# Knowledge Representation and Reasoning in Artificial Intelligence – First Order Predicate Logic

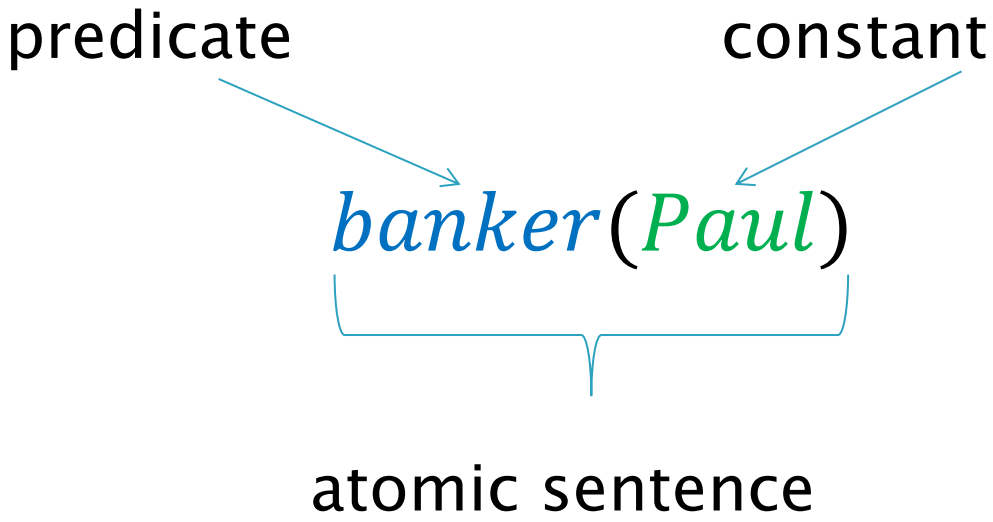
Panagiotis Stamatopoulos
Department of Informatics and Telecommunications
National and Kapodistrian University of Athens

# To discuss…

- First order predicate logic is a knowledge representation language
- Description of the world via well-formed formulas
- Transformation of formulas to conjunctive normal form
- Inference via a deduction mechanism (resolution)
- Constructing proofs by refutation
- A subset of first order predicate logic has been implemented as a programming language (Prolog)

# A simple example

*Paul is a banker*

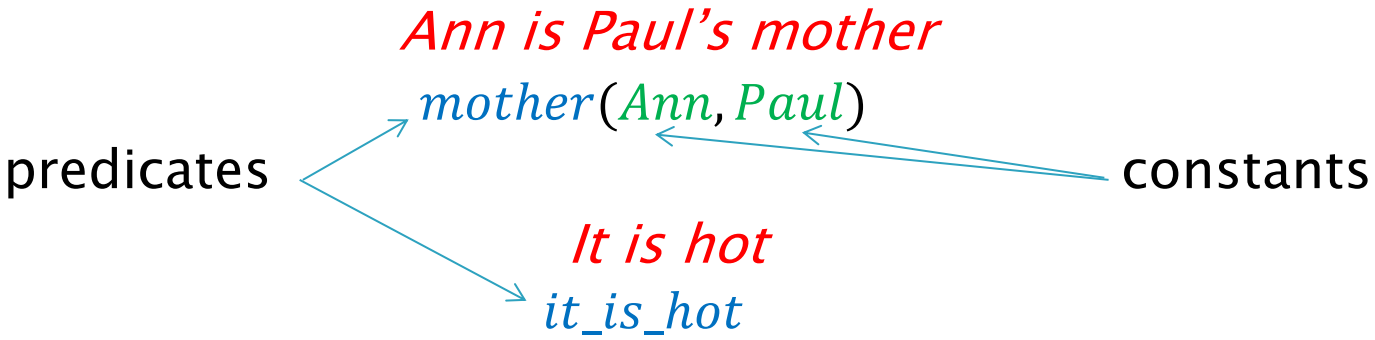


# The basic components

- **Constants** represent elementary entities, objects, abstract concepts, etc. of the world, e.g. $Paul$, $this\_table$, $happiness$, $2562$
- Each **predicate** has a **degree**, that is the number of its arguments – the degree of the predicate $banker$ is1
- A predicate of degree1 represents a property of its argument, e.g. $banker(Paul)$, as an **atomic sentence**
- Atomic sentences are the simplest form of **well-formed formulas** (sentences)
- A sentence has a **truth value**, that is **true** (T) or **false** (F), given a specific state of the world

