Artificial Intelligence

A course About Foundations

Automated Symbolic Calculus

Marco Piastra

Resolution by Refutation

Inference rule: Resolution

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

$\varphi \vee \psi$ is also called the *resolvent* of $\varphi \vee \chi$ and $\neg\chi \vee \psi$

The resolution rule is *correct*

That is, $\varphi \vee \chi, \neg\chi \vee \psi \vdash \varphi \vee \psi \quad \Rightarrow \quad \varphi \vee \chi, \neg\chi \vee \psi \models \varphi \vee \psi$

Proof:

φ	ψ	χ	$\varphi \vee \chi$	$\neg\chi \vee \psi$	$\varphi \vee \psi$
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

Normal forms

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

- **Conjunctive Normal Form** (CNF)

  A wff with a structure

  $\alpha_1 \land \alpha_2 \land \ldots \land \alpha_n$

  where each α_i has a structure

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

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

  Examples:

  $(B \lor D) \land (A \lor \neg C) \land C$

  $(B \lor \neg A \lor \neg C) \land (\neg D \lor \neg A \lor \neg C)$

- **Disjunctive Normal Form** (DNF)

  A wff with a structure

  $\beta_1 \lor \beta_2 \lor \ldots \lor \beta_n$

  where each β_i has a structure

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

  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

  $\alpha_1 \land \alpha_2 \land \ldots \land \alpha_n$

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

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

  Examples:

  $(B \lor D) \land (A \lor \neg C) \land C$

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

- **Set Notation**

  Each clause is transformed into a *set of literals*

  $\beta_1 \lor \beta_2 \lor \ldots \lor \beta_n$

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

  Example:

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

***Literals*:**
each β_i is either an *atom*
or the *negation* of an atom

A <u>set</u> of literals:
ordering is irrelevant
no multiple copies

Resolution by refutation

- An example:

$$\neg(\neg(\neg B \to D) \land \neg(A \to \neg C)),\ \neg B \to C,\ A \lor D,\ \neg B \vdash D$$

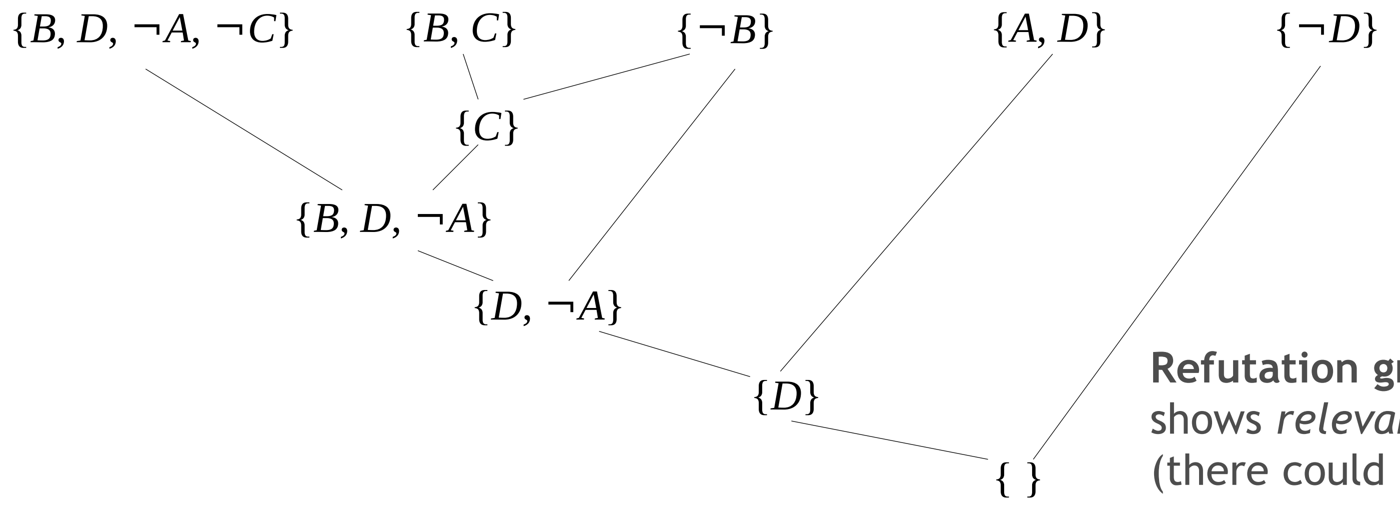
Refutation + rewrite in CNF:

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

Rewrite in CF:

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

Applying the resolution rule, _one complementary pair of literals_ at time:



Refutation graph:
shows *relevant* resolutions only
(there could be more)

Resolution by refutation

- An example:

