IN3070/4070 – Logic – Autumn 2023

Lecture 9: DPLL

Martin Giese

18th October 2023



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.

$$\frac{\Gamma \;\Rightarrow\; A, \Delta \qquad \Gamma, B \;\Rightarrow\; \Delta}{\Gamma, A \to B \;\Rightarrow\; \Delta}$$

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.

$$\frac{\Gamma \;\Rightarrow\; A, \Delta \qquad \Gamma, B \;\Rightarrow\; \Delta}{\Gamma, A \to B \;\Rightarrow\; \Delta}$$

γ-rules

Apply for all terms t, e.g.

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

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.

$$\frac{\Gamma \;\Rightarrow\; A, \Delta \qquad \Gamma, B \;\Rightarrow\; \Delta}{\Gamma, A \to B \;\Rightarrow\; \Delta}$$

γ-rules

Apply for all terms t, e.g.

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

δ-rules

Introduce new constant c, e.g.

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

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

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

A Naïve SAT Solver at Work

$$\dfrac{\dfrac{\neg p, p \;\Rightarrow \qquad\qquad\qquad\qquad q, p \;\Rightarrow}{p \;\Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

A Naïve SAT Solver at Work

$$\dfrac{\dfrac{\dfrac{}{\neg p, p \;\Rightarrow}}{p \;\Rightarrow}\qquad \dfrac{}{q, p \;\Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

A Naïve SAT Solver at Work

$$\dfrac{\dfrac{\dfrac{}{\neg p, p \;\Rightarrow}}{p \;\Rightarrow}\qquad \dfrac{}{q, p \;\Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

A Naïve SAT Solver at Work

$$\dfrac{\dfrac{\dfrac{}{\neg p, p \;\Rightarrow}}{p \;\Rightarrow} \qquad \dfrac{\dfrac{}{p, q, p \;\Rightarrow} \qquad \dfrac{}{\neg q, q, p \;\Rightarrow}}{q, p \;\Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

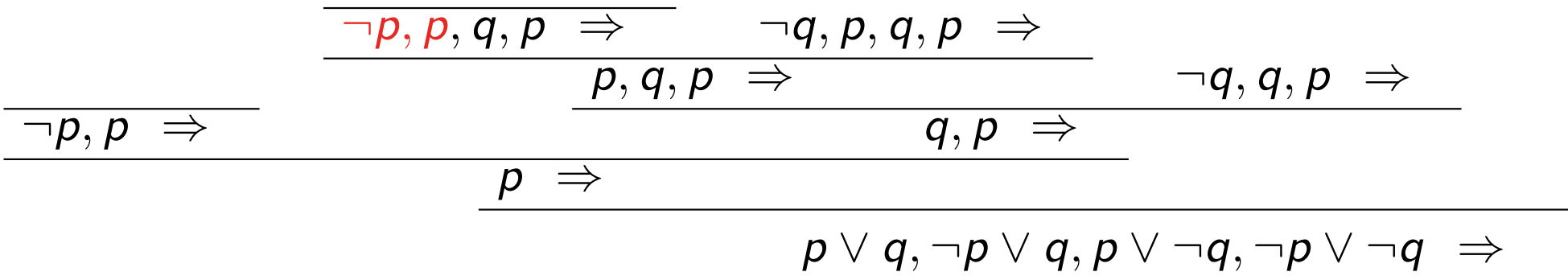
A Naïve SAT Solver at Work

$$\dfrac{\dfrac{\dfrac{}{\neg p, p \;\Rightarrow}}{p \;\Rightarrow}\qquad \dfrac{\dfrac{p, q, p \;\Rightarrow}{q, p \;\Rightarrow}\qquad \dfrac{}{\neg q, q, p \;\Rightarrow}}{p \;\Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

A Naïve SAT Solver at Work

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg p, p \;\Rightarrow}}{p \;\Rightarrow}\quad \dfrac{\dfrac{\dfrac{}{\neg p, p, q, p \;\Rightarrow}\quad \dfrac{}{\neg q, p, q, p \;\Rightarrow}}{p, q, p \;\Rightarrow}\quad \dfrac{}{\neg q, q, p \;\Rightarrow}}{q, p \;\Rightarrow}}{p \;\Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

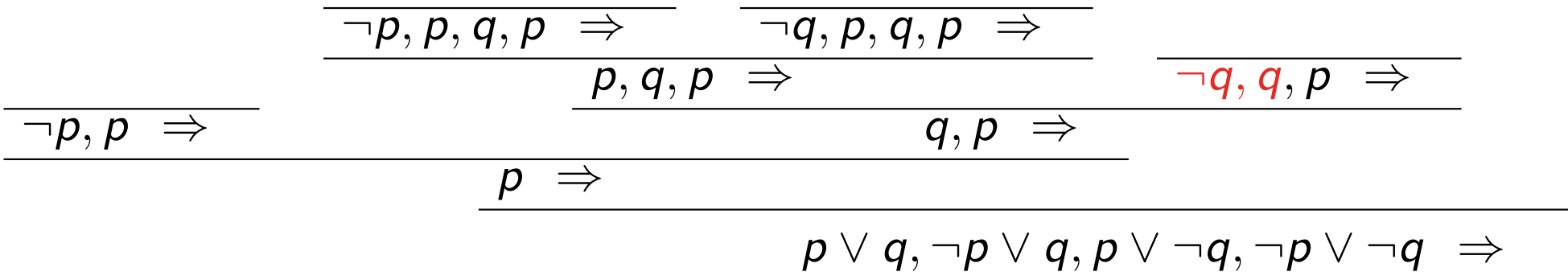
A Naïve SAT Solver at Work



A Naïve SAT Solver at Work

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

A Naïve SAT Solver at Work



A Naïve SAT Solver at Work

$$\dfrac{\dfrac{\dfrac{}{\neg q, q, p \;\Rightarrow}}{}\qquad q \;\Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

