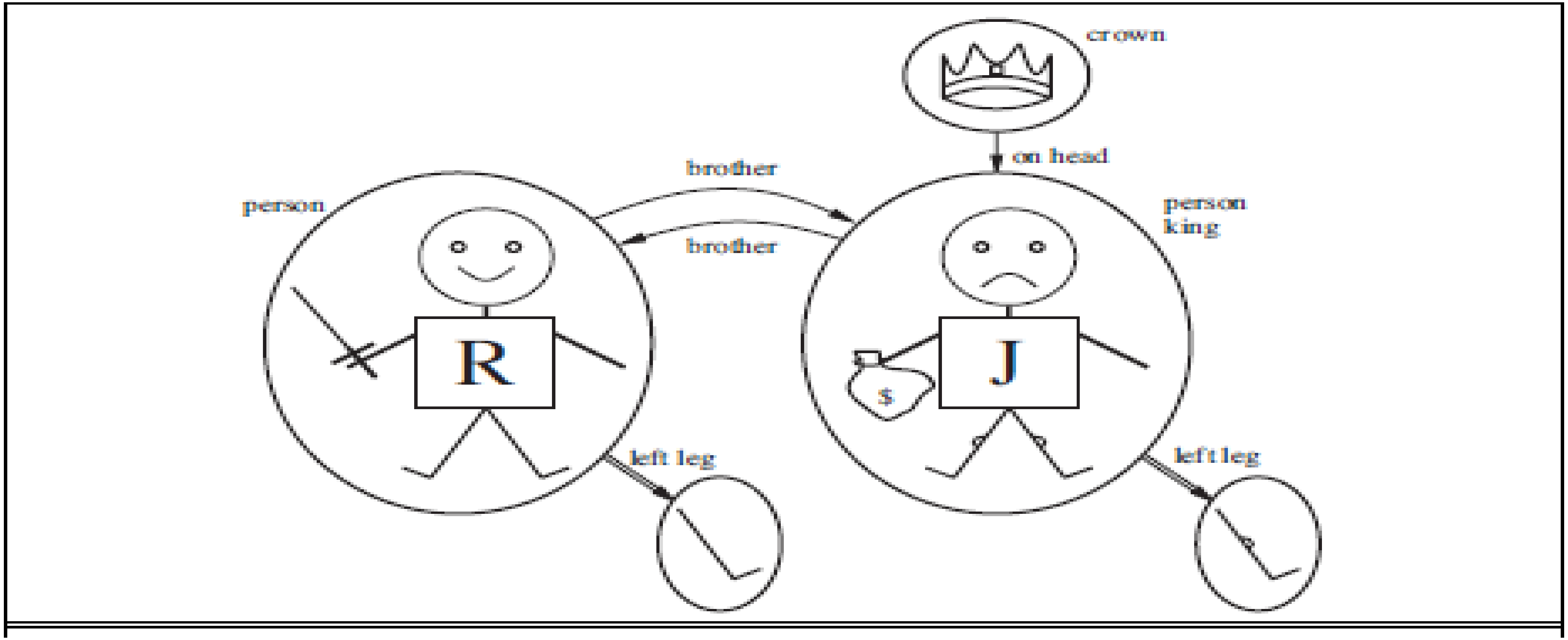
First Order Logic

Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief $\in [0, 1]$
Fuzzy logic	facts with degree of truth $\in [0, 1]$	known interval value

Formal languages and their ontological and epistemological commitments.

FOL



A model containing five objects, two binary relations, three unary relations (indicated by labels on the objects), and one unary function, left-leg.

