Informatics 2D · Agents and Reasoning · 2019/2020

Lecture 13 · Resolution-Based Inference

Claudia Chirita

School of Informatics, University of Edinburgh

11th February 2020

Based on slides by: Jacques Fleuriot, Michael Rovatsos, Michael Herrmann, Vaishak Belle

Previously on INF2D

Backward chaining

- if `Goal` is known (goal directed)
- can query for data

Forward chaining

- if specific `Goal` is not known, but the system needs to react to new facts (data driven)
- can make suggestions

What do users expect from the system?

Which direction has the larger branching factor?

Limitations

...due to restriction to definite clauses

In order to apply GMP

- premises of rules contain only non-negated symbols
- the conclusion of any rule is a non-negated symbol
- facts are non-negated atomic sentences

Possible solution: introduce more variables, e.g. $Q := \neg P$

What about: “If we cannot prove A, then $\neg A$ is true”?
(works only if there is a rule for each variable)

Resolution one more time

- Negate query α.
- Convert everything to CNF.
- Repeat: Choose clauses and resolve (based on unification).
- If resolution results in empty clause, α is proved.
- Return all substitutions (or Fail).

Ground Binary Resolution & Modus Ponens

Ground binary resolution

$$\frac{C \vee P \qquad D \vee \neg P}{C \vee D}$$

Suppose $C = \texttt{False}$.

$$\frac{P \qquad \neg P \vee D}{D}$$

i.e. P and $P \rightarrow D$ entails D.

Modus ponens is a special case of binary resolution.

Full Resolution & Generalised Modus Ponens

GMP with $p_i'\theta = p_i\theta$

$$\frac{p_1', p_2', \ldots, p_n' \quad (p_1 \wedge p_2 \wedge \ldots \wedge p_n \rightarrow q)}{q\theta}$$

$$\frac{p_1', p_2', \ldots, p_n' \quad (q \vee \neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n)}{q\theta}$$

Full resolution with θ mgu of all P_i and P_j'

$$\frac{C \vee P_1 \vee \ldots \vee P_m \qquad D \vee \neg P_1' \vee \ldots \vee \neg P_n'}{(C \vee D)\ \theta}$$

Resolution in Implication Form

Ground binary resolution

$$\frac{C \lor P \qquad D \lor \neg P}{C \lor D}$$

Set $C = \neg A$.

$$\frac{A \to P \qquad P \to D}{A \to D}$$

Example · Memes and Theorems

- Some students like all memes.

  $F_1: \quad \exists x. S(x) \land \forall y. M(y) \to \text{Likes}(x, y)$

- No student likes any theorem.

  $F_2: \quad \forall x, y. S(x) \land T(y) \to \neg\text{Likes}(x, y)$

- Show: No meme is a theorem.

  $F: \quad \forall x. M(x) \to \neg T(x)$

Example · Memes and Theorems

Example · Memes and Theorems

CNF · Eliminating implications

F_1 : $\exists x. S(x) \land \forall y. M(y) \rightarrow \text{Likes}(x, y)$

$\exists x. S(x) \land \forall y. \neg M(y) \lor \text{Likes}(x, y)$

F_2 : $\forall x, y. S(x) \land T(y) \rightarrow \neg \text{Likes}(x, y)$

$\forall x, y. \neg S(x) \lor \neg T(y) \lor \neg \text{Likes}(x, y)$

F : $\forall x. M(x) \rightarrow \neg T(x)$

$\forall x. \neg M(x) \lor \neg T(x)$

Example · Memes and Theorems

CNF · Standardising variables apart, skolemising, dropping universal quantifiers

F_1 : $\exists x. S(x) \wedge \forall y. \neg M(y) \vee \text{Likes}(x, y)$

$S(G) \wedge (\neg M(y) \vee \text{Likes}(G, y))$

F_2 : $\forall x, y. \neg S(x) \vee \neg T(y) \vee \neg \text{Likes}(x, y)$

$\neg S(w) \vee \neg T(z) \vee \neg \text{Likes}(w, z)$

F : $\forall x. \neg M(x) \vee \neg T(x)$

$\neg M(x) \vee \neg T(x)$

Example · Memes and Theorems

Unification

$F_1: \quad S(G) \land (\neg M(y) \lor \text{Likes}(G, y))$

$F_2: \quad \neg S(w) \lor \neg T(z) \lor \neg\text{Likes}(w, z)$

$w/G: \ \neg S(G) \lor \neg T(z) \lor \neg\text{Likes}(G, z)$

Negation of proof goal

$\neg(\neg M(x) \lor \neg T(x)) \equiv M(x) \land T(x)$

Example · Memes and Theorems

$S(G) \land (\neg M(y) \lor \text{Likes}(G, y))$
$\neg S(G) \lor \neg T(z) \lor \neg \text{Likes}(G, z)$
$M(x) \land T(x)$

