

IN3070/4070 – Logic – Autumn 2022

Lecture 10: DPLL

Martin Giese

25th October 2022



DEPARTMENT OF
INFORMATICS



UNIVERSITY OF
OSLO

Today's Plan

- ▶ Motivation
- ▶ Simplification Rules
- ▶ Atomic Cut
- ▶ The DPLL Algorithm
- ▶ Other Tricks

Outline

- ▶ Motivation
- ▶ Simplification Rules
- ▶ Atomic Cut
- ▶ The DPLL Algorithm
- ▶ Other Tricks

Smullyan's categories: $\alpha/\beta/\gamma/\delta$

- ▶ Many similar cases in proofs and implementations

Smullyan's categories: $\alpha/\beta/\gamma/\delta$

- ▶ Many similar cases in proofs and implementations
- ▶ Categorise rules, rule applications, and formulae

Smullyan's categories: $\alpha/\beta/\gamma/\delta$

- ▶ Many similar cases in proofs and implementations
- ▶ Categorise rules, rule applications, and formulae

α -rules

Propositional, one branch, e.g.

$$\frac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \wedge B \Rightarrow \Delta} \wedge\text{-left}$$

Smullyan's categories: $\alpha/\beta/\gamma/\delta$

- ▶ Many similar cases in proofs and implementations
- ▶ Categorise rules, rule applications, and formulae

α -rules

Propositional, one branch, e.g.

$$\frac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \wedge B \Rightarrow \Delta} \wedge\text{-left}$$

β -rules

Propositional, splitting, e.g.

$$\frac{\Gamma \Rightarrow A, \Delta \quad \Gamma, B \Rightarrow \Delta}{\Gamma, A \rightarrow B \Rightarrow \Delta}$$

Smullyan's categories: $\alpha/\beta/\gamma/\delta$

- ▶ Many similar cases in proofs and implementations
- ▶ Categorise rules, rule applications, and formulae

α -rules

Propositional, one branch, e.g.

$$\frac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \wedge B \Rightarrow \Delta} \wedge\text{-left}$$

β -rules

Propositional, splitting, e.g.

$$\frac{\Gamma \Rightarrow A, \Delta \quad \Gamma, B \Rightarrow \Delta}{\Gamma, A \rightarrow B \Rightarrow \Delta}$$

γ -rules

Apply for all terms t , e.g.

$$\frac{\Gamma, A[x \setminus t], \forall x A \Rightarrow \Delta}{\Gamma, \forall x A \Rightarrow \Delta} \forall\text{-left}$$

Smullyan's categories: $\alpha/\beta/\gamma/\delta$

- ▶ Many similar cases in proofs and implementations
- ▶ Categorise rules, rule applications, and formulae

α -rules

Propositional, one branch, e.g.

$$\frac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \wedge B \Rightarrow \Delta} \wedge\text{-left}$$

β -rules

Propositional, splitting, e.g.

$$\frac{\Gamma \Rightarrow A, \Delta \quad \Gamma, B \Rightarrow \Delta}{\Gamma, A \rightarrow B \Rightarrow \Delta}$$

γ -rules

Apply for all terms t , e.g.

$$\frac{\Gamma, A[x \setminus t], \forall x A \Rightarrow \Delta}{\Gamma, \forall x A \Rightarrow \Delta} \forall\text{-left}$$

δ -rules

Introduce new constant c , e.g.

$$\frac{\Gamma, A[x \setminus c], \Rightarrow \Delta}{\Gamma, \exists x A \Rightarrow \Delta} \exists\text{-left}$$

A Naïve SAT Solver at Work

$$p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow$$

A Naïve SAT Solver at Work

$$p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow$$

A Naïve SAT Solver at Work

$$p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow$$

A Naïve SAT Solver at Work

 $p \Rightarrow$

 $p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \neg p, p \Rightarrow \quad \quad \quad q, p \Rightarrow \\
 \hline
 p \Rightarrow \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \overline{\neg p, p \Rightarrow} \qquad \qquad \qquad q, p \Rightarrow \\
 \hline
 p \Rightarrow \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \overline{\neg p, p \Rightarrow} \qquad \qquad \qquad q, p \Rightarrow \\
 \hline
 p \Rightarrow \\
 \hline
 p \vee q, \neg p \vee q, \textcolor{red}{p} \vee \textcolor{red}{\neg q}, \neg p \vee \neg q \Rightarrow
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{}{\neg p, p \Rightarrow} \quad \frac{\frac{p, q, p \Rightarrow}{q, p \Rightarrow} \quad \frac{\neg q, q, p \Rightarrow}{}}{p \Rightarrow} \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{}{\neg p, p \Rightarrow} \quad \frac{\frac{p, q, p \Rightarrow}{q, p \Rightarrow} \quad \frac{\neg q, q, p \Rightarrow}{}}{p \Rightarrow} \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{}{\neg p, p \Rightarrow} \quad \frac{\frac{\neg p, p, q, p \Rightarrow \quad \neg q, p, q, p \Rightarrow}{p, q, p \Rightarrow} \quad \neg q, q, p \Rightarrow}{q, p \Rightarrow} \\
 \hline
 p \Rightarrow \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{}{\neg p, p \Rightarrow} \quad \frac{\frac{\frac{}{\neg p, p, q, p \Rightarrow} \quad \frac{}{\neg q, p, q, p \Rightarrow}}{p, q, p \Rightarrow} \quad \frac{}{\neg q, q, p \Rightarrow}}{q, p \Rightarrow} \\
 \hline
 p \Rightarrow \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{}{\neg p, p \Rightarrow} \quad \frac{\frac{\frac{}{\neg p, p, q, p \Rightarrow} \quad \frac{\frac{}{\neg q, p, q, p \Rightarrow}}{p, q, p \Rightarrow}}{q, p \Rightarrow} \quad \frac{}{\neg q, q, p \Rightarrow}}{p \Rightarrow} \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{}{\neg p, p \Rightarrow} \quad \frac{\frac{\frac{}{\neg p, p, q, p \Rightarrow} \quad \frac{}{\neg q, p, q, p \Rightarrow}}{p, q, p \Rightarrow} \quad \frac{}{q, p \Rightarrow} \quad \frac{}{\neg q, q, p \Rightarrow}}{p \Rightarrow} \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

A Naïve SAT Solver at Work

$$\frac{}{\neg q, q, p \Rightarrow}$$

$$q \Rightarrow$$

$$\frac{}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}$$

A Naïve SAT Solver at Work

$$\frac{}{\neg q, q, p \Rightarrow}$$

$$q \Rightarrow$$

$$p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{\overline{\neg q, q, p} \Rightarrow}{ \Rightarrow} \qquad \frac{\neg p, q \Rightarrow}{ \Rightarrow} \qquad \frac{q \Rightarrow}{ \Rightarrow} \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{\overline{\neg q, q, p} \Rightarrow}{ \Rightarrow} \\
 \frac{ \Rightarrow \quad \frac{\neg p, q \Rightarrow}{ \Rightarrow} \quad q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{}{\neg q, q, p \Rightarrow} \quad \frac{p, \neg p, q \Rightarrow \quad \frac{\neg q, \neg p, q \Rightarrow}{\neg p, q \Rightarrow}}{\quad} \quad q \Rightarrow \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{\neg q, q, p \Rightarrow}{} \quad \frac{\frac{\frac{p, \neg p, q \Rightarrow}{\neg p, q \Rightarrow} \quad \neg q, \neg p, q \Rightarrow}{q \Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{}{\frac{\neg q, q, p \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}} \quad \frac{}{\frac{\frac{p, \neg p, q \Rightarrow}{\neg p, q \Rightarrow} \quad \frac{\neg q, \neg p, q \Rightarrow}{q \Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}}
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{\overline{\neg q, \neg p, q \Rightarrow}}{q \Rightarrow} \qquad \qquad \qquad q, q \Rightarrow \\
 \hline
 \frac{q \Rightarrow}{p \vee q, \neg p \vee q, \textcolor{red}{p} \vee \textcolor{red}{\neg q}, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{\overline{\neg q, \neg p, q \Rightarrow}}{q \Rightarrow} \qquad \frac{p, q, q \Rightarrow}{q, q \Rightarrow} \qquad \frac{\neg q, q, q \Rightarrow}{q, q \Rightarrow} \\
 \hline
 \frac{q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{\overline{\neg q, \neg p, q \Rightarrow}}{q \Rightarrow} \qquad \frac{p, q, q \Rightarrow}{q, q \Rightarrow} \qquad \frac{\neg q, q, q \Rightarrow}{q, q \Rightarrow} \\
 \hline
 \frac{q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{\frac{\neg q, \neg p, q \Rightarrow}{q \Rightarrow}}{\quad} \quad \frac{\frac{\neg p, p, q, q \Rightarrow}{p, q, q \Rightarrow} \quad \frac{\neg q, p, q, q \Rightarrow}{q, q \Rightarrow}}{\neg q, q, q \Rightarrow} \\
 \frac{q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{\frac{\neg q, \neg p, q \Rightarrow}{q \Rightarrow}}{\quad} \quad \frac{\frac{\frac{\neg p, p, q, q \Rightarrow}{p, q, q \Rightarrow} \quad \frac{\neg q, p, q, q \Rightarrow}{\neg q, q, q \Rightarrow}}{q, q \Rightarrow}}{q \Rightarrow} \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{\frac{\neg q, \neg p, q \Rightarrow}{q \Rightarrow}}{\quad} \quad \frac{\frac{\neg p, p, q, q \Rightarrow}{p, q, q \Rightarrow} \quad \frac{\frac{\neg q, p, \textcolor{red}{q}, q \Rightarrow}{q, q \Rightarrow}}{\neg q, q, q \Rightarrow}}{q \Rightarrow} \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