FOL

Sentence → AtomicSentence | ComplexSentence

AtomicSentence → Predicate | Predicate(Term, . . .) | Term = Term

ComplexSentence → (Sentence) | [Sentence]

| ¬Sentence

| Sentence ∧ Sentence

| Sentence ∨ Sentence

| Sentence ⇒ Sentence

| Sentence ⇔ Sentence

| Quantifier Variable, . . . Sentence

FOL

Term → Function(Term, . . .)

| Constant

| Variable

Quantifier → ∀ | ∃

Constant → A | X1 | John | · · ·

Variable → a | x | s | ⋯

Predicate → True | False | After | Loves | Raining | ⋯

Function → Mother | LeftLeg | · · ·

OPERATOR PRECEDENCE : ¬,=, ∧, ∨,⇒,⇔

FOL

Atomic sentences

LeftLeg(John) refers to King John’s left leg. - Function

Brother (Richard , John). - Predicate

Married(Father (Richard),Mother (John)) - Complex terms as arguments

An atomic sentence is true in a given model if the relation referred to by the predicate symbol holds among the objects referred to by the arguments.

Complex Sentences

¬Brother (LeftLeg(Richard), John)

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

King(Richard) ∨ King(John)

¬King(Richard) ⇒ King(John)

FOL

$\forall$ x King(x) $\Rightarrow$ Person(x)

x $\rightarrow$ Richard the Lionheart,

x $\rightarrow$ King John,

x $\rightarrow$ Richard's left leg,

x $\rightarrow$ John's left leg,

x $\rightarrow$ the crown.

Richard the Lionheart is a king $\Rightarrow$ Richard the Lionheart is a person.

King John is a king $\Rightarrow$ King John is a person.

Richard's left leg is a king $\Rightarrow$ Richard's left leg is a person.

John's left leg is a king $\Rightarrow$ John's left leg is a person.

The crown is a king $\Rightarrow$ the crown is a person.

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

Implication is true when premise is True - intended interpretation

FOL

wrong

$\forall$ x King(x) $\wedge$ Person(x)

Richard the Lionheart is a king $\wedge$ Richard the Lionheart is a person,

King John is a king $\wedge$ King John is a person,

Richard’s left leg is a king $\wedge$ Richard’s left leg is a person,

Implication is true whenever premise is False – regardless of truth of conclusion - extended interpretation

FOL

$\exists$ x Crown(x) $\land$ OnHead(x, John)

$\exists$x is pronounced "There exists an x such that . . ." or "For some x . . ."

Richard the Lionheart is a crown $\land$ Richard the Lionheart is on John's head;

King John is a crown $\land$ King John is on John's head;

Richard's left leg is a crown $\land$ Richard's left leg is on John's head;

John's left leg is a crown $\land$ John's left leg is on John's head;

The crown is a crown $\land$ the crown is on John's head.

Just as $\Rightarrow$ appears to be the natural connective to use with $\forall$, $\land$ is the natural connective to use with $\exists$.

FOL - nested quantifiers

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

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

$\forall x \exists y \; Loves(x, y)$. Everybody loves somebody

$\exists y \forall x \; Loves(x, y)$. There is someone loved by everyone

$\forall x \; (Crown(x) \lor (\exists x \; Brother(Richard, x)))$ variable applies to innermost quantifier

Quantifiers

$\forall x \neg Likes(x, cabbage)$ is equivalent to $\neg\exists x\ Likes(x, cabbage)$.   Everyone dislikes cabbage

$\forall x\ Likes(x, IceCream)$ is equivalent to $\neg\exists x \neg Likes(x, IceCream)$.

$\forall x \neg P \equiv \neg\exists x\ P$	$\neg(P \lor Q) \equiv \neg P \land \neg Q$
$\neg\forall x\ P \equiv \exists x \neg P$	$\neg(P \land Q) \equiv \neg P \lor \neg Q$
$\forall x\ P \equiv \neg\exists x \neg P$	$P \land Q \equiv \neg(\neg P \lor \neg Q)$
$\exists x\ P \equiv \neg\forall x \neg P$	$P \lor Q \equiv \neg(\neg P \land \neg Q)$.

FOL - Equality

$Father(John) = Henry$

Richard has two brothers, John and Geoffrey.

$\exists\, x, y\ Brother(x, Richard) \land Brother(y, Richard)$?