Clauses: $S(G)$, $M(x)$, $T(x)$, $\neg M(y) \lor \text{Likes}(G, y)$,
$\neg S(G) \lor \neg T(z) \lor \neg \text{Likes}(G, z)$

$$\frac{S(G) \qquad \neg S(G) \lor \neg T(z) \lor \neg \text{Likes}(G, z)}{\neg T(z) \lor \neg \text{Likes}(G, z)}$$

$$\frac{\neg M(y) \lor \text{Likes}(G, y) \qquad \neg T(z) \lor \neg \text{Likes}(G, z)}{\neg M(z) \lor \neg T(z)}$$

Substitute z/x

$$\frac{\neg M(x) \lor \neg T(x) \qquad M(x)}{\neg T(x)} \quad \text{and} \quad \frac{\neg T(x) \qquad T(x)}{\square}$$

Therefore, $\neg M(x) \lor \neg T(x)$, i.e. $M(x) \to \neg T(x)$.

Example · Memes and Theorems 2.0

- Some students like all memes.

  F_1 : $\exists x. S(x) \wedge \forall y. M(y) \rightarrow \text{Likes}(x, y)$

- No student likes any theorem.

  F_2 : $\forall x. S(x) \rightarrow \forall y. T(y) \rightarrow \neg\text{Likes}(x, y)$

- Show: No meme is a theorem.

  F : $\forall x. M(x) \rightarrow \neg T(x)$

Example · Memes and Theorems 2.0

CNF · Eliminating implications

F_1 : $\exists x.S(x) \land \forall y.M(y) \to \mathrm{Likes}(x, y)$

$\exists x.S(x) \land \forall y.\neg M(y) \lor \mathrm{Likes}(x, y)$

F_2 : $\forall x.S(x) \to \forall y.T(y) \to \neg\mathrm{Likes}(x, y)$

$\forall x.\neg S(x) \lor \forall y.\neg T(y) \lor \neg\mathrm{Likes}(x, y)$

F : $\forall x.M(x) \to \neg T(x)$

$\forall x.\neg M(x) \lor \neg T(x)$

Example · Memes and Theorems 2.0

CNF · Standardising variables apart, skolemising, dropping universal quantifiers

F_1 : $\exists x. S(x) \land \forall y. \neg M(y) \lor \text{Likes}(x, y)$

$S(G) \land (\neg M(y) \lor \text{Likes}(G, y))$

F_2 : $\forall x. \neg S(x) \lor \forall y. \neg T(y) \lor \neg \text{Likes}(x, y)$

$\neg S(w) \lor (\neg T(z) \lor \neg \text{Likes}(w, z))$

F : $\forall x. \neg M(x) \lor \neg T(x)$

$\neg M(x) \lor \neg T(x)$

Resolution · Soundness and completeness

Resolution is sound and complete.
A set of clauses S is unsatisfiable if and only if one can derive the empty clause (false) from S.

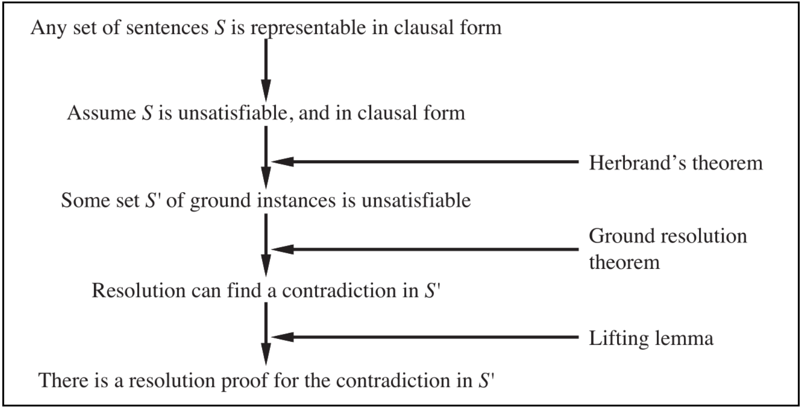
Soundness: derivability of empty clause implies unsatisfiability.

