Artificial Intelligence
CS 6364

Professor Dan Moldovan

Section 5

First Order Logic and Resolution

First Order Logic (First Order Predicate Calculus)

- There is need to access components of a sentence. Looking at a sentence as in Prepositional logic is too course. We need to deal with objects, their properties and functions of objects that may combine in various ways.

  We say that Prepositional logic has a very limited ontology.

- There is need to have variables. We want to be able to refer to a set of objects as having some property and avoid to indicate that each member of the set has that property.

Objects: cats, student, house

Relation: brother of, sister of

Properties: fat, red, cold

Functions: father of, best friend

Syntax and Semantics of FOL

Sentence: represents a fact (as in prepositional logic).

Constant symbols: A, B, John, Dallas,

refer to real objects.

Predicates: Round, Brother,

refer to relations between objects and functions of objects.
Round (Ball)

Term is a logical expression that refers to an object.

Leftlegof (John).

Atomic sentences: a sentence that is formed from a predicate followed by a list of term

Brother (John, Mary).

Atomic sentences with complex terms:

Married (Father of (John), Mother of (Mary))

Syntax and Semantics of FOL

An atomic sentence is true if the relation defined by predicate is true for the arguments.

Complex sentences—use connectives to link atomic sentences.

Brother (John, Mary) ∧ Brother (John, Richard)

says that John is brother of Mary and Richard.

Older (John, 30) ∨ Younger (John, 30)

Older (John, 30) ⇒ ¬Younger (John, 30)

¬Brother (Sue, John)

Quantifiers

There is need to express properties of sets rather than enumerating the same property for all members of the set.

Universal quantifier ∀

Indicates that a sentence is true for all values of its variables.

All cats are mammals. Is logically equivalent to:

If x is a cat than x is mammal.

$\forall x \; Cat(x) \Rightarrow Mammal(x)$

This replaces a large number of relations for all the cats in the world.

$Cat(Spot) \Rightarrow Mammal(Spot) \wedge$

$Cat(Felix) \Rightarrow Mammal(Felix) \wedge \ldots$

∀ is universal quantifier because the logical expression

$Cat \Rightarrow Mammal$

is true for all cats in the universe.

Quantifiers

Existential quantifier ∃

Indicates that a sentence is true for some values in the domain.

Spot has a sister who is a cat.

$\exists\ x\ Sister(x, Spot) \land Cat(x)$

Is equivalent to

$(Sister\ (Spot, Spot) \land Cat(Spot)) \lor$

$(Sister\ (Felix, Spot) \land Cat(Felix)) \lor$

$(Sister\ (Rebecca, Spot) \land Cat(Rebecca))...$

Note that

$\Rightarrow$ is used with $\forall$

$\land$ is used with $\exists$

Think why.

Quantifiers

Nested quantifiers

$\forall xy$ is equivalent to $\forall x \forall y$

$\forall xy$ Parent(x,y) $\Rightarrow$ child(y,x)

Everybody loves somebody

$\forall x \exists y$ Loves(x,y)

There is someone who is loved by everyone

$\exists y \forall x$ Loves(x,y)

Quantifiers

Relation between ∀ and ∃

∀ and ∃ are related through negation

Nobody likes taxes

$\forall x\, \neg Likes(x, Taxes) \equiv$ (equivalent)

$\neg \exists x\, Likes(x, Taxes)$

Everybody likes Ice cream

$\forall x\, Likes(x, \text{Ice cream}) \equiv$

$\neg \exists x\, \neg Likes(x, \text{Ice cream})$

Quantifiers

Some Equivalent Sentences

$\neg\exists x\ p(x) \equiv \forall x\ \neg p(x)$

$\neg\forall x\ p(x) \equiv \exists x\ \neg p(x)$

$\exists x\ p(x) \equiv \exists y\ p(y)$

$\forall x\ q(x) \equiv \forall y\ q(y)$

$\forall x\ (p(x) \wedge q(x)) \equiv \forall x\ p(x) \wedge \forall y\ q(y)$

$\exists x\ (p(x) \vee q(x)) \equiv \exists x\ p(x) \vee \exists y\ q(y)$

<u>Note:</u>

First order logic allows quantified variables to refer to objects in the discourse and <u>not</u> to predicates

Resolution Theorem Proving

Resolution finds contradictions in a KB of clauses. Resolution is a sound inference rule.

