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^{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

$$P \wedge Q \Rightarrow R \qquad \neg P \vee Q$$

Predicate Logic

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

- More symbols
  - $\forall$ for every — $\forall x$ Pompeian(x) $\Rightarrow$ died(x,79)
  - $\exists$ there exists — $\exists x$ Pompeian(x)

Ancient Pompei

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. $\forall x$ man(x) $\Rightarrow$ mortal(x)
5. $\forall x$ Pompeian(x) $\Rightarrow$ died(x,79)
6. erupted(Vesuvius,79)
7. $\forall x\ \forall t1\ \forall t2$ mortal(x) $\wedge$ born(x,t1) $\wedge$ gt(t2-t1,150)$\Rightarrow$ dead(x,t2)

Dead Guy in 2009

8. gt(now,79)

Some Rules of Inference

9. $\forall x \; \forall t \; [alive(x,t) \Rightarrow \neg dead(x,t)] \wedge$
$[\neg dead(x,t) \Rightarrow alive(x,t)]$

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

10. $\forall x \; \forall t1 \; \forall t2 \; died(x,t1) \wedge gt(t2,t1) \Rightarrow dead(x,t2)$

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. $\forall x\ \forall t1\ \forall t2\ mortal(x) \wedge born(x,t1) \wedge gt(t2-t1,150) \Rightarrow dead(x,t2)$
8. gt(now,79)
9. $\forall x\ \forall t\ [alive(x,t) \Rightarrow \neg dead(x,t)] \wedge [\neg dead(x,t) \Rightarrow alive(x,t)]$
10. $\forall x\ \forall t1\ \forall t2\ died(x,t1) \wedge gt(t2,t1) \Rightarrow dead(x,t2)$

Prove dead(Marcus,now)
Direct Proof

1. Pompeian(Marcus)

5. $\forall x$ Pompeian(x) $\Rightarrow$ died(x,79)

died(Marcus,79)

8. gt(now,79)

died(Marcus,79) $\wedge$ gt(now,79)

7. $\forall x$ $\forall t1$ $\forall t2$ died(x,t1) $\wedge$ gt(t2,t1) $\Rightarrow$ dead(x,t2)

dead(Marcus,now)

Proof by Contradiction

$\neg$ dead(Marcus,now)

$\forall x\ \forall t1\ \forall t2\ died(x,t1) \wedge gt(t2,t1) \Rightarrow dead(x,t2)$

$\forall t1\ \neg\ [died(Marcus,t1) \wedge gt(now,t1)]$

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

$\neg$ dead(Marcus,now)

$\forall$x $\forall$t1 $\forall$t2 died(x,t1) $\wedge$ gt(t2,t1) $\Rightarrow$ dead(x,t2)

$\forall$t1 $\neg$ [died(Marcus,t1) $\wedge$ gt(now,t1)]

$\forall$t1 $\neg$ died(Marcus,t1) $\vee$ $\neg$ gt(now,t1)

died(Marcus,79)*

$\neg$ gt(now,79)

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 $\vee(\neg A, B)$

- Example:

  man(x) $\Rightarrow$ mortal(x) is replaced by

  $\vee(\neg$ man(x),mortal(x)) or in infix notation

  $\neg$ man(x) $\vee$ mortal(x)

Preprocessing Continued

2. Reduce the scope of each $\neg$ to apply to at most one predicate by applying rules:

- Demorgan's Laws

  $\neg \vee(x1,\ldots,xn)$ is equivalent to $\wedge(\neg x1,\ldots, \neg xn)$

  $\neg \wedge(x1,\ldots,xn)$ is equivalent to $\vee(\neg x1,\ldots, \neg xn)$

- $\neg(\neg x) \Rightarrow x$
- $\neg(\forall x\ A) \Rightarrow \exists x(\neg A)$
- $\neg(\exists\ x\ A) \Rightarrow \forall\ x(\neg A)$

Preprocessing Continued

- Example

$\neg [\forall x\ \forall t1\ \forall t2\ [died(x,t1) \wedge gt(t2,t1)] \Rightarrow dead(x,t2)]$

