Announcements

Midterm 1 Exam

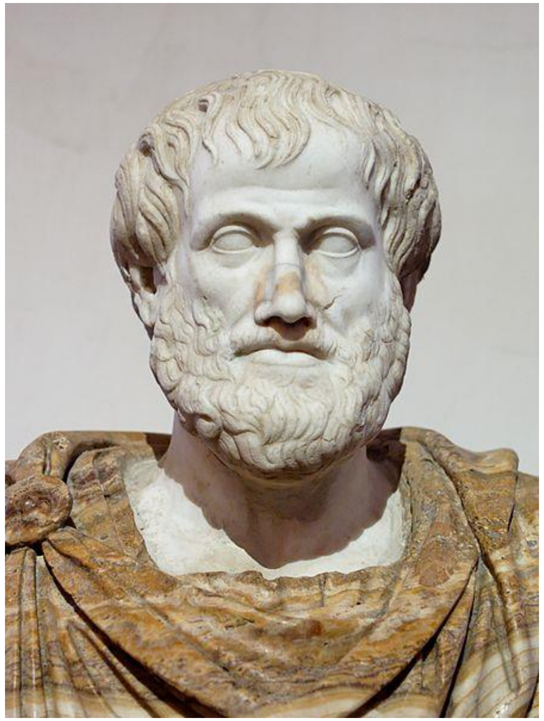
- Tue 10/1, in class
- See Piazza for details
  - Practice exam
  - Recitation Friday
  - Vote for Sunday review session time slot

Assignments:

- P2: Logic and Planning
  - Due Sat 10/5, 10 pm

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:

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

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	$KingJohn,\ 2,\ CMU, \ldots$
Predicates	$Brother,\ >, \ldots$
Functions	$Sqrt,\ LeftLegOf, \ldots$
Variables	$x,\ y,\ a,\ b, \ldots$
Connectives	$\wedge\ \vee\ \neg\ \Rightarrow\ \ \Leftrightarrow$
Equality	$=$
Quantifiers	$\forall\ \exists$