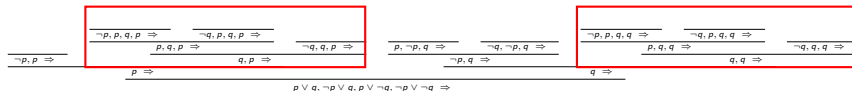
A Naïve SAT Solver at Work

$$\begin{array}{c}
 \frac{\frac{\neg q, \neg p, q \Rightarrow}{q \Rightarrow}}{\quad} \quad \frac{\frac{\neg p, p, q, q \Rightarrow}{p, q, q \Rightarrow} \quad \frac{\neg q, p, q, q \Rightarrow}{q, q \Rightarrow}}{q \Rightarrow} \quad \frac{\neg q, q, q \Rightarrow}{\quad} \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

Problems

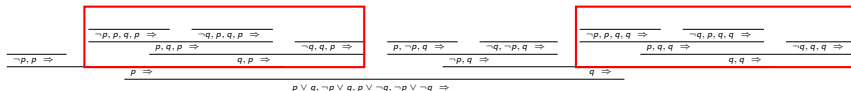
$$\begin{array}{c}
 \frac{\frac{\frac{}{\neg p, p, q, p \Rightarrow}}{\neg p, p \Rightarrow} \quad \frac{\frac{\frac{}{\neg q, p, q, p \Rightarrow}}{p, q, p \Rightarrow} \quad \frac{\frac{}{\neg q, q, p \Rightarrow}}{q, p \Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \quad \frac{\frac{\frac{}{p, \neg p, q \Rightarrow}}{\neg p, q \Rightarrow} \quad \frac{\frac{}{\neg q, \neg p, q \Rightarrow}}{p, q, q \Rightarrow} \quad \frac{\frac{\frac{}{\neg p, p, q, q \Rightarrow}}{p, q, q \Rightarrow} \quad \frac{\frac{\frac{}{\neg q, p, q, q \Rightarrow}}{q, q \Rightarrow}}{q \Rightarrow} \quad \frac{\frac{}{\neg q, q, q \Rightarrow}}{\neg q, q, q \Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

Problems



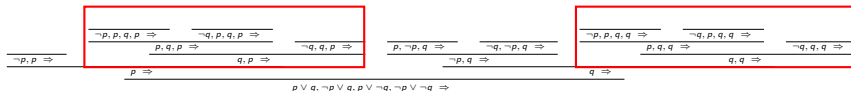
- Costly repetitions of identical proof trees

Problems



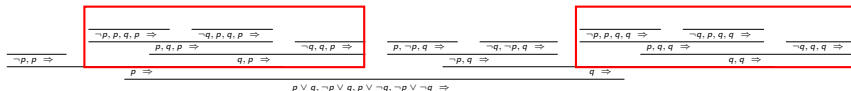
- Costly repetitions of identical proof trees
- 9 Branches

Problems

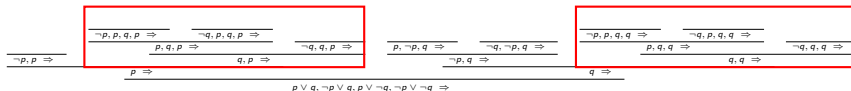


- Costly repetitions of identical proof trees
- 9 Branches
- Can often be avoided by using β rules in the “right” order

Problems



- ▶ Costly repetitions of identical proof trees
- ▶ 9 Branches
- ▶ Can often be avoided by using β rules in the “right” order
- ▶ But finding the best order is harder (!) than finding a proof



- Costly repetitions of identical proof trees
- 9 Branches
- Can often be avoided by using β rules in the “right” order
- But finding the best order is harder (!) than finding a proof
- Better: avoid using β (i.e. splitting) rules

Outline

- ▶ Motivation
- ▶ Simplification Rules
- ▶ Atomic Cut
- ▶ The DPLL Algorithm
- ▶ Other Tricks

Simplification Rules: Motivation

- ▶ Given two formulas p and $q \wedge (r \rightarrow s \wedge p)$

Simplification Rules: Motivation

- ▶ Given two formulas p and $q \wedge (r \rightarrow s \wedge p)$
- ▶ And an interpretation $\mathcal{I}$ with $\mathcal{I} \models p$

Simplification Rules: Motivation

- ▶ Given two formulas p and $q \wedge (r \rightarrow s \wedge p)$
- ▶ And an interpretation $\mathcal{I}$ with $\mathcal{I} \models p$
- ▶ $v_{\mathcal{I}}(q \wedge (r \rightarrow s \wedge p))$

Simplification Rules: Motivation

- ▶ Given two formulas p and $q \wedge (r \rightarrow s \wedge p)$
- ▶ And an interpretation $\mathcal{I}$ with $\mathcal{I} \models p$
- ▶ $v_{\mathcal{I}}(q \wedge (r \rightarrow s \wedge p)) = v_{\mathcal{I}}(q \wedge (r \rightarrow s \wedge \text{true}))$

Simplification Rules: Motivation

- ▶ Given two formulas p and $q \wedge (r \rightarrow s \wedge p)$
- ▶ And an interpretation $\mathcal{I}$ with $\mathcal{I} \models p$
- ▶ $v_{\mathcal{I}}(q \wedge (r \rightarrow s \wedge p)) = v_{\mathcal{I}}(q \wedge (r \rightarrow s \wedge \text{true})) = v_{\mathcal{I}}(q \wedge (r \rightarrow s))$

Simplification Rules: Motivation

- ▶ Given two formulas p and $q \wedge (r \rightarrow s \wedge p)$
- ▶ And an interpretation $\mathcal{I}$ with $\mathcal{I} \models p$
- ▶ $v_{\mathcal{I}}(q \wedge (r \rightarrow s \wedge p)) = v_{\mathcal{I}}(q \wedge (r \rightarrow s \wedge \text{true})) = v_{\mathcal{I}}(q \wedge (r \rightarrow s))$
- ▶ An interpretation $\mathcal{I}$ falsifies a sequent

$$p, q \wedge (r \rightarrow s \wedge p), \Gamma \vdash \Delta$$

Simplification Rules: Motivation

- ▶ Given two formulas p and $q \wedge (r \rightarrow s \wedge p)$
- ▶ And an interpretation $\mathcal{I}$ with $\mathcal{I} \models p$
- ▶ $v_{\mathcal{I}}(q \wedge (r \rightarrow s \wedge p)) = v_{\mathcal{I}}(q \wedge (r \rightarrow s \wedge \text{true})) = v_{\mathcal{I}}(q \wedge (r \rightarrow s))$
- ▶ An interpretation $\mathcal{I}$ falsifies a sequent

$$p, q \wedge (r \rightarrow s \wedge p), \Gamma \vdash \Delta$$

if and only if

Simplification Rules: Motivation

- ▶ Given two formulas p and $q \wedge (r \rightarrow s \wedge p)$
- ▶ And an interpretation $\mathcal{I}$ with $\mathcal{I} \models p$
- ▶ $v_{\mathcal{I}}(q \wedge (r \rightarrow s \wedge p)) = v_{\mathcal{I}}(q \wedge (r \rightarrow s \wedge \text{true})) = v_{\mathcal{I}}(q \wedge (r \rightarrow s))$
- ▶ An interpretation $\mathcal{I}$ falsifies a sequent

$$p, q \wedge (r \rightarrow s \wedge p), \Gamma \vdash \Delta$$

if and only if $\mathcal{I}$ falsifies the sequent

$$p, q \wedge (r \rightarrow s), \Gamma \vdash \Delta$$

Simplification

Definition 2.1 (Simplification).

Given two formulas A and B ,

Simplification

Definition 2.1 (Simplification).

Given two formulas A and B , where B does not have $\neg$ as top-symbol,

Simplification

Definition 2.1 (Simplification).

Given two formulas A and B , where B does not have $\neg$ as top-symbol, the simplification of A with B , written $A[B]$, is the result of

Simplification

Definition 2.1 (Simplification).

Given two formulas A and B , where B does not have $\neg$ as top-symbol, the simplification of A with B , written $A[B]$, is the result of

- ▶ *Replacing all occurrences of B in A by true, and*

Simplification

Definition 2.1 (Simplification).

Given two formulas A and B , where B does not have $\neg$ as top-symbol, the simplification of A with B , written $A[B]$, is the result of

- ▶ Replacing all occurrences of B in A by $true$, and
- ▶ Simplifying subformulae as long as possible using the rewritings

$A \vee true \mapsto true$	$A \vee false \mapsto A$
$A \wedge true \mapsto A$	$A \wedge false \mapsto false$
$A \rightarrow true \mapsto true$	$A \rightarrow false \mapsto \neg A$
$true \rightarrow A \mapsto A$	$false \rightarrow A \mapsto true$
$\neg true \mapsto false$	$\neg false \mapsto true$

Simplification

Definition 2.1 (Simplification).

Given two formulas A and B , where B does not have $\neg$ as top-symbol, the simplification of A with B , written $A[B]$, is the result of

- ▶ Replacing all occurrences of B in A by $true$, and
- ▶ Simplifying subformulae as long as possible using the rewritings

$$\begin{array}{ll}
 A \vee true \mapsto true & A \vee false \mapsto A \\
 A \wedge true \mapsto A & A \wedge false \mapsto false \\
 A \rightarrow true \mapsto true & A \rightarrow false \mapsto \neg A \\
 true \rightarrow A \mapsto A & false \rightarrow A \mapsto true \\
 \neg true \mapsto false & \neg false \mapsto true
 \end{array}$$

The simplification of A with $\neg B$, written $A[\neg B]$, is the result of

Simplification

Definition 2.1 (Simplification).