# Predicate degrees

- Predicates with degree 2 or greater are used to describe relations among their arguments
- Predicates with degree 0 represent a simple statement of the world
- Examples:




- A specialization of first order logic, that is **propositional logic**, involves predicates of degree 0 only

# Sentences

- More complex sentences (not atomic) are formed via **logical connectives**, which, in decreasing precedence, are:

| | |
|---|---|
| ¬ | negation |
| ∧ | conjunction |
| ∨ | disjunction |
| ⇒ | implication |
| ⇔ | equivalence |

- Examples:

$$\neg banker(George)$$
$$mother(Ann, Paul) \wedge \neg father(Paul, George)$$
$$man(Socrates) \Rightarrow mortal(Socrates)$$
$$happy(Mary) \Leftrightarrow healthy(Mary) \wedge \neg poor(Mary)$$

What is the meaning of the above sentences?

# Truth values

- The truth values of non-atomic sentences are defined at the table below, according to the involved logical connectives and the truth values of the sentences they comprise

| $p$ | $q$ | $\neg p$ | $p \wedge q$ | $p \vee q$ | $p \Rightarrow q$ | $p \Leftrightarrow q$ |
|---|---|---|---|---|---|---|
| T | T | F | T | T | T | T |
| T | F | F | F | T | F | F |
| F | T | T | F | T | T | F |
| F | F | T | F | F | T | T |

- Note the truth value of implication – it is false only when the antecedent is true and the consequent is false

# Functions – Terms

- **Functions** are used to construct **terms**, which represent compound entities, concepts, etc.
- Examples of terms:

$$father\_of(Paul)$$
$$next(0)$$
$$date(13, nov, 2018)$$
$$number\_of\_legs(my\_table)$$
$$mother\_of(father\_of(Ann))$$

- Functions have degrees as well, as is the case with predicates
- Constants are special cases of functions with degree 0
- Be careful: Functions **ARE NOT** predicates and terms **ARE NOT** atomic sentences

# Variables – Quantifiers

- In first order predicate logic, we use **variables** to represent unknown entities, objects, etc. of the world
- Variables are introduced in sentences through quantifiers:

  Universal quantifier: ∀ (for all)

  Existential quantifier: ∃ (exists)
- Examples:

  Everything is mortal

  $\forall x\ [mortal(x)]$

  There is a man

  $\exists x\ [man(x)]$

# Universal quantifier

- It is very common to represent a general rule of the world by an implication, where we quantify universally all variables of the sentence
- Examples:

Men are mortal

$$\forall x\ [man(x) \Rightarrow mortal(x)]$$

Bankers are rich

$$\forall x\ [banker(x) \Rightarrow rich(x)]$$

The parents love their children

$$\forall x\ \forall y\ [parent(x, y) \Rightarrow loves(x, y)]$$

# Existential quantifier

- It is applied, more or less, to a conjunction, in order to represent a statement that an entity, an object, etc. exists, where we quantify existentially all variables of the sentence
- Examples:

Some researchers are rich

$$\exists x\ [researcher(x) \wedge rich(x)]$$

Some countries border with Greece and with Romania

$$\exists x\ [country(x) \wedge borders(x, Greece) \wedge borders(x, Romania)]$$

Every man has a mother who is older than him

$$\forall x\ [man(x) \Rightarrow \exists y\ [mother(y, x) \wedge older(y, x)]]$$

# Equality

- In first order predicate logic, we often need the equality predicate $=$, or the inequality predicate $\neq$, which are usually used with an infix notation
- Example:

