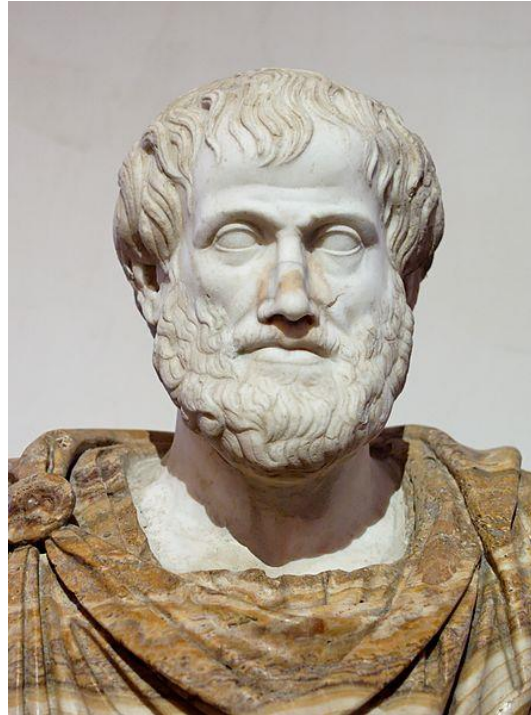


AI: Representation and Problem Solving

First-Order Logic



Instructors: Pat Virtue & Fei Fang

Slide credits: CMU AI, <http://aima.eecs.berkeley.edu>

Propositional Logic vs First-Order Logic

Rules of chess:

- 100,000 pages in propositional logic
- 1 page in first-order logic

Rules of pacman:

- $\forall x,y,t \text{ At}(x,y,t) \Leftrightarrow [\text{At}(x,y,t-1) \wedge \neg \exists u,v \text{ Reachable}(x,y,u,v,\text{Action}(t-1))] \vee [\exists u,v \text{ At}(u,v,t-1) \wedge \text{Reachable}(x,y,u,v,\text{Action}(t-1))]$

First-Order Logic (First-Order Predicate Calculus)

Whereas propositional logic assumes world contains **facts**,
first-order logic (like natural language) assumes the world contains

- **Objects:** people, houses, numbers, theories, Ronald McDonald, colors, baseball games, wars, centuries, ...
- **Relations:** red, round, bogus, prime, multistoried ..., mother of, bigger than, inside, part of, has color, occurred after, owns, ...
- **Functions:** mother of, best friend, third inning of, one more than, end of, ...

Logics in General

Language	What exists in the world	What an agent believes about facts
Propositional logic	Facts	true / false / unknown
First-order logic	facts, objects, relations	true / false / unknown
Probability theory	facts	degree of belief
Fuzzy logic	facts + degree of truth	known interval value

Syntax of FOL

Basic Elements

Constants *KingJohn, 2, CMU, ...*

Predicates *Brother, >, ...*

Functions *Sqrt, LeftLegOf, ...*

Variables *x, y, a, b, ...*

Connectives $\wedge \vee \neg \Rightarrow \Leftrightarrow$

Equality $=$

Quantifiers $\exists \forall$

Syntax of FOL

Atomic sentence = $\text{predicate}(\text{term}_1, \dots, \text{term}_n)$
or $\text{term}_1 = \text{term}_2$

Term = $\text{function}(\text{term}_1, \dots, \text{term}_n)$
or *constant*
or *variable*

Examples

$\text{Brother}(\text{KingJohn}, \text{RichardTheLionheart})$

$> (\text{Length}(\text{LeftLegOf}(\text{Richard})), \text{Length}(\text{LeftLegOf}(\text{KingJohn})))$

Syntax of FOL

Complex sentences are made from atomic sentences using connectives

$$\neg S, \quad S_1 \wedge S_2, \quad S_1 \vee S_2, \quad S_1 \Rightarrow S_2, \quad S_1 \Leftrightarrow S_2$$