Given two formulas A and B , where B does not have $\neg$ as top-symbol, the simplification of A with B , written $A[B]$, is the result of

- ▶ Replacing all occurrences of B in A by $true$, and
- ▶ Simplifying subformulae as long as possible using the rewritings

$$\begin{array}{ll}
 A \vee true \mapsto true & A \vee false \mapsto A \\
 A \wedge true \mapsto A & A \wedge false \mapsto false \\
 A \rightarrow true \mapsto true & A \rightarrow false \mapsto \neg A \\
 true \rightarrow A \mapsto A & false \rightarrow A \mapsto true \\
 \neg true \mapsto false & \neg false \mapsto true
 \end{array}$$

The simplification of A with $\neg B$, written $A[\neg B]$, is the result of

- ▶ Replacing all occurrences of B in A by $false$, and

Simplification

Definition 2.1 (Simplification).

Given two formulas A and B , where B does not have $\neg$ as top-symbol, the simplification of A with B , written $A[B]$, is the result of

- ▶ Replacing all occurrences of B in A by $true$, and
- ▶ Simplifying subformulae as long as possible using the rewritings

$$\begin{array}{ll}
 A \vee true \mapsto true & A \vee false \mapsto A \\
 A \wedge true \mapsto A & A \wedge false \mapsto false \\
 A \rightarrow true \mapsto true & A \rightarrow false \mapsto \neg A \\
 true \rightarrow A \mapsto A & false \rightarrow A \mapsto true \\
 \neg true \mapsto false & \neg false \mapsto true
 \end{array}$$

The simplification of A with $\neg B$, written $A[\neg B]$, is the result of

- ▶ Replacing all occurrences of B in A by $false$, and
- ▶ Applying the same rewritings.

Simplification Examples

- To compute $(q \wedge (r \rightarrow s \wedge p))[p]$

Simplification Examples

- ▶ To compute $(q \wedge (r \rightarrow s \wedge p))[p]$
 - ▶ Do the replacement: $q \wedge (r \rightarrow s \wedge \text{true})$

Simplification Examples

- ▶ To compute $(q \wedge (r \rightarrow s \wedge p))[p]$
 - ▶ Do the replacement: $q \wedge (r \rightarrow s \wedge \text{true})$
 - ▶ Then simplify $q \wedge (r \rightarrow s \wedge \text{true}) \mapsto$

Simplification Examples

- ▶ To compute $(q \wedge (r \rightarrow s \wedge p))[p]$
 - ▶ Do the replacement: $q \wedge (r \rightarrow s \wedge \text{true})$
 - ▶ Then simplify $q \wedge (r \rightarrow s \wedge \text{true}) \mapsto q \wedge (r \rightarrow s)$

Simplification Examples

- ▶ To compute $(q \wedge (r \rightarrow s \wedge p))[p]$
 - ▶ Do the replacement: $q \wedge (r \rightarrow s \wedge \text{true})$
 - ▶ Then simplify $q \wedge (r \rightarrow s \wedge \text{true}) \mapsto q \wedge (r \rightarrow s)$
- ▶ So $(q \wedge (r \rightarrow s \wedge p))[p] = q \wedge (r \rightarrow s)$

Simplification Examples

- ▶ To compute $(q \wedge (r \rightarrow s \wedge p))[p]$
 - ▶ Do the replacement: $q \wedge (r \rightarrow s \wedge \text{true})$
 - ▶ Then simplify $q \wedge (r \rightarrow s \wedge \text{true}) \mapsto q \wedge (r \rightarrow s)$
- ▶ So $(q \wedge (r \rightarrow s \wedge p))[p] = q \wedge (r \rightarrow s)$
- ▶ To compute $(q \wedge (r \rightarrow s \wedge p))[\neg p]$

Simplification Examples

- ▶ To compute $(q \wedge (r \rightarrow s \wedge p))[p]$
 - ▶ Do the replacement: $q \wedge (r \rightarrow s \wedge \text{true})$
 - ▶ Then simplify $q \wedge (r \rightarrow s \wedge \text{true}) \mapsto q \wedge (r \rightarrow s)$
- ▶ So $(q \wedge (r \rightarrow s \wedge p))[p] = q \wedge (r \rightarrow s)$
- ▶ To compute $(q \wedge (r \rightarrow s \wedge p))[\neg p]$
 - ▶ Do the replacement: $q \wedge (r \rightarrow s \wedge \text{false})$

Simplification Examples

- ▶ To compute $(q \wedge (r \rightarrow s \wedge p))[p]$
 - ▶ Do the replacement: $q \wedge (r \rightarrow s \wedge \text{true})$
 - ▶ Then simplify $q \wedge (r \rightarrow s \wedge \text{true}) \mapsto q \wedge (r \rightarrow s)$
- ▶ So $(q \wedge (r \rightarrow s \wedge p))[p] = q \wedge (r \rightarrow s)$
- ▶ To compute $(q \wedge (r \rightarrow s \wedge p))[\neg p]$
 - ▶ Do the replacement: $q \wedge (r \rightarrow s \wedge \text{false})$
 - ▶ Then simplify $q \wedge (r \rightarrow s \wedge \text{false}) \mapsto$

Simplification Examples

- ▶ To compute $(q \wedge (r \rightarrow s \wedge p))[p]$
 - ▶ Do the replacement: $q \wedge (r \rightarrow s \wedge \text{true})$
 - ▶ Then simplify $q \wedge (r \rightarrow s \wedge \text{true}) \mapsto q \wedge (r \rightarrow s)$
- ▶ So $(q \wedge (r \rightarrow s \wedge p))[p] = q \wedge (r \rightarrow s)$
- ▶ To compute $(q \wedge (r \rightarrow s \wedge p))[\neg p]$
 - ▶ Do the replacement: $q \wedge (r \rightarrow s \wedge \text{false})$
 - ▶ Then simplify $q \wedge (r \rightarrow s \wedge \text{false}) \mapsto q \wedge (r \rightarrow \text{false}) \mapsto$

Simplification Examples

- ▶ To compute $(q \wedge (r \rightarrow s \wedge p))[p]$
 - ▶ Do the replacement: $q \wedge (r \rightarrow s \wedge \text{true})$
 - ▶ Then simplify $q \wedge (r \rightarrow s \wedge \text{true}) \mapsto q \wedge (r \rightarrow s)$
- ▶ So $(q \wedge (r \rightarrow s \wedge p))[p] = q \wedge (r \rightarrow s)$
- ▶ To compute $(q \wedge (r \rightarrow s \wedge p))[\neg p]$
 - ▶ Do the replacement: $q \wedge (r \rightarrow s \wedge \text{false})$
 - ▶ Then simplify $q \wedge (r \rightarrow s \wedge \text{false}) \mapsto q \wedge (r \rightarrow \text{false}) \mapsto q \wedge \neg r$

Simplification Examples

- ▶ To compute $(q \wedge (r \rightarrow s \wedge p))[p]$
 - ▶ Do the replacement: $q \wedge (r \rightarrow s \wedge \text{true})$
 - ▶ Then simplify $q \wedge (r \rightarrow s \wedge \text{true}) \mapsto q \wedge (r \rightarrow s)$
- ▶ So $(q \wedge (r \rightarrow s \wedge p))[p] = q \wedge (r \rightarrow s)$

- ▶ To compute $(q \wedge (r \rightarrow s \wedge p))[\neg p]$
 - ▶ Do the replacement: $q \wedge (r \rightarrow s \wedge \text{false})$
 - ▶ Then simplify $q \wedge (r \rightarrow s \wedge \text{false}) \mapsto q \wedge (r \rightarrow \text{false}) \mapsto q \wedge \neg r$
- ▶ So $(q \wedge (r \rightarrow s \wedge p))[\neg p] = q \wedge \neg r$

Main property of Simplification

Lemma 2.1.

Given a formula A that contains a subformula B , and let $B \equiv B'$. Then A is logically equivalent to the result of replacing B by B' in A .

Main property of Simplification

Lemma 2.1.

Given a formula A that contains a subformula B , and let $B \equiv B'$. Then A is logically equivalent to the result of replacing B by B' in A .

Proof.

Easily shown by structural induction over A . □

Main property of Simplification

Lemma 2.1.

Given a formula A that contains a subformula B , and let $B \equiv B'$. Then A is logically equivalent to the result of replacing B by B' in A .

Proof.

Easily shown by structural induction over A . □

Theorem 2.1.

Given formulas A and B

Main property of Simplification

Lemma 2.1.

Given a formula A that contains a subformula B , and let $B \equiv B'$. Then A is logically equivalent to the result of replacing B by B' in A .

Proof.

Easily shown by structural induction over A . □

Theorem 2.1.

Given formulas A and B and an interpretation $\mathcal{I}$ with $\mathcal{I} \models B$.

Main property of Simplification

Lemma 2.1.

Given a formula A that contains a subformula B , and let $B \equiv B'$. Then A is logically equivalent to the result of replacing B by B' in A .

Proof.

Easily shown by structural induction over A . □

Theorem 2.1.

Given formulas A and B and an interpretation $\mathcal{I}$ with $\mathcal{I} \models B$. Then $\mathcal{I} \models A$ if and only if $\mathcal{I} \models A[B]$.

Main property of Simplification

Lemma 2.1.

Given a formula A that contains a subformula B , and let $B \equiv B'$. Then A is logically equivalent to the result of replacing B by B' in A .

Proof.

Easily shown by structural induction over A . □

Theorem 2.1.

