KRR Lecture 5

Resolution

November 5^{th}, 2012

Outline

1 Resolution in PL

Outline

Outline

Outline

Outline

1. Resolution in PL
   - General idea
   - Clausal form
   - Resolution proofs
2. Resolution in FOPL
3. Herbrand's Theorem
4. Resolution Strategies

Outline

Intuition

- consider the following formula

$$\phi = (p \vee q) \wedge (\neg p \vee r) \rightarrow (q \vee r)$$

Intuition

- consider the following formula

$$\phi = (p \vee q) \wedge (\neg p \vee r) \to (q \vee r)$$

- if p is true, then r must be true

Intuition

- consider the following formula

$$\phi = (p \vee q) \wedge (\neg p \vee r) \rightarrow (q \vee r)$$

- if p is true, then r must be true
- if p is false, then q must be true

Intuition

- consider the following formula

$$\phi = (p \vee q) \wedge (\neg p \vee r) \rightarrow (q \vee r)$$

- if p is true, then r must be true
- if p is false, then q must be true
- ϕ is a valid formula (a tautology)

Intuition

- consider the following formula

$$\phi = (p \vee q) \wedge (\neg p \vee r) \rightarrow (q \vee r)$$

- if p is true, then r must be true
- if p is false, then q must be true
- ϕ is a valid formula (a tautology)
- we can write the following sound inference rule based on ϕ

$$\frac{\begin{array}{r} p \vee q \\ \neg p \vee r \end{array}}{q \vee r}$$

Resolution inference rule

$$\frac{\begin{array}{r} p \vee \phi_1 \vee \phi_2 \vee \ldots \vee \phi_m \\ \neg p \vee \psi_1 \vee \psi_2 \vee \ldots \vee \psi_n \end{array}}{\phi_1 \vee \phi_2 \vee \ldots \vee \phi_m \vee \psi_1 \vee \psi_2 \vee \ldots \vee \psi_n}$$

- ϕ_i and ψ_j are literals (propositions or negated propositions)
- disjunction of literals = **clause**

Resolution inference rule

$$\frac{\begin{array}{r} p \vee \phi_1 \vee \phi_2 \vee \ldots \vee \phi_m \\ \neg p \vee \psi_1 \vee \psi_2 \vee \ldots \vee \psi_n \end{array}}{\phi_1 \vee \phi_2 \vee \ldots \vee \phi_m \vee \psi_1 \vee \psi_2 \vee \ldots \vee \psi_n}$$

- ϕ_i and ψ_j are literals (propositions or negated propositions)
- disjunction of literals = **clause**
- if two clauses contain a pair of complement literals, we say that they resolve (we can use them as premises for resolution)

Resolution inference rule

$$\frac{\begin{array}{r} p \vee \phi_1 \vee \phi_2 \vee \ldots \vee \phi_m \\ \neg p \vee \psi_1 \vee \psi_2 \vee \ldots \vee \psi_n \end{array}}{\phi_1 \vee \phi_2 \vee \ldots \vee \phi_m \vee \psi_1 \vee \psi_2 \vee \ldots \vee \psi_n}$$

- ϕ_i and ψ_j are literals (propositions or negated propositions)
- disjunction of literals = **clause**
- if two clauses contain a pair of complement literals, we say that they resolve (we can use them as premises for resolution)
- the result of applying resolution on the two clauses is called their resolvent

Resolution inference rule

$$\frac{\begin{array}{r} p \vee \phi_1 \vee \phi_2 \vee \ldots \vee \phi_m \\ \neg p \vee \psi_1 \vee \psi_2 \vee \ldots \vee \psi_n \end{array}}{\phi_1 \vee \phi_2 \vee \ldots \vee \phi_m \vee \psi_1 \vee \psi_2 \vee \ldots \vee \psi_n}$$

- ϕ_i and ψ_j are literals (propositions or negated propositions)
- disjunction of literals = **clause**
- if two clauses contain a pair of complement literals, we say that they resolve (we can use them as premises for resolution)
- the result of applying resolution on the two clauses is called their resolvent
- if several pairs of complement literals exist, they must be considered separately