Every man has exactly one father

$$\forall x \left[ man(x) \Rightarrow \exists y \left[ father(y, x) \wedge \forall z \left[ father(z, x) \Rightarrow z = y \right] \right] \right]$$

There are at least two elephants

$$\exists x \left[ elephant(x) \wedge \exists y \left[ elephant(y) \wedge y \neq x \right] \right]$$

There are exactly two cats

$$\exists x \left[ cat(x) \wedge \exists y \left[ cat(y) \wedge y \neq x \wedge \forall z \left[ cat(z) \Rightarrow z = x \vee z = y \right] \right] \right]$$

# A more complex example

Paul is a banker

$banker(Paul)$

Bankers are rich

$\forall x\ [banker(x) \Rightarrow rich(x)]$

Everybody has a mother

$\forall x\ \exists y\ [mother(y, x)]$

The mothers of the rich people are happy

$\forall x\ \forall y\ [mother(y, x) \wedge rich(x) \Rightarrow happy(y)]$

Happy mothers love their mothers

$\forall x\ [happy(x) \wedge \exists y\ [mother(x, y)] \Rightarrow \forall z\ [mother(z, x) \Rightarrow loves(x, z)]]$

# Transformation to conjunctive normal form

- Atomic sentences and negations of atomic sentences are **literals**
- A sentence in conjunctive normal form (CNF) is a conjunction of disjunctions of literals, where all variables are universally quantified at the beginning of the sentence, that is why the quantification is omitted
- CNF is needed in order to be able to apply a deduction procedure, called resolution, for drawing conclusions
- What we need to transform a sentence to CNF:

| | | |
|---|---|---|
| $\neg(\neg p) \equiv p$ | $\neg(p \wedge q) \equiv \neg p \vee \neg q$ | $\neg(p \vee q) \equiv \neg p \wedge \neg q$ |
| $p \Rightarrow q \equiv \neg p \vee q$ | $\neg\exists x\,[p(x)] \equiv \forall x\,[\neg p(x)]$ | $\neg\forall x\,[p(x)] \equiv \exists x\,[\neg p(x)]$ |
| $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ | | $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$ |

# Transformation examples

- $banker(Paul)$ is already in CNF
- $\forall x\ [banker(x) \Rightarrow rich(x)]$ is transformed to CNF:

$$\forall x\ [\neg banker(x) \vee rich(x)]$$
$$\neg banker(x) \vee rich(x)$$

- $\forall x\ \exists y\ [mother(y, x)]$ needs Skolemization to get rid of the existential quantifier, so it is transformed to:

$$\forall x\ [mother(mother\_of\,(x), x)]$$
$$mother(mother\_of\,(u), u)$$

!!! Note the difference between predicate $mother$ and function $mother\_of$ !!!

- $\forall x\ \forall y\ [mother(y, x) \wedge rich(x) \Rightarrow happy(y)]$ is transformed to:

$$\forall x\ \forall y\ [\neg(mother(y, x) \wedge rich(x)) \vee happy(y)]$$
$$\forall x\ \forall y\ [\neg mother(y, x) \vee \neg rich(x) \vee happy(y)]$$
$$\neg mother(y, w) \vee \neg rich(w) \vee happy(y)$$

# Transformation examples (cont.)

- And the last sentence:

$$\forall x \left[ happy(x) \wedge \exists y \left[ mother(x,y) \right] \Rightarrow \forall z \left[ mother(z,x) \Rightarrow loves(x,z) \right] \right]$$
$$\forall x \left[ \neg (happy(x) \wedge \exists y \left[ mother(x,y) \right]) \vee \forall z \left[ mother(z,x) \Rightarrow loves(x,z) \right] \right]$$
$$\forall x \left[ \neg (happy(x) \wedge \exists y \left[ mother(x,y) \right]) \vee \forall z \left[ \neg mother(z,x) \vee loves(x,z) \right] \right]$$
$$\forall x \left[ \neg happy(x) \vee \neg \exists y \left[ mother(x,y) \right] \vee \forall z \left[ \neg mother(z,x) \vee loves(x,z) \right] \right]$$
$$\forall x \left[ \neg happy(x) \vee \forall y \left[ \neg mother(x,y) \right] \vee \forall z \left[ \neg mother(z,x) \vee loves(x,z) \right] \right]$$
$$\forall x \forall y \forall z \left[ \neg happy(x) \vee \neg mother(x,y) \vee \neg mother(z,x) \vee loves(x,z) \right]$$
$$\neg happy(t) \vee \neg mother(t,v) \vee \neg mother(z,t) \vee loves(t,z)$$