A Naïve SAT Solver at Work

$$\dfrac{\dfrac{\dfrac{}{\neg q, q, p \;\Rightarrow}}{}\qquad \dfrac{}{q \;\Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

A Naïve SAT Solver at Work

$$\dfrac{\dfrac{\dfrac{}{\neg q, q, p \;\Rightarrow}}{} \qquad \dfrac{\dfrac{}{\neg p, q \;\Rightarrow}}{q \;\Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

A Naïve SAT Solver at Work

$$\dfrac{\dfrac{\dfrac{}{\neg q, q, p \;\Rightarrow}}{} \qquad \dfrac{\dfrac{\neg p, q \;\Rightarrow}{q \;\Rightarrow}}{}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

A Naïve SAT Solver at Work

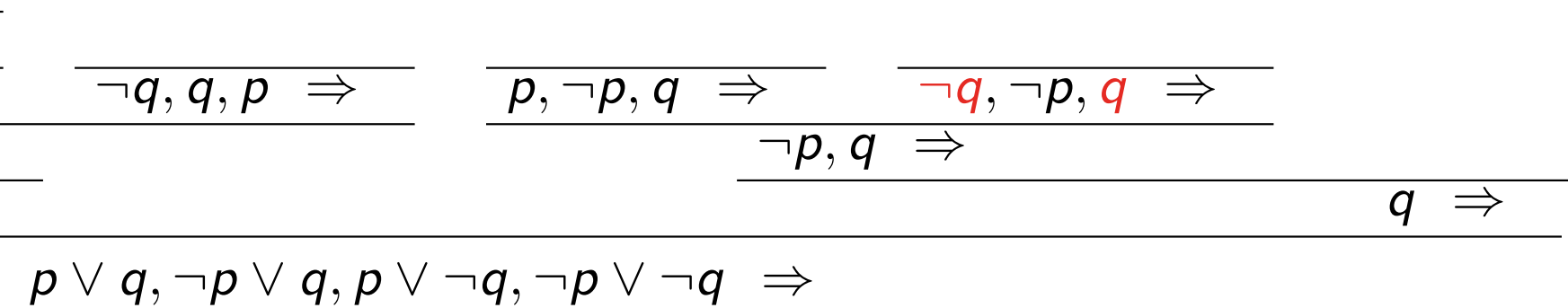
$$\dfrac{\dfrac{\dfrac{}{\neg q, q, p \;\Rightarrow}\qquad \dfrac{p, \neg p, q \;\Rightarrow \qquad \neg q, \neg p, q \;\Rightarrow}{\neg p, q \;\Rightarrow}}{q \;\Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

A Naïve SAT Solver at Work

$$\dfrac{\dfrac{\dfrac{}{\neg q, q, p \;\Rightarrow}\qquad\dfrac{}{p, \neg p, q \;\Rightarrow}}{\neg p, q \;\Rightarrow}\qquad\dfrac{}{\neg q, \neg p, q \;\Rightarrow}}{q \;\Rightarrow}$$

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

A Naïve SAT Solver at Work



A Naïve SAT Solver at Work

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg q, \neg p, q \;\Rightarrow}}{q \;\Rightarrow} \qquad \dfrac{}{q, q \;\Rightarrow}}{q \;\Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

A Naïve SAT Solver at Work

$$\dfrac{\dfrac{\dfrac{}{\neg q, \neg p, q \;\Rightarrow}}{q \;\Rightarrow} \qquad \dfrac{\dfrac{}{p, q, q \;\Rightarrow} \qquad \dfrac{}{\neg q, q, q \;\Rightarrow}}{q, q \;\Rightarrow}}{q \;\Rightarrow}$$

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

A Naïve SAT Solver at Work

$$\dfrac{\dfrac{\dfrac{}{\neg q, \neg p, q \;\Rightarrow}}{q \;\Rightarrow} \qquad \dfrac{\dfrac{}{p, q, q \;\Rightarrow} \qquad \dfrac{}{\neg q, q, q \;\Rightarrow}}{q, q \;\Rightarrow}}{q \;\Rightarrow}$$

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

A Naïve SAT Solver at Work

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg q, \neg p, q \;\Rightarrow}}{q \;\Rightarrow} \qquad \dfrac{\dfrac{\neg p, p, q, q \;\Rightarrow}{p, q, q \;\Rightarrow} \qquad \dfrac{\neg q, p, q, q \;\Rightarrow}{}}{q, q \;\Rightarrow} \qquad \neg q, q, q \;\Rightarrow}{q \;\Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