The set notation

- disjunction is commutative
- duplicate literals can be removed:

$$p \vee p \leftrightarrow p$$

$$\neg p \vee \neg p \leftrightarrow \neg p$$

- we may use sets of literals to denote clauses:

$$\frac{\{p, \phi_1, \phi_2, \ldots, \phi_m\} \quad \{\neg p, \psi_1, \psi_2, \ldots, \psi_n\}}{\{\phi_1, \phi_2, \ldots, \phi_m\} \cup \{\psi_1, \psi_2, \ldots, \psi_n\}}$$

Outline

Conversion to clausal form

$$\neg p \vee \neg(\neg p \rightarrow q)$$

Conversion to clausal form

$$\neg p \vee \neg(\neg p \to q)$$

- implications out

$$(\phi \to \psi) \leftrightarrow (\neg\phi \vee \psi)$$
$$(\phi \leftrightarrow \psi) \leftrightarrow (\neg\phi \vee \psi) \wedge (\phi \vee \neg\psi)$$

Conversion to clausal form

$$\neg p \vee \neg(\neg p \rightarrow q)$$

- implications out

$$(\phi \rightarrow \psi) \leftrightarrow (\neg\phi \vee \psi)$$
$$(\phi \leftrightarrow \psi) \leftrightarrow (\neg\phi \vee \psi) \wedge (\phi \vee \neg\psi)$$

Conversion to clausal form

$$\neg p \vee \neg(\neg\neg p \vee q)$$

- implications out

$$(\phi \rightarrow \psi) \leftrightarrow (\neg\phi \vee \psi)$$
$$(\phi \leftrightarrow \psi) \leftrightarrow (\neg\phi \vee \psi) \wedge (\phi \vee \neg\psi)$$

Conversion to clausal form

$$\neg p \vee \neg(\neg\neg p \vee q)$$

- implications out

$$(\phi \rightarrow \psi) \leftrightarrow (\neg\phi \vee \psi)$$
$$(\phi \leftrightarrow \psi) \leftrightarrow (\neg\phi \vee \psi) \wedge (\phi \vee \neg\psi)$$

- negations in

$$\neg(\phi \wedge \psi) \leftrightarrow (\neg\phi \vee \neg\psi)$$
$$\neg(\phi \vee \psi) \leftrightarrow (\neg\phi \wedge \neg\psi)$$
$$\neg\neg\phi \leftrightarrow \phi$$

Conversion to clausal form

$$\neg p \vee \neg(\neg\neg p \vee q)$$

- implications out

$$(\phi \rightarrow \psi) \leftrightarrow (\neg\phi \vee \psi)$$
$$(\phi \leftrightarrow \psi) \leftrightarrow (\neg\phi \vee \psi) \wedge (\phi \vee \neg\psi)$$

- negations in

$$\neg(\phi \wedge \psi) \leftrightarrow (\neg\phi \vee \neg\psi)$$
$$\neg(\phi \vee \psi) \leftrightarrow (\neg\phi \wedge \neg\psi)$$
$$\neg\neg\phi \leftrightarrow \phi$$

Conversion to clausal form

$$\neg p \vee (\neg\neg\neg p \wedge \neg q)$$

- implications out

$$(\phi \rightarrow \psi) \leftrightarrow (\neg\phi \vee \psi)$$
$$(\phi \leftrightarrow \psi) \leftrightarrow (\neg\phi \vee \psi) \wedge (\phi \vee \neg\psi)$$

- negations in

$$\neg(\phi \wedge \psi) \leftrightarrow (\neg\phi \vee \neg\psi)$$
$$\neg(\phi \vee \psi) \leftrightarrow (\neg\phi \wedge \neg\psi)$$
$$\neg\neg\phi \leftrightarrow \phi$$

Conversion to clausal form

$$\neg p \vee (\neg\neg\neg p \wedge \neg q)$$

- implications out

$$(\phi \rightarrow \psi) \leftrightarrow (\neg\phi \vee \psi)$$
$$(\phi \leftrightarrow \psi) \leftrightarrow (\neg\phi \vee \psi) \wedge (\phi \vee \neg\psi)$$

- negations in

$$\neg(\phi \wedge \psi) \leftrightarrow (\neg\phi \vee \neg\psi)$$
$$\neg(\phi \vee \psi) \leftrightarrow (\neg\phi \wedge \neg\psi)$$
$$\neg\neg\phi \leftrightarrow \phi$$