Examples

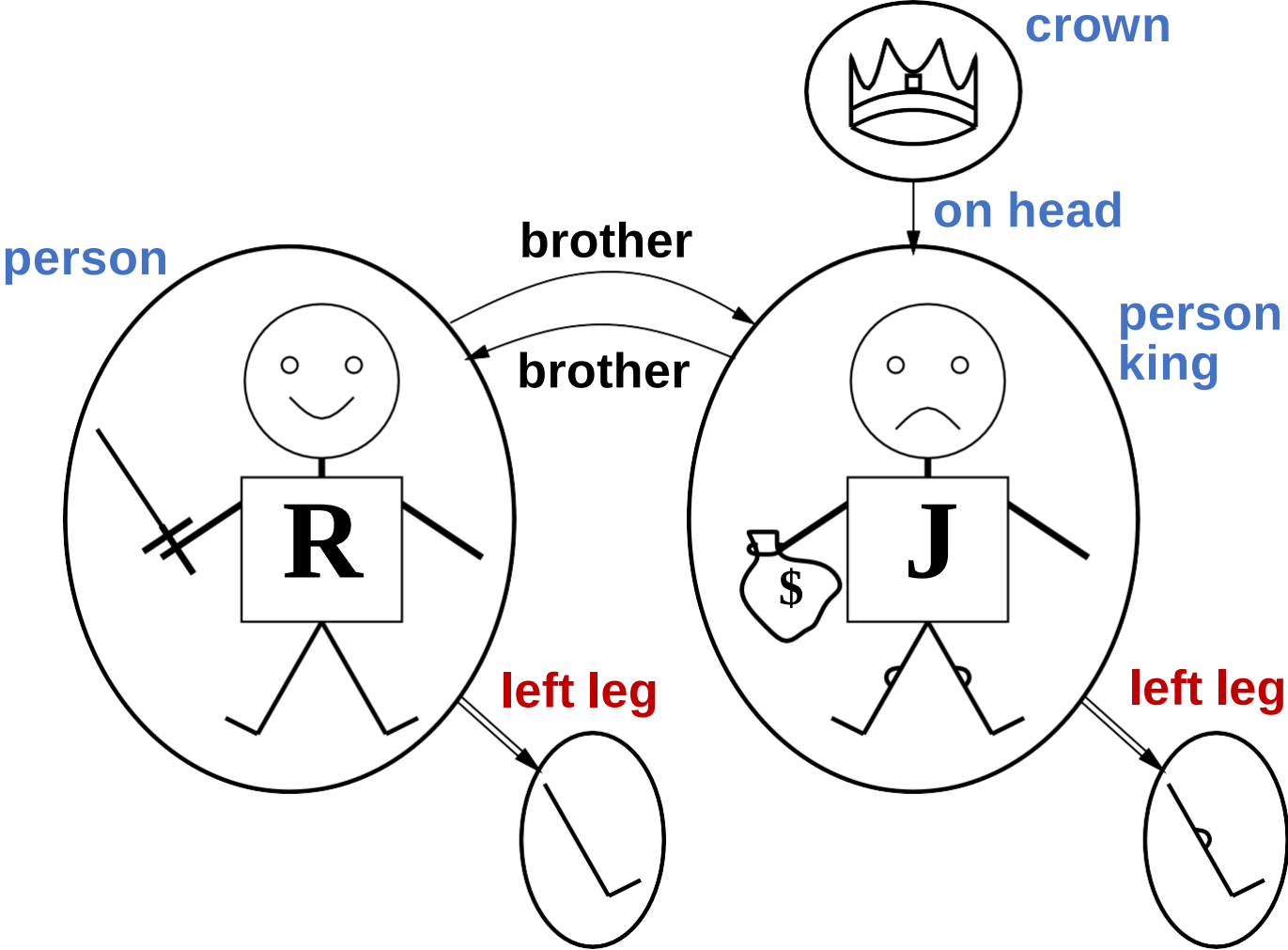
$$\textit{Sibling}(\textit{KingJohn}, \textit{Richard}) \Rightarrow \textit{Sibling}(\textit{Richard}, \textit{KingJohn})$$