- Finally, our knowledge in CNF is:

$$banker(Paul)$$
$$\neg banker(x) \vee rich(x)$$
$$mother(mother\_of(u), u)$$
$$\neg mother(y,w) \vee \neg rich(w) \vee happy(y)$$
$$\neg happy(t) \vee \neg mother(t,v) \vee \neg mother(z,t) \vee loves(t,z)$$

# Some remarks

- What is the difference between the following?

$\forall x\ \exists y\ [mother(y, x)]$ and $\exists y\ \forall x\ [mother(y, x)]$

After Skolemization they are transformed to, respectively:

$\forall x\ [mother(mother\_of(x), x)]$ and $\forall x\ [mother(universal\_mother, x)]$

That is:

*«Everybody has a mother»* (function of him/herself)

*«All have the same mother»* (constant, the same for all)

- When the result of a transformation to CNF is not a literal or a disjunction of literals, but a conjunction of disjunctions, we create one sentence for each conjunct, e.g.

$$(\neg A \vee \neg B \vee C) \wedge (A \vee B) \wedge (B \vee \neg C \vee \neg D)$$

$$\left\{ \begin{array}{c} \neg A \vee \neg B \vee C \\ A \vee B \\ B \vee \neg C \vee \neg D \end{array} \right\}$$

# Resolution and unification

- Assume we have the sentences:

$$A_1 \vee B \qquad \text{and} \qquad \neg A_2 \vee C$$

- If the atomic sentences $A_1$ and $A_2$ are **unifiable**, that is they might be the same after appropriate variable instantiations (let $\sigma$ be the set of instantiations), then, according to the **resolution** inference rule, the following sentence is deduced:

$$(B \vee C)\sigma$$

- The above notation means that the set of instantiations $\sigma$ is applied to the sentence $B \vee C$
- Example: From the sentences

$$\neg p(x) \vee q(x) \qquad \text{and} \qquad p(a) \vee r(b)$$

the sentence $q(a) \vee r(b)$ is deduced with $\sigma = \{a/x\}$

- The notation $a/x$ means that the variable $x$ is instantiated to the value $a$

# Deduction of new knowledge – Inference

- Given knowledge in CNF:

$$banker(Paul)$$
$$\neg banker(x) \vee rich(x)$$
$$mother(mother\_of(u), u)$$
$$\neg mother(y, w) \vee \neg rich(w) \vee happy(y)$$
$$\neg happy(t) \vee \neg mother(t, v) \vee \neg mother(z, t) \vee loves(t, z)$$

- Question:

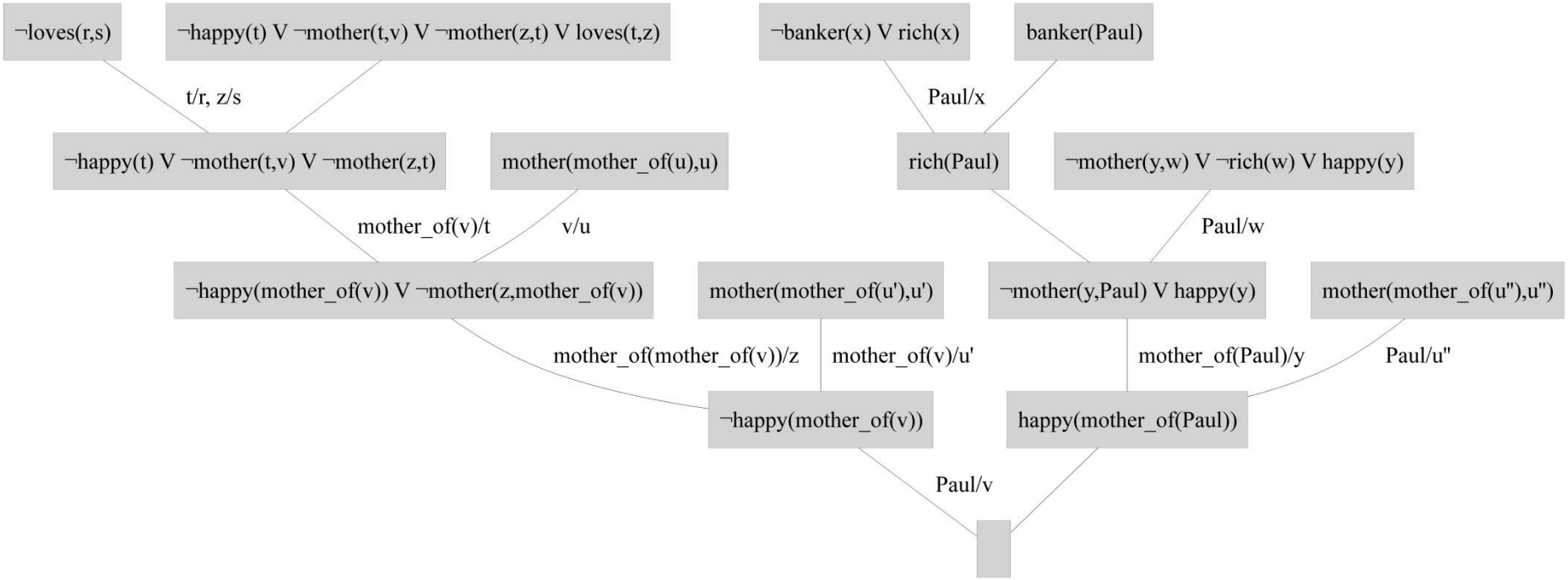
Is there anybody that loves somebody?

$$\exists r \exists s\ [loves(r, s)]$$

- Refutation of the sentence to be proved in CNF:

$$\neg \exists r \exists s\ [loves(r, s)]$$
$$\forall r\ [\neg \exists s\ [loves(r, s)]]$$
$$\forall r \forall s\ [\neg loves(r, s)]$$
$$\neg loves(r, s)$$

# Resolution via the deduction of contradiction




- By combining the instantiations $t/r$, $mother\_of(v)/t$, $Paul/v$ και $z/s$, $mother\_of(mother\_of(v))/z$, $Paul/v$, we have $mother\_of(Paul)/r$ and $mother\_of(mother\_of(Paul))/s$, that is *«Paul's mother loves her mother»*

# Some more examples

- All mushrooms are poisonous

$$\forall x\ [mushroom(x) \Rightarrow poisonous(x)]$$

- At least one mushroom is poisonous

$$\exists x\ [mushroom(x) \wedge poisonous(x)]$$

- All mushrooms except one are poisonous

$$\exists x\ \Big[mushroom(x) \wedge \neg poisonous(x) \wedge \forall y\ [mushroom(y) \wedge x \neq y \Rightarrow poisonous(y)]\Big]$$

- There exists exactly one mushroom

$$\exists x\ [mushroom(x) \wedge \forall y\ [mushroom(y) \Rightarrow x = y]]$$

- There exist at least two mushrooms

$$\exists x\ [mushroom(x) \wedge \exists y\ [mushroom(y) \wedge x \neq y]]$$

- There exist exactly two poisonous mushrooms

$$\exists x\ [mushroom(x) \wedge poisonous(x) \wedge \exists y\ [mushroom(y) \wedge poisonous(y) \wedge x \neq y \wedge \forall z\ [mushroom(z) \wedge poisonous(z) \Rightarrow z = x \vee z = y]]]$$

# Some more examples (cont.)

- A man loves every woman that hates Jim

$$\exists x\,[man(x) \wedge \forall y\,[woman(y) \wedge hates(y, \mathrm{Jim}) \Rightarrow loves(x, y)]]$$

