Propositional Resolution

Marco Piastra

Deductive systems and automation

- Is problem $\Gamma \vdash \varphi$ *decidible*?

  A deductive system 'a la Hilbert' (i.e. derivation using axiom schemas and MP) does <u>not</u> translate into an algorithm

  In fact, when trying to find a demonstration of $\Gamma \vdash \varphi$:

  We can use all $\psi \in \Gamma$ (if Γ is finite) (OK)

  We can apply the inference rule MP whenever possible (OK)

  We <u>cannot</u> generate all axiom *instances* from Axn (KO)

  Moral: *the problem is the infinite set of axioms*

Resolution rule

(*Just another inference rule*)

$$\varphi \lor \chi, \neg\chi \lor \psi \vdash \varphi \lor \psi$$

$\varphi \lor \psi$ is also called the *resolvent* of $\varphi \lor \chi$ e $\neg\chi \lor \psi$

The resolution rule is *correct*

φ	ψ	χ	$\varphi \lor \chi$	$\neg\chi \lor \psi$	$\varphi \lor \psi$
0	0	0	0	1	0
0	0	1	1	0	0
0	1	0	0	1	1
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	1

The rules *MP* can be seen a special case *resolution*

$\chi \rightarrow \psi, \chi \vdash \psi$ can be rewritten as $\chi, \neg\chi \lor \psi \vdash \psi$

Normal forms

= translation of each wff into an equivalent wff having a specific structure

- **Conjunctive Normal Form** (CNF)

  A wff with a structure

  $\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n$

  where each α_i has a structure

  $(\beta_1 \vee \beta_2 \vee \ldots \vee \beta_n)$

  where each β_j is a *literal* (i.e. an atomic symbol or the negation of an atomic symbol)

  Examples:

  $(B \vee D) \wedge (A \vee \neg C) \wedge C$

  $(B \vee \neg A \vee \neg C) \wedge (\neg D \vee \neg A \vee \neg C)$

- **Disjunctive Normal Form** (DNF)

  A wff with a structure

  $\beta_1 \vee \beta_2 \vee \ldots \vee \beta_n$

  where each β_i has a structure

  $(\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n)$

  where each α_j is a *literal*

Conjunctive Normal Form

- **Translation into CNF** (it can be automated)

  Exhaustive application of the following rules:

  1) Rewrite $\rightarrow$ and $\leftrightarrow$ using $\wedge$, $\vee$, $\neg$

  2) Move $\neg$ inside composite formulae

     "De Morgan laws": $\neg(\varphi \wedge \psi) \equiv (\neg\varphi \vee \neg\psi)$

     $\neg(\varphi \vee \psi) \equiv (\neg\varphi \wedge \neg\psi)$

  3) Eliminate double negations: $\neg\neg$

  4) Distribute $\vee$

     $$((\varphi \wedge \psi) \vee \chi) \equiv ((\varphi \vee \chi) \wedge (\psi \vee \chi))$$

Examples:

$(\neg B \rightarrow D) \vee \neg(A \wedge C)$

$B \vee D \vee \neg(A \wedge C)$	(rewrite $\rightarrow$)
$B \vee D \vee \neg A \vee \neg C$	(De Morgan)

$\neg(B \rightarrow D) \vee \neg(A \wedge C)$

$\neg(\neg B \vee D) \vee \neg(A \wedge C)$	(rewrite $\rightarrow$)
$(B \wedge \neg D) \vee (\neg A \vee \neg C)$	(De Morgan)
$(B \vee \neg A \vee \neg C) \wedge (\neg D \vee \neg A \vee \neg C)$	(distribute $\vee$)

Clausal Forms

= each wff is translated into an equivalent <u>set</u> of wffs having a specific structure

- **Clausal Form** (CF)

  Starting from a wff in CNF

  $\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n$

  the clausal form is simply the set of all *clauses*

  $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$

  Examples:

