First-Order Inference

Burr H. Settles
CS-540, UW-Madison
www.cs.wisc.edu/~cs540-1
Summer 2003

Announcements (7/1)

- Discussion list topic #2: pick a favorite saying and translate it to FOL
  - Must be well-formed
  - Have at least 1 quantified variable
  - Have at least 2 connectives (\(\land\lor\))

- If someone post a translation that you disagree with, say so! (it is a discussion list, after all… you’re not being graded on being wrong or right!)

Announcements (7/2)

- Excellent FOL translations going up on the discussion list! Feel free to post more than one!
- Plan for the next week and a half:
  - No class Friday (July 4)
  - Lectures Mon & Tues next week
  - Review session Wed (TAs will run it)
  - Midterm on Thursday (July 10, in class)
  - No class next Friday (July 11)
- Project papers/reports will be due Friday, August 1 (second to last week of classes)

Inference Rules for FOL

- All inference rules for PL also apply to FOL (MP, AE, AI, OE, DNE, UR, R, DML)
- Universal Elimination, UE
  \[\forall x \phi(x) \rightarrow \phi(a)\]
  \[\phi(a)\]
- Existential Elimination, EE
  \[\exists x \phi(x) \rightarrow \phi(c)\]
  \[\phi(c)\]
- Existential Introduction, EI
  \[\phi(a) \rightarrow \exists x \phi(x)\]
  \[\exists x \phi(x)\]

Proofs for FOL

- “Thom is a turtle.”
  1. turtle(Thom)
- “Rob is a rabbit.”
  2. rabbit(Rob)
- “Turtles outlast rabbits.”
  3. \(\forall x, y \text{ turtle}(x) \land \text{ rabbit}(y) \rightarrow \text{outlasts}(x, y)\)
- Prove: “Thom outlasts Rob.”
  \(? \text{outlasts}(Thom, Rob)\)

Proofs for FOL

- And Introduction: 1, 2
  4. \(\text{turtle}(Thom) \land \text{rabbit}(Rob)\)
- Universal Elimination: 3 \((a/Thom, y/Rob)\)
  5. \(\text{turtle}(Thom) \land \text{rabbit}(Rob) \rightarrow \text{outlasts}(Thom, Rob)\)
- Modus Ponens: 4, 5
  6. \(\text{outlasts}(Thom, Rob)\)
- AI, UE, MP is a common inference pattern
FOL Proof as Search

- **States:** the current KB
- **Actions:** inference rules
- **Goal test:** see if query is in KB
- **Problem:** huge branching factor (especially for UE)
- **Idea:** Find a substitution that makes the rule premise match known facts
  - **Make new, powerful inference rule!**

Automated Inference in FOL

- **Automated inference is harder for FOL than it is for PL**
  - Variables can take on a potentially infinite number of possible values from the domain
  - Thus... UE can be applied in a potentially infinite number of ways to KB!

Generalized Modus Ponens (GMP)

- **Unify rule premises with known facts and apply unifier to conclusion**
  - **Rule:** ∀x,y. turtle(x) ∧ rabbit(y) ⇒ outrun(x,y)
    - Known facts: turtle(Thom), and rabbit(Rob)
    - Unifier: {x/Thom, y/Rob}
  - **Apply unifier to conclusion:** outrun(Thom, Rob)

Generalized Modus Ponens (GMP)

- Combines AI, UE, and MP into a single rule

  \[
  p_1; p_2; \ldots; p_k; (p_k \land p_{k} \land \ldots \land p_n) \Rightarrow q
  \]

  \[
  \text{SUBST}(\theta q)
  \]

  (where SUBST(θp_i) = SUBST(θp_i) for all i)

- **SUBST(θa)** means apply substitutions in θ to sentence a
- **Substitution list** θ = {v_1/t_1, v_2/t_2, ... , v_n/t_n} means
  - Replace all occurrences of variable v_i with term t_i
  - Substitutions are made in left to right order

Generalized Modus Ponens (GMP)

- All variables assumed to be universally quantified
- Used with a KB in Horn normal form (HNF):
  - Horn sentence: disjunction with one positive literal
    - single atomic sentence: P(x)
    - (conj. of atoms) ⇒ atom: ¬φ(x) ∨ ¬ψ(x) ∨ R(x)