$$>(1, 2) \vee \leq(1, 2)$$

$$>(1, 2) \wedge \neg >(1, 2)$$

Models for FOL

Example



Models for FOL

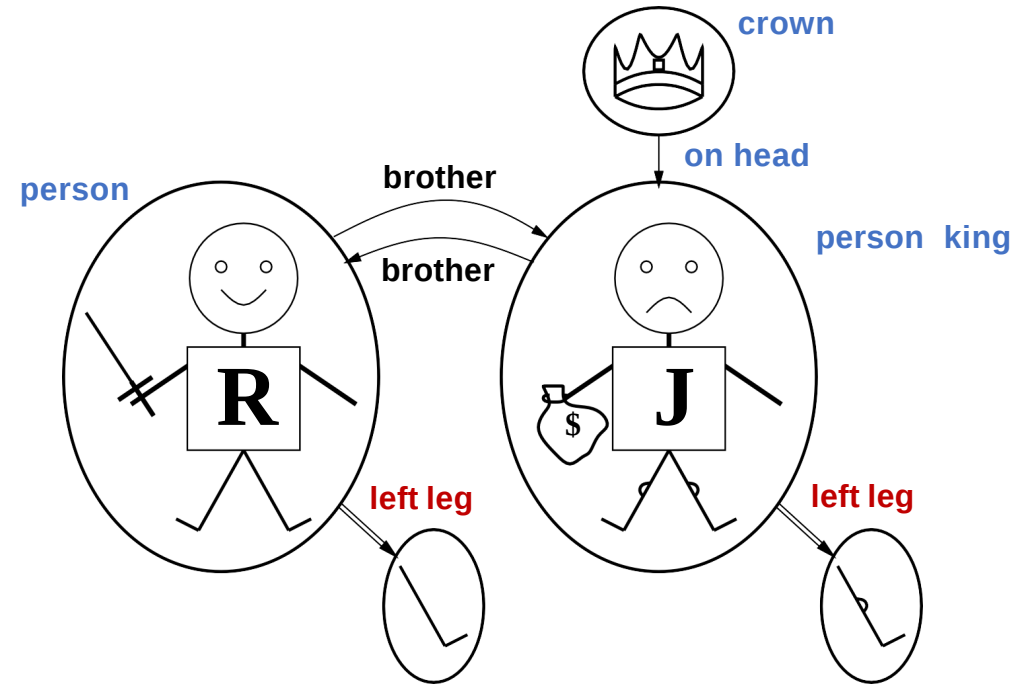
Brother(Richard, John)