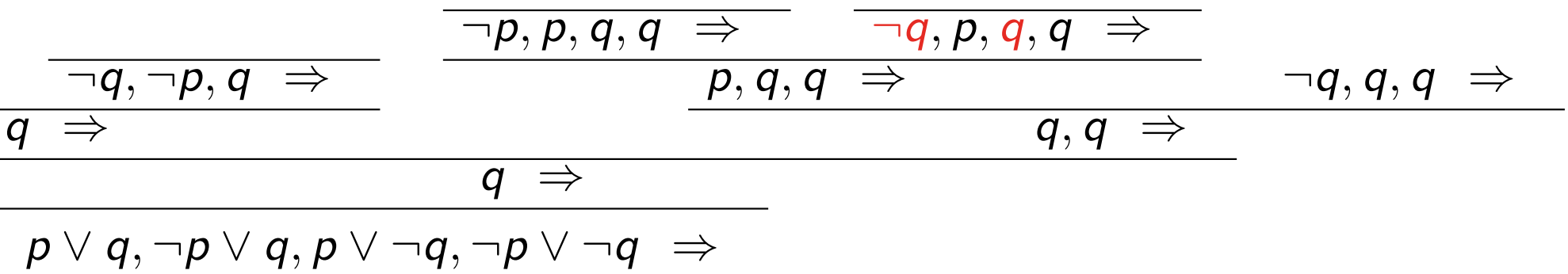
A Naïve SAT Solver at Work

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg p, p, q, q \;\Rightarrow}}{p, q, q \;\Rightarrow}}{q, q \;\Rightarrow}}{q \;\Rightarrow} \qquad \dfrac{\dfrac{\dfrac{}{\neg q, p, q, q \;\Rightarrow}}{\neg q, q, q \;\Rightarrow}}{\neg q, \neg p, q \;\Rightarrow}$$

$$\dfrac{q \;\Rightarrow \qquad \neg q, \neg p, q \;\Rightarrow}{q \;\Rightarrow}$$

$$\dfrac{q \;\Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

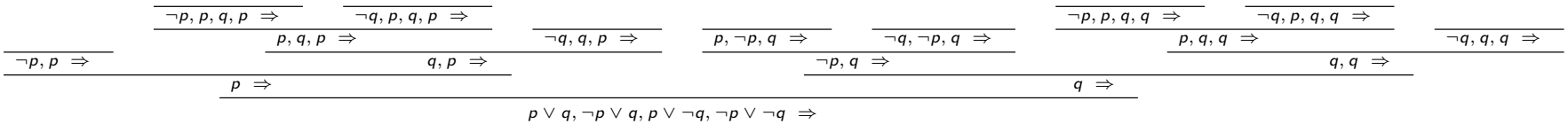
A Naïve SAT Solver at Work



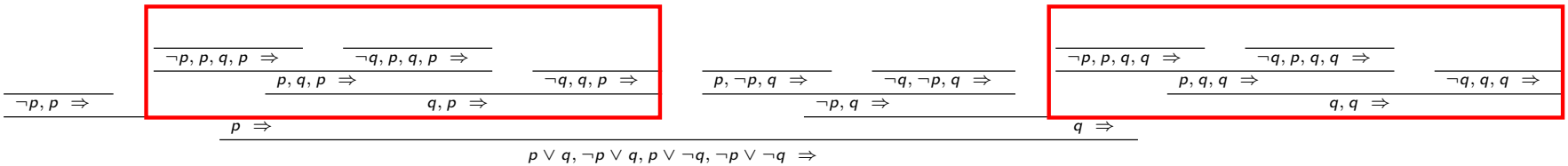
A Naïve SAT Solver at Work

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg q, \neg p, q \;\Rightarrow}}{q \;\Rightarrow} \quad \dfrac{\dfrac{\dfrac{}{\neg p, p, q, q \;\Rightarrow} \quad \dfrac{}{\neg q, p, q, q \;\Rightarrow}}{p, q, q \;\Rightarrow}}{q, q \;\Rightarrow} \quad \dfrac{}{\neg q, q, q \;\Rightarrow}}{q \;\Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

Problems

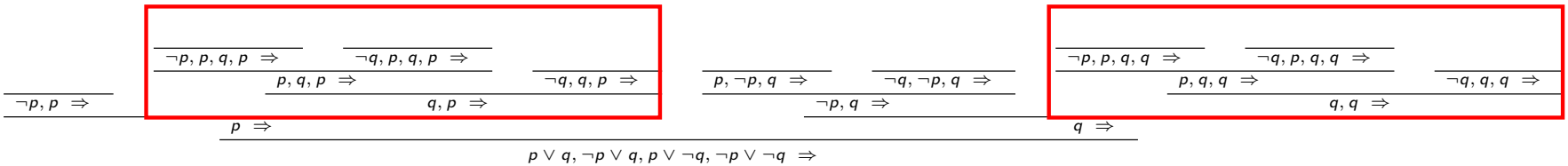

¬p, p, q, p ⇒
¬q, p, q, p ⇒
¬p, p, q, q ⇒
¬q, p, q, q ⇒
p, q, p ⇒
¬q, q, p ⇒
p, ¬p, q ⇒
¬q, ¬p, q ⇒
p, q, q ⇒
¬q, q, q ⇒
¬p, p ⇒
q, p ⇒
¬p, q ⇒
q, q ⇒
p ⇒
q ⇒
p ∨ q, ¬p ∨ q, p ∨ ¬q, ¬p ∨ ¬q ⇒