Can be proved by noticing that every model that satisfies the premises of resolution satisfies also satisfies its conclusion.

Completeness: every unsatisfiable clause can be refuted by resolution.

Can be proved using completeness of propositional resolution and lifting (as in the following slides; the full proof is beyond the scope of this course).

Resolution · Completeness proof



Completeness proof · Step 1

For a set of clauses S, we call the Herbrand universe of S the set H_S of all ground terms that can be constructed from the function symbols in S.

Example

For $S = \{\neg P(x, F(x, A)) \vee \neg Q(x, A) \vee R(x, B)\}$ we have

$H_S = \{A, B, F(A, A), F(A, B), F(B, A), F(B, B), F(A, F(A, A)), \dots\}$

Completeness proof · Step 1

For a set of clauses S and P a set of ground terms, $P(S)$, the saturation of S with respect to P, is the set of all ground clauses obtained by applying all possible consistent substitutions of variables in S with ground terms from P.

The saturation of a set S with respect to its Herbrand universe is called the Herbrand base of S and denoted $H_S(S)$.

Example

$$
\begin{aligned}
H_S(S) = \{ & \neg P(A, F(A, A)) \lor \neg Q(A, A) \lor R(A, B), \\
& \neg P(B, F(B, A)) \lor \neg Q(B, A) \lor R(B, B), \\
& \neg P(F(A, A), F(F(A, A), A)) \lor \neg Q(F(A, A), A) \lor R(F(A, A), B), \\
& \neg P(F(A, B), F(F(A, B), A)) \lor \neg Q(F(A, B), A) \lor R(F(A, B), B)\ , \dots \}
\end{aligned}
$$

Completeness proof · Step 1

Herbrand's theorem (1930)

If a set S of clauses is unsatisfiable, then there exists a finite subset of $H_S(S)$ that is also unsatisfiable.

Completeness proof · Step 2

Let S' be that finite unsatisfiable subset of ground sentences.

Running propositional resolution to completion on S' will derive a contradiction.

Completeness proof · Step 3

Lifting lemma

Let C_1 and C_2 be two clauses with no shared variables, and let C_1' and C_2' ground instances of C_1 and C_2.

If C' is a resolvent of C_1' and C_2', then there exists a clause C such that:

C is a resolvent of C_1 and C_2

C' is a ground instance of C.

$$
\begin{array}{ccc}
C_1, C_2 & \text{instantiation} & C_1', C_2' \\
\downarrow & & \downarrow \\
\frac{C_1 \quad C_2}{C} & \text{lifting} & \frac{C_1' \quad C_2'}{C'} \\
\downarrow & & \downarrow \\
C & \text{instantiation} & C'
\end{array}
$$

Completeness proof · Step 3

Example

$C_1 = \neg P(x, F(x, A)) \vee \neg Q(x, A) \vee R(x, B)$

$C_2 = \neg N(G(y), z) \vee P(H(y), z)$

$C_1' = \neg P(H(B), F(H(B), A)) \vee \neg Q(H(B), A) \vee R(H(B), B)$

$C_2' = \neg N(G(B), F(H(B), A)) \vee P(H(B), F(H(B), A))$

$C' = \neg N(G(B), F(H(B), A)) \vee \neg Q(H(B), A) \vee R(H(B), B)$

$C = \neg N(G(y), F(H(y), A)) \vee \neg Q(H(y), A) \vee R(H(y), B)$

Efficient algorithms for resolution

Heuristics to make resolution more efficient:

Unit preference: prefer clauses with only one symbol.

Pure clauses: a pure clause contains symbol A which does not occur in any other clause. Cannot lead to contradiction.

Tautology: clauses containing A and $\neg A$.

Set of support: identify *useful* clauses and ignore the rest.

Input resolution: intermediately generated clauses can only be combined with original input clauses.

Subsumption: if a clause contains another one, use only the shorter clause. Prune unnecessary facts from the KB.

Including heuristics, resolution is more efficient than DPLL.

Summary

- Limitations of GMP
- Relationship between inference rules
- Completeness of resolution – the lifting lemma
- Efficient algorithms for resolution