Consider the interpretation in which:

Richard → Richard the Lionheart

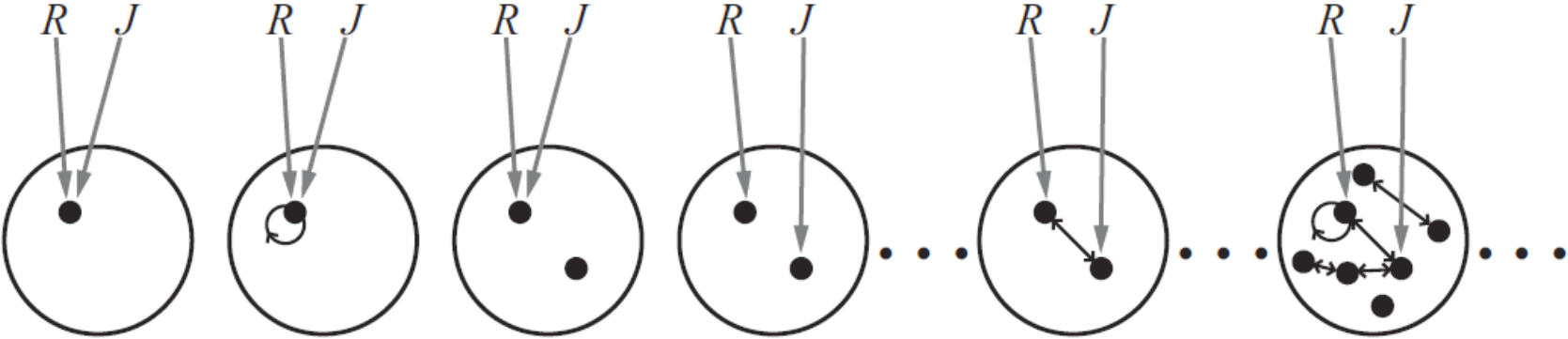
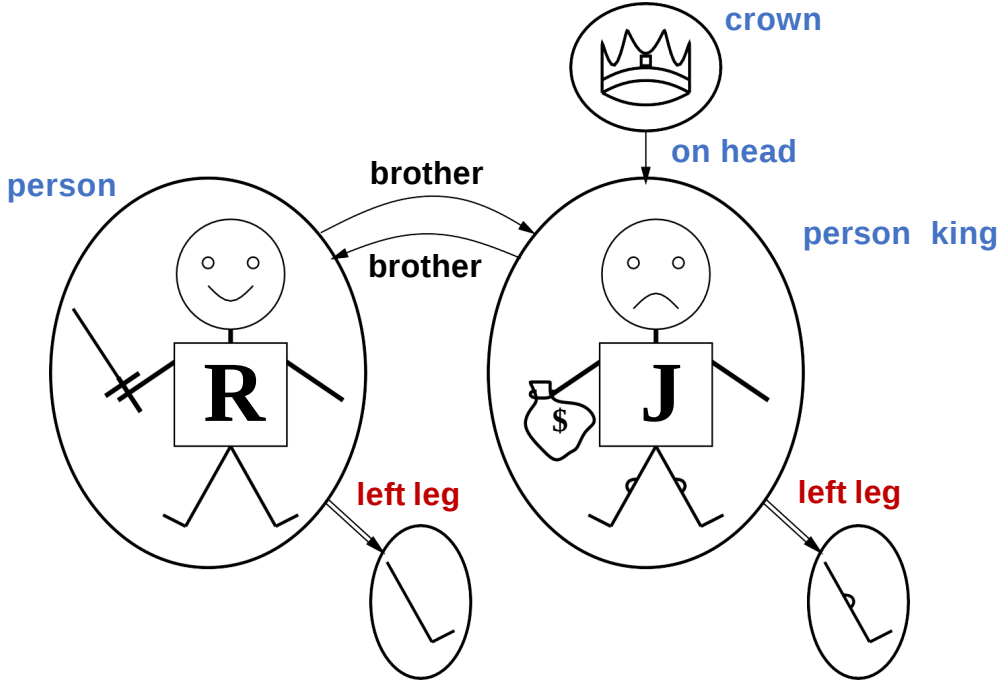
John → the evil King John