Problems



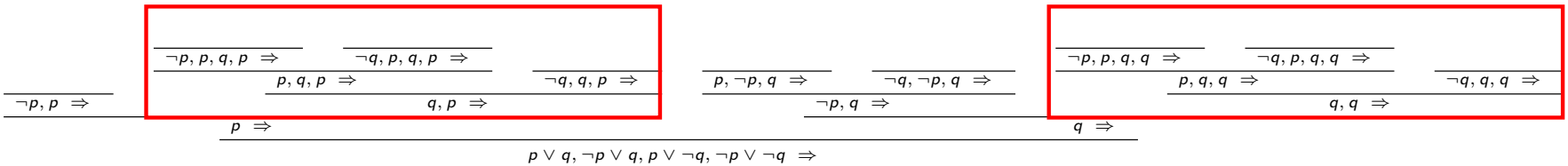
- Costly repetitions of identical proof trees

Problems



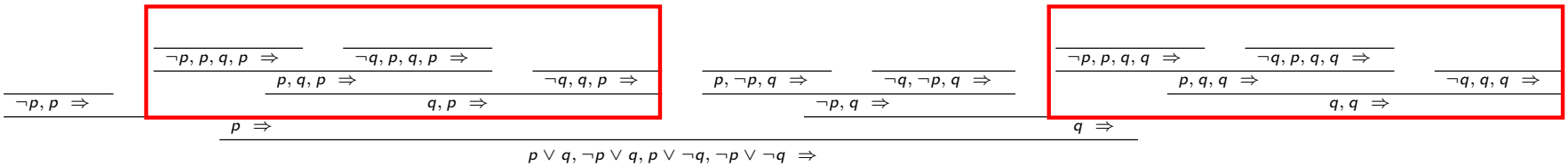
- Costly repetitions of identical proof trees
- 9 Branches

Problems



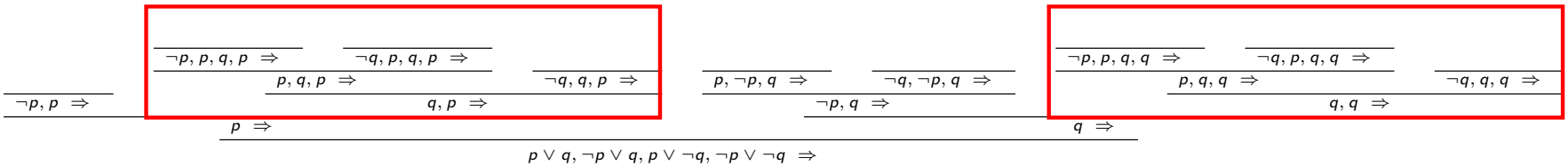
- Costly repetitions of identical proof trees
- 9 Branches
- Can often be avoided by using β rules in the "right" order

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

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
- 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 $\mathcal{I} \models p$
- $v_{\mathcal{I}}(q \wedge (r \rightarrow s \wedge p)) = v_{\mathcal{I}}(q \wedge (r \rightarrow s \wedge \mathit{true}))$

Simplification Rules: Motivation

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

Simplification Rules: Motivation

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

- 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 $\mathcal{I} \models p$
- $v_{\mathcal{I}}(q \wedge (r \rightarrow s \wedge p)) = v_{\mathcal{I}}(q \wedge (r \rightarrow s \wedge \mathit{true})) = v_{\mathcal{I}}(q \wedge (r \rightarrow s))$

- 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 $\mathcal{I} \models p$
- $v_{\mathcal{I}}(q \wedge (r \rightarrow s \wedge p)) = v_{\mathcal{I}}(q \wedge (r \rightarrow s \wedge \mathit{true})) = v_{\mathcal{I}}(q \wedge (r \rightarrow s))$

- 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*

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

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*

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

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*

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

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*

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

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: $q \wedge (r \rightarrow s \wedge \mathit{true})$

Simplification Examples

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

Simplification Examples

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

Simplification Examples

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

Simplification Examples

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

Simplification Examples

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

Simplification Examples

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

Simplification Examples

- To compute $(q \wedge (r \rightarrow s \wedge p))[p]$
  - Do the replacement: $q \wedge (r \rightarrow s \wedge \mathit{true})$
  - Then simplify $q \wedge (r \rightarrow s \wedge \mathit{true}) \mapsto q \wedge (r \rightarrow s)$
- 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: $q \wedge (r \rightarrow s \wedge \mathit{false})$
  - Then simplify $q \wedge (r \rightarrow s \wedge \mathit{false}) \mapsto q \wedge (r \rightarrow \mathit{false}) \mapsto q \wedge \neg r$
- 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 logicaly 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 logicaly equivalent to the next, due to the preceding lemma. □

Simplification Rules

We add the following four "simplification rules" to LK:

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

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

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

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

Example: (one-sided) LK with Simplification Rules

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

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{p, (\neg p \vee q)[p], p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow} \qquad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{p, \qquad\qquad q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow} \qquad\qquad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{p, \qquad\qquad q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow} \qquad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{\dfrac{p,\ q,\ (p \vee \neg q)[p], \neg p \vee \neg q \ \Rightarrow}{p, \qquad q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow} \qquad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}$$

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{\dfrac{p,\ q,\quad true, \neg p \vee \neg q \ \Rightarrow}{p,\quad q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow} \qquad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}$$

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{\dfrac{p,\ q,\quad true, \neg p \vee \neg q \ \Rightarrow}{p,\quad q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow} \qquad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}$$