Conversion to clausal form

$$\neg p \vee (\neg p \wedge \neg q)$$

- implications out

$$(\phi \rightarrow \psi) \leftrightarrow (\neg\phi \vee \psi)$$
$$(\phi \leftrightarrow \psi) \leftrightarrow (\neg\phi \vee \psi) \wedge (\phi \vee \neg\psi)$$

- negations in

$$\neg(\phi \wedge \psi) \leftrightarrow (\neg\phi \vee \neg\psi)$$
$$\neg(\phi \vee \psi) \leftrightarrow (\neg\phi \wedge \neg\psi)$$
$$\neg\neg\phi \leftrightarrow \phi$$

Conversion to clausal form (continued)

$$\neg p \vee (\neg p \wedge \neg q)$$

- distribution rules

$$\phi \vee (\psi_1 \wedge \psi_2) \leftrightarrow (\phi \vee \psi_1) \wedge (\phi \vee \psi_2)$$
$$(\phi_1 \wedge \phi_2) \vee \psi \leftrightarrow (\phi_1 \vee \psi) \wedge (\phi_2 \vee \psi)$$

Conversion to clausal form (continued)

$$\neg p \vee (\neg p \wedge \neg q)$$

- distribution rules

$$\phi \vee (\psi_1 \wedge \psi_2) \leftrightarrow (\phi \vee \psi_1) \wedge (\phi \vee \psi_2)$$
$$(\phi_1 \wedge \phi_2) \vee \psi \leftrightarrow (\phi_1 \vee \psi) \wedge (\phi_2 \vee \psi)$$

Conversion to clausal form (continued)

$$(\neg p \vee \neg p) \wedge (\neg p \vee \neg q)$$

- distribution rules

$$\phi \vee (\psi_1 \wedge \psi_2) \leftrightarrow (\phi \vee \psi_1) \wedge (\phi \vee \psi_2)$$
$$(\phi_1 \wedge \phi_2) \vee \psi \leftrightarrow (\phi_1 \vee \psi) \wedge (\phi_2 \vee \psi)$$

Conversion to clausal form (continued)

$$(\neg p \vee \neg p) \wedge (\neg p \vee \neg q)$$

- distribution rules

$$\phi \vee (\psi_1 \wedge \psi_2) \leftrightarrow (\phi \vee \psi_1) \wedge (\phi \vee \psi_2)$$
$$(\phi_1 \wedge \phi_2) \vee \psi \leftrightarrow (\phi_1 \vee \psi) \wedge (\phi_2 \vee \psi)$$

- convert to clauses

$$(\phi_1 \vee \phi_2) \wedge (\psi_1 \vee \psi_2) \Rightarrow \{\phi_1, \phi_2\}, \{\psi_1, \psi_2\}$$

Conversion to clausal form (continued)

$$(\neg p \vee \neg p) \wedge (\neg p \vee \neg q)$$

- distribution rules

$$\phi \vee (\psi_1 \wedge \psi_2) \leftrightarrow (\phi \vee \psi_1) \wedge (\phi \vee \psi_2)$$
$$(\phi_1 \wedge \phi_2) \vee \psi \leftrightarrow (\phi_1 \vee \psi) \wedge (\phi_2 \vee \psi)$$

- convert to clauses

$$(\phi_1 \vee \phi_2) \wedge (\psi_1 \vee \psi_2) \Rightarrow \{\phi_1, \phi_2\}, \{\psi_1, \psi_2\}$$

Conversion to clausal form (continued)

$$\{\neg p\}, \{\neg p, \neg q\}$$

- distribution rules

$$\phi \vee (\psi_1 \wedge \psi_2) \leftrightarrow (\phi \vee \psi_1) \wedge (\phi \vee \psi_2)$$
$$(\phi_1 \wedge \phi_2) \vee \psi \leftrightarrow (\phi_1 \vee \psi) \wedge (\phi_2 \vee \psi)$$

- convert to clauses

$$(\phi_1 \vee \phi_2) \wedge (\psi_1 \vee \psi_2) \Rightarrow \{\phi_1, \phi_2\}, \{\psi_1, \psi_2\}$$

Outline

Example - initial formulation

$$\begin{array}{r} p \rightarrow q \\ q \rightarrow r \\ r \rightarrow s \\ \hline p \rightarrow s \end{array}$$