  $(B \vee D) \wedge (A \vee \neg C) \wedge C$

  $\{(B \vee D), (A \vee \neg C), C\}$

- **Special notation**

  Each clause is usually written as a *set*

  $\beta_1 \vee \beta_2 \vee \ldots \vee \beta_n$

  $\{\beta_1, \beta_2, \ldots, \beta_n\}$

  Example:

  $\{\{B, D\}, \{A, \neg C\}, \{C\}\}$

  A set of *literals*:
  ordering is irrelevant
  no multiple copies

Resolution by refutation

- Algorithm
  - Problem: "$\Gamma \vdash \varphi$" ?
  - The problem is transformed into: is "$\Gamma \cup \{\neg\varphi\}$" *coherent*?
    - If $\Gamma \vdash \varphi$ then $\Gamma \cup \{\neg\varphi\}$ is incoherent and therefore a contradiction can be derived
  - $\Gamma \cup \{\neg\varphi\}$ is translated into CNF hence in CF

The resolution algorithm is applied to the set of *clauses* $\Gamma \cup \{\neg\varphi\}$

At each step:

a) Select a pair of clauses $\{C_1, C_2\}$ containing a pair of *complementary literals* making sure that this combination has never been selected before

b) Compute C as the *resolvent* of $\{C_1, C_2\}$ according to the resolution rule.

c) Add C to the set of clauses

Termination:

When C is the empty clause $\{\ \}$

or there are no more combinations to be selected in step a)

Advantages:

No axioms. Only one operation (i.e. the resolution rule). It is a *native* algorithm

Resolution by refutation

- The same example as before

  $B \lor D \lor \neg A \lor \neg C, B \lor C, \; A \lor D, \; \neg B \; \vdash \; D$

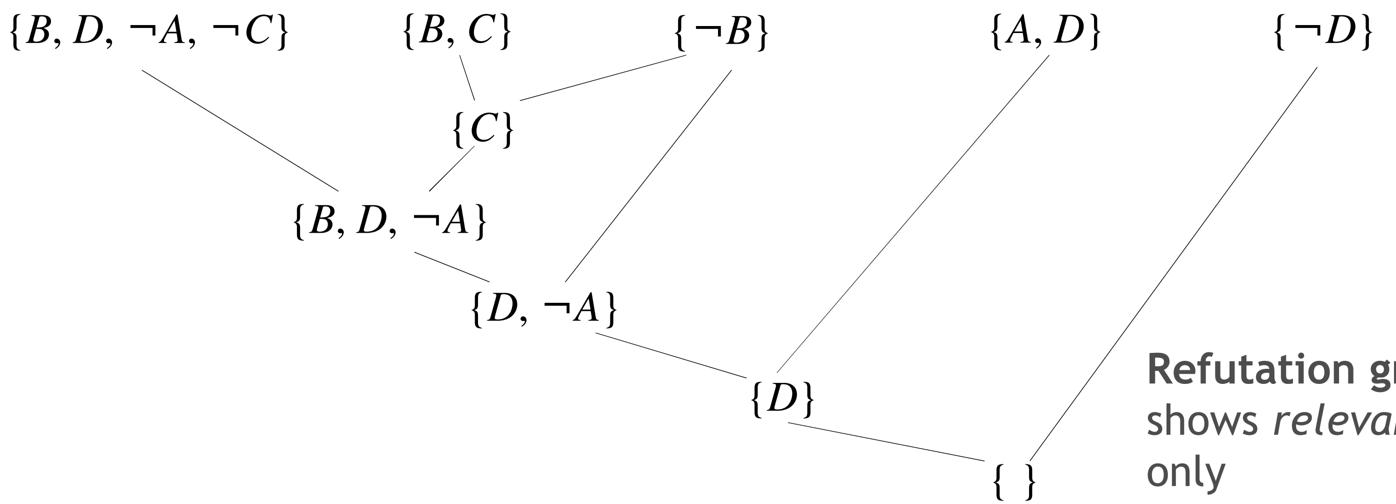
  Refutation + rewrite in CNF:

  $B \lor D \lor \neg A \lor \neg C, B \lor C, \; A \lor D, \; \neg B, \neg D$

  Rewrite in CF:

  $\{B, D, \neg A, \neg C\}, \{B, C\}, \{A, D\}, \{\neg B\}, \{\neg D\}$

  Applying the resolution rule:



Refutation graph:
shows *relevant* resolutions only

Resolution by refutation

- The same example as before

  $B \vee D \vee \neg A \vee \neg C, B \vee C, \; A \vee D, \; \neg B \; \vdash \; D$

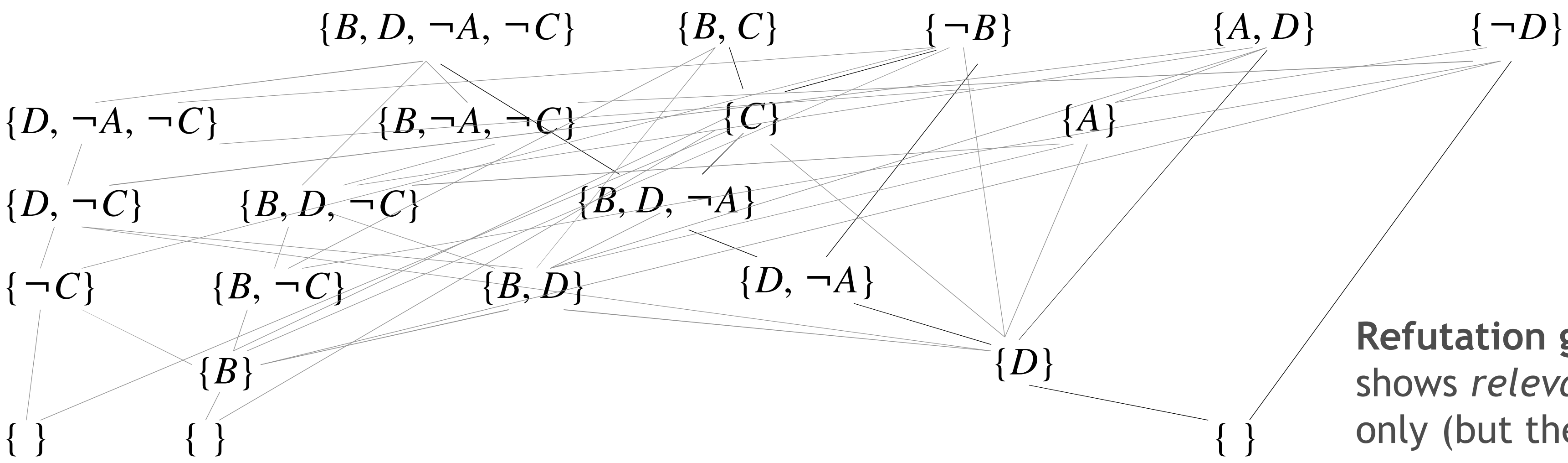
  Refutation + rewrite in CNF:

  $B \vee D \vee \neg A \vee \neg C, B \vee C, \; A \vee D, \; \neg B, \neg D$

  Rewrite in CF:

  $\{B, D, \neg A, \neg C\}, \{B, C\}, \{A, D\}, \{\neg B\}, \{\neg D\}$

  Applying the resolution rule:



Refutation graph: shows *relevant* resolutions only (but there are more)

Resolution by refutation

- Resolution by refutation for propositional logic

  Is correct: $\Gamma \vdash \varphi \;\Rightarrow\; \Gamma \models \varphi$

  Is complete: $\Gamma \models \varphi \;\Rightarrow\; \Gamma \vdash \varphi$

  In this sense: if $\Gamma \models \varphi$ then there exists a refutation graph

- Algorithm

  It is a decision procedure for the problem $\Gamma \models \varphi$

  It has time complexity $O(2^n)$

  where n is the number of propositional symbols in $\Gamma \cup \{\neg\varphi\}$