- Get rid of the implication

$\neg [\forall x\ \forall t1\ \forall t2\ \neg [died(x,t1) \wedge gt(t2,t1)] \vee dead(x,t2)]$

- Apply the rule for $\neg[\forall$

$\exists x\ \exists t1\ \exists t2\ \neg(\neg [died(x,t1) \wedge gt(t2,t1)] \vee dead(x,t2))$

- Apply DeMorgan's Law

$\exists x\ \exists t1\ \exists t2\ \neg \neg [died(x,t1) \wedge gt(t2,t1)] \wedge \neg dead(x,t2)$

$\exists x\ \exists t1\ \exists t2\ died(x,t1) \wedge gt(t2,t1) \wedge \neg dead(x,t2)]$

Preprocessing Continued

3. Standardize Variables

Rename variables so that each quantifier binds a unique variable

$\forall x\ [P(x) \wedge \exists x\ Q(x)]$

becomes

$\forall x\ [P(x) \wedge \exists y\ Q(y)]$

Preprocessing Continued

- 4. Eliminate existential qualifiers by introducing 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 $\exists$.

$\forall x\ \forall y\ P(x,y,f(x,y))$

Preprocessing Continued

5. Rewrite the result in Conjunctive Normal Form (CNF)

$\wedge$ (x1,…,xn) where the xi can be

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

This uses the rule

$$\vee(x1, \wedge(x2, \ldots, xn)) = \wedge(\vee(x1,x2), \ldots, \vee(x1,xn))$$

Preprocessing Continued

6. Since all the variables are now only universally quantified, eliminate the $\forall$ as understood.

$\forall x\ \forall t1\ \forall t2\ \neg died(x,t1) \lor \neg gt(t2,t1) \lor dead(x,t2)$

becomes

$\neg died(x,t1) \lor \neg gt(t2,t1) \lor dead(x,t2)$

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 NIL is one of these clauses you produce, then F' is unsatisfiable and the conjecture is proved.
4. You get 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 C1 = L1 ∨ L2 ∨ … ∨ Ln
- Let C2 = L1’ ∨ L2’ ∨ … ∨ Ln’
- If C1 has a literal L and C2 has the opposite literal ¬L, they cancel each other and produce

  resolvent(C1,C2) =

  L1 ∨ L2 ∨ … ∨ Ln ∨ L1’ ∨ L2’ ∨ … ∨ Ln’

  with both L and ¬L removed
- If no 2 literals cancel, nothing is removed

Propositional Logic Example

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

Conjecture: R

Negation of conjecture: $\neg R$

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

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

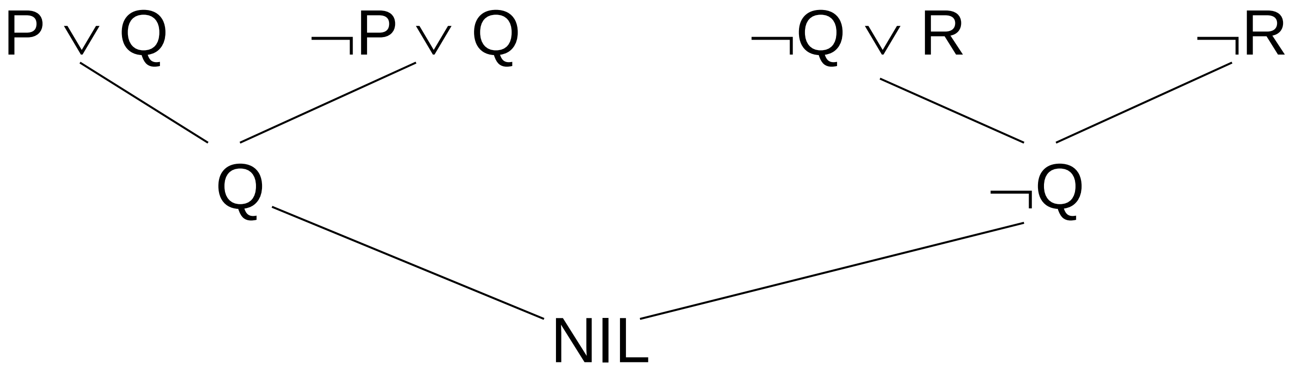
Resolvent($\neg Q \vee R$, $\neg R$) is $\neg Q$. Add $\neg Q$ to clauses.