- convert to clausal form – logically equivalent
- use the resolution inference rule to obtain the conclusion from the premises – direct (generative) proof
- prove that the union of the premises with the negation of the conclusion is inconsistent – indirect (refutation) proof

Direct (generative) proof

$$1.\{\neg p, q\} \leftarrow \textit{premise}$$
$$2.\{\neg q, r\} \leftarrow \textit{premise}$$
$$3.\{\neg r, s\} \leftarrow \textit{premise}$$
$$4.\{\neg p, r\} \leftarrow 1, 2, (q, \neg q)$$
$$5.\{\neg p, s\} \leftarrow 3, 4, (\neg r, r)$$

Indirect (refutation) proof

1. $\{\neg p, q\} \leftarrow$ *premise*
2. $\{\neg q, r\} \leftarrow$ *premise*
3. $\{\neg r, s\} \leftarrow$ *premise*
4. $\{p\} \leftarrow$ *negated conclusion*
5. $\{\neg s\} \leftarrow$ *negated conclusion*
6. $\{q\} \leftarrow 1, 4, (\neg p, p)$
7. $\{r\} \leftarrow 2, 6, (\neg q, q)$
8. $\{s\} \leftarrow 3, 7, (\neg r, r)$
9. $\{\} \leftarrow 5, 8, (\neg s, s)$

Completeness of resolution

- can we prove all conclusions that are logically entailed by a set of premises?
- resolution is not generatively complete
  - $p \vee \neg p$
  - $p \vee q$, given p
- resolution is refutationally complete

Outline

1. Resolution in PL
2. Resolution in FOPL
   - Conversion to clausal form
   - Unification
   - Factors
3. Herbrand's Theorem
4. Resolution Strategies

Outline

Rules for conversion

- implications out – same as for PL
- negations in – add the following rules for quantifiers

$$\neg\forall X.\Phi \leftrightarrow \exists X.\neg\Phi$$
$$\neg\exists X.\Phi \leftrightarrow \forall X.\neg\Phi$$

- standardize variables – rename variables so that each quantifier uses a unique variable, e.g.:

$$\forall X.P(X) \vee \exists X.Q(X) \Rightarrow \forall X.P(X) \vee \exists Y.Q(Y)$$

Rules for conversion (continued)

- existentials out – replace variable with new constant or new Skolem function, e.g.:

$$\exists X.P(X) \Rightarrow P(a)$$
$$\forall X.\exists Y.Q(X,Y) \Rightarrow \forall X.Q(X,f(X))$$

- alls out – remove universal quantifiers
- distribution – convert to conjunctive normal form: literals are predicates or negated predicates
- convert to clauses using set notation

Example

- $H = Human$, $F = Female$, $P = Parent$,

Example

- $H = Human$, $F = Female$, $P = Parent$,
- $\forall X.(H(X) \rightarrow \exists Y.(H(Y) \wedge F(Y) \wedge P(Y, X)))$

Example

- $H = Human$, $F = Female$, $P = Parent$,
- $\forall X.(H(X) \rightarrow \exists Y.(H(Y) \wedge F(Y) \wedge P(Y,X)))$
- $\forall X.(\neg H(X) \vee \exists Y.(H(Y) \wedge F(Y) \wedge P(Y,X)))$

Example

- $H = Human$, $F = Female$, $P = Parent$, $m = mother$
- $\forall X.(H(X) \rightarrow \exists Y.(H(Y) \wedge F(Y) \wedge P(Y, X)))$
- $\forall X.(\neg H(X) \vee \exists Y.(H(Y) \wedge F(Y) \wedge P(Y, X)))$
- $\forall X.(\neg H(X) \vee (H(m(X)) \wedge F(m(X)) \wedge P(m(X), X)))$

Example

- $H = Human$, $F = Female$, $P = Parent$, $m = mother$
- $\forall X.(H(X) \rightarrow \exists Y.(H(Y) \wedge F(Y) \wedge P(Y, X)))$
- $\forall X.(\neg H(X) \vee \exists Y.(H(Y) \wedge F(Y) \wedge P(Y, X)))$
- $\forall X.(\neg H(X) \vee (H(m(X)) \wedge F(m(X)) \wedge P(m(X), X)))$
- $\neg H(X) \vee (H(m(X)) \wedge F(m(X)) \wedge P(m(X), X))$