Resolution refutation proves a theorem by negating the statement to be proved and adding this negated goal to the set of axioms. Refutation is complete, meaning that always we get empty clause if exists. Thus resolution refutation is sound and complete.

<u>Steps</u> (for Resolution Refutation Proofs)

1. Put the premises or axioms into clause form. All axioms are expressed as disjunction of literals.
2. Add the negation of what is to be proved, in clause form, to the set of axioms.
3. Resolve these clauses together producing new clauses that follow logically from them.
4. Produce a contradiction by generating the empty clause.

Some useful steps to convert axioms into clause form (conversion to normal form)

- Eliminate implications

  $P \Rightarrow Q \equiv \neg P \vee Q$
- Move negations down to atomic formulas

  $\neg(P \wedge Q) \equiv (\neg P) \vee (\neg Q)$

  $\neg(P \vee Q) \equiv (\neg P) \wedge (\neg Q)$

  $\neg\forall x(P(x)) \equiv \exists x(\neg P(x))$

  $\neg\exists x(P(x)) \equiv \forall x(\neg P(x))$
- Rename variables as necessary so that no two variables are the same.

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

  $(\forall x P(x)) \vee (\exists x\, Q(x)) \equiv (\forall x\, P(x)) \vee (\exists y\, Q(y))$
- Move quantifiers left

  $P \vee (\forall x Q(x)) \equiv \forall x\, [P \vee Q(x)]$
- Eliminate existential quantifiers.

  (called Skolemization)

  a) $\exists x\, P(x)$ replaced by $P(A)$

  where A is a constant.

Some useful steps to convert axioms into clause form (conversion to normal form)

b) $\exists y \forall x \, P(x,y)$ replaced by $\forall x P(x,A)$

c) $\forall x \exists y \, P(x,y)$ replaced by $\forall x P(x,f(x))$

where $f(x)$ is a Skolem function.

Case c) requires more discussion. Variable y cannot be replaced by a constant.

Example: "Everyone has a heart".

$\forall x \, Person(x) \Rightarrow \exists y \, Heart(y) \wedge Has(x,y)$

If we replaced y by a constant A we have

$\forall x \, Person(x) \Rightarrow Heart(A) \wedge Has(x,A)$

This means that everyone has the same heart A, which is not true. Instead we replace y with f(x), meaning that a heart y is linked by x by f(X).

$\forall x \, Person(x) \Rightarrow Heart(f(x)) \wedge Has(x,f(x))$

Skolemization eliminates existentially quantified variables so we can drop universal quantifiers.

Some useful steps to convert axioms into clause form (conversion to normal form)

- Move disjunctions down to literals

  $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$

- Eliminate conjunctions

  $(P(x) \vee Q(x)) \wedge (R(x) \vee S(x))$

  Is decomposed into two clauses

  a) $P(x) \vee Q(x)$

  b) $R(x) \vee S(x)$

  In fact we do not eliminate conjunctions, we only write differently.

- Eliminate universal quantifiers.

  We do not actually eliminate but instead we consider that all variables are universally quantified.

  $\forall x\, P(x)$ becomes $P(x)$

Logical Proof with Modus Ponens

Resolution produces a proof similar to the one produced by modus ponens.