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

The conjecture is proved.

Refutation Graph

Original Clauses: $\{P \vee Q, \neg P \vee Q, \neg Q \vee R, \neg R\}$



Exercise

- Given $P \Longrightarrow R$ and $R \Longrightarrow Q$, prove that $P \Longrightarrow Q$

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.

  C1 = eats(Tom x)
  C2 = eats(Tom, ice cream)

- The substituion ice cream/x (read “ice cream for x”) makes C1 = C2.
- You can substitute constants for variables and variables for variables, but nothing for constants.

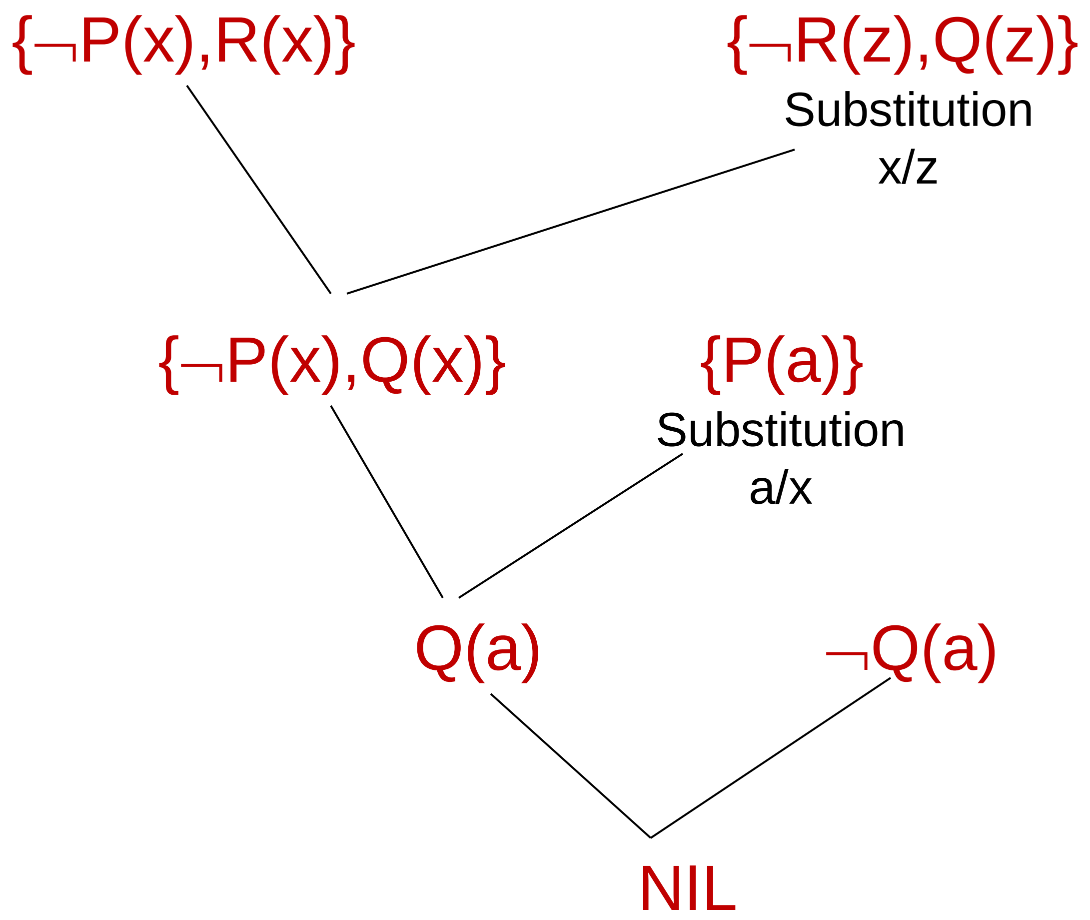
Proof Using Unification

- Given $\forall x\ P(x) \Rightarrow R(x)$ $\{\neg P(x), R(x)\}$

  $\forall z\ R(z) \Rightarrow Q(z)$ $\{\neg R(z), Q(z)\}$