Example

- $H = Human$, $F = Female$, $P = Parent$, $m = mother$
- $\forall X.(H(X) \rightarrow \exists Y.(H(Y) \wedge F(Y) \wedge P(Y, X)))$
- $\forall X.(\neg H(X) \vee \exists Y.(H(Y) \wedge F(Y) \wedge P(Y, X)))$
- $\forall X.(\neg H(X) \vee (H(m(X)) \wedge F(m(X)) \wedge P(m(X), X)))$
- $\neg H(X) \vee (H(m(X)) \wedge F(m(X)) \wedge P(m(X), X))$
- $(\neg H(X) \vee H(m(X))) \wedge (\neg H(X) \vee F(m(X))) \wedge (\neg H(X) \vee P(m(X), X))$

Example

- $H = Human$, $F = Female$, $P = Parent$, $m = mother$
- $\forall X.(H(X) \rightarrow \exists Y.(H(Y) \wedge F(Y) \wedge P(Y, X)))$
- $\forall X.(\neg H(X) \vee \exists Y.(H(Y) \wedge F(Y) \wedge P(Y, X)))$
- $\forall X.(\neg H(X) \vee (H(m(X)) \wedge F(m(X)) \wedge P(m(X), X)))$
- $\neg H(X) \vee (H(m(X)) \wedge F(m(X)) \wedge P(m(X), X))$
- $(\neg H(X) \vee H(m(X))) \wedge (\neg H(X) \vee F(m(X))) \wedge (\neg H(X) \vee P(m(X), X))$
- $\{\neg H(X), H(m(X))\}, \{\neg H(X), F(m(X))\}, \{\neg H(X), P(m(X), X)\}$

Outline

Complement pairs in FOPL

- $P(X, Y)$ and $\neg P(X, Y)$
- $P(X, f(Y), g(X, f(Y)))$ and $\neg P(X, f(Y), g(X, f(X)))$

Complement pairs in FOPL

- $P(X, Y)$ and $\neg P(X, Y)$
- $P(X, f(Y), g(X, f(Y)))$ and $\neg P(X, f(Y), g(X, f(X)))$
- $P(X, b)$ and $\neg P(f(Y), Z)$ might work too, if X is replaced by $f(Y)$ and Z is replaced by b

Complement pairs in FOPL

- $P(X, Y)$ and $\neg P(X, Y)$
- $P(X, f(Y), g(X, f(Y)))$ and $\neg P(X, f(Y), g(X, f(X)))$
- $P(X, b)$ and $\neg P(f(Y), Z)$ might work too, if X is replaced by $f(Y)$ and Z is replaced by b
- substitution = replacement of all occurrences of a variable with a term (variable, constant or function)

$$P(X, b)\{X \leftarrow f(Y)\} = P(f(Y), b)$$

- unification = perform the right substitution α such that two predicates become identical; α in this case is called a unifier of the two predicates

$$P(X, b)\alpha = P(f(Y), Z)\alpha = P(f(Y), b)$$
$$\alpha = \{X \leftarrow f(Y), Z \leftarrow b\}$$

Most general unifier

- composition of substitutions
- α is more general than β if there exists a substitution γ such that $\beta = \alpha\gamma$
- most general unifier (MGU)
- if predicates have different names or arities, they cannot be unified
- otherwise, we compute the MGU starting from the empty substitution $\{\}$ and using a recursive approach for comparing corresponding arguments of the predicate in the two instances

Algorithm for computing the MGU

- identical terms $\Rightarrow$ nothing to do

Algorithm for computing the MGU

- identical terms $\Rightarrow$ nothing to do
- one is a constant and the other is not a variable $\Rightarrow$ fail

Algorithm for computing the MGU

- identical terms $\Rightarrow$ nothing to do
- one is a constant and the other is not a variable $\Rightarrow$ fail
- both are functions, but the names or arities differ $\Rightarrow$ fail

Algorithm for computing the MGU

- identical terms $\Rightarrow$ nothing to do
- one is a constant and the other is not a variable $\Rightarrow$ fail
- both are functions, but the names or arities differ $\Rightarrow$ fail
- both are functions and the names and arities coincide $\Rightarrow$ compare their arguments one by one