Given formulas A and B and an interpretation $\mathcal{I}$ with $\mathcal{I} \models B$. Then $\mathcal{I} \models A$ if and only if $\mathcal{I} \models A[B]$.

Proof.

For the first step (replacing B by *true* or *false*), the proof is by structural induction on A .

Main property of Simplification

Lemma 2.1.

Given a formula A that contains a subformula B , and let $B \equiv B'$. Then A is logically equivalent to the result of replacing B by B' in A .

Proof.

Easily shown by structural induction over A . □

Theorem 2.1.

Given formulas A and B and an interpretation $\mathcal{I}$ with $\mathcal{I} \models B$. Then $\mathcal{I} \models A$ if and only if $\mathcal{I} \models A[B]$.

Proof.

For the first step (replacing B by *true* or *false*), the proof is by structural induction on A . For the simplification steps, each formula is logically equivalent to the next,

Main property of Simplification

Lemma 2.1.

Given a formula A that contains a subformula B , and let $B \equiv B'$. Then A is logically equivalent to the result of replacing B by B' in A .

Proof.

Easily shown by structural induction over A . □

Theorem 2.1.

Given formulas A and B and an interpretation $\mathcal{I}$ with $\mathcal{I} \models B$. Then $\mathcal{I} \models A$ if and only if $\mathcal{I} \models A[B]$.

Proof.

For the first step (replacing B by *true* or *false*), the proof is by structural induction on A . For the simplification steps, each formula is logically equivalent to the next, due to the preceding lemma. □

Simplification Rules

We add the following four “simplification rules” to LK:

$$\frac{B, A[B], \Gamma \Rightarrow \Delta}{B, A, \Gamma \Rightarrow \Delta}$$

$$\frac{A[\neg B], \Gamma \Rightarrow B, \Delta}{A, \Gamma \Rightarrow B, \Delta}$$

$$\frac{B, \Gamma \Rightarrow A[B], \Delta}{B, \Gamma \Rightarrow A, \Delta}$$

$$\frac{\Gamma \Rightarrow B, A[\neg B], \Delta}{\Gamma \Rightarrow B, A, \Delta}$$

Example: (one-sided) LK with Simplification Rules

$$p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow$$

Example: (one-sided) LK with Simplification Rules

$$p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow$$

Example: (one-sided) LK with Simplification Rules

$$\frac{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}$$

Example: (one-sided) LK with Simplification Rules

$$\frac{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}$$

Example: (one-sided) LK with Simplification Rules

$$\frac{p, (\neg p \vee q)[p], p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \quad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow$$

$$p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow$$

Example: (one-sided) LK with Simplification Rules

$$\frac{
 \frac{
 p, \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 }{
 p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 }
 \quad
 q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 }{
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 }$$

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 p, \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 \frac{p, q, (p \vee \neg q)[p], \neg p \vee \neg q \Rightarrow}{p, \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \frac{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 \frac{p, q, \quad \text{true}, \neg p \vee \neg q \Rightarrow}{p, \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \frac{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 \textcolor{red}{p}, q, \quad \text{true}, \textcolor{red}{\neg p} \vee \textcolor{red}{\neg q} \Rightarrow \\
 \hline
 p, \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 \frac{p, q, \text{true}, (\neg p \vee \neg q)[p] \Rightarrow}{p, q, \text{true}, \neg p \vee \neg q \Rightarrow} \\
 \frac{p, q, \text{true}, \neg p \vee \neg q \Rightarrow}{p, q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \frac{p, q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 \frac{p, q, \text{true}, \quad \neg q \Rightarrow}{p, q, \quad \text{true}, \neg p \vee \neg q \Rightarrow} \\
 \frac{p, \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 \frac{p, \textcolor{red}{q}, \text{true}, \quad \quad \quad \textcolor{red}{\neg q} \Rightarrow}{p, q, \quad \quad \quad \text{true}, \neg p \vee \neg q \Rightarrow} \\
 \frac{p, \quad \quad \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad \quad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 \hline
 p, q, \text{true}, \quad \neg q \Rightarrow \\
 \hline
 p, q, \quad \text{true}, \neg p \vee \neg q \Rightarrow \\
 \hline
 p, \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 \hline
 p, q, \text{true}, \quad \neg q \Rightarrow \\
 \hline
 p, q, \quad \text{true}, \neg p \vee \neg q \Rightarrow \\
 \hline
 p, \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 \frac{}{p, q, \text{true}, \quad \neg q \Rightarrow} \\
 \hline
 \frac{}{p, q, \quad \text{true}, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{p, \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \quad \frac{q, \text{true}, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 \hline
 p, q, \text{true}, \quad \neg q \Rightarrow \\
 \hline
 p, q, \quad \text{true}, \neg p \vee \neg q \Rightarrow \\
 \hline
 p, \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \qquad \frac{q, \text{true}, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 \frac{}{p, q, \text{true}, \quad \neg q \Rightarrow} \\
 \hline
 \frac{p, q, \quad \text{true}, \neg p \vee \neg q \Rightarrow}{p, \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{q, \text{true}, p, \neg p \vee \neg q \Rightarrow}{q, \text{true}, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 \frac{}{p, q, \text{true}, \quad \neg q \Rightarrow} \\
 \hline
 \frac{p, q, \quad \text{true}, \neg p \vee \neg q \Rightarrow}{p, \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{q, \text{true}, p, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{q, \text{true}, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 \frac{p, q, \text{true}, \quad \neg q \Rightarrow}{p, q, \quad \text{true}, \neg p \vee \neg q \Rightarrow} \\
 \frac{p, \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \frac{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{q, \text{true}, p, \neg p \Rightarrow}{q, \text{true}, p, \neg p \vee \neg q \Rightarrow} \\
 \frac{q, \text{true}, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 \frac{p, q, \text{true}, \quad \neg q \Rightarrow}{p, q, \quad \text{true}, \neg p \vee \neg q \Rightarrow} \\
 \frac{p, \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{q, \text{true}, \textcolor{red}{p}, \neg p \Rightarrow}{q, \text{true}, p, \neg p \vee \neg q \Rightarrow} \\
 \frac{q, \text{true}, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 \frac{p, q, \text{true}, \quad \neg q \Rightarrow}{p, q, \quad \text{true}, \neg p \vee \neg q \Rightarrow} \\
 \frac{p, \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \frac{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{q, \text{true}, p, \neg p \Rightarrow}{q, \text{true}, p, \neg p \vee \neg q \Rightarrow} \\
 \frac{q, \text{true}, p, \neg p \vee \neg q \Rightarrow}{q, \text{true}, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \frac{q, \text{true}, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 \frac{p, q, \text{true}, \quad \neg q \Rightarrow}{p, q, \quad \text{true}, \neg p \vee \neg q \Rightarrow} \\
 \frac{p, \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{q, \text{true}, p, \neg p \Rightarrow}{q, \text{true}, p, \neg p \vee \neg q \Rightarrow} \\
 \frac{q, \text{true}, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

- Strategy: Apply simplification as much as possible, before β rules

Example: (one-sided) LK with Simplification Rules

$$\begin{array}{c}
 \frac{p, q, \text{true}, \quad \neg q \Rightarrow}{p, q, \quad \text{true}, \neg p \vee \neg q \Rightarrow} \\
 \frac{p, \quad q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \frac{q, \text{true}, p, \neg p \vee \neg q \Rightarrow}{q, \text{true}, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \frac{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

- Strategy: Apply simplification as much as possible, before β rules
- In this case: from 9 branches down to 2.

Simplification for Clauses

- ▶ Simplify a clause C with a literal L

Simplification for Clauses

- ▶ Simplify a clause C with a literal L
- ▶ Case 1: C contains L ,

Simplification for Clauses

- ▶ Simplify a clause C with a literal L
- ▶ Case 1: C contains L , $C = A_1 \vee \dots \vee A_k \vee L$

Simplification for Clauses

- ▶ Simplify a clause C with a literal L
- ▶ Case 1: C contains L , $C = A_1 \vee \dots \vee A_k \vee L$
 - ▶ Then $C[L] =$

Simplification for Clauses

- ▶ Simplify a clause C with a literal L
- ▶ Case 1: C contains L , $C = A_1 \vee \dots \vee A_k \vee L$
 - ▶ Then $C[L] = \text{true}$
 - ▶ In refutation (left of sequent, resolution), *true* is useless and can be removed

Simplification for Clauses

- ▶ Simplify a clause C with a literal L
- ▶ Case 1: C contains L , $C = A_1 \vee \dots \vee A_k \vee L$
 - ▶ Then $C[L] = \text{true}$
 - ▶ In refutation (left of sequent, resolution), *true* is useless and can be removed
 - ▶ Removing C because $L \in C$ is called **unit subsumption**

Simplification for Clauses

- ▶ Simplify a clause C with a literal L
- ▶ Case 1: C contains L , $C = A_1 \vee \dots \vee A_k \vee L$
 - ▶ Then $C[L] = \text{true}$
 - ▶ In refutation (left of sequent, resolution), *true* is useless and can be removed
 - ▶ Removing C because $L \in C$ is called **unit subsumption**
- ▶ Case 2: C contains $\bar{L}$,

Simplification for Clauses

- ▶ Simplify a clause C with a literal L
- ▶ Case 1: C contains L , $C = A_1 \vee \dots \vee A_k \vee L$
 - ▶ Then $C[L] = \text{true}$
 - ▶ In refutation (left of sequent, resolution), *true* is useless and can be removed
 - ▶ Removing C because $L \in C$ is called **unit subsumption**
- ▶ Case 2: C contains $\bar{L}$, $C = A_1 \vee \dots \vee A_k \vee \bar{L}$