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{\dfrac{\dfrac{p,\ q,\ true, (\neg p \vee \neg q)[p] \ \Rightarrow}{p,\ q,\ true, \neg p \vee \neg q \ \Rightarrow}}{p,\ q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow} \qquad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}$$

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{\dfrac{\dfrac{p,\ q,\ true,\ \neg q\ \Rightarrow}{p,\ q,\ true, \neg p \vee \neg q\ \Rightarrow}}{p,\ q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow} \qquad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}$$

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{\dfrac{\dfrac{p, q, true, \neg q \Rightarrow}{p, q, true, \neg p \vee \neg q \Rightarrow}}{p, q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow} \qquad \dfrac{}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \Rightarrow}$$

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p,\ q,\ \mathit{true},\ \neg q\ \Rightarrow}}{p,\ q,\ \mathit{true},\ \neg p \vee \neg q\ \Rightarrow}}{p,\ q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}}{\dfrac{p,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow \qquad q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}{p \vee q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}}$$

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p,\ q,\ \mathit{true},\ \neg q\ \Rightarrow}}{p,\ q,\ \mathit{true},\ \neg p \vee \neg q\ \Rightarrow}}{p,\ q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{}{q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}}{p \vee q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}$$

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p,\ q,\ \mathit{true},\ \neg q\ \Rightarrow}}{p,\ q,\ \mathit{true},\ \neg p \vee \neg q\ \Rightarrow}}{p,\ q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{}{q,\ \mathit{true},\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}}{\dfrac{p,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow \qquad q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}{p \vee q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}}$$

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p,\ q,\ \mathit{true},\ \neg q\ \Rightarrow}}{p,\ q,\ \mathit{true},\ \neg p \vee \neg q\ \Rightarrow}}{p,\ q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{q,\ \mathit{true},\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}{q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}}{p \vee q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}$$

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p,\ q,\ true,\ \neg q\ \Rightarrow}}{p,\ q,\ true, \neg p \vee \neg q\ \Rightarrow}}{p,\ q, \neg p \vee \neg q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{\dfrac{\dfrac{}{q,\ true,\ p, \neg p \vee \neg q\ \Rightarrow}}{q, true, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}$$

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p,\ q,\ \mathit{true},\ \neg q\ \Rightarrow}}{p,\ q,\ \mathit{true},\ \neg p \vee \neg q\ \Rightarrow}}{p,\ q,\ \mathit{true},\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}}{p,\ q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{\dfrac{\dfrac{}{q,\ \mathit{true},\ p,\ \neg p \vee \neg q\ \Rightarrow}}{q,\ \mathit{true},\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}}{q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}$$

$$\dfrac{p,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow \qquad q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}{p \vee q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}$$

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p,\ q,\ true,\ \neg q\ \Rightarrow}}{p,\ q,\ true, \neg p \vee \neg q\ \Rightarrow}}{p,\ q, \neg p \vee q, \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{\dfrac{\dfrac{}{q,\ true, p, \neg p\ \Rightarrow}}{q,\ true,\ p, \neg p \vee \neg q\ \Rightarrow}}{q, true, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}$$

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

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p,\ q,\ true,\ \neg q\ \Rightarrow}}{p,\ q,\ true, \neg p \vee \neg q\ \Rightarrow}}{p,\ q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{\dfrac{\dfrac{}{q,\ true,\ p, \neg p\ \Rightarrow}}{q,\ true,\ p, \neg p \vee \neg q\ \Rightarrow}}{q, true, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}$$

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

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p,\ q,\ true,\ \neg q\ \Rightarrow}}{p,\ q,\ true, \neg p \vee \neg q\ \Rightarrow}}{p,\ q, \neg p \vee \neg q, \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{\dfrac{\dfrac{}{q,\ true, p, \neg p\ \Rightarrow}}{q,\ true,\ p, \neg p \vee \neg q\ \Rightarrow}}{q, true, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}$$

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

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p,\ q,\ true,\ \neg q\ \Rightarrow}}{p,\ q,\ true, \neg p \vee \neg q\ \Rightarrow}}{p,\ q, \neg p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{p, \neg p \vee q, \neg p \vee \neg q, \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{\dfrac{\dfrac{}{q,\ true, p, \neg p\ \Rightarrow}}{q,\ true,\ p, \neg p \vee \neg q\ \Rightarrow}}{q, true, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}$$

$$\dfrac{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow \qquad q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}$$

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

Example: (one-sided) LK with Simplification Rules

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p,\ q,\ true,\ \neg q\ \Rightarrow}}{p,\ q,\ true, \neg p \vee \neg q\ \Rightarrow}}{p,\ q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{p, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{q,\ true, p, \neg p\ \Rightarrow}}{q,\ true,\ p, \neg p \vee \neg q\ \Rightarrow}}{q, true, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}$$

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

- 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, $C = A_1 \vee \cdots \vee A_k \vee L$

Simplification for Clauses

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

Simplification for Clauses

- Simplify a clause C with a literal L
- Case 1: C contains L, $C = A_1 \vee \cdots \vee A_k \vee L$
  - Then $C[L] = \mathit{true}$
  - 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, $C = A_1 \vee \cdots \vee A_k \vee L$
  - Then $C[L] = \mathit{true}$
  - 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, $C = A_1 \vee \cdots \vee A_k \vee L$
  - Then $C[L] = \mathit{true}$
  - 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, $C = A_1 \vee \cdots \vee A_k \vee L$
  - Then $C[L] = \mathit{true}$
  - 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 $\bar{L}$, $C = A_1 \vee \cdots \vee A_k \vee \bar{L}$