Example:

  \[
  p_1 = \text{taller}(\text{Larry}, \text{Curly})
  \]

  \[
  p_2 = \text{taller}(\text{Curly}, \text{Moe})
  \]

  \[
  p_1 \land p_2 \Rightarrow q = \text{taller}(x,y) \land \text{taller}(y,z) \Rightarrow \text{taller}(x,z)
  \theta = {x/\text{Larry}, y/\text{Curly}, z/\text{Moe}}
  \]

  \[
  \text{SUBST}(\theta q) = \text{taller}(\text{Larry}, \text{Moe})
  \]
Unification

- Substitution $\theta$ is said to unify $p$ and $q$ if $\text{SUBST}(\theta p) = \text{SUBST}(\theta q)$

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>curtain(y)</td>
<td>curtain(y)</td>
<td>(y/Fish)</td>
</tr>
<tr>
<td>loves(Burr,x)</td>
<td>loves(Burr,x)</td>
<td>(x/Mother)</td>
</tr>
<tr>
<td>friends(Burr,x)</td>
<td>friends(y,Mark)</td>
<td>(y/Burr,x/Mark)</td>
</tr>
<tr>
<td>sees(x,y,z)</td>
<td>sees(x,y,z)</td>
<td></td>
</tr>
<tr>
<td>home(JD)</td>
<td>home(JD)</td>
<td></td>
</tr>
</tbody>
</table>

Unification Algorithm

- $\theta$ is a most general unifier (MGU)
  - Shortest length substitution list to make a match
  - In general, more than one MGU
- Our algorithm recursively explores the two expressions and simultaneously builds $\theta$
- We want to prevent replacing variables with terms that contains that variable (e.g. $\{x/y(x)\}$)
  - This slows down the algorithm
- Unification with this variable-substitution check has a time complexity of $O(n^2)$, where $n$ is the number of terms in the expressions

Soundness of GMP

- Note: $\theta \alpha \beta$ is the same as $\text{SUBST}(\theta, \alpha \beta)$
- We want to show:
  1. $p_1 \land \cdots \land p_n \land (p_1 \land \cdots \land p_n \Rightarrow q) \vDash q \theta$
     provided $p_i \theta = p_i \theta$ for all $i$
- Lemma: for any Horn clause $p; p \vDash q$ by UE
  1. $(p_1 \land \cdots \land p_n \Rightarrow q) \vDash (p_1 \land \cdots \land p_n \Rightarrow q) \theta = (p_1 \theta \land \cdots \land p_n \theta \Rightarrow q \theta)$
  2. $p_1 \land \cdots \land p_n \vDash p_1 \theta \land \cdots \land p_n \theta \vDash p_1 \theta \land \cdots \land p_n \theta$
  3. $q \theta$ (MP: 1, 2) since $p_i \theta = p_i \theta$ for all $i$

Inference Example

“The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, and enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is an American.”

1. american(x) ⇒ weapon(y) ⇒ sells(x,y,2) ⇒ hostile(z) ⇒ criminal(z)
2. enemy(Nono,America)
3. $\exists x$ owns(Nono,x) ∧ missile(x)
4. missile(M)
5. $\exists x$ owns(Nono,x) ∧ missile(West,x,Nono)
6. american(West)
7. $\exists x$ enemy(x, America) ∧ hostile(x)
8. missile(x) ⇒ weapon(x)
Forward Chaining with GMP

Backward Chaining with GMP

Completeness of General FOL

For full FC Algorithm, see page 282
- Sound and complete for first-order definite clauses
  - Proof similar to PL proof
- Datalog: FOL clauses with no functions (e.g. crime KB)
  - FC terminates for Datalog in at most \( p^{n^4} \) literals
- FC typically adds all sentences that can be inferred
  - Matching premises against known facts in NP-Hard
- Still, FC is used widely in deductive databases

For full BC Algorithm, see page 288
- DFS recursive proof search
- Incomplete due to possible infinite loops
  - Fix by checking current goal against every goal on the stack
- Inefficient due to repeated subgoals
  - Complications added to keep track of unifiers
  - Fix by caching previous results (but this uses up space!)
- Two versions: find any solution, find all solutions
- BC still used (without improvements!) extensively for logic programming systems (e.g. Prolog)