- Every man loves a woman that hates Jim

$$\forall x\,[man(x) \Rightarrow \exists y\,[woman(y) \wedge hates(y, \mathrm{Jim}) \wedge loves(x, y)]]$$

- A set is empty if and only if it contains no elements

$$\forall s\,[empty(s) \Leftrightarrow \neg\exists x\,[in(x, s)]]$$

- An element belongs to the difference of two sets if and only if it belongs to the first set and it does not belong to the second set

$$\forall x\,\forall s_1 \forall s_2\,[in\big(x, diff(s_1, s_2)\big) \Leftrightarrow in(x, s_1) \wedge \neg in(x, s_2)]$$

# An example of resolution

Sam, Clyde and Oscar are elephants. Sam is pink. Clyde is grey and likes Oscar. Oscar is pink or grey and likes Sam. Prove that a grey elephant likes a pink elephant.

$$elephant(Sam)$$
$$elephant(Clyde)$$
$$elephant(Oscar)$$
$$pink(Sam)$$
$$grey(Clyde) \wedge likes(Clyde, Oscar)$$
$$[pink(Oscar) \vee grey(Oscar)] \wedge likes(Oscar, Sam)$$

To prove:

$$\exists x\ [elephant(x) \wedge grey(x) \wedge \exists y\ [elephant(y) \wedge pink(y) \wedge likes(x, y)]]$$

# An example of resolution (cont.)

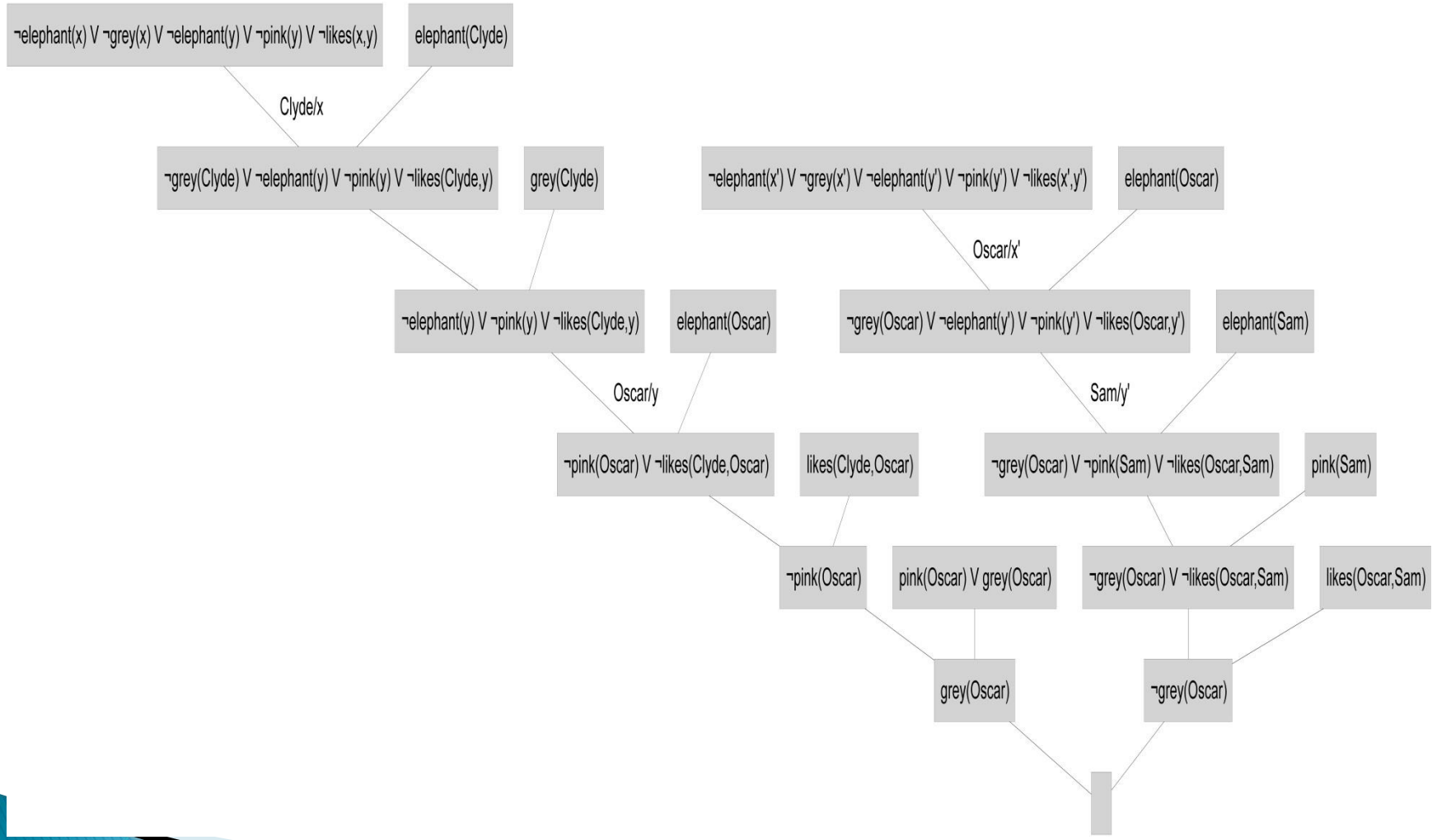
Conjunctive normal form (CNF):

