Inf2D 13: Resolution-Based Inference

Cristina Alexandru

School of Informatics, University of Edinburgh

13/02/18

Slide Credits: Jacques Fleuriot, Michael Rovatsos, Michael Herrmann
Backward chaining:
- If Goal is known (goal directed)
- Can query for data

Forward chaining:
- If specific Goal is not known, but system needs to react to new facts (data driven)
- Can make suggestions

Combination of forward and backward chaining: Opportunistic reasoning

Also relevant:
- 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 all rules contain only non-negated symbols
- Conclusions of all rules is a non-negated symbol
- Facts are non-negated propositions

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)
Negate query $\alpha$
Convert everything to CNF
Repeat: Choose clauses and resolve (based on unification)
If resolution results in empty clause, $\alpha$ is proved
Return all substitutions (or Fail)
Is the following sentence unsatisfiable?

\[ P(x, t), \neg P(s, x) \]

with \( x \): variable, \( s \neq t \): constants
Answer

- Is the following set of clauses unsatisfiable?
  
  \[ P(x, t), \neg P(s, x) \]

- Not necessarily, if this was a propositionalisation of
  
  \[ \forall x. (P(x, t) \land \neg P(s, x)) \]

- It is unsatisfiable, if this was a propositionalisation of
  
  \[ (\forall x. P(x, t)) , (\forall x. \neg P(s, x)) \]

Here we can standardise-away the clash of the variable \(x\) (which we should have done in the first place!) and obtain a contradiction by instantiation.
Ground binary resolution

\[
\frac{C \lor P \quad D \lor \neg P}{C \lor D}
\]

Suppose \( C = \text{False} \)

\[
\frac{P \quad \neg P \lor D}{D}
\]

i.e. \( P \) and \( P \rightarrow D \) entails \( D \).

Modus ponens is a special case of binary resolution.
GMP with $p'_i\theta \equiv p_i\theta \ \forall i$

\[
\frac{p'_1, \ldots, p'_n}{q\theta}
\]

\[
\frac{p'_1, \ldots, p'_n \quad (p_1 \land \cdots \land p_n \Rightarrow q)}{q\theta}
\]

Full resolution with $\theta$ MGU of all $P_i$ and $P'_i\theta \equiv P_i\theta \ \forall i$

\[
\frac{C \lor P_1 \lor \cdots \lor P_m \quad D \lor \neg P'_1 \lor \cdots \lor \neg P'_n}{(C \lor D)\theta}
\]
Ground binary resolution

\[
\begin{align*}
C \lor P & \quad D \lor \neg P \\
\hline
C \lor D
\end{align*}
\]

Set \( C = \neg A \)

\[
\begin{align*}
A \Rightarrow P & \quad P \Rightarrow D \\
\hline
A \Rightarrow D
\end{align*}
\]
Example: Quacks and Doctors

- Some patients like all doctors
  \[ F_1 \equiv \exists x. \ P(x) \land \forall y. \ D(y) \Rightarrow Likes(x, y) \]

- No patient likes any quack
  \[ F_2 \equiv \forall x. \ P(x) \land \forall y. \ Q(y) \Rightarrow \neg Likes(x, y) \]

- Show: No doctor is a quack
  \[ F \equiv \forall x. \ D(x) \Rightarrow \neg Q(x) \]
Example: Quacks and Doctors

**CNF**

\[ F_1 \equiv \exists x. P(x) \land \forall y. D(y) \Rightarrow Likes(x, y) \]
\[ \equiv \exists x. P(x) \land \forall y. \neg D(y) \lor Likes(x, y) \]

\[ F_2 \equiv \forall x. P(x) \land \forall y. Q(y) \Rightarrow \neg Likes(x, y) \]
\[ \equiv \forall x. P(x) \land \forall y. \neg Q(y) \lor \neg Likes(x, y) \]

\[ F \equiv \forall x. D(x) \Rightarrow \neg Q(x) \]
\[ \equiv \forall x. \neg D(x) \lor \neg Q(x) \]
Example: Quacks and Doctors

Propositionalisation

\[ F_1 \equiv \exists x. \ P(x) \land \forall y. \neg D(y) \lor Likes(x, y) \]
\[ \Rightarrow P(G) \land (\neg D(y) \lor Likes(G, y)) \]

\[ F_2 \equiv \forall x. \ P(x) \land \forall y. \neg Q(y) \lor \neg Likes(x, y) \]
\[ \Rightarrow P(w) \land \neg Q(z) \lor \neg Likes(w, z) \]

\[ F \equiv \forall x. \neg D(x) \lor \neg Q(x) \]
\[ \equiv \neg D(x) \lor \neg Q(x) \]
Example: Quacks and Doctors

Unification

\[ F'_1 \equiv P(G) \land \neg D(y) \lor Likes(G, y) \]

\[ F'_2 \equiv P(w) \land \neg Q(z) \lor \neg Likes(w, z) \]

\[ w/G \implies P(G) \land \neg Q(z) \lor \neg Likes(G, z) \]