Syntax of FOL

Atomic sentence = $predicate(term_1, \ldots, term_n)$
or $term_1 = term_2$

Term = $function(term_1, \ldots, term_n)$
or $constant$
or $variable$

Examples

$Brother(KingJohn, RichardTheLionheart)$

$> (Length(LeftLegOf(Richard)),\ Length(LeftLegOf(KingJohn)))$

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

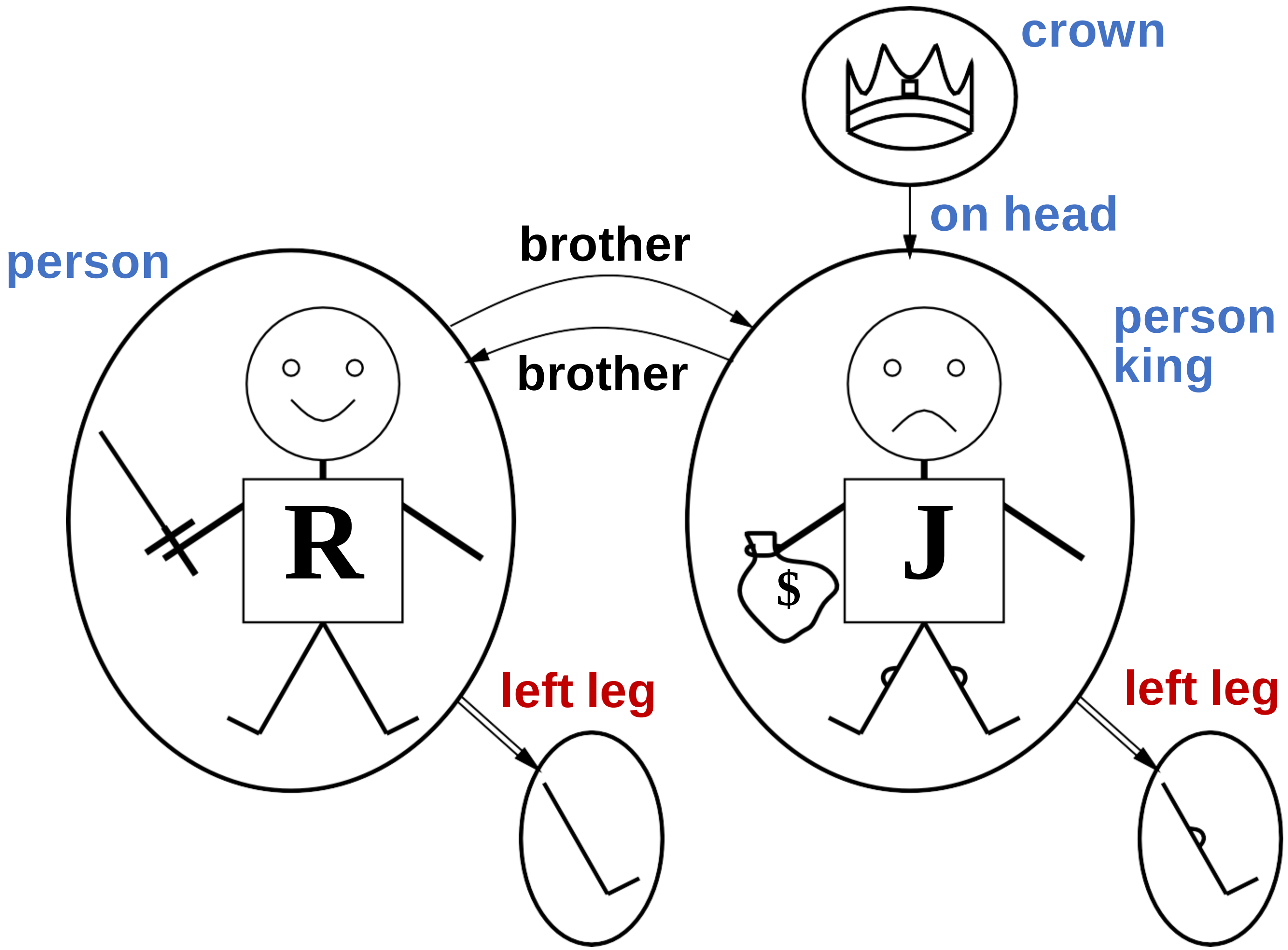
$Sibling(KingJohn, Richard) \Rightarrow Sibling(Richard, KingJohn)$

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

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

Models for FOL

Example

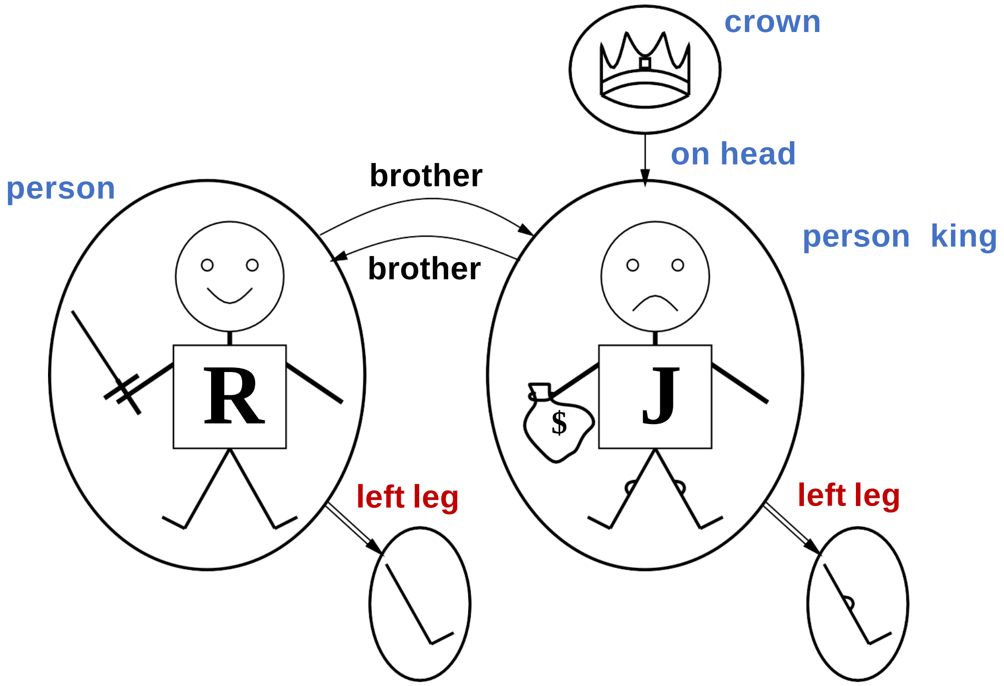


Models for FOL

Brother(Richard, John)