$\exists\, x, y\ Brother(x, Richard) \land Brother(y, Richard) \land \neg(x = y)$.

$Brother(John, Richard) \land Brother(Geoffrey, Richard)$?

$Brother(John, Richard) \land Brother(Geoffrey, Richard) \land John = Geoffrey$

$\land\ \forall x\ Brother(x, Richard) \Rightarrow (x = John \lor x = Geoffrey)$.

Database Semantics

Database Semantics

UNIQUE- NAMES assumption - every constant symbol refer to a distinct object

CLOSED-WORLD assumption - atomic sentences not known to be true are in fact false

Domain closure – Each model contains no more domain elements than those named by the constant symbols.

FOL assertions and queries

TELL(KB, King(John)) .

TELL(KB, Person(Richard)) .

TELL(KB, $\forall$ x King(x) $\Rightarrow$ Peson(x))

ASK(KB, King(John))

ASK(KB, Person(John))

ASK(KB, $\exists$ x Person(x)) . Returns T

ASKVARS(KB, Person(x))   Returns stream of answers

FOL - Kinship Domain

one’s mother is one’s female parent:

$\forall\, m, c\ Mother(c) = m \Leftrightarrow Female(m) \land Parent(m, c)$.

One’s husband is one’s male spouse:

$\forall\, w, h\ Husband(h, w) \Leftrightarrow Male(h) \land Spouse(h, w)$.

Male and female are disjoint categories:

$\forall\, x\ Male(x) \Leftrightarrow \neg Female(x)$.

Parent and child are inverse relations:

$\forall\, p, c\ Parent(p, c) \Leftrightarrow Child(c, p)$.

A grandparent is a parent of one’s parent:

$\forall\, g, c\ Grandparent(g, c) \Leftrightarrow \exists p\ Parent(g, p) \land Parent(p, c)$.

A sibling is another child of one’s parents:

$\forall\, x, y\ Sibling(x, y) \Leftrightarrow x = y \land \exists p\ Parent(p, x) \land Parent(p, y)$.

Wumpus World

Input – percept sentence with timestep

Percept ([Stench, Breeze, Glitter , None, None], 5) .

Actions in the wumpus world can be represented by logical terms:

Turn(Right), Turn(Left), Forward , Shoot , Grab, Climb .

To determine which is best, the agent program executes the query

ASKVARS(∃ a BestAction(a, 5))

The raw percept data implies certain facts about the current state (perception rules)

$\forall$ t, s, g, m, c Percept ([s, Breeze, g,m, c], t) $\Rightarrow$ Breeze(t) ,

$\forall$ t, s, b, m, c Percept ([s, b, Glitter,m, c], t) $\Rightarrow$ Glitter (t) ,

Reflex behavior can also be implemented by quantified implication sentences.

$\forall$ t Glitter (t) $\Rightarrow$ BestAction(Grab, t)

Wumpus world

Adjacency of any two squares :

$\forall$ x, y, a, b Adjacent ([x, y], [a, b]) $\Leftrightarrow$ (x = a $\land$ (y = b − 1 $\lor$ y = b + 1)) $\lor$ (y = b $\land$ (x = a − 1 $\lor$ x = a + 1)) .

Agent is at square s at time t.

At(Agent, s, t)

Given its current location, the agent can infer properties of the square from properties of its current percept.

$\forall$ s, t At(Agent, s, t) $\land$ Breeze(t) $\Rightarrow$ Breezy(s)

agent can deduce where the pits are with one axiom

$\forall$ s Breezy(s) $\Leftrightarrow$ $\exists$r Adjacent (r, s) $\land$ Pit(r)

Knowledge Engineering

1. Identify the task.
2. Assemble the relevant knowledge.
3. Decide on a vocabulary of predicates, functions, and constants.
4. Encode general knowledge about the domain.
5. Encode a description of the specific problem instance.
6. Pose queries to the inference procedure and get answers.
7. Debug the knowledge base.