Simplification for Clauses

- ▶ Simplify a clause C with a literal L
- ▶ Case 1: C contains L , $C = A_1 \vee \dots \vee A_k \vee L$
 - ▶ Then $C[L] = \text{true}$
 - ▶ In refutation (left of sequent, resolution), *true* is useless and can be removed
 - ▶ Removing C because $L \in C$ is called **unit subsumption**
- ▶ Case 2: C contains $\bar{L}$, $C = A_1 \vee \dots \vee A_k \vee \bar{L}$
 - ▶ Then $C[L] =$

Simplification for Clauses

- ▶ Simplify a clause C with a literal L
- ▶ Case 1: C contains L , $C = A_1 \vee \dots \vee A_k \vee L$
 - ▶ Then $C[L] = \text{true}$
 - ▶ In refutation (left of sequent, resolution), *true* is useless and can be removed
 - ▶ Removing C because $L \in C$ is called **unit subsumption**
- ▶ Case 2: C contains $\bar{L}$, $C = A_1 \vee \dots \vee A_k \vee \bar{L}$
 - ▶ Then $C[L] = A_1 \vee \dots \vee A_k$

Simplification for Clauses

- ▶ Simplify a clause C with a literal L
- ▶ Case 1: C contains L , $C = A_1 \vee \dots \vee A_k \vee L$
 - ▶ Then $C[L] = \text{true}$
 - ▶ In refutation (left of sequent, resolution), *true* is useless and can be removed
 - ▶ Removing C because $L \in C$ is called **unit subsumption**
- ▶ Case 2: C contains $\bar{L}$, $C = A_1 \vee \dots \vee A_k \vee \bar{L}$
 - ▶ Then $C[L] = A_1 \vee \dots \vee A_k$
 - ▶ $C[L]$ is the resolvent of C and L !

Simplification for Clauses

- ▶ Simplify a clause C with a literal L
- ▶ Case 1: C contains L , $C = A_1 \vee \dots \vee A_k \vee L$
 - ▶ Then $C[L] = \text{true}$
 - ▶ In refutation (left of sequent, resolution), true is useless and can be removed
 - ▶ Removing C because $L \in C$ is called **unit subsumption**
- ▶ Case 2: C contains $\bar{L}$, $C = A_1 \vee \dots \vee A_k \vee \bar{L}$
 - ▶ Then $C[L] = A_1 \vee \dots \vee A_k$
 - ▶ $C[L]$ is the resolvent of C and L !
 - ▶ Replacing C by $A_1 \vee \dots \vee A_k$ is called **unit resolution**

Simplification for Clauses

- ▶ Simplify a clause C with a literal L
- ▶ Case 1: C contains L , $C = A_1 \vee \dots \vee A_k \vee L$
 - ▶ Then $C[L] = \text{true}$
 - ▶ In refutation (left of sequent, resolution), true is useless and can be removed
 - ▶ Removing C because $L \in C$ is called **unit subsumption**
- ▶ Case 2: C contains $\bar{L}$, $C = A_1 \vee \dots \vee A_k \vee \bar{L}$
 - ▶ Then $C[L] = A_1 \vee \dots \vee A_k$
 - ▶ $C[L]$ is the resolvent of C and L !
 - ▶ Replacing C by $A_1 \vee \dots \vee A_k$ is called **unit resolution**
 - ▶ Note that C is subsumed by $A_1 \vee \dots \vee A_k$

Simplification for Clauses

- ▶ Simplify a clause C with a literal L
- ▶ Case 1: C contains L , $C = A_1 \vee \dots \vee A_k \vee L$
 - ▶ Then $C[L] = \text{true}$
 - ▶ In refutation (left of sequent, resolution), true is useless and can be removed
 - ▶ Removing C because $L \in C$ is called **unit subsumption**
- ▶ Case 2: C contains $\bar{L}$, $C = A_1 \vee \dots \vee A_k \vee \bar{L}$
 - ▶ Then $C[L] = A_1 \vee \dots \vee A_k$
 - ▶ $C[L]$ is the resolvent of C and L !
 - ▶ Replacing C by $A_1 \vee \dots \vee A_k$ is called **unit resolution**
 - ▶ Note that C is subsumed by $A_1 \vee \dots \vee A_k$
- ▶ Unit subsumption and unit resolution together: **unit propagation**

Simplification for Clauses

- ▶ Simplify a clause C with a literal L
- ▶ Case 1: C contains L , $C = A_1 \vee \dots \vee A_k \vee L$
 - ▶ Then $C[L] = \text{true}$
 - ▶ In refutation (left of sequent, resolution), true is useless and can be removed
 - ▶ Removing C because $L \in C$ is called **unit subsumption**
- ▶ Case 2: C contains $\bar{L}$, $C = A_1 \vee \dots \vee A_k \vee \bar{L}$
 - ▶ Then $C[L] = A_1 \vee \dots \vee A_k$
 - ▶ $C[L]$ is the resolvent of C and L !
 - ▶ Replacing C by $A_1 \vee \dots \vee A_k$ is called **unit resolution**
 - ▶ Note that C is subsumed by $A_1 \vee \dots \vee A_k$
- ▶ Unit subsumption and unit resolution together: **unit propagation**
- ▶ Given a literal L , every clause can be either removed completely, or shortened by removing $\bar{L}$, unit propagation can be used to remove L from every other clause containing L or $\bar{L}$.

Outline

- ▶ Motivation
- ▶ Simplification Rules
- ▶ Atomic Cut
- ▶ The DPLL Algorithm
- ▶ Other Tricks

Atomic Cut: Motivation

- ▶ For a sequent with n different β -formulas,

Atomic Cut: Motivation

- ▶ For a sequent with n different β -formulas,
- ▶ each of them has to be expanded on every branch. . .

Atomic Cut: Motivation

- ▶ For a sequent with n different β -formulas,
- ▶ each of them has to be expanded on every branch. . .
- ▶ . . . which gives 2^n branches. . .

Atomic Cut: Motivation

- ▶ For a sequent with n different β -formulas,
- ▶ each of them has to be expanded on every branch. . .
- ▶ . . . which gives 2^n branches. . .
- ▶ even though there might be only $k < n$ propositional variables,

Atomic Cut: Motivation

- ▶ For a sequent with n different β -formulas,
- ▶ each of them has to be expanded on every branch. . .
- ▶ . . . which gives 2^n branches. . .
- ▶ even though there might be only $k < n$ propositional variables,
- ▶ and therefore only 2^k different interpretations!

Atomic Cut: Motivation

- ▶ For a sequent with n different β -formulas,
- ▶ each of them has to be expanded on every branch. . .
- ▶ . . . which gives 2^n branches. . .
- ▶ even though there might be only $k < n$ propositional variables,
- ▶ and therefore only 2^k different interpretations!

- ▶ E.g. in the motivating example:
9 branches for 4 interpretations for 2 prop. variables.

Atomic Cut: Motivation

- ▶ For a sequent with n different β -formulas,
- ▶ each of them has to be expanded on every branch. . .
- ▶ . . . which gives 2^n branches. . .
- ▶ even though there might be only $k < n$ propositional variables,
- ▶ and therefore only 2^k different interpretations!

- ▶ E.g. in the motivating example:
9 branches for 4 interpretations for 2 prop. variables.

- ▶ Idea: max. 1 split per propositional variable

The Cut Rule

- The cut rule for LK:

$$\frac{\Gamma \Rightarrow A, \Delta \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

The Cut Rule

- ▶ The cut rule for LK:

$$\frac{\Gamma \Rightarrow A, \Delta \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

- ▶ The rule is sound (exercise) but not needed for completeness.

The Cut Rule

- ▶ The cut rule for LK:

$$\frac{\Gamma \Rightarrow A, \Delta \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

- ▶ The rule is sound (exercise) but not needed for completeness.
- ▶ It is a bit like proving a lemma A and then using it.

The Cut Rule

- The cut rule for LK:

$$\frac{\Gamma \Rightarrow A, \Delta \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

- The rule is sound (exercise) but not needed for completeness.
- It is a bit like proving a lemma A and then using it.
- Using cut can make proofs non-elementarily shorter (in first order logic)

The Cut Rule

- The cut rule for LK:

$$\frac{\Gamma \Rightarrow A, \Delta \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

- The rule is sound (exercise) but not needed for completeness.
- It is a bit like proving a lemma A and then using it.
- Using cut can make proofs non-elementarily shorter (in first order logic)
- I.e. size $O(k)$ with cut but $O(\underbrace{2^{2^{\cdot^{\cdot^2}}}}_k)$ without.

The Cut Rule

- ▶ The cut rule for LK:

$$\frac{\Gamma \Rightarrow A, \Delta \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

- ▶ The rule is sound (exercise) but not needed for completeness.
- ▶ It is a bit like proving a lemma A and then using it.
- ▶ Using cut can make proofs non-elementarily shorter (in first order logic)
- ▶ I.e. size $O(k)$ with cut but $O(\underbrace{2^{2^{\cdot^{\cdot^2}}}}_k)$ without.
- ▶ Not useful for automated proof search, because A has to be guessed.

The Cut Rule

- ▶ The cut rule for LK:

$$\frac{\Gamma \Rightarrow A, \Delta \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

- ▶ The rule is sound (exercise) but not needed for completeness.
- ▶ It is a bit like proving a lemma A and then using it.
- ▶ Using cut can make proofs non-elementarily shorter (in first order logic)
- ▶ I.e. size $O(k)$ with cut but $O(\underbrace{2^{2^{\cdot^{\cdot^2}}}}_k)$ without.
- ▶ Not useful for automated proof search, because A has to be guessed.
- ▶ The essence of human theorem proving: introducing the right lemmas!