- Prove $\forall x\ P(x) \Rightarrow Q(x)$
- Negation $\neg\ \forall x\ P(x) \Rightarrow Q(x)$
- $\exists x\ \neg(P(x) \Rightarrow Q(x))$
- $\exists x\ \neg\ (\neg P(x) \vee Q(x))$
- $\exists x\ P(x) \wedge \neg\ Q(x)$
- $P(a) \wedge \neg\ Q(a)$* $\{P(a)\}$ $\{\neg\ Q(a)\}$

* Skolem function for a single variable is just a constant

Refutation Graph with Unification



Another Pompeian Example

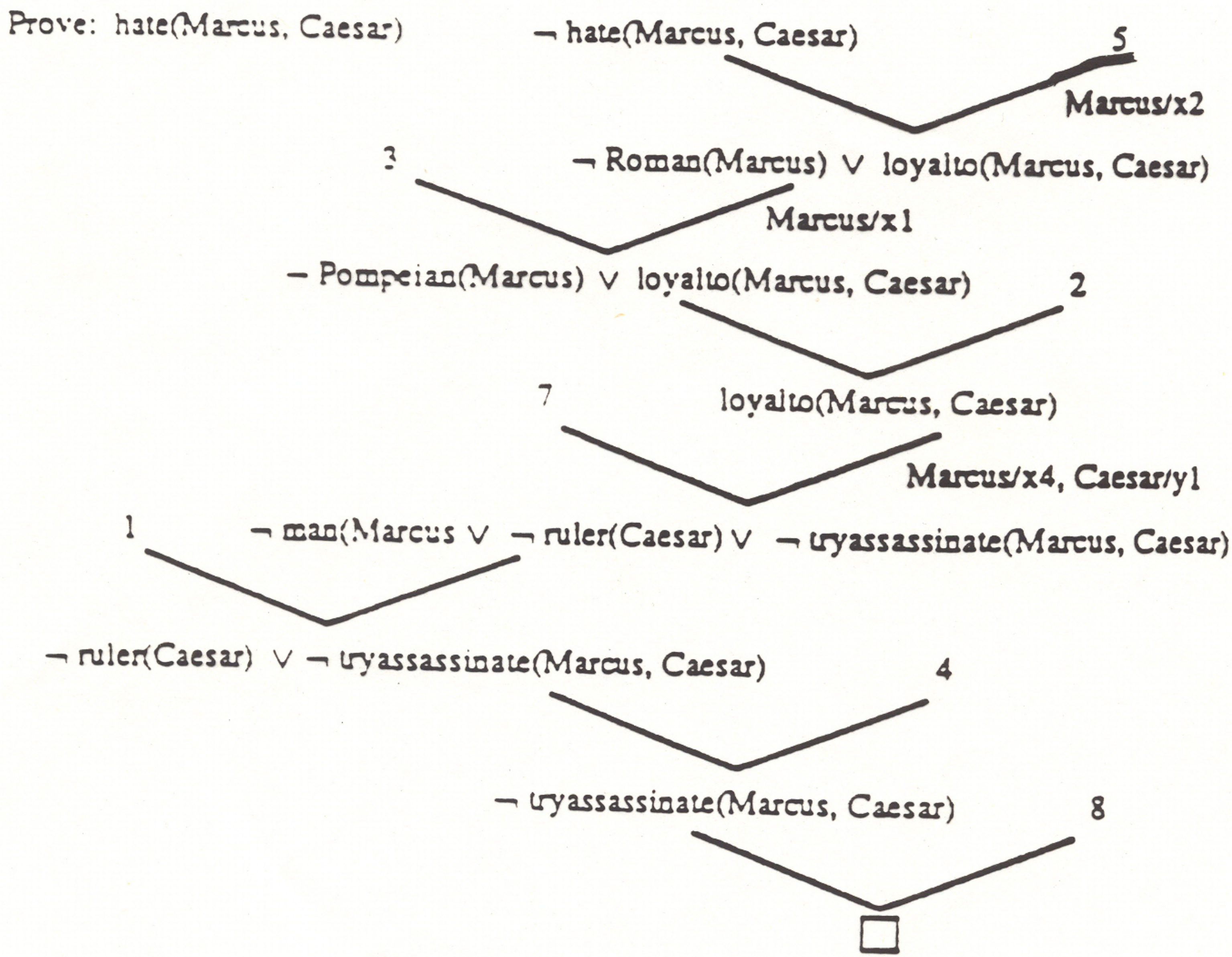
1. man(Marcus)
2. Pompeian(Marcus)
3. $\neg$Pompeian(x1) $\vee$ Roman(x1)
4. ruler(Caesar)
5. $\neg$Roman(x2) $\vee$ loyalto(x2,Caesar) $\vee$ hate(x2,Caesar)
6. loyalto(x3,f1(x3))
7. $\neg$man(x4) $\vee$ $\neg$ruler(y1) $\vee$ $\neg$tryassissinate(x4,y1) $\vee$ $\neg$loyalto(x4,y1)
8. tryassissinate(Marcus,Caesar)