Example

1. Leena is a professor
2. All professors are people.
3. Frank is the dean.
4. Deans are professors.
5. All professors consider the dean a friend or don’t know him.
6. Everyone is a friend of someone.
7. People only criticize people that are not their friends.
8. Leena criticized Frank.

KB

prof(leena)

$\forall$ x (prof(x) => person(x))

dean(frank)

$\forall$ x (dean(x) => prof(x))

$\forall$ x ($\forall$ y (prof(x) $\wedge$ dean(y) => friend(y, x) $\vee$ $\neg$ knows(x, y)))

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

$\forall$ x ($\forall$ y (person(x) $\wedge$ person(y) criticize (x,y) => $\neg$ friend (y,x)))

criticize(leena,frank)

Question: Is Frank no friend of Leena?

$\neg$ friend-of(frank,leena)

Unit resolution inference rule

each l is a literal, li and m are complementary

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m}{\ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k},$$

Unit resolution inference rule takes a clause - a disjunction of literals

and a literal as input

and produces a new clause

Full resolution rule: *li* and *mj* are complementary

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m_1 \vee \cdots \vee m_n}{\ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n},$$

Ex:

$$\frac{P_{1,1} \vee P_{3,1}, \qquad \neg P_{1,1} \vee \neg P_{2,2}}{P_{3,1} \vee \neg P_{2,2}}$$

Factoring: removal of multiple copies of literals: by resolving (A V B) with (A V ¬ B) , we get (A V A), reduced to A

Resolution rule forms basis for Sound , complete inference procedures

A resolution-based theorem prover can, for any sentences α and β in propositional logic, decide whether α |= β.

Resolution

Resolution Rule of Inference

$$\frac{P_{1,1} \vee P_{3,1}, \qquad \neg P_{1,1} \vee \neg P_{2,2}}{P_{3,1} \vee \neg P_{2,2}}$$

Algorithm: Resolution Proof

Negate the theorem to be proved, and add the result to the knowledge base.

Bring knowledge base into conjunctive normal form (CNF) -

CNF: conjunctions of disjunctions. Each disjunction is called a clause.

Until there is no resolvable pair of clauses,

Find resolvable clauses and resolve them.

Add the results of resolution to the knowledge base.

If NIL (empty clause) is produced, stop and report that the (original) theorem is true.

Report that the (original) theorem is false.

Resolution

Resolution Example: Propositional Logic

- To prove: ¬ P
- Transform Knowledge Base into CNF
- Proof

1. $\neg P \vee Q$	Sentence 1
2. $\neg Q \vee R$	Sentence 2
3. $\neg R$	Sentence 3
4. P	Assume opposite
5. Q	Resolve 4 and 1
6. R	Resolve 5 and 2
7. nil	Resolve 6 with 3

More examples on FOL

Atomic sentences: Predicate | Predicate (Term, …) | Term = Term

Term = function(Term, …)

Function appears only as an attribute in a predicate.

Predicate has a truth value for the domains of attributes, when instantiated to object values.

1. Everyone loves its mother $\forall x\ \exists y\ Mother(x,y) \land Loves(x,y)$: Mother, Loves :Predicates

   $\forall x\ Loves(x, Mother(x))$: Mother : function

2. Not all students take both History and Biology

   predicates: student(x) : x is a student

   Takes (x, y) : subject x is taken by y

   constants: History, Biology

   $\neg [\forall x\ student(x) \Rightarrow Takes(History,x) \land Takes(Biology,x)]$

   $\exists x\ student(x) \land [\neg Takes(History,x) \lor \neg Takes(Biology,x)]$

More examples on FOL

3. Only one student Failed in History

predicate: Failed(x,y) : student y failed in subject x

$\exists x$ student(x) $\wedge$ Failed(History, x) : There is somebody who failed in History

$\exists x$ [student(x) $\wedge$ Failed(History, x) $\wedge$ $\forall y$ [$\neg$ (x=y) $\wedge$ student(y) => $\neg$ Failed (History, y)]]