Atomic Cut

- The atomic cut rule is just the cut rule

$$\frac{\Gamma \Rightarrow A, \Delta \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

Atomic Cut

- The atomic cut rule is just the cut rule

$$\frac{\Gamma \Rightarrow A, \Delta \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

- Where A is restricted to be an **atomic formula**.

Atomic Cut

- The atomic cut rule is just the cut rule

$$\frac{\Gamma \Rightarrow A, \Delta \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

- Where A is restricted to be an **atomic formula**.
- No nonelementary speedup :-)

Atomic Cut

- ▶ The atomic cut rule is just the cut rule

$$\frac{\Gamma \Rightarrow A, \Delta \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

- ▶ Where A is restricted to be an **atomic formula**.
- ▶ No nonelementary speedup :-)
- ▶ But we don't need more atomic cuts than we have prop. variables :-)

Atomic Cut

- ▶ The atomic cut rule is just the cut rule

$$\frac{\Gamma \Rightarrow A, \Delta \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

- ▶ Where A is restricted to be an **atomic formula**.
- ▶ No nonelementary speedup :-)
- ▶ But we don't need more atomic cuts than we have prop. variables :-)
- ▶ We can replace β rules in LK by atomic cut. . .

Atomic Cut

- ▶ The atomic cut rule is just the cut rule

$$\frac{\Gamma \Rightarrow A, \Delta \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

- ▶ Where A is restricted to be an **atomic formula**.
- ▶ No nonelementary speedup :-)
- ▶ But we don't need more atomic cuts than we have prop. variables :-)
- ▶ We can replace β rules in LK by atomic cut...
- ▶ ...if we add the simplification rules to deal with β formulas.

Atomic Cut

- ▶ The atomic cut rule is just the cut rule

$$\frac{\Gamma \Rightarrow A, \Delta \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

- ▶ Where A is restricted to be an **atomic formula**.
- ▶ No nonelementary speedup :-)
- ▶ But we don't need more atomic cuts than we have prop. variables :-)
- ▶ We can replace β rules in LK by atomic cut. . .
- ▶ . . . if we add the simplification rules to deal with β formulas.
- ▶ Atomic cut for single sided calculi:

Atomic Cut

- ▶ The atomic cut rule is just the cut rule

$$\frac{\Gamma \Rightarrow A, \Delta \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

- ▶ Where A is restricted to be an **atomic formula**.
- ▶ No nonelementary speedup :-)
- ▶ But we don't need more atomic cuts than we have prop. variables :-)
- ▶ We can replace β rules in LK by atomic cut. . .
- ▶ . . . if we add the simplification rules to deal with β formulas.
- ▶ Atomic cut for single sided calculi:

Atomic Cut

- ▶ The atomic cut rule is just the cut rule

$$\frac{\Gamma \Rightarrow A, \Delta \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

- ▶ Where A is restricted to be an **atomic formula**.
- ▶ No nonelementary speedup :-)
- ▶ But we don't need more atomic cuts than we have prop. variables :-)
- ▶ We can replace β rules in LK by atomic cut. . .
- ▶ . . . if we add the simplification rules to deal with β formulas.
- ▶ Atomic cut for single sided calculi:

$$\frac{A, \Gamma \Rightarrow \quad \neg A, \Gamma \Rightarrow}{\Gamma \Rightarrow}$$

Atomic Cut

- ▶ The atomic cut rule is just the cut rule

$$\frac{\Gamma \Rightarrow A, \Delta \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

- ▶ Where A is restricted to be an **atomic formula**.
- ▶ No nonelementary speedup :-)
- ▶ But we don't need more atomic cuts than we have prop. variables :-)
- ▶ We can replace β rules in LK by atomic cut. . .
- ▶ . . . if we add the simplification rules to deal with β formulas.
- ▶ Atomic cut for single sided calculi:

$$\frac{A, \Gamma \Rightarrow \quad \neg A, \Gamma \Rightarrow}{\Gamma \Rightarrow}$$

$$\frac{\Rightarrow A, \Delta \quad \Rightarrow \neg A, \Delta}{\Rightarrow \Delta}$$

Atomic Cut + Unit Propagation

$$p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow$$

Atomic Cut + Unit Propagation

$$p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow$$

Atomic Cut + Unit Propagation

$$\frac{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}$$

Atomic Cut + Unit Propagation

$$\begin{array}{c}
 q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

Atomic Cut + Unit Propagation

$$\frac{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \quad \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow$$

$$p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow$$

Atomic Cut + Unit Propagation

$$\frac{
 \frac{
 q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 }{
 q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 }
 \quad
 \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 }{
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 }$$

Atomic Cut + Unit Propagation

$$\begin{array}{c}
 \frac{q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \frac{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

Atomic Cut + Unit Propagation

$$\begin{array}{c}
 \frac{q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \frac{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

Atomic Cut + Unit Propagation

$$\begin{array}{c}
 \frac{q, p, \neg p \vee \neg q \Rightarrow}{q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \frac{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \frac{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

Atomic Cut + Unit Propagation

$$\begin{array}{c}
 \frac{q, p, \neg p \vee \neg q \Rightarrow}{q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \frac{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \frac{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

Atomic Cut + Unit Propagation

$$\begin{array}{c}
 \frac{q, p, \neg p \Rightarrow}{q, p, \neg p \vee \neg q \Rightarrow} \\
 \frac{q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \frac{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}
 \quad
 \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow$$

Atomic Cut + Unit Propagation

$$\begin{array}{c}
 \frac{q, p, \neg p \Rightarrow}{q, p, \neg p \vee \neg q \Rightarrow} \\
 \frac{q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \frac{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}
 \quad
 \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow$$

Atomic Cut + Unit Propagation

$$\begin{array}{c}
 \hline
 q, p, \neg p \Rightarrow \\
 \hline
 q, p, \neg p \vee \neg q \Rightarrow \\
 \hline
 q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

Atomic Cut + Unit Propagation

$$\begin{array}{c}
 \hline
 q, p, \neg p \Rightarrow \\
 \hline
 q, p, \neg p \vee \neg q \Rightarrow \\
 \hline
 q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \quad \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

Atomic Cut + Unit Propagation

$$\begin{array}{c}
 \hline
 q, p, \neg p \Rightarrow \\
 \hline
 q, p, \neg p \vee \neg q \Rightarrow \\
 \hline
 q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}
 \qquad
 \begin{array}{c}
 \neg q, p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

Atomic Cut + Unit Propagation

$$\begin{array}{c}
 \hline
 q, p, \neg p \Rightarrow \\
 \hline
 q, p, \neg p \vee \neg q \Rightarrow \\
 \hline
 q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}
 \qquad
 \begin{array}{c}
 \neg q, p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow \\
 \hline
 \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow
 \end{array}$$

Atomic Cut + Unit Propagation

$$\begin{array}{c}
 \frac{}{q, p, \neg p \Rightarrow} \\
 \hline
 \frac{}{q, p, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{}{\neg q, p, \neg p, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{\neg q, p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{\neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{\neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

Atomic Cut + Unit Propagation

$$\begin{array}{c}
 \frac{}{q, p, \neg p \Rightarrow} \\
 \hline
 \frac{}{q, p, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{}{\neg q, p, \neg p, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{\neg q, p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{\neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{\neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

Atomic Cut + Unit Propagation

$$\begin{array}{c}
 \frac{}{q, p, \neg p \Rightarrow} \\
 \hline
 \frac{}{q, p, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{}{\neg q, p, \neg p, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{\neg q, p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \\
 \hline
 \frac{}{\neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}
 \end{array}$$

Outline

- ▶ Motivation
- ▶ Simplification Rules
- ▶ Atomic Cut
- ▶ The DPLL Algorithm
- ▶ Other Tricks

The DPLL Algorithm

- ▶ DPLL stands for Davis-Putnam-Logemann-Loveland

The DPLL Algorithm

- ▶ DPLL stands for Davis-Putnam-Logemann-Loveland
- ▶ Introduced in 1962 by Martin Davis, George Logemann and Donald W. Loveland

The DPLL Algorithm

- ▶ DPLL stands for Davis-Putnam-Logemann-Loveland
- ▶ Introduced in 1962 by Martin Davis, George Logemann and Donald W. Loveland
- ▶ A refinement of an earlier algorithm, invented by Martin Davis and Hilary Putnam in (1960)

The DPLL Algorithm

- ▶ DPLL stands for Davis-Putnam-Logemann-Loveland
- ▶ Introduced in 1962 by Martin Davis, George Logemann and Donald W. Loveland
- ▶ A refinement of an earlier algorithm, invented by Martin Davis and Hilary Putnam in (1960)
- ▶ Made propositional theorem proving (“SAT solving”) practically viable

The DPLL Algorithm

- ▶ DPLL stands for Davis-Putnam-Logemann-Loveland
- ▶ Introduced in 1962 by Martin Davis, George Logemann and Donald W. Loveland
- ▶ A refinement of an earlier algorithm, invented by Martin Davis and Hilary Putnam in (1960)
- ▶ Made propositional theorem proving (“SAT solving”) practically viable
- ▶ After almost 60 years, still the basis of most efficient SAT solvers

The DPLL Algorithm

- ▶ DPLL stands for Davis-Putnam-Logemann-Loveland
- ▶ Introduced in 1962 by Martin Davis, George Logemann and Donald W. Loveland
- ▶ A refinement of an earlier algorithm, invented by Martin Davis and Hilary Putnam in (1960)
- ▶ Made propositional theorem proving (“SAT solving”) practically viable
- ▶ After almost 60 years, still the basis of most efficient SAT solvers
- ▶ DPLL works on a set of propositional clauses

