 0 if it has at least one blue arc above it
- −1 if it has no blue arc and an odd number of red arcs above it
- 1 otherwise

The coloring algorithm for determining polarity

$A = \neg((p \to q) \wedge (p \wedge q \to r) \to (p \leftrightarrow (r \to q)))$

- Color in blue all arcs below an equivalence
- Color in red all uncolored arcs exiting a negation or the left-hand side of an implication



The polarity of a position is

- 0 if it has at least one blue arc above it
- −1 if it has no blue arc and an odd number of red arcs above it
- 1 otherwise

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 $A[B]_\pi$ denotes, indifferently:

- A formula A having subformula B at position π
- The result of replacing the subformula of A at position π by B

Lemma 2 (Monotonic Replacement)

Let A, B, B' be formulas, $\mathcal{I}$ be an interpretation such that $\mathcal{I} \models B \rightarrow B'$.

1. *If $pol(A, \pi) = 1$, then $\mathcal{I} \models A[B]_\pi \rightarrow A[B']_\pi$.*
2. *If $pol(A, \pi) = -1$, then $\mathcal{I} \models A[B']_\pi \rightarrow A[B]_\pi$.*

Monotonic replacement

Notation $A[B]_\pi$ denotes, indifferently:

- A formula A having subformula B at position π
- The result of replacing the subformula of A at position π by B

Lemma 2 (Monotonic Replacement)

Let A, B, B' *be formulas,* $\mathcal{I}$ *be an interpretation such that* $\mathcal{I} \models B \rightarrow B'$.

1. *If* $pol(A, \pi) = 1$, *then* $\mathcal{I} \models A[B]_\pi \rightarrow A[B']_\pi$.
2. *If* $pol(A, \pi) = -1$, *then* $\mathcal{I} \models A[B']_\pi \rightarrow A[B]_\pi$.

Monotonic replacement

Theorem 3 (Monotonic Replacement)

Let A, B, B' be formulas such that $\models B \rightarrow B'$. Let A^-, resp. A^+, be the formula obtained from A by replacing one or more negative, resp. positive, occurrences of B by B'. Then,

$$\models A^- \rightarrow A \quad \text{and} \quad \models A \rightarrow A^+ .$$

Corollary 4

Let A, B, B', A^-, A^+ be as above. Then, the following holds.

1. *If A^- is satisfiable, so is A.*
2. *If A^+ is unsatisfiable, so is A.*

Monotonic replacement

Theorem 3 (Monotonic Replacement)

Let A, B, B' be formulas such that $\models B \rightarrow B'$. Let A^-, resp. A^+, be the formula obtained from A by replacing one or more negative, resp. positive, occurrences of B by B'. Then,

$$\models A^- \rightarrow A \quad \text{and} \quad \models A \rightarrow A^+ .$$

Corollary 4

Let A, B, B', A^-, A^+ be as above. Then, the following holds.

1. *If A^- is satisfiable, so is A.*
2. *If A^+ is unsatisfiable, so is A.*

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

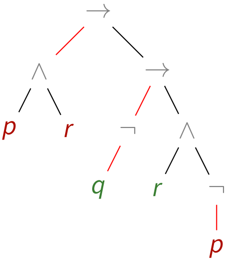
$$p \wedge r \to (\neg q \to (r \wedge \neg p))$$

- 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

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

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

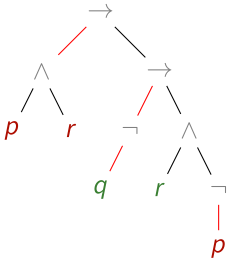


- Both occurrences of p are negative, so p is pure
- The only occurrence of q is positive, so q is pure
- r is not pure, since it has both negative and positive occurrences

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
- 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 5 (Pure Atom)

Suppose variable p has only positive occurrences in A and $\mathcal{I} \models A$. Define

$$\mathcal{I}' \stackrel{\text{def}}{=} \mathcal{I} + (p \mapsto 1) \quad \textit{(maps } p \textit{ to 1 and is otherwise identical to } \mathcal{I}\textit{)}$$

Then $\mathcal{I}' \models A$.

