Knowledge & Reasoning

• Logical Reasoning: to have a computer automatically perform deduction or prove theorems

• Knowledge Representations: modern ways of representing large bodies of knowledge
Logical Reasoning

• In order to communicate, we need a formal language in which to express
  – axioms
  – theorems
  – hypotheses
  – rules

• Common languages include
  – propositional logic
  – 1\textsuperscript{st} order predicate logic
Propositional Logic

• Propositions are statements that are true or false.
  – P: Sierra is a dog
  – Q: Muffy is a cat
  – R: Sierra and Muffy are not friends

• Propositions can be combined using logic symbols

  \[ P \land Q \Rightarrow R \quad \neg P \lor Q \]
Predicate Logic

• Formulas have predicates with variables and constants:
  – man(Marcus)
  – Pompeian(Marcus)
  – born(Marcus, 40)

• More symbols
  – ∀ for every \( \forall x \text{ Pompeian}(x) \Rightarrow \text{died}(x,79) \)
  – ∃ there exists \( \exists x \text{ Pompeian}(x) \)
Vesuvius
Ancient Pompei and Vesuvius

What happened to ancient Pompei?

Vesuvius erupted and killed everyone.

When?

79 A.D.
Predicate Logic Example

1. Pompeian(Marcus)
2. born(Marcus,40)
3. man(Marcus)
4. $\forall x \text{ man}(x) \Rightarrow \text{mortal}(x)$
5. $\forall x \text{ Pompeian}(x) \Rightarrow \text{died}(x,79)$
6. erupted(Vesuvius,79)
7. $\forall x \forall t1 \forall t2 \text{ mortal}(x) \land \text{born}(x,t1) \land \text{gt}(t2-t1,150) \Rightarrow \text{dead}(x,t2)$
Dead Guy in 2009

8. gt(now, 79)
Some Rules of Inference

9. $\forall x \forall t \ [\text{alive}(x,t) \Rightarrow \neg \text{dead}(x,t)] \land \neg \text{dead}(x,t) \Rightarrow \text{alive}(x,t)$

If x is alive at time t, he’s not dead at time t, and vice versa.

10. $\forall x \forall t1 \forall t2 \text{died}(x,t1) \land \text{gt}(t2,t1) \Rightarrow \text{dead}(x,t2)$

If x died at time t1 and t2 is later, x is still dead at t2.
Prove dead(Marcus, now)

1. Pompeian(Marcus)
2. born(Marcus, 40)
3. man(Marcus)
4. \( \forall x \) man(x) \( \Rightarrow \) mortal(x)
5. \( \forall x \) Pompeian(x) \( \Rightarrow \) died(x, 79)
6. erupted(Vesuvius, 79)
7. \( \forall x \) \( \forall t_1 \) \( \forall t_2 \) mortal(x) \( \land \) born(x, t1) \( \land \) gt(t2-t1, 150) \( \Rightarrow \) dead(x, t2)
8. gt(now, 79)
9. \( \forall x \) \( \forall t \) [alive(x, t) \( \Rightarrow \) \( \neg \)dead(x, t)] \( \land \) [\( \neg \)dead(x, t) \( \Rightarrow \) alive(x, t)]
10. \( \forall x \) \( \forall t_1 \) \( \forall t_2 \) died(x, t1) \( \land \) gt(t2, t1) \( \Rightarrow \) dead(x, t2)
Prove dead(Marcus, now)

Direct Proof

1. Pompeian(Marcus)

5. \( \forall x \text{ Pompeian}(x) \implies \text{died}(x,79) \)

\[
\begin{align*}
\text{died}(\text{Marcus},79)
\end{align*}
\]

8. \( \text{gt}(\text{now},79) \)

\[
\begin{align*}
\text{died}(\text{Marcus},79) \land \text{gt}(\text{now},79)
\end{align*}
\]

7. \( \forall x \ \forall t_1 \ \forall t_2 \ \text{died}(x,t_1) \land \text{gt}(t_2,t_1) \implies \text{dead}(x,t_2) \)

\[
\begin{align*}
\text{dead}(\text{Marcus},\text{now})
\end{align*}
\]
Proof by Contradiction

\[ \neg \text{dead(Marcus, now)} \]

\[ \forall x \forall t1 \forall t2 \text{ died}(x, t1) \land \text{gt}(t2, t1) \Rightarrow \text{dead}(x, t2) \]

\[ \forall t1 \neg [\text{died}(\text{Marcus, } t1) \land \text{gt}(\text{now, } t1)] \]

What substitutions were made here?
What rule of inference was used?