Example:

	Natural Language	Predicate form	Clause form
1.	All dogs are animals.:	$\forall x(dog(x) \Rightarrow animal(x))$	$\neg dog(x) \vee animal(x)$
2.	Fido is a dog.:	dog(fido)	dog(fido)
3.	All animals will die.	$\forall y(animal(y) \Rightarrow (die)y))$	$\neg animal(y) \vee die(y)$

We wish to prove that “Fido will die”: die(fido)

Proof with modus ponens: (use predictae form)

1 and 2: modus ponens & unify fido with x

$\rightarrow$ animal(fido)

3 and animal(fido) via modus ponens and unify

{fido/y} $\rightarrow$ die(fido)

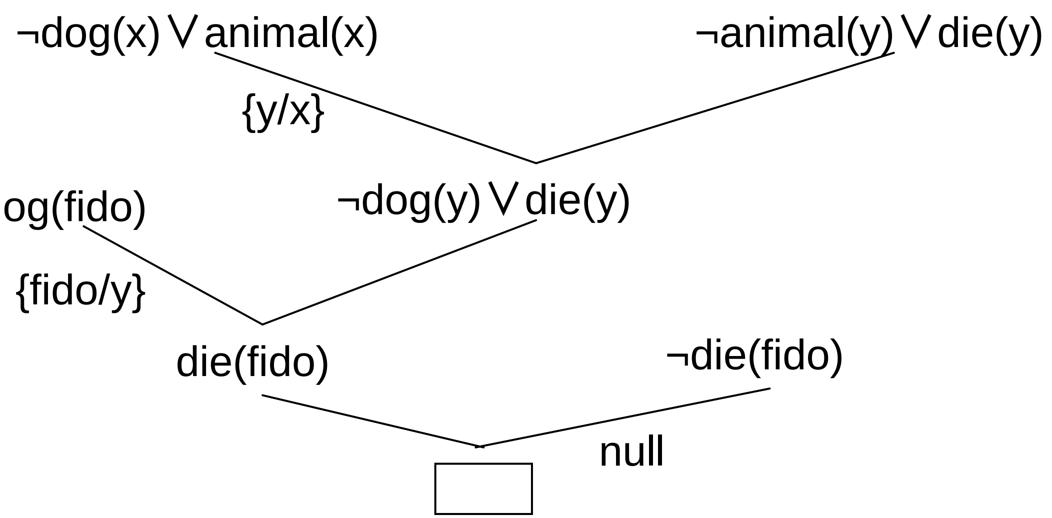
We proved that “fido will die” using modus ponens.

Proof with resolution

Step 1. Transform axioms into clause form.

Step 2. Add negation ¬die(fido)

Step 3. Resolve these clauses together



Note: From substitutions we determine more information:

fido/y meaning fido is an animal

The process of reading back the unifications is called answer extraction.

Refutation

- Definition - one complete inference procedure using resolution!

  → also known as - proof by contradiction

  - reductio ad absurdum

- The idea: to prove P, assume P is false (i.e. add $\neg$P to KB) and prove by contradiction

  $(KB \wedge \neg P \Rightarrow False) \Leftrightarrow (KB \Rightarrow P)$

Example Proof

- Refutation on a more complex example:

- In English:

  Everyone who loves all animals is loved by someone.
  Anyone who kills an animal is loved by no one.
  Jack loves all animals.
  Either Jack or Curiosity killed the cat who is named Tuna.
  Did Curiosity kill the cat?

Translation in FOL

A. $\forall x\ [\forall y\ \text{Animal}(y) \Rightarrow \text{Loves}(x, y)] \Rightarrow$
$[\exists y\ \text{Loves}(y, x)]$

B. $\forall x\ [\exists y\ \text{Animal}(y) \wedge \text{Kills}(x,y)] \Rightarrow$
$[\forall z\ \neg\text{Loves}(z,x)]$

C. $\forall x\ \text{Animal}(x) \Rightarrow \text{Loves}(\text{Jack}, x)$

D. Kills(Jack, Tuna) $\vee$ Kills(Curiosity, Tuna)

E. Cat(Tuna)

F. $\forall x\ \text{Cat}(x) \Rightarrow \text{Animal}(x)$

$\neg$G. $\neg$Kills(Curiosity, Tuna)