Brother → the brotherhood relation



Model for FOL

Lots of models!



Model for FOL

Lots of models!

Entailment in propositional logic can be computed by enumerating models

We **can** enumerate the FOL models for a given KB vocabulary:

For each number of domain elements n from 1 to ∞

For each k -ary predicate P_k in the vocabulary

For each possible k -ary relation on n objects

For each constant symbol C in the vocabulary

For each choice of referent for C from n objects . . .

Computing entailment by enumerating FOL models is not easy!

Truth in First-Order Logic

Sentences are true with respect to a **model** and an **interpretation**

Model contains ≥ 1 objects (**domain elements**) and relations among them

Interpretation specifies referents for

constant symbols $\rightarrow$ objects

predicate symbols $\rightarrow$ relations

function symbols $\rightarrow$ functional relations

An atomic sentence *predicate*(*term*₁, ..., *term*_{*n*}) is true:

iff the **objects** referred to by *term*₁, ..., *term*_{*n*}

are in the **relation** referred to by *predicate*

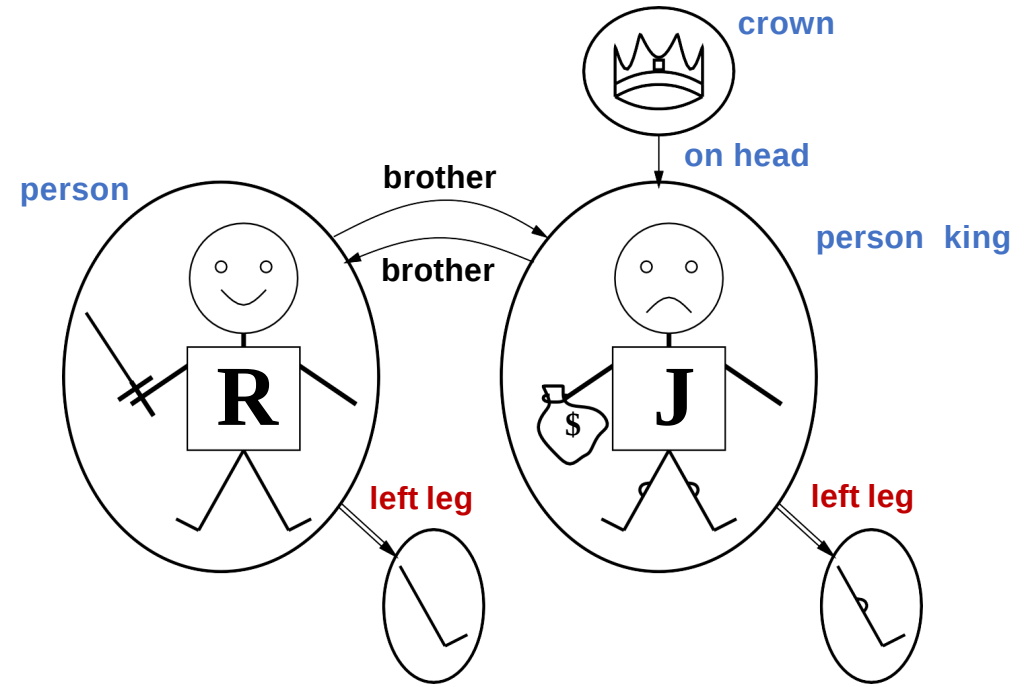
Models for FOL

Consider the interpretation in which:

Richard → Richard the Lionheart

John → the evil King John

Brother → the brotherhood relation



Under this interpretation, *Brother(Richard, John)* is true just in the case Richard the Lionheart and the evil King John are in the brotherhood relation in the model

Universal Quantification

$\forall(\text{variables}) \quad (\text{sentence})$

Everyone at the banquet is hungry:

$\forall x \quad \text{At}(x, \text{Banquet}) \Rightarrow \text{Hungry}(x)$

$\forall x \quad P$ is true in a model m iff P is true with x being each possible object in the model

Roughly speaking, equivalent to the conjunction of instantiations of P

$$\begin{aligned} & (\text{At}(\text{KingJohn}, \text{Banquet}) \Rightarrow \text{Hungry}(\text{KingJohn})) \\ & \wedge (\text{At}(\text{Richard}, \text{Banquet}) \Rightarrow \text{Hungry}(\text{Richard})) \\ & \wedge (\text{At}(\text{Banquet}, \text{Banquet}) \Rightarrow \text{Hungry}(\text{Banquet})) \\ & \wedge \dots \end{aligned}$$

Universal Quantification

Common mistake

Typically, $\Rightarrow$ is the main connective with $\forall$

Common mistake: using $\wedge$ as the main connective with $\forall$:

$$\forall x \text{ } At(x, \textit{Banquet}) \wedge \textit{Hungry}(x)$$

means “Everyone is at the banquet and everyone is hungry”

Existential Quantification

$\exists$ (*variables*) (*sentence*)

Someone at the tournament is hungry:

$\exists x \text{At}(x, \text{Tournament}) \wedge \text{Hungry}(x)$

$\exists x P$ is true in a model m iff P is true with x being
some possible object in the model

Roughly speaking, equivalent to the disjunction of instantiations of P

$(\text{At}(\text{KingJohn}, \text{Tournament}) \wedge \text{Hungry}(\text{KingJohn}))$
 $\vee (\text{At}(\text{Richard}, \text{Tournament}) \wedge \text{Hungry}(\text{Richard}))$
 $\vee (\text{At}(\text{Tournament}, \text{Tournament}) \wedge \text{Hungry}(\text{Tournament}))$
 $\vee \dots$

Existential Quantification

Common mistake

Typically, $\wedge$ is the main connective with $\exists$

Common mistake: using $\Rightarrow$ as the main connective with $\exists$:

$$\exists x At(x, Tournament) \Rightarrow Hungry(x)$$

is true if there is anyone who is not at the tournament!