Simplification for Clauses

- Simplify a clause C with a literal L
- Case 1: C contains L, $C = A_1 \vee \cdots \vee A_k \vee L$
  - Then $C[L] = \mathit{true}$
  - 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 $\bar{L}$, $C = A_1 \vee \cdots \vee A_k \vee \bar{L}$
  - Then $C[L] =$

Simplification for Clauses

- Simplify a clause C with a literal L
- Case 1: C contains L, $C = A_1 \vee \cdots \vee A_k \vee L$
  - Then $C[L] = \mathit{true}$
  - 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 $\bar{L}$, $C = A_1 \vee \cdots \vee A_k \vee \bar{L}$
  - Then $C[L] = A_1 \vee \cdots \vee A_k$

Simplification for Clauses

- Simplify a clause C with a literal L
- Case 1: C contains L, $C = A_1 \vee \cdots \vee A_k \vee L$
  - Then $C[L] = \mathit{true}$
  - 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 $\bar{L}$, $C = A_1 \vee \cdots \vee A_k \vee \bar{L}$
  - Then $C[L] = A_1 \vee \cdots \vee A_k$
  - $C[L]$ is the resolvent of C and L!

Simplification for Clauses

- Simplify a clause C with a literal L
- Case 1: C contains L, $C = A_1 \vee \cdots \vee A_k \vee L$
  - Then $C[L] = \mathit{true}$
  - 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 $\bar{L}$, $C = A_1 \vee \cdots \vee A_k \vee \bar{L}$
  - Then $C[L] = A_1 \vee \cdots \vee A_k$
  - $C[L]$ is the resolvent of C and L!
  - Replacing C by $A_1 \vee \cdots \vee A_k$ is called unit resolution

Simplification for Clauses

- Simplify a clause C with a literal L
- Case 1: C contains L, $C = A_1 \vee \cdots \vee A_k \vee L$
  - Then $C[L] = \mathit{true}$
  - 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 $\bar{L}$, $C = A_1 \vee \cdots \vee A_k \vee \bar{L}$
  - Then $C[L] = A_1 \vee \cdots \vee A_k$
  - $C[L]$ is the resolvent of C and L!
  - Replacing C by $A_1 \vee \cdots \vee A_k$ is called unit resolution
  - Note that C is subsumed by $A_1 \vee \cdots \vee A_k$

Simplification for Clauses

- Simplify a clause C with a literal L
- Case 1: C contains L, $C = A_1 \vee \cdots \vee A_k \vee L$
  - Then $C[L] = \mathit{true}$
  - 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 $\bar{L}$, $C = A_1 \vee \cdots \vee A_k \vee \bar{L}$
  - Then $C[L] = A_1 \vee \cdots \vee A_k$
  - $C[L]$ is the resolvent of C and L!
  - Replacing C by $A_1 \vee \cdots \vee A_k$ is called unit resolution
  - Note that C is subsumed by $A_1 \vee \cdots \vee A_k$
- Unit subsumption and unit resolution together: unit propagation

Simplification for Clauses

- Simplify a clause C with a literal L
- Case 1: C contains L, $C = A_1 \vee \cdots \vee A_k \vee L$
  - Then $C[L] = \mathit{true}$
  - 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 $\bar{L}$, $C = A_1 \vee \cdots \vee A_k \vee \bar{L}$
  - Then $C[L] = A_1 \vee \cdots \vee A_k$
  - $C[L]$ is the resolvent of C and L!
  - Replacing C by $A_1 \vee \cdots \vee A_k$ is called unit resolution
  - Note that C is subsumed by $A_1 \vee \cdots \vee A_k$
- 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:

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

The Cut Rule

- The cut rule for LK:

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

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

The Cut Rule

- The cut rule for LK:

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

- 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:

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

- 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:

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

- 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:

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

- 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:

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

- 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

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

Atomic Cut

- The atomic cut rule is just the cut rule

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

- Where A is resricted to be an atomic formula.

Atomic Cut

- The atomic cut rule is just the cut rule

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

- Where A is resricted to be an atomic formula.
- No nonelementary speedup :-(

Atomic Cut

- The atomic cut rule is just the cut rule

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

- Where A is resricted 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

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

- Where A is resricted 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

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

- Where A is resricted 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

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

- Where A is resricted 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

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

- Where A is resricted 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

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

- Where A is resricted 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:

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

Atomic Cut

- The atomic cut rule is just the cut rule

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

- Where A is resricted 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:

$$\frac{A, \Gamma \Rightarrow \qquad \neg A, \Gamma \Rightarrow}{\Gamma \Rightarrow} \qquad \frac{\Rightarrow A, \Delta \qquad \Rightarrow \neg A, \Delta}{\Rightarrow \Delta}$$

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

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

Atomic Cut + Unit Propagation

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

Atomic Cut + Unit Propagation

$$\dfrac{\dfrac{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow} \qquad \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

Atomic Cut + Unit Propagation

$$\dfrac{\dfrac{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow} \qquad \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

Atomic Cut + Unit Propagation

$$\dfrac{\dfrac{\dfrac{}{q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow} \qquad \dfrac{}{\neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

Atomic Cut + Unit Propagation

$$\dfrac{\dfrac{\dfrac{q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}}{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow} \qquad \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \;\Rightarrow}$$