Conversion to CNF

A. $\forall x\, [\forall y\, \text{Animal}(y) \Rightarrow \text{Loves}(x, y)] \Rightarrow [\exists y\, \text{Loves}(y, x)]$

B. $\forall x\, [\exists y\, \text{Animal}(y) \wedge \text{Kills}(x,y)] \Rightarrow [\forall z\, \neg\text{Loves}(z,x)]$

C. $\forall x\, \text{Animal}(x) \Rightarrow \text{Loves}(\text{Jack}, x)$

D. $\text{Kills}(\text{Jack}, \text{Tuna}) \vee \text{Kills}(\text{Curiosity}, \text{Tuna})$

E. $\text{Cat}(\text{Tuna})$

F. $\forall x\, \text{Cat}(x) \Rightarrow \text{Animal}(x)$

$\neg$G. $\neg\text{Kills}(\text{Curiosity}, \text{Tuna})$

A1. $\text{Animal}(F(x)) \vee \text{Loves}(G(x), x)$

A2. $\neg\text{Loves}(x, F(x)) \vee \text{Loves}(G(x), x)$

B. $\neg\text{Animal}(y) \vee \neg\text{Kills}(x,y) \vee \neg\text{Loves}(z,x)]$

C. $\neg\text{Animal}(x) \vee \text{Loves}(\text{Jack}, x)$

D. $\text{Kills}(\text{Jack}, \text{Tuna}) \vee \text{Kills}(\text{Curiosity}, \text{Tuna})$

E. $\text{Cat}(\text{Tuna})$

F. $\neg\text{Cat}(x) \vee \text{Animal}(x)$

$\neg$G. $\neg\text{Kills}(\text{Curiosity}, \text{Tuna})$

Proof

A1.Animal(F(x)) $\vee$ Loves(G(x), x)

A2.$\neg$Loves(x, F(x)) $\vee$ Loves(G(x), x)