Dually, Suppose p has only negative occurrences in A and $\mathcal{I} \models A$. Define

$$\mathcal{I}' \stackrel{\text{def}}{=} \mathcal{I} + (p \mapsto 0) \quad \textit{(maps } p \textit{ to 0 and is otherwise identical to } \mathcal{I}\textit{)}$$

Then $\mathcal{I}' \models A$.

Theorem 6 (Pure Atom)

Suppose variable p has only positive (respectively, only negative) occurrences in A. Then A is satisfiable iff so is $A_p^{\top}$ (respectively, $A_p^{\perp}$).

Properties of pure atoms

Lemma 5 (Pure Atom)

Suppose variable p has only positive occurrences in A and $\mathcal{I} \models A$. Define

$$\mathcal{I}' \stackrel{\text{def}}{=} \mathcal{I} + (p \mapsto 1) \quad \textit{(maps p to 1 and is otherwise identical to } \mathcal{I}\textit{)}$$

Then $\mathcal{I}' \models A$.

Dually, Suppose p has only negative occurrences in A and $\mathcal{I} \models A$. Define

$$\mathcal{I}' \stackrel{\text{def}}{=} \mathcal{I} + (p \mapsto 0) \quad \textit{(maps p to 0 and is otherwise identical to } \mathcal{I}\textit{)}$$

Then $\mathcal{I}' \models A$.

Theorem 6 (Pure Atom)

Suppose variable p has only positive (respectively, only negative) occurrences in A. Then A is satisfiable iff so is $A_p^\top$ (respectively, $A_p^\bot$).

Properties of pure atoms

Lemma 5 (Pure Atom)

Suppose variable p has only positive occurrences in A and $\mathcal{I} \models A$. Define

$$\mathcal{I}' \stackrel{\text{def}}{=} \mathcal{I} + (p \mapsto 1) \quad \text{(maps } p \text{ to 1 and is otherwise identical to } \mathcal{I}\text{)}$$

Then $\mathcal{I}' \models A$.

Dually, Suppose p has only negative occurrences in A and $\mathcal{I} \models A$. Define

$$\mathcal{I}' \stackrel{\text{def}}{=} \mathcal{I} + (p \mapsto 0) \quad \text{(maps } p \text{ to 0 and is otherwise identical to } \mathcal{I}\text{)}$$

Then $\mathcal{I}' \models A$.

Theorem 6 (Pure Atom)

Suppose variable p has only positive (respectively, only negative) occurrences in A. Then A is satisfiable iff so is $A_p^\top$ (respectively, $A_p^\perp$).

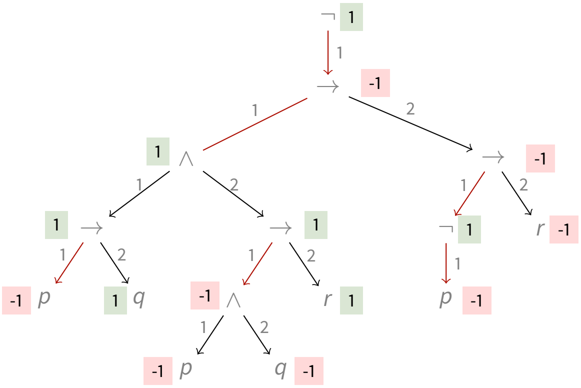
Pure atom, example

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

All occurrences of p are negative, so to check for satisfiability we can replace p by $\bot$

Pure atom, example

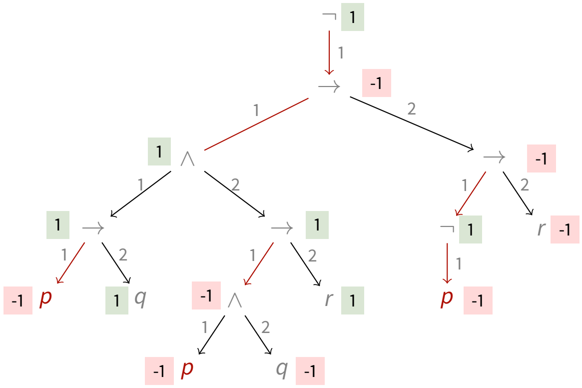
$$\neg((p \rightarrow q) \wedge (p \wedge q \rightarrow r) \rightarrow (\neg p \rightarrow r))$$