Marcus for \( x \); now for \( t2 \)

If \( x \Rightarrow y \) then \( \neg y \Rightarrow \neg x \)
Proof by Contradiction

\( \neg \text{dead(Marcus, now)} \)

\( \forall x \ \forall t1 \ \forall t2 \ \text{died}(x, t1) \land \text{gt}(t2, t1) \Rightarrow \text{dead}(x, t2) \)

\( \forall t1 \ \neg [\text{died}(\text{Marcus}, t1) \land \text{gt}(\text{now}, t1)] \)

\( \forall t1 \ \neg \text{died}(\text{Marcus}, t1) \lor \neg \text{gt}(\text{now}, t1) \)

\( \neg \text{gt}(\text{now}, 79) \)

\( \text{died(Marcus, 79)} \)

\( \text{gt(now, 79)} \)

contradiction

*assume we proved this separately
Resolution Theorem Provers for Predicate Logic

• Given:
  – F: a set of axioms represented as formulas
  – S: a conjecture represented as a formula

• Prove: F logically implies S

• Technique
  – Construct \( \neg S \), the negated conjecture
  – Show that \( F' = F \cup \{\neg S\} \) leads to a contradiction
  – Conclude: \( \neg \neg S \) or \( S \)
Part I: Preprocessing to express in Conjunctive Normal Form

1. Eliminate implication operator \( \Rightarrow \)
   - Replace \( A \Rightarrow B \) by \( \lor (\neg A, B) \)

   - Example:
     \( \text{man}(x) \Rightarrow \text{mortal}(x) \) is replaced by
     \( \lor (\neg \text{man}(x), \text{mortal}(x)) \) or in infix notation
     \( \neg \text{man}(x) \lor \text{mortal}(x) \)
2. Reduce the scope of each \( \neg \) to apply to at most one predicate by applying rules:

- Demorgan’s Laws
  \[ \neg \lor(x_1,\ldots,x_n) \text{ is equivalent to } \land(\neg x_1,\ldots, \neg x_n) \]
  \[ \neg \land(x_1,\ldots,x_n) \text{ is equivalent to } \lor(\neg x_1,\ldots, \neg x_n) \]
- \( \neg(\neg x) \Rightarrow x \)
- \( \neg(\forall x \ A) \Rightarrow \exists x(\neg A) \)
- \( \neg(\exists x \ A) \Rightarrow \forall x(\neg A) \)
Preprocessing Continued

• Example
\[ \neg [\forall x \forall t1 \forall t2 [died(x,t1) \land gt(t2,t1)] \Rightarrow dead(x,t2)] \]

• Get rid of the implication
\[ \neg [\forall x \forall t1 \forall t2 \neg [died(x,t1) \land gt(t2,t1)] \lor dead(x,t2)] \]

• Apply the rule for \( \neg [\forall \exists x \exists t1 \exists t2 \neg (\neg [died(x,t1) \land gt(t2,t1)] \lor dead(x,t2)) \]

• Apply DeMorgan’s Law
\[ \exists x \exists t1 \exists t2 \neg \neg [died(x,t1) \land gt(t2,t1)] \land \neg dead(x,t2) \]
\[ \exists x \exists t1 \exists t2 died(x,t1) \land gt(t2,t1) \land \neg dead(x,t2) \]
3. Standardize Variables

Rename variables so that each quantifier binds a unique variable

\( \forall x [P(x) \land \exists x Q(x)] \)

becomes

\( \forall x [P(x) \land \exists y Q(y)] \)
• 4. Eliminate existential qualifiers by introducing \textbf{Skolem functions}.