Properties of Quantifiers

$\forall x \forall y$ is the same as $\forall y \forall x$

$\exists x \exists y$ is the same as $\exists y \exists x$

$\exists x \forall y$ is **not** the same as $\forall y \exists x$

$\exists x \forall y \text{ Loves}(x, y)$

“There is a person who loves everyone in the world”

$\forall y \exists x \text{ Loves}(x, y)$

“Everyone in the world is loved by at least one person”

Quantifier duality: each can be expressed using the other

$\forall x \text{ Likes}(x, \text{IceCream}) \quad \neg \exists x \neg \text{Likes}(x, \text{IceCream})$

$\exists x \text{ Likes}(x, \text{Broccoli}) \quad \neg \forall x \neg \text{Likes}(x, \text{Broccoli})$

Example Sentences

Brothers are siblings

$$\forall x, y \text{Brother}(x, y) \Rightarrow \text{Sibling}(x, y).$$

“Sibling” is symmetric

$$\forall x, y \text{Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x).$$

A first cousin is a child of a parent’s sibling

$$\forall x, y \text{FirstCousin}(x, y) \Leftrightarrow \exists p, ps \text{Parent}(p, x) \wedge \text{Sibling}(ps, p) \wedge \text{Parent}(ps, y)$$

Equality

$term_1 = term_2$ is true under a given interpretation
if and only if $term_1$ and $term_2$ refer to the same object

E.g., $1 = 2$ and $\forall x \times (Sqrt(x), Sqrt(x)) = x$ are satisfiable
 $2 = 2$ is valid

E.g., definition of (full) *Sibling* in terms of *Parent*:

$$\begin{aligned} \forall x, y \quad Sibling(x, y) \Leftrightarrow \\ [\neg(x = y) \wedge \exists m, f \neg(m = f) \wedge \\ Parent(m, x) \wedge Parent(f, x) \wedge Parent(m, y) \wedge Parent(f, y)] \end{aligned}$$

Interacting with FOL KBs

Suppose a wumpus-world agent is using an FOL KB and perceives a smell and a breeze (but no glitter) at $t = 5$:

$Tell(KB, Percept([Smell, Breeze, None], 5))$

$Ask(KB, \exists a Action(a, 5))$

i.e., does KB entail any action at $t = 5$?

Answer: $Yes, \{a/Shoot\} \leftarrow$ substitution (binding list)

Notation Alert!

Given a sentence S and a substitution θ ,
 $S\theta$ denotes the result of plugging θ into S ; e.g.,

Notation Alert!

$S = Smarter(x, y)$

$\theta = \{x/EVE, y/WALL-E\}$

$S\theta = Smarter(EVE, WALL-E)$

$Ask(KB, S)$ returns some/all θ such that $KB \models S\theta$

Inference in First-Order Logic

A) Reducing first-order inference to propositional inference

- Removing $\forall$
- Removing $\exists$
- Unification

B) *Lifting* propositional inference to first-order inference

- Generalized Modus Ponens
- FOL forward chaining

Universal Instantiation

Every instantiation of a universally quantified sentence is entailed by it:

$$\forall v a$$

$$\text{Subst}(\{v/g\}, a)$$

for any variable v and ground term g

E.g., $\forall x \text{ King}(x) \wedge \text{Greedy}(x) \Rightarrow \text{Evil}(x)$ yields

$$\text{King}(\text{John}) \wedge \text{Greedy}(\text{John}) \Rightarrow \text{Evil}(\text{John})$$

$$\text{King}(\text{Richard}) \wedge \text{Greedy}(\text{Richard}) \Rightarrow \text{Evil}(\text{Richard})$$

$$\text{King}(\text{Father}(\text{John})) \wedge \text{Greedy}(\text{Father}(\text{John})) \Rightarrow \text{Evil}(\text{Father}(\text{John}))$$

Existential Instantiation

For any sentence a , variable v , and constant symbol k
that does not appear elsewhere in the knowledge base:

$$\begin{array}{l} \exists v \quad a \\ \text{Subst}(\{v/k\}, a) \end{array}$$

E.g., $\exists x \quad \text{Crown}(x) \wedge \text{OnHead}(x, \text{John})$ yields

$$\text{Crown}(C_1) \wedge \text{OnHead}(C_1, \text{John})$$

provided C_1 is a new constant symbol, called a **Skolem constant**

Reduction to Propositional Inference

Suppose the KB contains just the following:

$$\forall x \text{ King}(x) \wedge \text{Greedy}(x) \Rightarrow \text{Evil}(x)$$

$\text{King}(\text{John})$

$\text{Greedy}(\text{John})$

$\text{Brother}(\text{Richard}, \text{John})$

Instantiating the universal sentence in *all possible* ways, we have

$$\text{King}(\text{John}) \wedge \text{Greedy}(\text{John}) \Rightarrow \text{Evil}(\text{John})$$

$$\text{King}(\text{Richard}) \wedge \text{Greedy}(\text{Richard}) \Rightarrow \text{Evil}(\text{Richard})$$

$\text{King}(\text{John})$

$\text{Greedy}(\text{John})$

$\text{Brother}(\text{Richard}, \text{John})$

The new KB is **propositionalized**: proposition symbols are

$\text{King}(\text{John})$, $\text{Greedy}(\text{John})$, $\text{Evil}(\text{John})$, $\text{King}(\text{Richard})$ etc.