4. Only one student failed in both History and Biology

$\exists x$ [student(x) $\wedge$ Failed(History,x) $\wedge$ Failed(Biology,x) $\wedge$

$\forall y$[$\neg$ (x=y) $\wedge$ student(y) => $\neg$ Failed(History, y) $\wedge$ Failed(Biology,y)]]

5. The best score in History is better than best score in Biology

function: score(subject, student)

Predicate: Greater(x,y) : x>y

More examples on FOL

5. The best score in History is better than best score in Biology

function: Score(subject, student)

Predicate: Greater(x,y) : x>y

$\forall x$ [student(x) $\wedge$ Takes(Biology,x) => $\exists y$ student(y) $\wedge$ Takes(History,x) $\wedge$
Greater(Score(History,y), Score(Biology,x))]

6. No Person likes a Professor unless the Professor is Smart

Predicate: Smart (x)

Likes(x, y) : y likes x

$\forall x$ Professor(x) $\wedge$ $\neg$Smart(x) => $\forall y$ $\neg$Likes(x,y)

More examples on FOL

Russels Paradox:

There is a single barber in Town. Those and those who do not shave themselves are shaved by the barber. Who shaves barber?

Predicates: Barber(x), Shaves(x,y) : y shaves x

$\exists x$ [Barber(x) $\wedge$ $\forall y$ $\neg$(x=y) => $\neg$Barber(y)] :There is a single barber in town

$\forall x$ [$\neg$ Shaves(x,x) <=> Shaves (x,y) $\wedge$ Barber(y)]

Homework:

Politicians can fool some of the persons all the time, and they can fool all the people some of the time, but can not fool all the people all the time.

Inference Rules for Quantifiers

Universal Quantifier $\forall$

KB: axiom: All greedy kings are evil

Predicates: King(x), Greedy (x), Evil (x)

Sentence: $\forall$ x King (x) $\land$ Greedy (x) => Evil (x)

King(John) $\land$ Greedy(John) $\Rightarrow$ Evil(John)	: {x/John}
King(Richard) $\land$ Greedy(Richard) $\Rightarrow$ Evil(Richard)	: {x/Richard}
King(Father (John)) $\land$ Greedy(Father (John)) $\Rightarrow$ Evil(Father (John))	: {x/Father(John)}

Universal Instantiation (UI) : we can infer any sentence obtained by substituting a ground term (a term without variables)

$$\frac{\forall v \quad \alpha}{\mathrm{SUBST}(\{v/g\}, \alpha)}$$

For any variable v and ground term g. SUBST (θ, α) denotes the result of applying the substitution θ to the sentence α

Inference Rules for Quantifiers

Existential Instantiation:

for any sentence α, variable v, and constant symbol k that does not appear elsewhere in the knowledge base,

$$\frac{\exists v \quad \alpha}{\text{SUBST}(\{v/k\}, \alpha)}$$

Variable v is replaced by a new constant symbol, k

From the sentence,

$\exists x \; Crown(x) \land OnHead(x, John)$

we can infer the sentence

$Crown(C1) \land OnHead(C1, John)$

as long as C1 does not appear elsewhere in the knowledge base.

New name for the object is Skolem Constant.

Existential Instantiation is a special case of a more general process called skolemization

Inference Rules for Quantifiers

Universal Instantiation can be applied many times to produce many different Consequences.

Existential Instantiation can be applied once, and then the existentially quantified sentence can be discarded.

Inference Rules for Quantifiers

Reduction to propositional inference

our knowledge base contains just the sentences

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

King(John)

Greedy(John)

Brother (Richard, John)

apply UI to the first sentence using all possible ground-term substitutions from the vocabulary of the knowledge base

{x/John} and {x/Richard }.

We obtain

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

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

universally quantified sentence is now discarded.

ground atomic sentences (assuming King(John),… as PL symbols, Propositional Logic)

Use PL inference to obtain conclusion.