• Example

\[ \forall x \; \forall y \; \exists z \; P(x,y,z) \]

• The variable \( z \) depends on \( x \) and \( y \).

• So \( z \) is a function of \( x \) and \( y \).

• We choose an arbitrary function name, say \( f \), and replace \( z \) by \( f(x,y) \), eliminating the \( \exists \).

\[ \forall x \; \forall y \; P(x,y,f(x,y)) \]
5. Rewrite the result in Conjunctive Normal Form (CNF)

\( \land (x_1, \ldots, x_n) \) where the \( x_i \) can be

- atomic formulas \( A(x) \)
- negated atomic formulas \( \neg A(x) \)
- disjunctions \( A(x) \lor P(y) \)

This uses the rule

\( \lor (x_1, \land (x_2, \ldots, x_n)) = \land (\lor (x_1, x_2), \ldots, \lor (x_1, x_n)) \)
6. Since all the variables are now only universally quantified, eliminate the $\forall$ as understood.

$$\forall x \; \forall t1 \; \forall t2 \; \neg \text{died}(x,t1) \lor \neg \text{gt}(t2,t1) \lor \text{dead}(x,t2)$$

becomes

$$\neg \text{died}(x,t1) \lor \neg \text{gt}(t2,t1) \lor \text{dead}(x,t2)$$
Clause Form

- The clause form of a set of original formulas consists of a set of clauses as follows.
  - A literal is an atom or negation of atom.
  - A clause is a disjunction of literals.
  - A formula is a conjunction of clauses.

Example

Clause 1: \{A(x), \neg P(g(x,y),z), \neg R(z)\} (implicit or)
Clause 2: \{C(x,y), Q(x,y,z)\} (another implicit or)
Steps in Proving a Conjecture

1. Given a set of axioms $F$ and a conjecture $S$, let $F' = F \cup \neg S$ and find the clause form $C$ of $F'$.

2. Iteratively try to find new clauses that are logically implied by $C$.

3. If $\text{NIL}$ is one of these clauses you produce, then $F'$ is unsatisfiable and the conjecture is proved.

4. You get $\text{NIL}$ when you produce something that has $A$ and also has $\neg A$. 
Resolution Procedure

1. Convert F to clause form: a set of clauses.
2. Negate S, convert it to clause form, and add it to your set of clauses.
3. Repeat until a contradiction or no progress
   a. Select two parent clauses.
   b. Produce their resolvent.
   c. If the resolvent = NIL, we are done.
   d. Else add the resolvent to the set of clauses.
Resolution for Propositions

- Let $C_1 = L_1 \lor L_2 \lor ... \lor L_n$
- Let $C_2 = L_1' \lor L_2' \lor ... \lor L_n'$
- If $C_1$ has a literal $L$ and $C_2$ has the opposite literal $\neg L$, they cancel each other and produce
  
  \[
  \text{resolvent}(C_1, C_2) = L_1 \lor L_2 \lor ... \lor L_n \lor L_1' \lor L_2' \lor ... \lor L_n'
  \]
  
  with both $L$ and $\neg L$ removed.
- If no 2 literals cancel, nothing is removed.
Propositional Logic Example

Formulas: $P \lor Q$, $P \Rightarrow Q$, $Q \Rightarrow R$

Conjecture: $R$

Negation of conjecture: $\neg R$

Clauses: \{ $P \lor Q$, $\neg P \lor Q$, $\neg Q \lor R$, $\neg R$ \}

Resolvent($P \lor Q$, $\neg P \lor Q$) is $Q$. Add $Q$ to clauses.

Resolvent($\neg Q \lor R$, $\neg R$) is $\neg Q$. Add $\neg Q$ to clauses.

Resolvent($Q$, $\neg Q$) is NIL.

The conjecture is proved.
Refutation Graph

Original Clauses: \{P \lor Q, \neg P \lor Q, \neg Q \lor R, \neg R\}
Exercise

• Given $P \Rightarrow R$ and $R \Rightarrow Q$, prove that $P \Rightarrow Q$
Resolution for Predicates

• Requires a matching procedure that compares 2 literals and determines whether there is a set of substitutions that makes them identical.