All occurrences of p are negative, so to check for satisfiability we can replace p by $\bot$

Pure atom, example

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



All occurrences of p are negative, so to check for satisfiability we can replace p by $\bot$

Example, continued

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

$\neg\top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

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

Example, continued

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

$\neg\top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

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

Example, continued

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

$\neg \top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg \bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

Example, continued

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

$\neg \top$	$\Rightarrow \bot$
$\top \land A$	$\Rightarrow A$
$\top \lor A$	$\Rightarrow \top$
$A \to \top$	$\Rightarrow \top$
$\top \to A$	$\Rightarrow A$
$A \leftrightarrow \top$	$\Rightarrow A$
$\top \leftrightarrow A$	$\Rightarrow A$
$\neg \bot$	$\Rightarrow \top$
$\bot \land A$	$\Rightarrow \bot$
$\bot \lor A$	$\Rightarrow A$
$A \to \bot$	$\Rightarrow \neg A$
$\bot \to A$	$\Rightarrow \top$
$A \leftrightarrow \bot$	$\Rightarrow \neg A$
$\bot \leftrightarrow A$	$\Rightarrow \neg A$

Example, continued

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

$$
\begin{array}{rcl}
\neg \top & \Rightarrow & \bot \\
\top \land A & \Rightarrow & A \\
\top \lor A & \Rightarrow & \top \\
A \to \top & \Rightarrow & \top \\
\top \to A & \Rightarrow & A \\
A \leftrightarrow \top & \Rightarrow & A \\
\top \leftrightarrow A & \Rightarrow & A \\
\hline
\neg \bot & \Rightarrow & \top \\
\bot \land A & \Rightarrow & \bot \\
\bot \lor A & \Rightarrow & A \\
A \to \bot & \Rightarrow & \neg A \\
\bot \to A & \Rightarrow & \top \\
A \leftrightarrow \bot & \Rightarrow & \neg A \\
\bot \leftrightarrow A & \Rightarrow & \neg A
\end{array}
$$

Example, continued

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

$\neg\top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

Example, continued

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

$\neg\top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

Example, continued

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

$\neg\top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

Example, continued

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

$\neg\top \Rightarrow \bot$	$\neg\bot \Rightarrow \top$
$\top \wedge A \Rightarrow A$	$\bot \wedge A \Rightarrow \bot$
$\top \vee A \Rightarrow \top$	$\bot \vee A \Rightarrow A$
$A \to \top \Rightarrow \top$	$A \to \bot \Rightarrow \neg A$
$\top \to A \Rightarrow A$	$\bot \to A \Rightarrow \top$
$A \leftrightarrow \top \Rightarrow A$	$A \leftrightarrow \bot \Rightarrow \neg A$
$\top \leftrightarrow A \Rightarrow A$	$\bot \leftrightarrow A \Rightarrow \neg A$

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

Example, continued

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

$\neg\top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg\bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

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

Example, continued

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

$\neg \top$	$\Rightarrow$	$\bot$
$\top \land A$	$\Rightarrow$	A
$\top \lor A$	$\Rightarrow$	$\top$
$A \to \top$	$\Rightarrow$	$\top$
$\top \to A$	$\Rightarrow$	A
$A \leftrightarrow \top$	$\Rightarrow$	A
$\top \leftrightarrow A$	$\Rightarrow$	A
$\neg \bot$	$\Rightarrow$	$\top$
$\bot \land A$	$\Rightarrow$	$\bot$
$\bot \lor A$	$\Rightarrow$	A
$A \to \bot$	$\Rightarrow$	$\neg A$
$\bot \to A$	$\Rightarrow$	$\top$
$A \leftrightarrow \bot$	$\Rightarrow$	$\neg A$
$\bot \leftrightarrow A$	$\Rightarrow$	$\neg A$

We have shown the satisfiability of this formula deterministically (no guesses), using only the pure atom rule