$$elephant(Sam)$$
$$elephant(Clyde)$$
$$elephant(Oscar)$$
$$pink(Sam)$$
$$grey(Clyde)$$
$$likes(Clyde, Oscar)$$
$$pink(Oscar) \vee grey(Oscar)$$
$$likes(Oscar, Sam)$$

Refutation of the query:

$$\neg elephant(x) \vee \neg grey(x) \vee \neg elephant(y) \vee \neg pink(y) \vee \neg likes(x, y)$$

# An example of resolution (cont.)



# Another example

A murder was committed in town. Victor was the victim. The police arrested three suspects, Abbott, Babbitt and Cabot. Abbott claimed that Babbitt and Victor were friends and that Cabot hated Victor. Babbitt denied that he was in town at the day of the murder and said that he didn't know Victor. Cabot testified that he saw Abbott and Babbitt with Victor just before the crime. The police is confident that exactly one of the three, Abbott, Babbitt and Cabot, is guilty and they assume that the other two are telling the truth. Who is the murderer?

# Representation in first order logic

$$innocent(A) \rightarrow friend(B,V)$$
$$innocent(A) \rightarrow hates(C,V)$$

$$innocent(B) \rightarrow \neg in\_town(B)$$
$$innocent(B) \rightarrow \neg knows(B,V)$$

$$innocent(C) \rightarrow with(A,V)$$
$$innocent(C) \rightarrow with(B,V)$$

$$\forall x\ [with(x,V) \rightarrow in\_town(x)]$$
$$\forall x\ \forall y\ [friend(x,y) \rightarrow knows(x,y)]$$
$$\forall x\ \forall y\ [hates(x,y) \rightarrow knows(x,y)]$$

$$\neg innocent(A)\ \vee \neg innocent(B) \vee \neg innocent(C)$$
$$(innocent(A) \wedge\ innocent(B)) \vee (innocent(A) \wedge innocent(C)) \vee (innocent(B) \wedge innocent(C))$$

To prove:

$$\exists x\ [\neg innocent(x)]$$

# Conjunctive normal form (CNF)

$$\neg innocent(A) \vee friend(B,V)$$
$$\neg\ innocent(A) \vee hates(C,V)$$

$$\neg\ innocent(B) \vee \neg in\_town(B)$$
$$\neg innocent(B) \vee \neg knows(B,V)$$