Reduction to Propositional Inference

Claim: a ground sentence* is entailed by new KB iff entailed by original KB

Claim: every FOL KB can be propositionalized so as to preserve entailment

Idea: propositionalize KB and query, apply resolution, return result Problem: with function symbols, there are infinitely many ground terms,

e.g., *Father(Father(Father(John)))*

Theorem: Herbrand (1930). If a sentence *a* is entailed by an FOL KB, it is entailed by a finite subset of the propositional KB

Idea: For $n = 0$ to ∞ do

create a propositional KB by instantiating with depth- n terms see if *a* is entailed by this KB

Problem: works if *a* is entailed, loops if *a* is not entailed

Theorem: Turing (1936), Church (1936), entailment in FOL is semidecidable

Problems with Propositionalization

Propositionalization seems to generate lots of irrelevant sentences. E.g., from

$\forall x \text{ King}(x) \wedge \text{Greedy}(x) \Rightarrow \text{Evil}(x)$

$\text{King}(\text{John})$

$\forall y \text{ Greedy}(y)$

$\text{Brother}(\text{Richard}, \text{John})$

it seems obvious that $\text{Evil}(\text{John})$, but propositionalization produces lots of facts such as $\text{Greedy}(\text{Richard})$ that are irrelevant

Unification

We can get the inference immediately if we can find a substitution θ such that $King(x)$ and $Greedy(x)$ match $King(John)$ and $Greedy(y)$

$\theta = \{x/John, y/John\}$ works

$Unify(a, \beta) = \theta$ if $a\theta = \beta\theta$

p	q	θ
$Knows(John, x)$	$Knows(John, Jane)$	$\{x/Jane\}$
$Knows(John, x)$	$Knows(y, Sam)$	$\{x/Sam, y/John\}$
$Knows(John, x)$	$Knows(y, Mother(y))$	$\{y/John, x/Mother(John)\}$
$Knows(John, x)$	$Knows(x, Sam)$	$fail$

Standardizing apart eliminates overlap of variables, e.g., $Knows(z_{17}, OJ)$

Generalized Modus Ponens (GMP)

$$\frac{p'_1, p'_2, \dots, p'_n, \quad (p_1 \wedge p_2 \dots \wedge p_n \Rightarrow q)}{q\theta} \quad \text{where } p'_i\theta = p_i\theta \text{ for all } i$$

Example

p'_1 is *King(John)*

p'_2 is *Greedy(y)*

p_1 is *King(x)*

p_2 is *Greedy(x)*

q is *Evil(x)*

θ is $\{x/\text{John}, y/\text{John}\}$

$q\theta$ is *Evil(John)*

GMP used with KB of **definite clauses** (**exactly** one positive literal)

All variables assumed universally quantified

FOL Forward Chaining

```
function FOL-FC-Ask( $KB$ ,  $\alpha$ ) returns a substitution or false

  repeat until new is empty
     $new \leftarrow \{ \}$ 
    for each sentence  $r$  in  $KB$  do
       $(p_1 \wedge \dots \wedge p_n \Rightarrow q) \leftarrow \text{Standardize-Apart}(r)$ 
      for each  $\theta$  such that  $(p_1 \wedge \dots \wedge p_n)\theta = (p_1^t \wedge \dots \wedge p_n^t)\theta_n$ 
        for some  $p_1^t, \dots, p_n^t$  in  $KB$ 
           $q^t \leftarrow \text{Subst}(\theta, q)$ 
          if  $q^t$  is not a renaming of a sentence already in  $KB$  or new then do
            add  $q^t$  to new
             $\varphi \leftarrow \text{Unify}(q^t, \alpha)$ 
            if  $\varphi$  is not fail then return  $\varphi$ 
    add new to  $KB$ 
  return false
```