Negation of proof goal

\[ \neg(\neg D(x) \lor \neg Q(x)) \equiv D(x) \land Q(x) \]
Example: Quacks and Doctors

Resolution

\[ P \left( G \right) \land \neg D \left( y \right) \lor Likes \left( G, y \right), \]
\[ P \left( G \right) \land \neg Q \left( z \right) \lor \neg Likes \left( G, z \right), \]
\[ D \left( x \right) \land Q \left( x \right) \]

Clauses: \[ P \left( G \right), D \left( y \right), Q \left( z \right), \]
\[ \neg D \left( y \right) \lor Likes \left( G, y \right), \neg Q \left( z \right) \lor \neg Likes \left( G, z \right) \]

\[ \neg D \left( y \right) \lor Likes \left( G, z \right) \quad \neg Q \left( z \right) \lor \neg Likes \left( G, z \right) \]
\[ \therefore \neg D \left( y \right) \lor \neg Q \left( z \right) \]

Substitute \( y/x \) and \( z/x \)

\[ \neg D \left( x \right) \lor \neg Q \left( x \right) \quad D \left( x \right) \quad \text{and} \quad \neg Q \left( x \right) \quad Q \left( x \right) \]

\[ \therefore \neg Q \left( x \right) \]

Therefore: \[ \neg D \left( x \right) \lor \neg Q \left( x \right), \text{ i.e. } D \left( x \right) \Rightarrow \neg Q \left( x \right) \] for any \( x \) such that \( \forall x. D \left( x \right) \Rightarrow \neg Q \left( x \right) \)
Example: Quacks and Doctors (Variant)

- Some patients like all doctors
  
  \[ F_1 \equiv \exists x. P(x) \land \forall y. D(y) \Rightarrow Likes(x, y) \]

- No patient likes any quack
  
  \[ F_2 \equiv \forall x. P(x) \Rightarrow \forall y. Q(y) \Rightarrow \neg Likes(x, y) \]

- Show: No doctor is a quack
  
  \[ F \equiv \forall x. D(x) \Rightarrow \neg Q(x) \]
Example: Quacks and Doctors (Variant)

CNF

\[ F_1 \equiv \exists x. P(x) \land \forall y. D(y) \Rightarrow Likes(x, y) \]
\[ \equiv \exists x. P(x) \land \forall y. \neg D(y) \lor Likes(x, y) \]

\[ F_2 \equiv \forall x. P(x) \Rightarrow \forall y. Q(y) \Rightarrow \neg Likes(x, y) \]
\[ \equiv \forall x. \neg P(x) \lor \forall y. \neg Q(y) \lor \neg Likes(x, y) \]

\[ F \equiv \forall x. D(x) \Rightarrow \neg Q(x) \]
\[ \equiv \forall x. \neg D(x) \lor \neg Q(x) \]
Propositionalisation

\[ F_1 \equiv \exists x. P(x) \land \forall y. \neg D(y) \lor Likes(x, y) \]
\[ \Rightarrow P(G) \land (\neg D(y) \lor Likes(G, y)) \]

\[ F_2 \equiv \forall x. \neg P(x) \lor \forall y. \neg Q(y) \lor \neg Likes(x, y) \]
\[ \Rightarrow \neg P(w) \lor \neg Q(z) \lor \neg Likes(w, z) \]

\[ F \equiv \forall x. \neg D(x) \lor \neg Q(x) \]
\[ \equiv \neg D(x) \lor \neg Q(x) \]
Example: Quacks and Doctors (Variant)

Unification

\[ F_1' \equiv P(G) \land \neg D(y) \lor \text{Likes}(G, y) \]

\[ F_2' \equiv \neg P(w) \lor \neg Q(z) \lor \neg \text{Likes}(w, z) \]

\[ w/G \Rightarrow \neg P(G) \lor \neg Q(z) \lor \neg \text{Likes}(G, z) \]

Negation of proof goal

\[ \neg (\neg D(x) \lor \neg Q(x)) \equiv D(x) \land Q(x) \]
Example: Quacks and Doctors (Variant)

Resolution

\[ P(G) \land \neg D(y) \lor Likes(G, y), \]
\[ \neg P(G) \lor \neg Q(z) \lor \neg Likes(G, z), \]
\[ D(x) \land Q(x) \]

Clauses: \[ P(G), D(y), Q(z), \]
\[ \neg D(y) \lor Likes(G, y), \neg P(G) \lor \neg Q(z) \lor \neg Likes(G, z) \]

\[ P(G) \quad \neg P(G) \lor \neg Q(z) \lor \neg Likes(G, z) \]
\[ \hline \]
\[ \neg Q(z) \lor \neg Likes(G, z) \]

\[ \neg D(y) \lor Likes(G, z) \quad \neg Q(z) \lor \neg Likes(G, z) \]
\[ \hline \]
\[ \neg D(y) \lor \neg Q(z) \]

Substitute \( y/x \) and \( z/x \)

\[ \neg D(x) \lor \neg Q(x) \quad D(x) \quad \text{and} \quad \neg Q(x) \quad Q(x) \]
\[ \hline \]
\[ \neg Q(x) \]

Therefore: \[ \neg D(x) \lor \neg Q(x), \text{ i.e. } D(x) \Rightarrow \neg Q(x) \text{ for any } x \]
such that \( \forall x. D(x) \Rightarrow \neg Q(x) \)
Resolution is sound and complete: If a set of clauses is unsatisfiable, then one can derive an empty clause from this set.