FC and BC are complete for KBs in Horn form, but incomplete for general FOL:
- \( \text{owns}(Burr, x) \land \text{shoe}(x) \Rightarrow \text{stinky}(x) \)
- \( \text{shoe}(x) \land \text{stinky}(x) \Rightarrow \neg \text{allowed}(x) \)
- \( \neg \text{shoe}(x) \land \neg \text{owns}(Burr, x) \land \neg \text{allowed}(x) \)
- Can’t prove query with FC or BC... why?
- Does a complete algorithm for FOL exist?

450BC Stoics PL, inference (?)  
32BC Aristotle inference rules (syllogisms), quantifiers  
1565 Cardano PL + uncertainty (probability theory)  
1847 Boole PL (again)  
1879 Frege FOL  
1922 Wittgenstein proof using truth table  
1930 Gödel complete algo for FOL exists  
1930 Herbrand complete algo for FOL (reduce to PL)  
1931 Gödel no complete algo for number theory  
1960 Davis/Putnam practical algo for PL  
1965 Robinson practical algo for FOL (resolution)
**Resolution**

- Entailment in general FOL is only semi-decidable:
  - Can prove $\alpha$ if $\text{KB} \models \alpha$
  - Cannot always prove that $\text{KB}$ doesn’t $\models \alpha$ (halting)
- Resolution is a refutation technique:
  - To prove $\text{KB} \not\models \alpha$ show that $\text{KB} \land \neg \alpha$ is unsatisfiable
- Resolution uses $\text{KB}$ and $\neg \alpha$ in CNF:
  - Conjunction of clauses that are disjunction of literals
- Resolution repeatedly combines two clauses to make a new one until an empty clause is derived (a contradiction)

**Resolution Refutation Example**

Recycling the “West is a criminal” example, let’s begin by making sure that all the facts and rules in our KB are in CNF. The following are already in CNF:

- enemy(Nono, America)
- owns(Nono, M)
- missile(M)
- American(West)

The remaining four need to be converted to CNF:

- enemy(x, America) $\implies$ hostile(x)
- missile(x) $\implies$ weapon(x)
- American(West) $\implies$ criminal(x)
- owns(x, Nono) $\land$ sells(x, West, Nono)

And we also need to negate our query: $\neg$criminal(West)

**Resolution**

- Resolution in PL equivalently:
  \[
  \alpha \lor \beta, \neg \beta \lor \gamma \equiv \alpha \lor \gamma \\
  \neg \alpha \lor \beta, \neg \beta \lor \gamma \equiv \neg \alpha \lor \gamma
  \]
- Generalized Resolution (GR) for FOL:
  where $p_j$ and $q_i$ are literals for all $i$
  \[
  p_j \lor \cdots \lor p_{n_j} \land q_1 \lor \cdots \lor q_{n_i} \lor \cdots \lor q_{n_k}
  \]
  \[
  SUBST(\theta, p_j \lor \cdots \lor p_{n_j} \land q_1 \lor \cdots \lor q_{n_i} \lor \cdots \lor q_{n_k})
  \]
  \[
  \]

- Can be thought of as search
  - Reversed construction of search tree (leaves to root)
  - Leaves are KB clauses and $\neg$-query
  - Resolvent is new node with arcs to parent clauses
  - Root is a clause containing false
- A search is complete if it guarantees that the empty clause (i.e. false) can be derived whenever $\text{KB} \not\models \alpha$
  - Goal is to design a complete search that efficiently finds a contradiction (i.e. empty clause)
Converting FOL to CNF