B. $\neg$Animal(y) $\vee$ $\neg$Kills(x,y) $\vee$ $\neg$Loves(z,x)]

C. $\neg$Animal(x) $\vee$ Loves(Jack, x)

D. Kills(Jack, Tuna) $\vee$ Kills(Curiosity, Tuna)

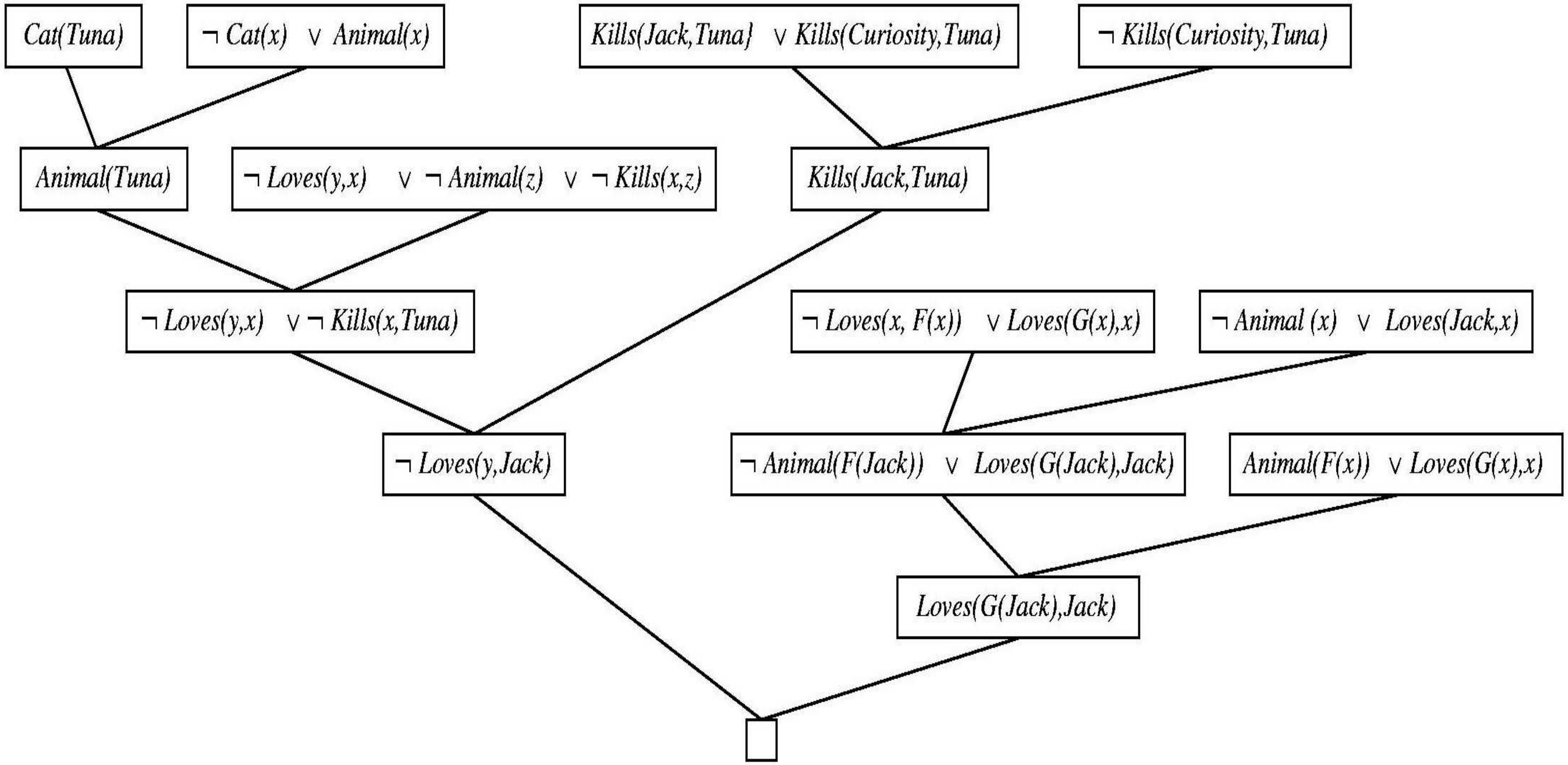
E. Cat(Tuna)

F. $\neg$Cat(x) $\vee$ Animal(x)

$\neg$G.$\neg$Kills(Curiosity, Tuna)

R1: E & F $\Rightarrow$ Animal(Tuna) {Tuna/x}
R2: R1&B $\Rightarrow$ $\neg$Loves(y, x) $\vee$ $\neg$Kills(x, Tuna) {Tuna/z}
R3: D & $\neg$G $\Rightarrow$ Kills(Jack, Tuna)
R4: R3 & R2 $\Rightarrow$ $\neg$Loves(y, Jack)
R5: A2 & C $\Rightarrow$ $\neg$Animal(F(Jack)) $\vee$ Loves(G(Jack), Jack) {Jack/x, F(x) = $\perp$}
R6: R5 & A1 $\Rightarrow$ Loves(G(Jack), Jack) {Jack/x}
R7: R4 & R5 $\Rightarrow$ FALSE {G(Jack)/y}

Resolution Proof



Example 2

In English:

The custom officials searched everyone who entered the country and was not a VIP. Some of the drug pushers entered this country and they were only searched by drug pushers. No drug pusher was a VIP.

At least one of the custom officials was a drug pusher.

Translation in FOL

"The custom officials searched everyone who entered the country and was not a VIP"

$$\forall x\, \exists y\, (entered\_country(x) \wedge \neg VIP(x)) \Rightarrow (official(y) \wedge searched(y,x))$$

with VIP(x) - x is a VIP
official(y) - y is a custom official
searched(x,y) x has searched y

"Some of the drug pushers entered this country and they were only searched by drug pushers"

$$\exists x\, \forall y\, [entered\_country(x) \wedge d\_pusher(x)] \wedge [searched(y,x) \Rightarrow d\_pusher(y)]$$

More translations!