Soundness is evident since the conclusion of any inference rule is a logical consequence of its premises.

Completeness can be proved using completeness of propositional resolution and lifting (see the following slides, but full proof is beyond the scope of this course).
Reminder: Factoring

\[ A \lor B \quad A \lor \neg B \]

\[ \frac{A \lor A}{A \lor A} \]

More generally

\[ C \lor P_1 \lor \ldots \lor P_m \]

\[ \frac{(C \lor P_1)\theta}{(C \lor P_1)\theta} \]

where \( \theta \) is the MGU of the \( P_i \)

**Soundness:** by universal instantiation and deletion of duplicates.
Let $C_1$ and $C_2$ be two clauses. A resolvent is a

- binary resolvent of $C_1$ and $C_2$
- binary resolvent of a factor of $C_1$ and $C_2$
- binary resolvent of $C_1$ and a factor of $C_2$
- binary resolvent of a factor of $C_1$ and a factor of $C_2$
Factoring example

$$C_1 = P(x) \lor P(f(y)) \lor R(g(y))$$

$$C_2 = \neg P(f(g(a))) \lor Q(b)$$

$$C'_1 = P(f(y)) \lor R(g(y))$$ is a factor of $$C_1$$

$$C = R(g(g(a))) \lor Q(b)$$ is a resolvent of $$C'_1$$ and $$C_2$$ and thus a resolvent of $$C_1$$ and $$C_2$$.

Choose $$\theta = \{x/f(y), y/g(a)\}$$
The lifting lemma

If $C_1$ and $C_2$ are two clauses with no shared variables and $C'_1$ and $C'_2$ their respective instances and $C'$ is a resolvent of $C'_1$ and $C'_2$, then there exists a clause $C$ such that both $C$ is a resolvent of $C_1$ and $C_2$ and $C'$ is an instance of $C$. 

\[\begin{array}{c}
C_1, C_2 \xrightarrow{\text{Instantiation}} C' \\
\downarrow \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \downarrow \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \downarrow
C_1 \quad C_2 \\
\downarrow \\
C \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \downarrow \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \downarrow
C \\
\end{array}\]

\[\begin{array}{c}
C_1, C_2 \xleftarrow{\text{Lifting}} C'_1, C'_2 \\
\downarrow \quad \quad \quad \quad \downarrow \quad \quad \quad \quad \downarrow \quad \quad \quad \quad \downarrow \quad \quad \quad \quad \downarrow
C_1 \quad C_2 \\
\downarrow \\
C' \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \downarrow \\
C' \\
\end{array}\]
Lifting lemma: Example

- **Clauses**
  - \( C_1 = P(x) \lor Q(x) \)
  - \( C_2 = \neg P(f(y)) \lor R(y) \)

- Resolving \( C_1 \) and \( C_2 \) gives \( C = Q(f(y)) \lor R(y) \)

- **Instances:**
  - \( C'_1 = P(f(a)) \lor Q(f(a)) \) using \( x/f(a) \)
  - \( C'_2 = \neg P(f(a)) \lor R(a) \) using \( y/a \)

- Resolving \( C'_1 \) and \( C'_2 \) gives \( C' = Q(f(a)) \lor R(a) \)

- \( C' \) is an instance of \( C \) as predicted by the Lifting Lemma!
Remark: Resolution for 3SAT

\[
(X_i \lor \neg X_j \lor X_k) \land (X_j \lor \neg X_r \lor X_s) \land \cdots \land (\neg X_k \lor X_l \lor \neg X_m)
\]

with \( i, j, k, r, s, l, m, \ldots \in \{1, \ldots, N\} \)

Assuming \( X_i = 0 \):

\[
\frac{\neg X_i}{X_i \lor \neg X_j \lor X_k} \quad \frac{X_i \lor \neg X_j \lor X_k}{\neg X_j \lor X_k}
\]

(we can’t really use the (negated) goal at this stage)

First two clauses:

\[
\frac{\neg X_j \lor X_k}{X_k \lor \neg X_r \lor X_s} \quad \frac{X_j \lor \neg X_r \lor X_s}{X_k \lor \neg X_r \lor X_s}
\]

More assumptions? More symbols per clause?
Efficient algorithms for resolution

Heuristics to make resolution more efficient (compare DPLL!)

- Unit preference: Prefer clauses with only one symbol
- Pure clauses: Pure clauses 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” rules and ignore the rest.
- Input resolution: Intermediately generated sentences can only be combined with original inputs or original rules.
- Subsumption: If a clause contains another one, use only the shorter clause. Prune unnecessary facts from the database.

Including heuristics, resolution is more efficient than DPLL.
Summary

- Forward chaining
- Backward chaining
- Resolution