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

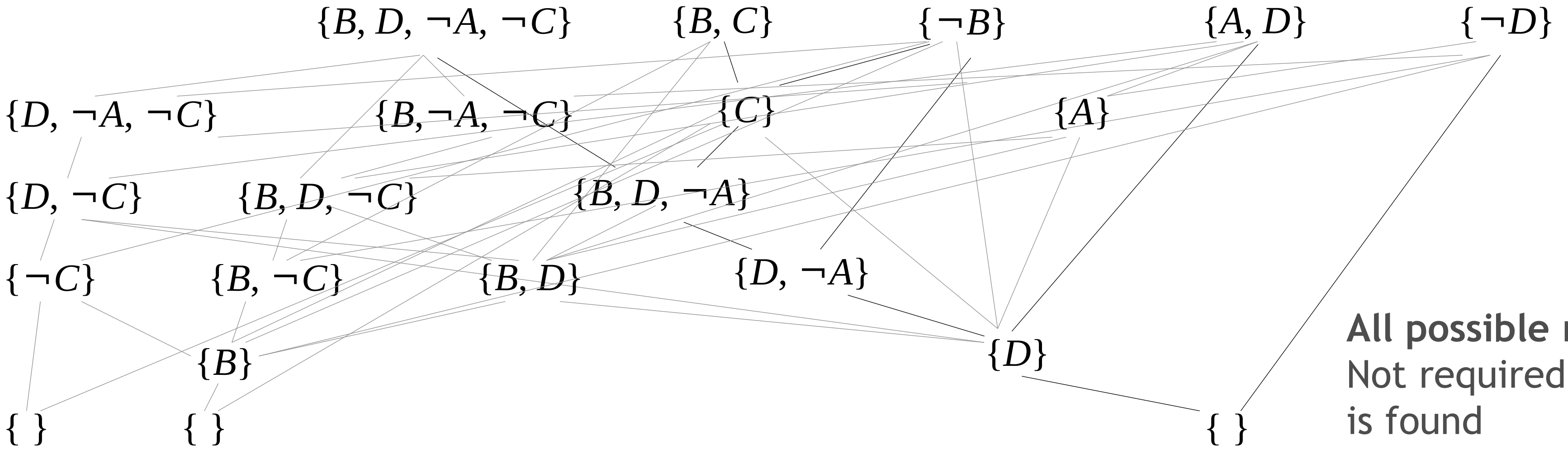
  Refutation + rewrite in CNF:

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

  Rewrite in CF:

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

  Applying the resolution rule:



All possible resolutions
Not required, once an *empty clause* is found

Resolution by refutation

- Algorithm

  Problem: "$\Gamma \vdash \varphi$" ?

  The problem is transformed into: is "$\Gamma \cup \{\neg\varphi\}$" *coherent (=satisfiable)*?

  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 into CF, in set notation

  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 such pair has never been selected before

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

  c) Add C_r to the set of clauses

  Termination:

  When C_r is the empty clause $\{\ \}$ *(success)*

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

Resolution by refutation - Algorithm properties

- **Termination**

  The algorithm never *diverges* (it never enters an infinite loop)

  Each application of the resolution rule 'uses' a pair of clauses in a finite set, each resolvent is smaller than the union of the resolved clauses. Therefore, it cannot go on forever

- ***Symbolic derivation***

  As already stated, despite its name, this is a *symbolic* method

  We write

  $$\Gamma \vdash \varphi$$

  iff the resolution method is successful (= the empty clause is derived) for $\Gamma \cup \{\neg\varphi\}$

  How do we know that $\Gamma \vdash \varphi \;\Rightarrow\; \Gamma \models \varphi$?

  (*Soundness* - also *correctness* - of the method)

  Exercise: prove it
  (*hint*: consider the condition on $\Gamma \cup \{\neg\varphi\}$ and think about how it relates to the *inference rule*)

  How do we know that $\Gamma \models \varphi \;\Rightarrow\; \Gamma \vdash \varphi$?

  (*Completeness* of the method)

  Proving it is a bit more difficult: see textbook (Ben-Ari's)

Resolution by refutation - Algorithm properties

- **Termination**

  The algorithm never *diverges* (it never enters an infinite loop)

  Each application of the resolution rule 'uses' a pair of clauses in a finite set, each resolvent is smaller than the union of the resolved clauses. Therefore, it cannot go on forever

- ***Soundness***

  $\Gamma \vdash_{ST} \varphi \;\Rightarrow\; \Gamma \models \varphi$

- **Completeness**

  $\Gamma \models \varphi \;\Rightarrow\; \Gamma \vdash_{ST} \varphi$

- **Termination + Soundness + Completeness = *Decision Algorithm***

  (for propositional logic)

Resolution by refutation

- Resolution by refutation for propositional logic

  Is correct: $\Gamma \vdash_{RES} \varphi \;\Rightarrow\; \Gamma \models \varphi$

  Is complete: $\Gamma \models \varphi \;\Rightarrow\; \Gamma \vdash_{RES} \varphi$

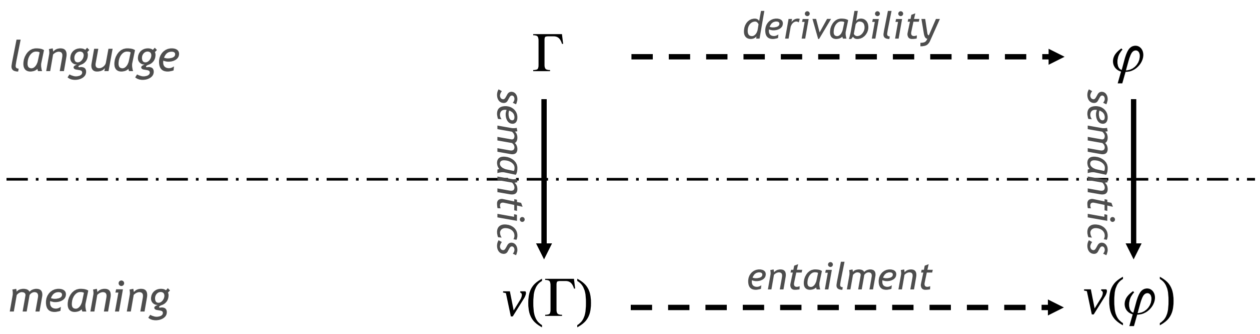
  In this sense: iff $\Gamma \models \varphi$ then there exists a refutation graph

- 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 $\Gamma \cup \{\neg\varphi\}$



derivability
language
Γ
φ
semantics
semantics
entailment
meaning
v(Γ)
v(φ)