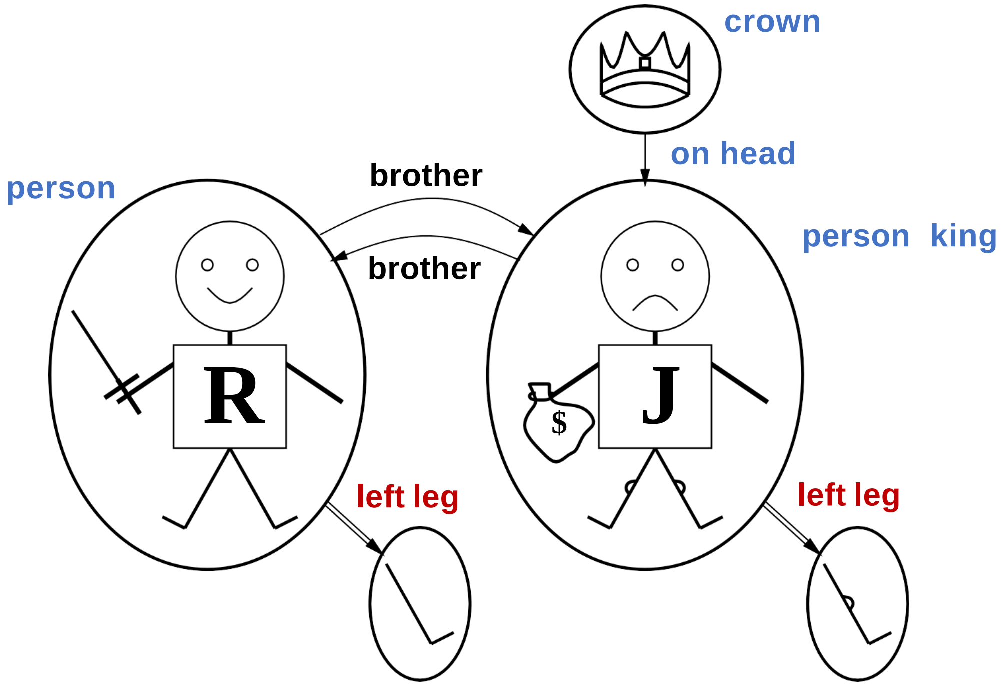
Consider the interpretation in which:

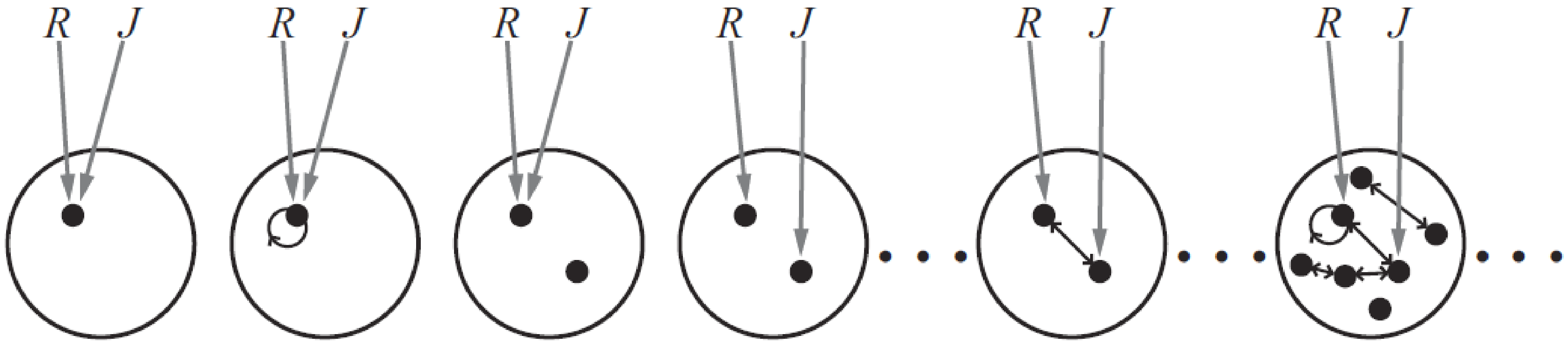
Richard → Richard the Lionheart
John → the evil King John
Brother → the brotherhood relation