The DPLL Algorithm

- ▶ DPLL stands for Davis-Putnam-Logemann-Loveland
- ▶ Introduced in 1962 by Martin Davis, George Logemann and Donald W. Loveland
- ▶ A refinement of an earlier algorithm, invented by Martin Davis and Hilary Putnam in (1960)
- ▶ Made propositional theorem proving (“SAT solving”) practically viable
- ▶ After almost 60 years, still the basis of most efficient SAT solvers

- ▶ DPLL works on a set of propositional clauses
- ▶ DPLL Consists of

The DPLL Algorithm

- ▶ DPLL stands for Davis-Putnam-Logemann-Loveland
- ▶ Introduced in 1962 by Martin Davis, George Logemann and Donald W. Loveland
- ▶ A refinement of an earlier algorithm, invented by Martin Davis and Hilary Putnam in (1960)
- ▶ Made propositional theorem proving (“SAT solving”) practically viable
- ▶ After almost 60 years, still the basis of most efficient SAT solvers

- ▶ DPLL works on a set of propositional clauses
- ▶ DPLL Consists of
 - ▶ Atomic Cut (with a heuristic for choosing the atom)

The DPLL Algorithm

- ▶ DPLL stands for Davis-Putnam-Logemann-Loveland
- ▶ Introduced in 1962 by Martin Davis, George Logemann and Donald W. Loveland
- ▶ A refinement of an earlier algorithm, invented by Martin Davis and Hilary Putnam in (1960)
- ▶ Made propositional theorem proving (“SAT solving”) practically viable
- ▶ After almost 60 years, still the basis of most efficient SAT solvers

- ▶ DPLL works on a set of propositional clauses
- ▶ DPLL Consists of
 - ▶ Atomic Cut (with a heuristic for choosing the atom)
 - ▶ Unit Propagation

The DPLL Algorithm

- ▶ DPLL stands for Davis-Putnam-Logemann-Loveland
- ▶ Introduced in 1962 by Martin Davis, George Logemann and Donald W. Loveland
- ▶ A refinement of an earlier algorithm, invented by Martin Davis and Hilary Putnam in (1960)
- ▶ Made propositional theorem proving (“SAT solving”) practically viable
- ▶ After almost 60 years, still the basis of most efficient SAT solvers

- ▶ DPLL works on a set of propositional clauses
- ▶ DPLL Consists of
 - ▶ Atomic Cut (with a heuristic for choosing the atom)
 - ▶ Unit Propagation
 - ▶ Pure Literal Elimination (exercise!)

Outline

- ▶ Motivation
- ▶ Simplification Rules
- ▶ Atomic Cut
- ▶ The DPLL Algorithm
- ▶ Other Tricks

Lemma Generation

- Remember the exercise sheet 2?

$$\frac{\Gamma \Rightarrow A, \Delta \quad \Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \wedge B, \Delta} \wedge\text{-lg}$$

Lemma Generation

- Remember the exercise sheet 2?

$$\frac{\Gamma \Rightarrow A, \Delta \quad \Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \wedge B, \Delta} \wedge\text{-lg}$$

- Closing the left branch, we “learnt the Lemma A ”

Lemma Generation

- Remember the exercise sheet 2?

$$\frac{\Gamma \Rightarrow A, \Delta \quad \Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \wedge B, \Delta} \wedge\text{-lg}$$

- Closing the left branch, we “learnt the Lemma A ”
- With single-sided sequents:

$$\frac{A, \Gamma \Rightarrow \quad \neg A, B, \Gamma \Rightarrow}{A \vee B, \Gamma \Rightarrow} \vee\text{-lg}$$

Lemma Generation

- Remember the exercise sheet 2?

$$\frac{\Gamma \Rightarrow A, \Delta \quad \Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \wedge B, \Delta} \wedge\text{-lg}$$

- Closing the left branch, we “learnt the Lemma A ”
- With single-sided sequents:

$$\frac{A, \Gamma \Rightarrow \quad \neg A, B, \Gamma \Rightarrow}{A \vee B, \Gamma \Rightarrow} \vee\text{-lg}$$

- We refuted A , so now we may assume $\neg A$.

Lemma Generation

- Remember the exercise sheet 2?

$$\frac{\Gamma \Rightarrow A, \Delta \quad \Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \wedge B, \Delta} \wedge\text{-lg}$$

- Closing the left branch, we “learnt the Lemma A ”
- With single-sided sequents:

$$\frac{A, \Gamma \Rightarrow \quad \neg A, B, \Gamma \Rightarrow}{A \vee B, \Gamma \Rightarrow} \vee\text{-lg}$$

- We refuted A , so now we may assume $\neg A$.
- Whenever we close a branch, we learn that a certain combination of literals $L_1, \dots, L_k$ leads to a contradiction

Lemma Generation

- ▶ Remember the exercise sheet 2?

$$\frac{\Gamma \Rightarrow A, \Delta \quad \Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \wedge B, \Delta} \wedge\text{-lg}$$

- ▶ Closing the left branch, we “learnt the Lemma A ”
- ▶ With single-sided sequents:

$$\frac{A, \Gamma \Rightarrow \quad \neg A, B, \Gamma \Rightarrow}{A \vee B, \Gamma \Rightarrow} \vee\text{-lg}$$

- ▶ We refuted A , so now we may assume $\neg A$.
- ▶ Whenever we close a branch, we learn that a certain combination of literals $L_1, \dots, L_k$ leads to a contradiction
- ▶ We can add a clause $\overline{L_1} \vee \dots \vee \overline{L_k}$ to capture this.

Lemma Generation

- Remember the exercise sheet 2?

$$\frac{\Gamma \Rightarrow A, \Delta \quad \Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \wedge B, \Delta} \wedge\text{-lg}$$

- Closing the left branch, we “learnt the Lemma A ”
- With single-sided sequents:

$$\frac{A, \Gamma \Rightarrow \quad \neg A, B, \Gamma \Rightarrow}{A \vee B, \Gamma \Rightarrow} \vee\text{-lg}$$

- We refuted A , so now we may assume $\neg A$.
- Whenever we close a branch, we learn that a certain combination of literals $L_1, \dots, L_k$ leads to a contradiction
- We can add a clause $\overline{L_1} \vee \dots \vee \overline{L_k}$ to capture this.
- “Clause Learning”

Pruning

► Pruning

Pruning

- ▶ Pruning $\equiv$ Backjumping

Pruning

- ▶ Pruning $\equiv$ Backjumping $\equiv$ Intelligent Backtracking

Pruning

- ▶ Pruning $\equiv$ Backjumping $\equiv$ Intelligent Backtracking $\equiv$ Non-chronological Backtracking

Pruning

- ▶ Pruning $\equiv$ Backjumping $\equiv$ Intelligent Backtracking $\equiv$ Non-chronological Backtracking
- ▶ Consider the following derivation

$$\begin{array}{c}
 \text{needed B and G} \qquad \qquad \text{needed G} \\
 \hline
 p, q, \neg p, \neg r \Rightarrow \qquad p, q, r, \neg r \Rightarrow \\
 \hline
 p, q, \neg p \vee r, \neg r \Rightarrow \qquad \text{G} \qquad p, s, \dots \Rightarrow \\
 \hline
 p, q \vee s, \neg p \vee r, \neg r \Rightarrow \qquad \text{R} \qquad q, \dots \Rightarrow \\
 \hline
 p \vee q, q \vee s, \neg p \vee r, \neg r \Rightarrow \qquad \text{B}
 \end{array}$$

Pruning

- ▶ Pruning $\equiv$ Backjumping $\equiv$ Intelligent Backtracking $\equiv$ Non-chronological Backtracking
- ▶ Consider the following derivation

$$\frac{\frac{\text{needed B and G}}{p, q, \neg p, \neg r \Rightarrow} \quad \frac{\text{needed G}}{p, q, r, \neg r \Rightarrow} \quad \text{G}}{p, q, \neg p \vee r, \neg r \Rightarrow} \quad \frac{p, s, \dots \Rightarrow}{p, q \vee s, \neg p \vee r, \neg r \Rightarrow} \text{R} \quad \frac{q, \dots \Rightarrow}{p \vee q, q \vee s, \neg p \vee r, \neg r \Rightarrow} \text{B}$$

- ▶ No formulae introduced by **R** needed to close the two left branches

Pruning

- ▶ Pruning $\equiv$ Backjumping $\equiv$ Intelligent Backtracking $\equiv$ Non-chronological Backtracking
- ▶ Consider the following derivation

$$\begin{array}{c}
 \text{needed B and G} \qquad \qquad \text{needed G} \\
 \hline
 \frac{p, q, \neg p, \neg r \Rightarrow}{p, q, \neg p \vee r, \neg r \Rightarrow} \qquad \frac{p, q, r, \neg r \Rightarrow}{p, q \vee s, \neg p \vee r, \neg r \Rightarrow} \text{G} \qquad \frac{p, s, \dots \Rightarrow}{p \vee q, q \vee s, \neg p \vee r, \neg r \Rightarrow} \text{R} \qquad \frac{q, \dots \Rightarrow}{p \vee q, q \vee s, \neg p \vee r, \neg r \Rightarrow} \text{B}
 \end{array}$$

- ▶ No formulae introduced by R needed to close the two left branches
- ▶ Could have closed the branch without applying R