$$\neg innocent(C) \vee with(A,V)$$
$$\neg\ innocent(C) \vee with(B,V)$$

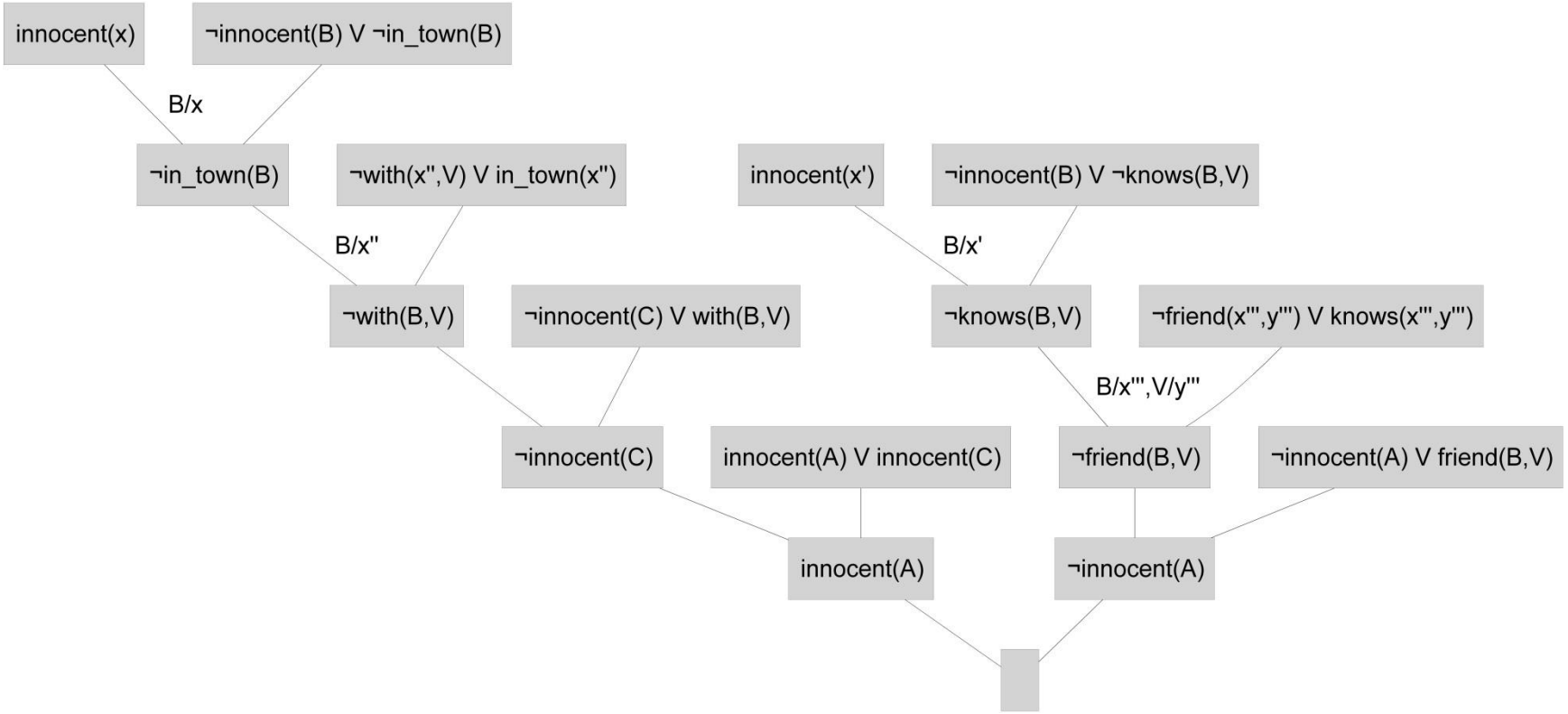
$$\neg\ with(x,V) \vee in\_town(x)$$
$$\neg friend(x,y) \vee knows(x,y)$$
$$\neg\ hates(x,y) \vee knows(x,y)$$

$$\neg innocent(A)\ \vee \neg innocent(B) \vee \neg innocent(C)$$
$$innocent(A) \vee\ innocent(B)$$
$$innocent(A) \vee\ innocent(C)$$
$$innocent(B) \vee\ innocent(C)$$

Refutation of the query:

$$innocent(x)$$

# Resolution




Thus, Babbitt is the murderer!