Prove: Marcus hates Caesar

Another Pompeian Example

5. $\neg$Roman(x2) $\vee$ loyalto(x2,Caesar) $\vee$ hate(x2,Caesar)
6. loyalto(x3,f1(x3))
7. $\neg$man(x4) $\vee$ $\neg$ruler(y1) $\vee$ $\neg$tryassissinate(x4,y1) $\vee$ $\neg$loyalto(x4,y1)
8. tryassissinate(Marcus,Caesar)

5. If x2 is Roman and not loyal to Caesar then x2 hates Caesar.
6. For every x3, there is someone he is loyal to.
7. If x4 is a man and y1 is a ruler and x4 tries to assassinate x1 then x4 is not loyal to y1.
8. Marcus tried to assassinate Caesar.


Prove: hate(Marcus, Caesar)
¬ hate(Marcus, Caesar)
5
Marcus/x2
3
¬ Roman(Marcus) ∨ loyalto(Marcus, Caesar)
Marcus/x1
¬ Pompeian(Marcus) ∨ loyalto(Marcus, Caesar)
2
7
loyalto(Marcus, Caesar)
Marcus/x4, Caesar/y1
1
¬ man(Marcus ∨ ¬ ruler(Caesar) ∨ ¬ tryassassinate(Marcus, Caesar)
¬ ruler(Caesar) ∨ ¬ tryassassinate(Marcus, Caesar)
4
¬ tryassassinate(Marcus, Caesar)
8
□

The Monkey-Bananas Problem (Simplified)

Axioms

1) $\forall x\ \forall s\{\neg ONBOX(s) \rightarrow AT(box, x, pushbox(x,s))\}$

   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 pushbox(x,s).

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

   For all states s, the monkey will be on the box in the state achieved by applying climbbox to s.