Pruning

- ▶ Pruning $\equiv$ Backjumping $\equiv$ Intelligent Backtracking $\equiv$ Non-chronological Backtracking
- ▶ Consider the following derivation

$$\begin{array}{c}
 \text{needed B and G} \qquad \qquad \text{needed G} \\
 \hline
 p, q, \neg p, \neg r \Rightarrow \qquad p, q, r, \neg r \Rightarrow \\
 \hline
 p, q, \neg p \vee r, \neg r \Rightarrow \qquad \qquad \qquad \text{G} \qquad p, s, \dots \Rightarrow \\
 \hline
 p, q \vee s, \neg p \vee r, \neg r \Rightarrow \qquad \qquad \qquad \text{R} \qquad q, \dots \Rightarrow \\
 \hline
 p \vee q, q \vee s, \neg p \vee r, \neg r \Rightarrow \qquad \qquad \qquad \text{B}
 \end{array}$$

- ▶ No formulae introduced by **R** needed to close the two left branches
- ▶ Could have closed the branch without applying **R**
- ▶ Pruning: after closing the left two branches, continue with

$$\begin{array}{c}
 \text{needed B and G} \qquad \qquad \text{needed G} \\
 \hline
 p, \neg p, \neg r \Rightarrow \qquad p, r, \neg r \Rightarrow \\
 \hline
 p, \neg p \vee r, \neg r \Rightarrow \qquad \qquad \qquad \text{G} \qquad q, \dots \Rightarrow \\
 \hline
 p \vee q, q \vee s, \neg p \vee r, \neg r \Rightarrow \qquad \qquad \qquad \text{B}
 \end{array}$$

Conflict-driven clause learning (CDCL)

- ▶ The modern take on DPLL

Conflict-driven clause learning (CDCL)

- ▶ The modern take on DPLL
- ▶ See e.g. the successful MiniSat implementation <http://minisat.se/>

Conflict-driven clause learning (CDCL)

- ▶ The modern take on DPLL
- ▶ See e.g. the successful MiniSat implementation <http://minisat.se/>
- ▶ A combination of

Conflict-driven clause learning (CDCL)

- ▶ The modern take on DPLL
- ▶ See e.g. the successful MiniSat implementation <http://minisat.se/>
- ▶ A combination of
 - ▶ Atomic cut

Conflict-driven clause learning (CDCL)

- ▶ The modern take on DPLL
- ▶ See e.g. the successful MiniSat implementation <http://minisat.se/>
- ▶ A combination of
 - ▶ Atomic cut
 - ▶ Unit propagation

Conflict-driven clause learning (CDCL)

- ▶ The modern take on DPLL
- ▶ See e.g. the successful MiniSat implementation <http://minisat.se/>
- ▶ A combination of
 - ▶ Atomic cut
 - ▶ Unit propagation
 - ▶ Clause learning

Conflict-driven clause learning (CDCL)

- ▶ The modern take on DPLL
- ▶ See e.g. the successful MiniSat implementation <http://minisat.se/>
- ▶ A combination of
 - ▶ Atomic cut
 - ▶ Unit propagation
 - ▶ Clause learning
 - ▶ Pruning

Stålmarck's Method

- ▶ Devised by Gunnar Stålmarck, applied for patent 1989

Stålmarck's Method

- Devised by Gunnar Stålmarck, applied for patent 1989
- The Dilemma Rule:

$$\frac{
 \begin{array}{c}
 \Gamma_1 \cap \Gamma_2 \Rightarrow \\
 \hline
 \Gamma_1 \Rightarrow \qquad \Gamma_2 \Rightarrow \\
 \vdots \qquad \qquad \vdots \\
 A, \Gamma \Rightarrow \qquad \neg A, \Gamma \Rightarrow \\
 \hline
 \Gamma \Rightarrow
 \end{array}
 }{
 \Gamma \Rightarrow
 }$$

Stålmarck's Method

- Devised by Gunnar Stålmarck, applied for patent 1989
- The Dilemma Rule:

$$\frac{
 \begin{array}{c}
 \Gamma_1 \cap \Gamma_2 \Rightarrow \\
 \hline
 \Gamma_1 \Rightarrow \qquad \Gamma_2 \Rightarrow \\
 \vdots \qquad \qquad \vdots \\
 A, \Gamma \Rightarrow \qquad \neg A, \Gamma \Rightarrow \\
 \hline
 \Gamma \Rightarrow
 \end{array}
 }{
 }$$

- After unit propagation, join branches generated by cut

Stålmarck's Method

- ▶ Devised by Gunnar Stålmarck, applied for patent 1989
- ▶ The Dilemma Rule:

$$\frac{
 \begin{array}{c}
 \Gamma_1 \cap \Gamma_2 \Rightarrow \\
 \hline
 \Gamma_1 \Rightarrow \qquad \Gamma_2 \Rightarrow \\
 \vdots \qquad \qquad \vdots \\
 A, \Gamma \Rightarrow \qquad \neg A, \Gamma \Rightarrow \\
 \hline
 \Gamma \Rightarrow
 \end{array}
 }{
 }$$

- ▶ After unit propagation, join branches generated by cut
- ▶ Stålmarck's discovery: often enough to consider max two branches

Stålmarck's Method

- ▶ Devised by Gunnar Stålmarck, applied for patent 1989
- ▶ The Dilemma Rule:

$$\frac{
 \begin{array}{c}
 \Gamma_1 \cap \Gamma_2 \Rightarrow \\
 \hline
 \Gamma_1 \Rightarrow \qquad \Gamma_2 \Rightarrow \\
 \vdots \qquad \qquad \vdots \\
 A, \Gamma \Rightarrow \qquad \neg A, \Gamma \Rightarrow \\
 \hline
 \Gamma \Rightarrow
 \end{array}
 }{
 }$$

- ▶ After unit propagation, join branches generated by cut
- ▶ Stålmarck's discovery: often enough to consider max two branches
- ▶ Not always.

Stålmarck's Method

- ▶ Devised by Gunnar Stålmarck, applied for patent 1989
- ▶ The Dilemma Rule:

$$\frac{
 \begin{array}{c}
 \Gamma_1 \cap \Gamma_2 \Rightarrow \\
 \hline
 \Gamma_1 \Rightarrow \qquad \Gamma_2 \Rightarrow \\
 \vdots \qquad \qquad \vdots \\
 A, \Gamma \Rightarrow \qquad \neg A, \Gamma \Rightarrow \\
 \hline
 \Gamma \Rightarrow
 \end{array}
 }{
 }$$

- ▶ After unit propagation, join branches generated by cut
- ▶ Stålmarck's discovery: often enough to consider max two branches
- ▶ Not always. Why?

Stålmarck's Method

- ▶ Devised by Gunnar Stålmarck, applied for patent 1989
- ▶ The Dilemma Rule:

$$\frac{
 \begin{array}{c}
 \Gamma_1 \cap \Gamma_2 \Rightarrow \\
 \hline
 \Gamma_1 \Rightarrow \qquad \Gamma_2 \Rightarrow \\
 \vdots \qquad \qquad \vdots \\
 A, \Gamma \Rightarrow \qquad \neg A, \Gamma \Rightarrow \\
 \hline
 \Gamma \Rightarrow
 \end{array}
 }{
 }$$

- ▶ After unit propagation, join branches generated by cut
- ▶ Stålmarck's discovery: often enough to consider max two branches
- ▶ Not always. Why?
- ▶ In general: nesting of Dilemma Rule.

Stålmarck's Method

- ▶ Devised by Gunnar Stålmarck, applied for patent 1989
- ▶ The Dilemma Rule:

$$\frac{
 \begin{array}{c}
 \Gamma_1 \cap \Gamma_2 \Rightarrow \\
 \hline
 \Gamma_1 \Rightarrow \qquad \Gamma_2 \Rightarrow \\
 \vdots \qquad \qquad \vdots \\
 A, \Gamma \Rightarrow \qquad \neg A, \Gamma \Rightarrow \\
 \hline
 \Gamma \Rightarrow
 \end{array}
 }{
 }$$

- ▶ After unit propagation, join branches generated by cut
- ▶ Stålmarck's discovery: often enough to consider max two branches
- ▶ Not always. Why?
- ▶ In general: nesting of Dilemma Rule.
- ▶ Still: deep nesting rarely needed.

Summary

- ▶ Efficient theorem provers *combine* formulas instead of just decomposing

Summary

- ▶ Efficient theorem provers *combine* formulas instead of just decomposing
 - ▶ The resolution rule is an example

Summary

- ▶ Efficient theorem provers *combine* formulas instead of just decomposing
 - ▶ The resolution rule is an example
 - ▶ The simplification rules are another

Summary

- ▶ Efficient theorem provers *combine* formulas instead of just decomposing
 - ▶ The resolution rule is an example
 - ▶ The simplification rules are another
- ▶ For propositional logic, unit propagation is very effective

Summary

- ▶ Efficient theorem provers *combine* formulas instead of just decomposing
 - ▶ The resolution rule is an example
 - ▶ The simplification rules are another
- ▶ For propositional logic, unit propagation is very effective
- ▶ Atomic cut and unit propagation are the main ingredients of DPLL

Summary

- ▶ Efficient theorem provers *combine* formulas instead of just decomposing
 - ▶ The resolution rule is an example
 - ▶ The simplification rules are another
- ▶ For propositional logic, unit propagation is very effective
- ▶ Atomic cut and unit propagation are the main ingredients of DPLL
- ▶ DPLL has been refined to CDCL

Summary

- ▶ Efficient theorem provers *combine* formulas instead of just decomposing
 - ▶ The resolution rule is an example
 - ▶ The simplification rules are another
- ▶ For propositional logic, unit propagation is very effective
- ▶ Atomic cut and unit propagation are the main ingredients of DPLL
- ▶ DPLL has been refined to CDCL
- ▶ CDCL incorporates clause learning and pruning