"No drug pusher was a VIP"

$\forall x\ [d\_pusher(x) \Rightarrow \neg\ VIP(x)]$

Goal: "At least one of the custom officials is a drug pusher"

$\exists x\ (official(x) \wedge d\_pusher(x))$

More details

Let us look again at the translation

"The custom officials searched everyone who entered the country and was not a VIP"

$$\forall x \, \exists y \, (entered\_country(x) \land \neg VIP(x)) \Rightarrow (official(y) \land searched(y,x))$$

Eliminate "$\Rightarrow$"

$$\forall x \, \exists y \, (\neg(entered\_country(x) \land \neg VIP(x)) \lor (official(y) \land searched(y,x))$$

$$\Downarrow$$

$$\forall x \, (\neg entered\_country(x) \lor VIP(x) \lor (official(f(x)) \land searched(f(x),x))$$

Skolem function

The axioms

1. $\neg$entered_country(x) $\vee$ VIP(x) $\vee$ official(f(x))
2. $\neg$entered_country(x) $\vee$ VIP(x) $\vee$ searched(f(x),x)

$\exists$x $\forall$y [entered_country(x) $\wedge$ d_pusher(x)] $\wedge$ [searched(y,x) $\Rightarrow$ d_pusher(y)]
eliminate "$\Rightarrow$"
$\exists$x $\forall$y [entered_country(x) $\wedge$ d_pusher(x)] $\wedge$ [$\neg$ searched(y,x) $\vee$ d_pusher(y)]
Then what?
x=a (a constant) + standardize variables y$\Rightarrow$x

3. entered_country(a)
4. d_pusher(a)
5. $\neg$searched(x,a) $\vee$ d_pusher(x)

Last axioms

From statement $\forall x\ (d\_pusher(x) \Rightarrow \neg VIP(x))$

6. $\neg d\_pusher(x) \vee \neg VIP(x)$

The goal: $\underline{\exists x\ (official(x) \wedge d\_pusher(x))}$

Negated: $\forall x\ (\neg official(x) \vee \neg d\_pusher(x))$

7. $\neg official(x) \vee \neg d\_pusher(x)$

The refutation

2: ¬entered_country(x) ∨ VIP(x) ∨ searched(f(x),x)

6. ¬d_pusher(x) ∨ ¬VIP(x)

8: ¬d_pusher(x) ∨ ¬entered_country(x) ∨ searched(f(x),x)

4. d_pusher(a)

a/x

9: ¬entered_country(a) ∨ searched(f(a),a)

3. entered_country(a)

10: searched(f(a),a)

The refutation

1: ¬entered_country(x) ∨ VIP(x) ∨ official(f(x))

6. ¬d_pusher(x) ∨ ¬VIP(x)

11: ¬d_pusher(x) ∨ ¬entered_country(x) ∨ official(f(x))

4. d_pusher(a)

a/x

12: ¬entered_country(a) ∨ official(f(a))

3. entered_country(a)

10: official(f(a))

The refutation

7: ¬official(x) ∨ ¬d_pusher(x)

4. d_pusher(a)

x/a

14: ¬official(f(a))

13. official(f(a))

NIL

The statement is valid.

Example 3

John likes all kinds of food.
Apple is food.
Chicken is food.
Anything anyone eats isn't killed by is food.
Bill eats peanuts and is alive.
Sue eats anything Bill eats.

Show that *John likes peanuts*.
What food does Sue eat?

Translation in FOL

"*John likes all kinds of food*."

$\forall x\ food(x) \Rightarrow likes(John, x)$

"Apple is food." ⇒ food(Apple)

"Chicken is food." ⇒ food(Chicken)

"Anything anyone eats isn't killed by is food."