Algorithm for computing the MGU

- identical terms $\Rightarrow$ nothing to do
- one is a constant and the other is not a variable $\Rightarrow$ fail
- both are functions, but the names or arities differ $\Rightarrow$ fail
- both are functions and the names and arities coincide $\Rightarrow$ compare their arguments one by one
- one is a variable X and there is a binding for X in the current substitution $\Rightarrow$ unify the binding of X with the other term

Algorithm for computing the MGU

- identical terms $\Rightarrow$ nothing to do
- one is a constant and the other is not a variable $\Rightarrow$ fail
- both are functions, but the names or arities differ $\Rightarrow$ fail
- both are functions and the names and arities coincide $\Rightarrow$ compare their arguments one by one
- one is a variable X and there is a binding for X in the current substitution $\Rightarrow$ unify the binding of X with the other term
- one is a variable X, there is no binding for X in the current substitution and the other term does not contain X $\Rightarrow$ add substitution that replaces X with the other term

Algorithm for computing the MGU

- identical terms ⇒ nothing to do
- one is a constant and the other is not a variable ⇒ fail
- both are functions, but the names or arities differ ⇒ fail
- both are functions and the names and arities coincide ⇒ compare their arguments one by one
- one is a variable X and there is a binding for X in the current substitution ⇒ unify the binding of X with the other term
- one is a variable X, there is no binding for X in the current substitution and the other term does not contain X ⇒ add substitution that replaces X with the other term
- else ⇒ fail

Unification example

$$P(f(g(a)), X, f(h(Z, Z)), h(Y, g(W)))$$
$$P(Y, g(Z), f(V), h(f(W), X))$$

Resolution with unification

$$\frac{\begin{array}{r}\{\Phi, \Phi_1, \Phi_2, \ldots, \Phi_m\} \\ \{\neg\Psi, \Psi_1, \Psi_2, \ldots, \Psi_n\}\end{array}}{(\{\Phi_1, \Phi_2, \ldots, \Phi_m\} \cup \{\Psi_1, \Psi_2, \ldots, \Psi_n\})\beta}$$

where $\beta = mgu(\Phi, \Psi)$

- variable renaming so that we can unify $P(a, X)$ with $P(X, b)$.

Outline

1. Resolution in PL
2. Resolution in FOPL
    - Conversion to clausal form
    - Unification
    - Factors
3. Herbrand's Theorem
4. Resolution Strategies

Factors

$$\neg Shaves(X, X) \rightarrow Shaves(barber, X)$$
$$Shaves(barber, Y) \rightarrow \neg Shaves(Y, Y)$$

- clearly there is a contradiction, but we cannot infer it using resolution
- if a subset of literals from a clause Φ has a most general unifier β, then $\Phi\beta$ is a factor of Φ
- resolution with factors

$$\frac{\begin{array}{c}\Phi \\ \Psi\end{array}}{((\Phi' \setminus \{\phi\}) \cup (\Psi' \setminus \{\neg\psi\}))\beta}$$

where Φ' is a factor of Φ, Ψ' is a factor of Ψ and $\beta = mgu(\phi, \psi)$

Outline

Herbrand base / universe / interpretation

Given a set of clauses *S* we can define

- Herbrand universe = minimal domain of values that we can refer using the language
- Herbrand base = instances of every predicate for each element of the Herbrand base
- Herbrand interpretation = truth assignment for every element of the universe
- **Herbrand's theorem:** a set of clauses is unsatisfiable iff it is false for any H-interpretation of the set

Semantics trees

- semantic tree
- complete semantics tree – paths contain occurrences of all ground instances of predicates from S
- failure node – node fails to satisfy S, while parent does satisfy S
- closed semantics tree – all paths end in failure nodes
- **Herbrand's theorem (version 1):** for any semantic tree there exists a finite closed semantic tree

Second version of HT

There exists a finite set of ground instances of S that is unsatisfiable.

Completeness of resolution

Lemma: if C'_1 and C'_2 are ground instances of clauses C_1 and C_2 and $C' = rez(C'_1, C'_2)$ then there exists a clause C such that $C = rez(C_1, C_2)$ and C' is a ground instance of C.

Theorem: Resolution is refutationally complete – a set of clauses S is unsatisfiable iff there exists a deduction of the empty clause from S.

Outline

Resolution strategies

- Linear resolution
- Linear input resolution
- Level resolution
- Support set resolution
- ...