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 $\geq$ 1 objects (domain elements) and relations among them

Interpretation specifies referents for

- constant symbols → objects
- predicate symbols → relations
- function symbols → functional relations

An atomic sentence $predicate(term_1, \ldots, term_n)$ is true:

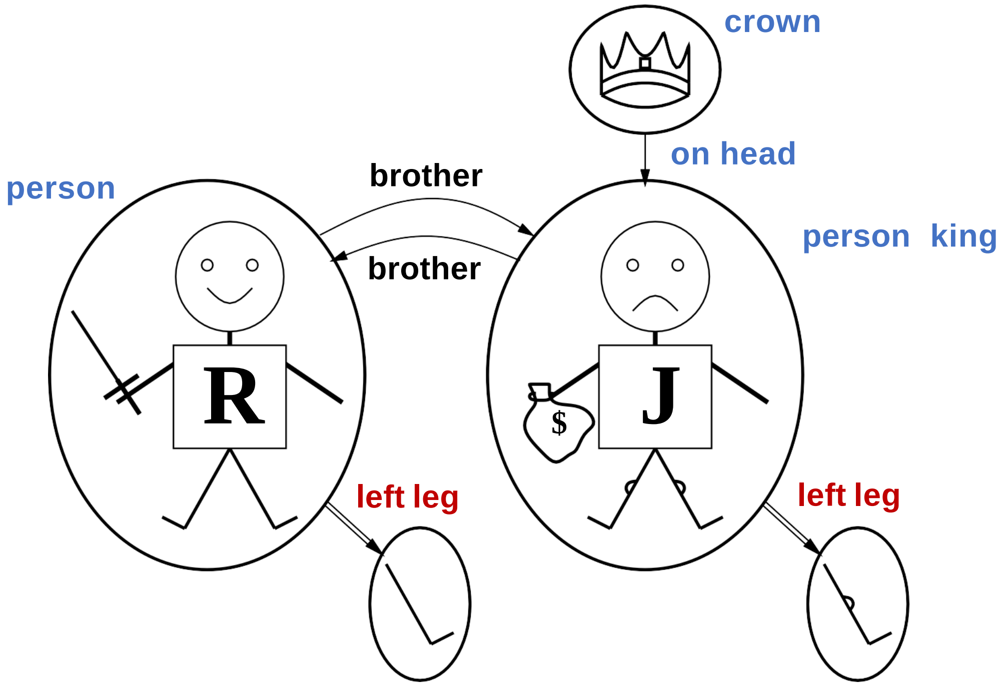
iff the objects referred to by $term_1, \ldots, term_n$

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

$\forall \langle variables \rangle \quad \langle sentence \rangle$

Everyone at the banquet is hungry:

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

$\forall x \quad P$ 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

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

Universal Quantification

Common mistake

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

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

$$\forall x\ At(x,\ Banquet) \wedge Hungry(x)$$

means "Everyone is at the banquet and everyone is hungry"

Existential Quantification

$\exists \langle variables \rangle \quad \langle sentence \rangle$

Someone at the tournament is hungry:
$\exists x At(x, Tournament) \wedge Hungry(x)$

$\exists x P$ 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

$$
\begin{aligned}
& (At(KingJohn, Tournament) \wedge Hungry(KingJohn)) \\
\vee\ & (At(Richard, Tournament) \wedge Hungry(Richard)) \\
\vee\ & (At(Tournament, Tournament) \wedge Hungry(Tournament)) \\
\vee\ & \ldots
\end{aligned}
$$

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 $\forall y \; \exists x$

$\exists x \, \forall y \, Loves(x, y)$
"There is a person who loves everyone in the world"

$\forall y \, \exists x \, Loves(x, y)$
"Everyone in the world is loved by at least one person"

Quantifier duality: each can be expressed using the other

$\forall x \, Likes(x, IceCream) \qquad \neg \exists x \, \neg Likes(x, IceCream)$

$\exists x \, Likes(x, Broccoli) \qquad \neg \forall x \, \neg Likes(x, Broccoli)$