$\forall x\ \forall y\ (eats(x,y) \wedge \neg\ killed(x,y)) \Rightarrow food(y)$

More translations

"*Bill eats peanuts and is alive*."

I assume $alive(x) \equiv \forall y\ (\neg killed(x,y))$ is too general for the context of the sentence

I prefer the conversion:

$eats(Bill,Peanuts) \wedge \neg\ killed(Bill, Peanuts)$

I could have used:

$eats(Bill,Peanuts) \wedge \forall x\ \neg killed(Bill, x)$

"*Sue eats anything Bill eats*."

$\forall x\ eats(Bill,x) \Rightarrow eats(Sue,x)$

Refute "John likes peanuts"

Transformation to CNF:

1. $\neg$food(x) $\vee$ likes(John, x)
2. food(Apple)
3. food(Chicken)
4. $\neg$eats(x,y) $\vee$ killed(x,y)) $\vee$ food(y)
5. eats(Bill, Peanuts)
6. $\neg$killed(Bill, Peanuts)
7. $\neg$eats(Bill,x) $\vee$ eats(Sue,x)

+Goal: 8. $\neg$likes(John,Peanuts)

The proof

9. $\neg$food(Peanuts) from 1&8 Peanuts/x

10. killed(Bill, Peanuts) $\vee$ food(Peanuts)
from 4&5 Bill/x
Peanuts/y

11. food(Peanuts) 6&10

12. NIL 9&11

$\Rightarrow$ This resolution theorem proving showed that the clause "*John likes peanuts*" is valid

“What food does Sue eat?”

We do again theorem proving, changing the goal clause to:

$\neg$likes(John, Peanuts) $\vee$ $\neg$eats(Sue,z)

Axioms

1. $\neg food(x) \vee likes(John, x)$
2. $food(Apple)$
3. $food(Chicken)$
4. $\neg eats(x,y) \vee killed(x,y) \vee food(y)$
5. $eats(Bill, Peanuts)$
6. $\neg killed(Bill, x)$
7. $\neg eats(Bill,x) \vee eats(Sue,x)$
8. $\neg likes(John,Peanuts) \vee \neg eats(Sue,z)$

Proof

9. $\neg$food(Peanuts) $\vee$ $\neg$eats(Sue,z)
   from 1&8   Peanuts/x
10. eats(Sue, Peanuts)   from 5&7   Peanuts/x
11. $\neg$food(Peanuts)   from 9&10   z/Peanuts
12. killed(Bill, Peanuts) $\vee$ food(Peanuts)
    from 4&5   Billy/x
    Peanuts/y
13. food(Peanuts)   from 6&12   x/Peanuts
14. NIL   from 11&13

From the substitution in 11, we see that z=Peanuts $\Rightarrow$ Sue eats Peanuts

Another proof

9. $\neg$food(Peanuts) $\vee$ $\neg$eats(Sue,z)		
	from 1&8	Peanuts/x
10. killed(Bill, Peanuts) $\vee$ food(Peanuts)		
	from 4&5	Billy/x Peanuts/y
11. food(Peanuts)	from 6&10	x/Peanuts
12. $\neg$eats(Sue, z)	from 9&11	
13. eats(Sue, Peanuts)	from 5&7	Peanuts/x
14. NIL	from 12&13	Peanuts/z

$\downarrow$

eats(Sue, Peanuts)

Answering questions

"*What food does Sue eat*?"

Set the goal to:

8′. $\forall x$ (eats(Sue,x) $\vee$ $\neg$eats(Sue,x))

New proof

9. $\neg$eats(Bill,Peanuts) $\vee$ eats(Sue, Peanuts)
from 7&8   Peanuts/x

$\boxed{\text{10. eats(Sue, Peanuts)}}$ from 5&9

Lessons learned

- 4 examples
- Several kinds of proofs
- Formal method of answering questions
- Search and unification