Atomic Cut + Unit Propagation

$$\dfrac{\dfrac{\dfrac{\dfrac{}{q,\ p,\ \neg p \vee \neg q \ \Rightarrow}}{q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow} \qquad \dfrac{}{\neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}$$

Atomic Cut + Unit Propagation

$$\dfrac{\dfrac{\dfrac{\dfrac{}{q,\ p,\ \neg p \vee \neg q \ \Rightarrow}}{q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow} \qquad \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q \ \Rightarrow}$$

Atomic Cut + Unit Propagation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{q,\ p,\ \neg p\ \Rightarrow}}{q,\ p,\ \neg p \vee \neg q\ \Rightarrow}}{q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow} \qquad \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}$$

Atomic Cut + Unit Propagation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{q,\ p,\ \neg p\ \Rightarrow}}{q,\ p,\ \neg p \vee \neg q\ \Rightarrow}}{q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow} \qquad \neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}$$

Atomic Cut + Unit Propagation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{q,\ p,\ \neg p\ \Rightarrow}}{q,\ p,\ \neg p \vee \neg q\ \Rightarrow}}{q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}}{q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}}{q,\ p \vee q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow} \qquad \neg q,\ p \vee q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow$$

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

Atomic Cut + Unit Propagation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{q,\ p,\ \neg p\ \Rightarrow}}{q,\ p,\ \neg p \vee \neg q\ \Rightarrow}}{q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}}{q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}}{q,\ p \vee q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow} \qquad \neg q,\ p \vee q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow$$

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

Atomic Cut + Unit Propagation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{q,\ p,\ \neg p\ \Rightarrow}}{q,\ p,\ \neg p \vee \neg q\ \Rightarrow}}{q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}}{q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{}{\neg q,\ p,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}}{q,\ p \vee q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{}{\neg q,\ p \vee q,\ \neg p \vee q,\ p \vee \neg q,\ \neg p \vee \neg q\ \Rightarrow}$$

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

Atomic Cut + Unit Propagation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{q,\ p,\ \neg p\ \Rightarrow}}{q,\ p,\ \neg p \vee \neg q\ \Rightarrow}}{q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{\neg q,\ p,\ \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}{\neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}$$

Atomic Cut + Unit Propagation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{q,\ p,\ \neg p\ \Rightarrow}}{q,\ p,\ \neg p \vee \neg q\ \Rightarrow}}{q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\neg q,\ p,\ \neg p, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{\neg q,\ p,\ \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{\neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}$$

Atomic Cut + Unit Propagation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{q,\ p,\ \neg p\ \Rightarrow}}{q,\ p,\ \neg p \vee \neg q\ \Rightarrow}}{q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{\dfrac{\dfrac{}{\neg q,\ p,\ \neg p, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{\neg q,\ p,\ \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{}{\neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}$$

Atomic Cut + Unit Propagation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{q,\ p,\ \neg p\ \Rightarrow}}{q,\ p,\ \neg p \vee \neg q\ \Rightarrow}}{q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{\dfrac{\dfrac{}{\neg q,\ p,\ \neg p, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{\neg q,\ p,\ \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow} \qquad \dfrac{}{\neg q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}}{q, p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\ \Rightarrow}$$

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

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?

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

Lemma Generation

- Remember the exercise sheet 2?

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

- Closing the left branch, we "learnt the Lemma A"

Lemma Generation

- Remember the exercise sheet 2?

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

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

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

Lemma Generation

- Remember the exercise sheet 2?

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

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

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

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

Lemma Generation

- Remember the exercise sheet 2?

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

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

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

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

Lemma Generation

- Remember the exercise sheet 2?

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

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

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

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

Lemma Generation

- Remember the exercise sheet 2?

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

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

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

- We refuted A, so now we may assume $\neg A$.
- Whenever we close a branch, we learn that a certain combination of literals $L_1, \ldots, L_k$ leads to a contradiction
- We can add a clause $\overline{L_1} \vee \cdots \vee \overline{L_k}$ 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 ≡ Backjumping ≡ Intelligent Backtracking ≡ Non-chronological Backtracking
- Consider the following derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{needed B and G}}{p, q, \neg p, \neg r \;\Rightarrow} \qquad \dfrac{\text{needed G}}{p, q, r, \neg r \;\Rightarrow}}{p, q, \neg p \vee r, \neg r \;\Rightarrow}\,\mathsf{G} \qquad p, s, \ldots \;\Rightarrow}{p, q \vee s, \neg p \vee r, \neg r \;\Rightarrow}\,\mathsf{R} \qquad q, \ldots \;\Rightarrow}{p \vee q, q \vee s, \neg p \vee r, \neg r \;\Rightarrow}\,\mathsf{B}$$

Pruning

- Pruning ≡ Backjumping ≡ Intelligent Backtracking ≡ Non-chronological Backtracking
- Consider the following derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{needed B and G}}{p, q, \neg p, \neg r \;\Rightarrow} \qquad \dfrac{\text{needed G}}{p, q, r, \neg r \;\Rightarrow}}{p, q, \neg p \vee r, \neg r \;\Rightarrow}\,\mathsf{G} \qquad p, s, \ldots \;\Rightarrow}{p, q \vee s, \neg p \vee r, \neg r \;\Rightarrow}\,\mathsf{R} \qquad q, \ldots \;\Rightarrow}{p \vee q, q \vee s, \neg p \vee r, \neg r \;\Rightarrow}\,\mathsf{B}$$