Example Sentences

Brothers are siblings

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

"Sibling" is symmetric

$$\forall x, y Sibling(x, y) \Leftrightarrow Sibling(y, x).$$

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

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

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*:

$$\forall x, y \ \ Sibling(x, y) \Leftrightarrow$$
$$[\neg(x = y) \ \wedge \ \exists m, f \ \neg(m = f) \ \wedge$$
$$Parent(m, x) \wedge Parent(f, x) \wedge Parent(m, y) \wedge Parent(f, y)]$$

Piazza Poll 1

Given the following two FOL sentences:

$$\gamma: \quad \forall\, x \quad Hungry(x)$$

$$\delta: \quad \exists\, x \quad Hungry(x)$$

Which of these is true?

A) $\gamma \vDash \delta$

B) $\delta \vDash \gamma$

C) Both

D) Neither

Piazza Poll 1

Given the following two FOL sentences:

$$\gamma: \quad \forall x \quad Hungry(x)$$

$$\delta: \quad \exists x \quad Hungry(x)$$

Which of these is true?

A) $\gamma \vDash \delta$

B) $\delta \vDash \gamma$

C) Both

D) Neither

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: $Yes, \{a/Shoot\}$ ← 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.,
$S = Smarter(x, y)$
$\theta = \{x/EVE, y/WALL\text{-}E\}$
$S\theta = Smarter(EVE, WALL\text{-}E)$

Notation Alert!

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

Inference in First-Order Logic

A) Reducing first-order inference to propositional inference

- Removing ∀
- Removing ∃
- 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., $\forall x \quad King(x) \land Greedy(x) \Rightarrow Evil(x)$ yields

$$King(John) \land Greedy(John) \Rightarrow Evil(John)$$

$$King(Richard) \land Greedy(Richard) \Rightarrow Evil(Richard)$$

$$King(Father(John)) \land Greedy(Father(John)) \Rightarrow Evil(Father(John))$$

Existential Instantiation

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

$$\frac{\exists v \quad a}{\mathrm{Subst}(\{v/k\}, a)}$$

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

$Crown(C_1) \wedge OnHead(C_1, John)$

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

Reduction to Propositional Inference

Suppose the KB contains just the following:

$$\forall x \;\; King(x) \wedge Greedy(x) \;\Rightarrow\; Evil(x)$$

$$King(John)$$
$$Greedy(John)$$
$$Brother(Richard, John)$$

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

$$King(John) \wedge Greedy(John) \;\Rightarrow\; Evil(John)$$

$$King(Richard) \wedge Greedy(Richard) \Rightarrow Evil(Richard)$$

$$King(John)$$
$$Greedy(John)$$
$$Brother(Richard, John)$$

The new KB is propositionalized: proposition symbols are

$King(John)$, $Greedy(John)$, $Evil(John)$, $King(Richard)$ 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 α 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 α is entailed by this KB

Problem: works if α is entailed, loops if α 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

$$\forall x\ King(x) \land Greedy(x) \Rightarrow Evil(x)$$

$$King(John)$$

$$\forall y\ Greedy(y)$$

$$Brother(Richard, John)$$

it seems obvious that $Evil(John)$, but propositionalization produces lots of facts such as $Greedy(Richard)$ 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

$\text{Unify}(\alpha, \beta) = \theta$ if $\alpha\theta = \beta\theta$

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., $Knows(z_{17}, Sam)$

Generalized Modus Ponens (GMP)

$$\frac{p_1', p_2', \ldots, p_n', \qquad (p_1 \wedge p_2 \;\ldots\; \wedge p_n \Rightarrow q)}{q\theta}$$

where $p_i'\theta = p_i\theta$ 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 $\{x/John, y/John\}$

$q\theta$ 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, α) returns a substitution or false

    repeat until new is empty
        new ← { }
        for each sentence r in KB do
            ( p1 ∧ . . . ∧ pn  ⇒  q) ← Standardize-Apart(r)
            for each θ such that (p1 ∧ . . . ∧ pn)θ = (p'1 ∧ . . . ∧ p'n)θ
                            for some p'1, . . . , p'n in KB
                q' ← Subst(θ, q)
                if q' is not a renaming of a sentence already in KB or new then do
                    add q' to new
                    φ ← Unify(q', α)
                    if φ is not fail then return φ
        add new to KB
    return false
```