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

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

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

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

5) $\neg ONBOX(s_o)$

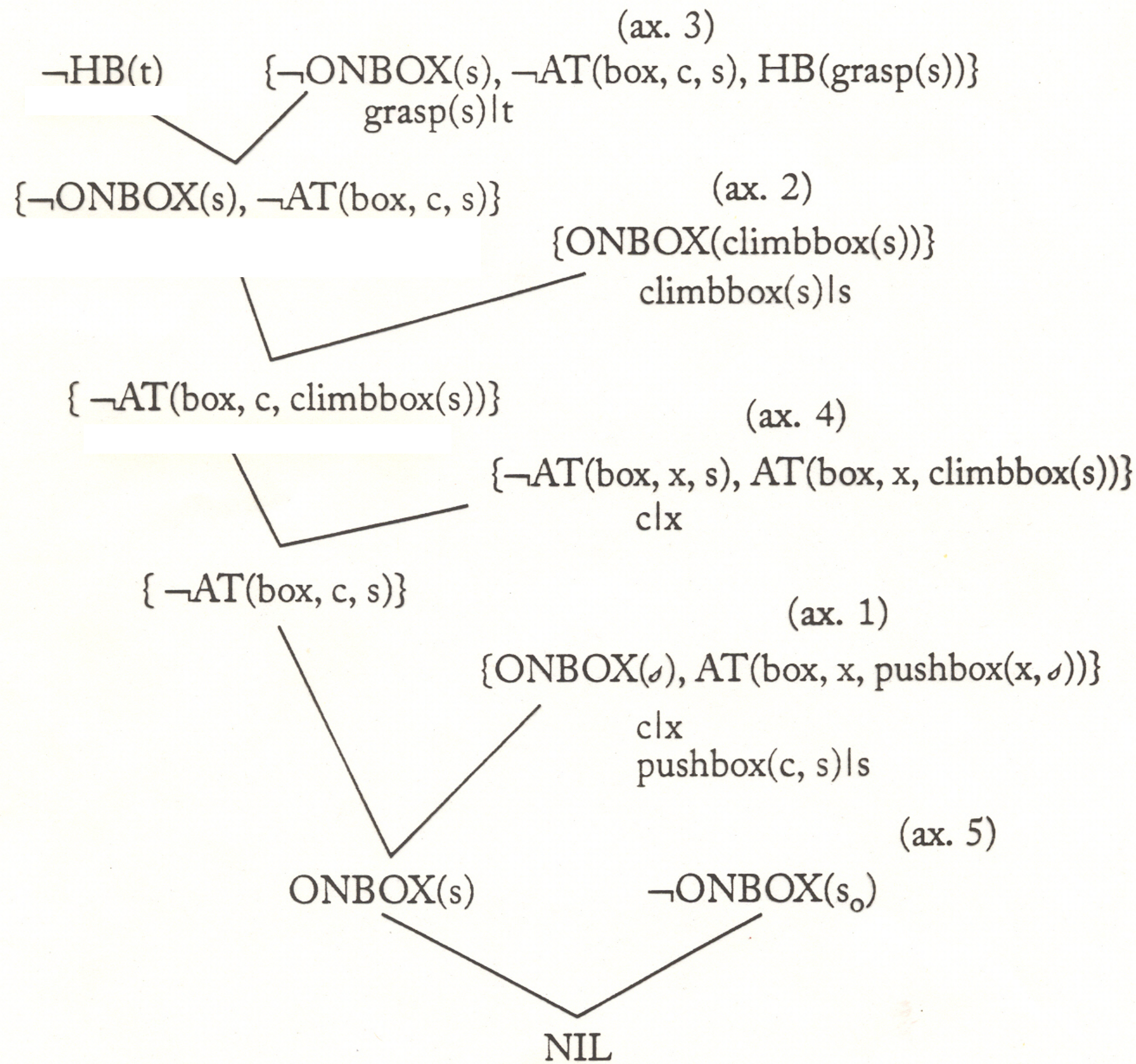
Conjecture

$\exists s\ HB(s)$

Negation

$\forall s \neg HB(s)$ or $\neg HB(s)$

Refutation Graph



Monkey Solution

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

  HB(grasp(climbbox(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.