1. Replace $\equiv$ with equivalent: $P$ $\iff$ $Q$ becomes $P \implies Q \land Q \implies P$
2. Replace $\equiv$ with equivalent: $P$ $\iff$ $Q$ becomes $P \lor \neg Q \land \neg Q \lor P$
3. Reduce scope of \textbf{$\forall$} to single literals:
   - $\neg \neg P$ becomes $P$
   - $\neg (P \lor Q)$ becomes $P \land \neg Q$ (De Morgan’s)
   - $\neg (P \land Q)$ becomes $\neg P \lor \neg Q$ (De Morgan’s)
4. Standardize variables apart:
   - Each quantifier must have a unique variable name
   - Avoids confusion in steps 5 and 6
   - $e.g.$ $\forall x P \lor \exists y Q$ becomes $\forall x P \lor \exists y Q$
5. Eliminate existential quantifiers (Skolemize):
   - $\forall x P(x)$ becomes $P(K)$ (EE)
   - $K$ is some new constant (Skolem constant)
   - $e.g.$ $\forall x \exists y P(x, y)$ becomes $\forall x P(K, P(x))$
   - $P()$ must be a new function (Skolem function) with arguments that are all enclosing universally quantified variables
   - Everyone has a name.
   - $\forall x \text{person}(x) \implies \exists y \text{name}(y) \land \text{has}(x, y)$
   - $\forall x \text{person}(x) \implies \exists y \text{name}(y) \land \text{has}(x, y)$
   - Everyone has the same name K!
   - We want everyone to have a name based on who they are
   - Right: $\forall x \text{person}(x) \implies \exists y \text{name}(y) \land \text{has}(x, y)$
5. Reduce scope of $\land$ and $\lor$ to literals

   - $\forall x \neg \text{animal}(y) \lor \text{loves}(x,y)$ becomes $\exists y \neg \text{animal}(y) \lor \text{loves}(x,y)$
   - $\forall x \neg \text{animal}(y) \lor \text{loves}(x,y)$ becomes $\exists y \neg \text{animal}(y) \lor \text{loves}(x,y)$
5. Distribute $\lor$ over $\land$:
   - $P \land Q \lor R$ becomes $(P \lor R) \land (Q \lor R)$
6. Group conjunctions/disjunctions:
   - $(P \land Q) \lor R$ becomes $(P \land Q \land R)$
   - $(P \lor Q) \lor R$ becomes $(P \lor Q \lor R)$
FOL-CNF Conversion Example

“Everyone who loves all animals is loved by someone.”

$\forall x [\forall y \text{animal}(y) \implies \text{loves}(x,y)] \implies [\exists y \text{loves}(x,y)]$

1&2. Eliminate biconditionals and implications

$\forall x \neg (\forall y \neg \text{animal}(y) \lor \text{loves}(x,y)) \lor [\exists y \text{loves}(x,y)]$

3. Reduce scope of $\forall$ to single literals

$\forall x [\exists y \neg \text{animal}(y) \lor \text{loves}(x,y)] \lor [\exists y \text{loves}(x,y)]$

4. Standardize variables apart

$\forall x [\exists y \text{animal}(y) \land \neg \text{loves}(x,y)] \lor [\exists y \text{loves}(x,y)]$

5. Eliminate Existentials by skolemizing

$\forall x [\text{animal}(F(x)) \land \neg \text{loves}(x,F(x))] \lor [\exists y \text{loves}(x,y)]$

6. Drop universals

$[\text{animal}(F(x)) \land \neg \text{loves}(x,F(x))] \lor [\exists y \text{loves}(x,y)]$

7&8. Distribute $\lor$ over $\land$, group conjunctions/disjunctions:

$[\text{animal}(F(x)) \lor \text{loves}(G(x),x)] \land [\neg \text{loves}(x,F(x)) \lor \text{loves}(G(x),x)]$
Summary

- First-Order logic is a language that is very expressive, but difficult to perform inference with
  - One inference method is removing all variables/quantifiers (i.e. propositionalizing), which is slow
  - We can also use unification to identify appropriate substitutions with generalized Modus Ponens (GMP)
  - The Forward Chaining and Backward Chaining algorithms use GMP to KBs in HNF
  - GMP complete for definite clauses (HNF), but not general FOL domains

Summary

- Generalized resolution inference provides a complete proof system for all of FOL
  - KBs must be converted to CNF
  - Resolution refutation is an efficient strategy for proving a query \( q \) by showing that it’s negation \( \neg q \) is inconsistent with the KB
  - i.e. show that \( KB \land \neg q \) is unsatisfiable