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

Pruning

- Pruning ≡ Backjumping ≡ Intelligent Backtracking ≡ Non-chronological Backtracking
- Consider the following derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{needed B and G}}{p, q, \neg p, \neg r \;\Rightarrow} \quad \dfrac{\text{needed G}}{p, q, r, \neg r \;\Rightarrow}}{p, q, \neg p \vee r, \neg r \;\Rightarrow}\,\mathsf{G} \quad p, s, \ldots \;\Rightarrow}{p, q \vee s, \neg p \vee r, \neg r \;\Rightarrow}\,\mathsf{R} \quad q, \ldots \;\Rightarrow}{p \vee q, q \vee s, \neg p \vee r, \neg r \;\Rightarrow}\,\mathsf{B}$$

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

Pruning

- Pruning ≡ Backjumping ≡ Intelligent Backtracking ≡ Non-chronological Backtracking
- Consider the following derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{needed B and G}}{p, q, \neg p, \neg r \;\Rightarrow} \quad \dfrac{\text{needed G}}{p, q, r, \neg r \;\Rightarrow}}{p, q, \neg p \vee r, \neg r \;\Rightarrow}\,\mathrm{G} \quad p, s, \ldots \;\Rightarrow}{p, q \vee s, \neg p \vee r, \neg r \;\Rightarrow}\,\mathrm{R} \quad q, \ldots \;\Rightarrow}{p \vee q, q \vee s, \neg p \vee r, \neg r \;\Rightarrow}\,\mathrm{B}$$

- 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

$$\dfrac{\dfrac{\dfrac{\text{needed B and G}}{p, \neg p, \neg r \;\Rightarrow} \quad \dfrac{\text{needed G}}{p, r, \neg r \;\Rightarrow}}{p, \neg p \vee r, \neg r \;\Rightarrow}\,\mathrm{G} \quad q, \ldots \;\Rightarrow}{p \vee q, q \vee s, \neg p \vee r, \neg r \;\Rightarrow}\,\mathrm{B}$$

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:

$$\dfrac{\Gamma_1 \cap \Gamma_2 \;\Rightarrow}{\begin{array}{cc} \Gamma_1 \;\Rightarrow & \Gamma_2 \;\Rightarrow \\ \vdots & \vdots \\ A, \Gamma \;\Rightarrow & \neg A, \Gamma \;\Rightarrow \end{array}}{\Gamma \;\Rightarrow}$$

Stålmarck's Method

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

$$\dfrac{\dfrac{\Gamma_1 \cap \Gamma_2 \;\Rightarrow}{\Gamma_1 \;\Rightarrow} \qquad \dfrac{\Gamma_1 \cap \Gamma_2 \;\Rightarrow}{\Gamma_2 \;\Rightarrow}}{\dfrac{\vdots}{A, \Gamma \;\Rightarrow} \qquad \dfrac{\vdots}{\neg A, \Gamma \;\Rightarrow}}{\Gamma \;\Rightarrow}$$

- After unit propagation, join branches generated by cut

Stålmarck's Method

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

$$\dfrac{\Gamma_1 \cap \Gamma_2 \;\Rightarrow}{\dfrac{\Gamma_1 \;\Rightarrow}{\vdots}\;\dfrac{\Gamma_2 \;\Rightarrow}{\vdots}}$$
$$\dfrac{A, \Gamma \;\Rightarrow \qquad \neg A, \Gamma \;\Rightarrow}{\Gamma \;\Rightarrow}$$

- 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:

$$\dfrac{\Gamma_1 \cap \Gamma_2 \;\Rightarrow}{\dfrac{\Gamma_1 \;\Rightarrow \qquad\qquad \Gamma_2 \;\Rightarrow}{\begin{array}{cc} \vdots & \vdots \\ A, \Gamma \;\Rightarrow & \neg A, \Gamma \;\Rightarrow \end{array}}}{\Gamma \;\Rightarrow}$$

- 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:

$$\dfrac{\Gamma_1 \cap \Gamma_2 \;\Rightarrow}{\begin{array}{cc} \Gamma_1 \;\Rightarrow & \Gamma_2 \;\Rightarrow \\ \vdots & \vdots \\ A, \Gamma \;\Rightarrow & \neg A, \Gamma \;\Rightarrow \end{array}}{\Gamma \;\Rightarrow}$$

- 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:

$$\dfrac{\Gamma_1 \cap \Gamma_2 \;\Rightarrow}{\dfrac{\Gamma_1 \;\Rightarrow \qquad\qquad \Gamma_2 \;\Rightarrow}{\begin{array}{cc} \vdots & \vdots \\ A, \Gamma \;\Rightarrow & \neg A, \Gamma \;\Rightarrow \end{array}}}{\Gamma \;\Rightarrow}$$

- 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:

$$\dfrac{\Gamma_1 \cap \Gamma_2 \;\Rightarrow}{\dfrac{\Gamma_1 \;\Rightarrow \qquad\qquad \Gamma_2 \;\Rightarrow}{\begin{array}{cc} \vdots & \vdots \\ A, \Gamma \;\Rightarrow & \neg A, \Gamma \;\Rightarrow \end{array}}}{\Gamma \;\Rightarrow}$$

- 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