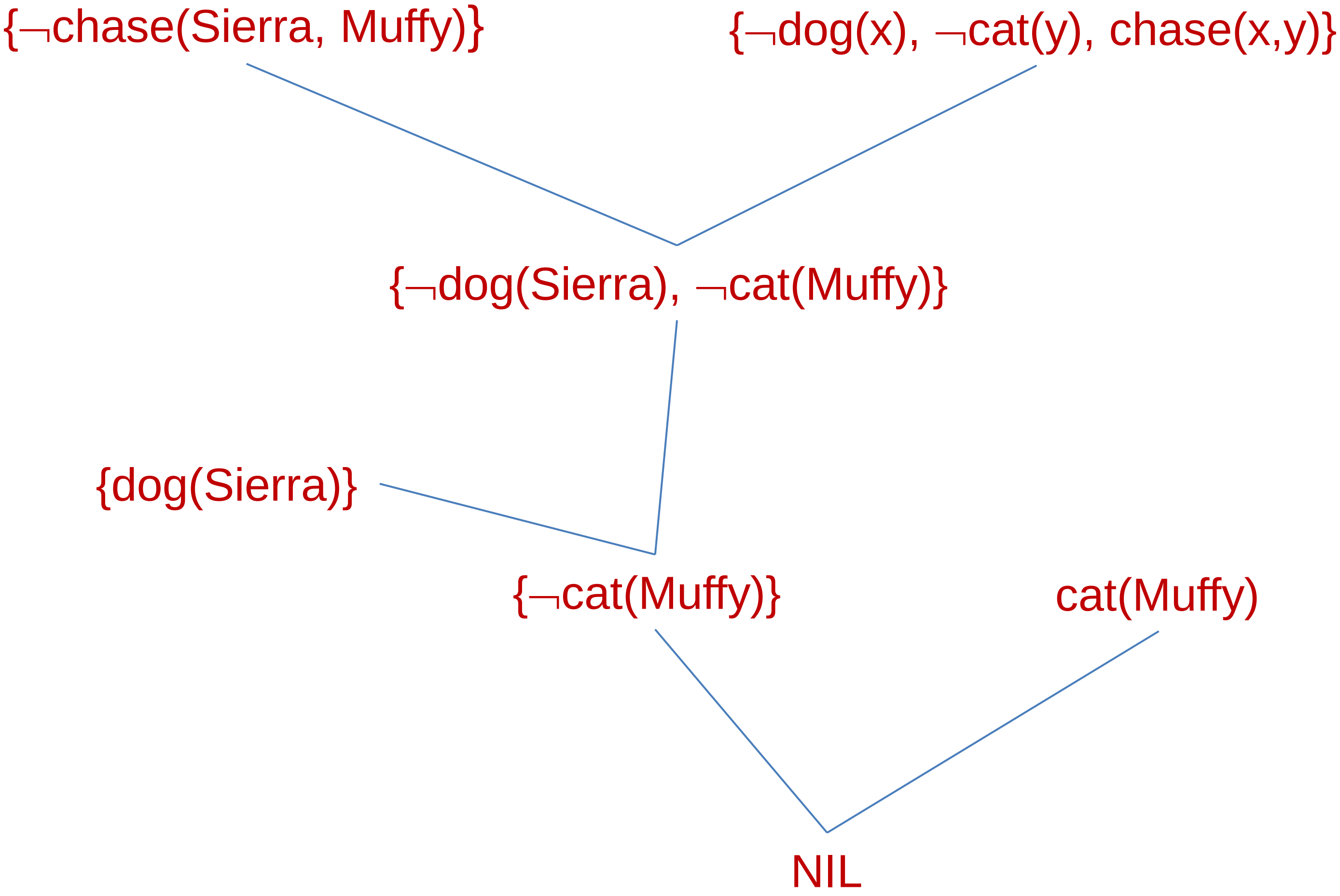
  $\forall x\ \forall y\ (dog(x) \wedge cat(y)) \rightarrow chase(x,y)$

  $\forall x\ \forall y\ \neg(dog(x) \wedge cat(y)) \vee chase(x,y)$

  $\forall x\ \forall y\ \neg dog(x) \vee \neg cat(y) \vee chase(x,y)$

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

- Prove: Sierra chases Muffy
- Negate: $\{\neg chase(Sierra, Muffy)\}$


{¬chase(Sierra, Muffy)}
{¬dog(x), ¬cat(y), chase(x,y)}
{¬dog(Sierra), ¬cat(Muffy)}
{dog(Sierra)}
{¬cat(Muffy)}
cat(Muffy)
NIL