• This procedure is called unification.

\[ C_1 = \text{eats(Tom, x)} \]
\[ C_2 = \text{eats(Tom, ice cream)} \]

• The substitution \text{ice cream/x} (read “ice cream for x”) makes \( C_1 = C_2 \).

• You can substitute constants for variables and variables for variables, but nothing for constants.
Proof Using Unification

• Given $\forall x \ P(x) \Rightarrow R(x)$

$\forall z \ R(z) \Rightarrow Q(z)$

• Prove $\forall x \ P(x) \Rightarrow Q(x)$

• Negation $\neg \forall x \ P(x) \Rightarrow Q(x)$

• $\exists x \ \neg (P(x) \Rightarrow Q(x))$

• $\exists x \ \neg (\neg P(x) \lor Q(x))$

• $\exists x \ P(x) \land \neg Q(x)$

• $P(a) \land \neg Q(a)^*$

$\{P(a)\} \ {\neg Q(a)}$

* Skolem function for a single variable is just a constant
Refutation Graph with Unification

\{\neg P(x), R(x)\} \quad \{\neg R(z), Q(z)\}

Substitution
\ x/z

\{\neg P(x), Q(x)\} \quad \{P(a)\}

Substitution
\ a/x

Q(a) \quad \neg Q(a)

NIL
Another Pompeian Example

1. $\text{man}(\text{Marcus})$
2. $\text{Pompeian}(\text{Marcus})$
3. $\neg \text{Pompeian}(x_1) \lor \text{Roman}(x_1)$
4. $\text{ruler}(\text{Caesar})$
5. $\neg \text{Roman}(x_2) \lor \text{loyalto}(x_2, \text{Caesar}) \lor \text{hate}(x_2, \text{Caesar})$
6. $\text{loyalto}(x_3, f_1(x_3))$
7. $\neg \text{man}(x_4) \lor \neg \text{ruler}(y_1) \lor \neg \text{tryassissinate}(x_4, y_1) \lor$
   $\neg \text{loyalto}(x_4, y_1)$
8. $\text{tryassissinate}(\text{Marcus}, \text{Caesar})$

Prove: Marcus hates Caesar
Another Pompeian Example

5. \( \neg \text{Roman}(x_2) \lor \text{loyalto}(x_2, \text{Caesar}) \lor \text{hate}(x_2, \text{Caesar}) \)
6. \( \text{loyalto}(x_3, f_1(x_3)) \)
7. \( \neg \text{man}(x_4) \lor \neg \text{ruler}(y_1) \lor \neg \text{tryassissinate}(x_4, y_1) \lor \neg \text{loyalto}(x_4, y_1) \)
8. \( \text{tryassissinate}(\text{Marcus}, \text{Caesar}) \)

5. If \( x_2 \) is Roman and not loyal to Caesar then \( x_2 \) hates Caesar.
6. For every \( x_3 \), there is someone he is loyal to.
7. If \( x_4 \) is a man and \( y_1 \) is a ruler and \( x_4 \) tries to assassinate \( x_1 \) then \( x_4 \) is not loyal to \( y_1 \).
8. Marcus tried to assassinate Caesar.
Prove: \( \text{hate(Marcus, Caesar)} \)

\[ \neg \text{hate(Marcus, Caesar)} \]

5

\[ \text{Marcus/x2} \]

3

\[ \neg \text{Roman(Marcus)} \lor \text{loyalto(Marcus, Caesar)} \]

\[ \text{Marcus/x1} \]

2

\[ \neg \text{Pompeian(Marcus)} \lor \text{loyalto(Marcus, Caesar)} \]

7

\[ \text{loyalto(Marcus, Caesar)} \]

\[ \text{Marcus/x4, Caesar/y1} \]

1

\[ \neg \text{man(Marcus)} \lor \neg \text{ruler(Caesar)} \lor \neg \text{tryassassinate(Marcus, Caesar)} \]

\[ \neg \text{ruler(Caesar)} \lor \neg \text{tryassassinate(Marcus, Caesar)} \]

4

\[ \text{tryassassinate(Marcus, Caesar)} \]

8

\[ \square \]
The Monkey-Bananas Problem (Simplified)

Axioms

1) \( \forall x \forall s \{-\text{ONBOX}(s) \rightarrow \text{AT}(\text{box, } x, \text{pushbox}(x,s))\} \)

For each position \( x \) and state \( s \), if the monkey isn’t on the box in state \( s \), then the box will be pushed to position \( x \) and the new state is \( \text{pushbox}(x,s) \).

2) \( \forall s \{\text{ONBOX(\text{climbbox}(s))}\} \)

For all states \( s \), the monkey will be on the box in the state achieved by applying \( \text{climbbox} \) to \( s \).

3) \( \forall s \{\text{ONBOX}(s) \land \text{AT}(\text{box, c, s}) \rightarrow \text{HB(\text{grasp}(s))}\} \)

For all states \( s \), if the monkey is on the box and the box is at position \( c \) in state \( s \), then \( \text{HB} \) is true of the state attained by applying \( \text{grasp} \) to \( s \).

4) \( \forall x \forall s \{\text{AT}(\text{box, } x, s) \rightarrow \text{AT}(\text{box, } x, \text{climbbox}(s))\} \)

The position of the box does not change when the monkey climbs on it, but the state does.

5) \( -\text{ONBOX}(s_0) \)
Conjecture: \( \exists s \ HB(s) \)

Negation: \( \forall s \ \neg HB(s) \) or \( \neg HB(s) \)

Refutation Graph:

- \( \neg HB(t) \)
  - \( \neg ONBOX(s), \neg AT(box, c, s), HB(grasp(s)) \)
  - \( grasp(s) \| t \)

- \( \neg ONBOX(s), \neg AT(box, c, s) \)
  - \( ONBOX(climbbox(s)) \)
  - \( climbbox(s) \| s \)

- \( \neg AT(box, c, climbbox(s)) \)
  - \( \neg AT(box, x, s), AT(box, x, climbbox(s)) \)
  - \( c \| x \)

- \( \neg AT(box, c, s) \)
  - \( ONBOX(o), AT(box, x, pushbox(x, o)) \)
  - \( c \| x \)
  - \( pushbox(c, s) \| s \)

- \( ONBOX(s) \)
- \( \neg ONBOX(s_0) \)
- \( NIL \)
Monkey Solution

- If we change the conjecture to \( \{\neg \text{HB}(s), \text{HB}(s)\} \) the result of the refutation becomes:

\[ \text{HB(grasp(climbbbox(pushbox(c,s0))))} \]
Propositional Logic Resolution

Exercise

• Given: P V Q
  P -> R
  Q -> R

• Prove R
Predicate Logic Resolution Exercise

• Given: Sierra is a dog
  Muffy is a cat
  All dogs chase all cats.

• Prove: Sierra chases Muffy
Predicate Logic Resolution Exercise

• Given: Sierra is a dog {dog(Sierra)}
  Muffy is a cat {cat(Muffy)}
  All dogs chase all cats.
    \( \forall x \forall y (\text{dog}(x) \land \text{cat}(y)) \rightarrow \text{chase}(x,y) \)
    \( \forall x \forall y \neg (\text{dog}(x) \land \text{cat}(y)) \lor \text{chase}(x,y) \)
    \( \forall x \forall y \neg \text{dog}(x) \lor \neg \text{cat}(y) \lor \text{chase}(x,y) \)
    \{\neg \text{dog}(x), \neg \text{cat}(y), \text{chase}(x,y)\}

• Prove: Sierra chases Muffy

• Negate: \( \neg \text{chase}(\text{Sierra}, \text{Muffy}) \)
\{\neg \text{chase}(\text{Sierra, Muffy})\}

\{\neg \text{dog}(x), \neg \text{cat}(y), \text{chase}(x,y)\}

\{\neg \text{dog}(\text{Sierra}), \neg \text{cat}(\text{Muffy})\}

\{\text{dog}(\text{Sierra})\}

\{\neg \text{cat}(\text{Muffy})\}

\text{cat}(\text{Muffy})

\text{NIL}