Outline

- Why FOL?
- Syntax and semantics of FOL
- Using FOL
- Wumpus world in FOL
- Knowledge engineering in FOL

Last Week: Propositional Logic

- Propositions; Syntax, Semantics

- Entailment: $\alpha \models \beta$ iff $M(\alpha) \subseteq M(\beta)$
- Model checking for Wumpus world

$KB \models g$ iff $M(KB) \subseteq M(g)$

- Inference: $\alpha \vdash_i \beta$ = sentence β can be derived from α (sound?, complete?)
- Resolution: show $KB \wedge \neg \alpha$ unsatisfiable
- Horn clauses; Back/Forward Chaining

Propositional logic: limitations

declarative

partial/disjunctive/negated information

compositional:

context-independent

very limited expressive power

First-order logic

- Whereas propositional logic assumes world contains facts

- Objects
- Relations
- Functions

Syntax of FOL: Basic elements

- Constants KingJohn, 2, NUS,...
- Predicates Brother, >,...
- Functions Sqrt, LeftLegOf,...
- Variables x, y, a, b,...
- Connectives ¬, ⇒, ∧, ∨, ⇔
- Equality =
- Quantifiers ∀, ∃

Atomic sentences

Atomic sentence = $predicate(term_1,...,term_n)$ or

$term_1 = term_2$

Term = $function(term_1,...,term_n)$ or

constant or variable

- E.g.

  Brother(..... *)*

- E.g.

  >(Length(LeftLeg(.....)),

  Length(LeftLeg(.....)))

Complex analysis: Complex numbers

Introduction

model

interpretation

constant symbols

predicate symbols

function symbols

objects

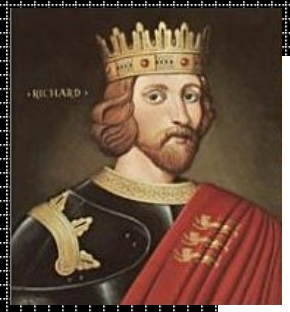
relations

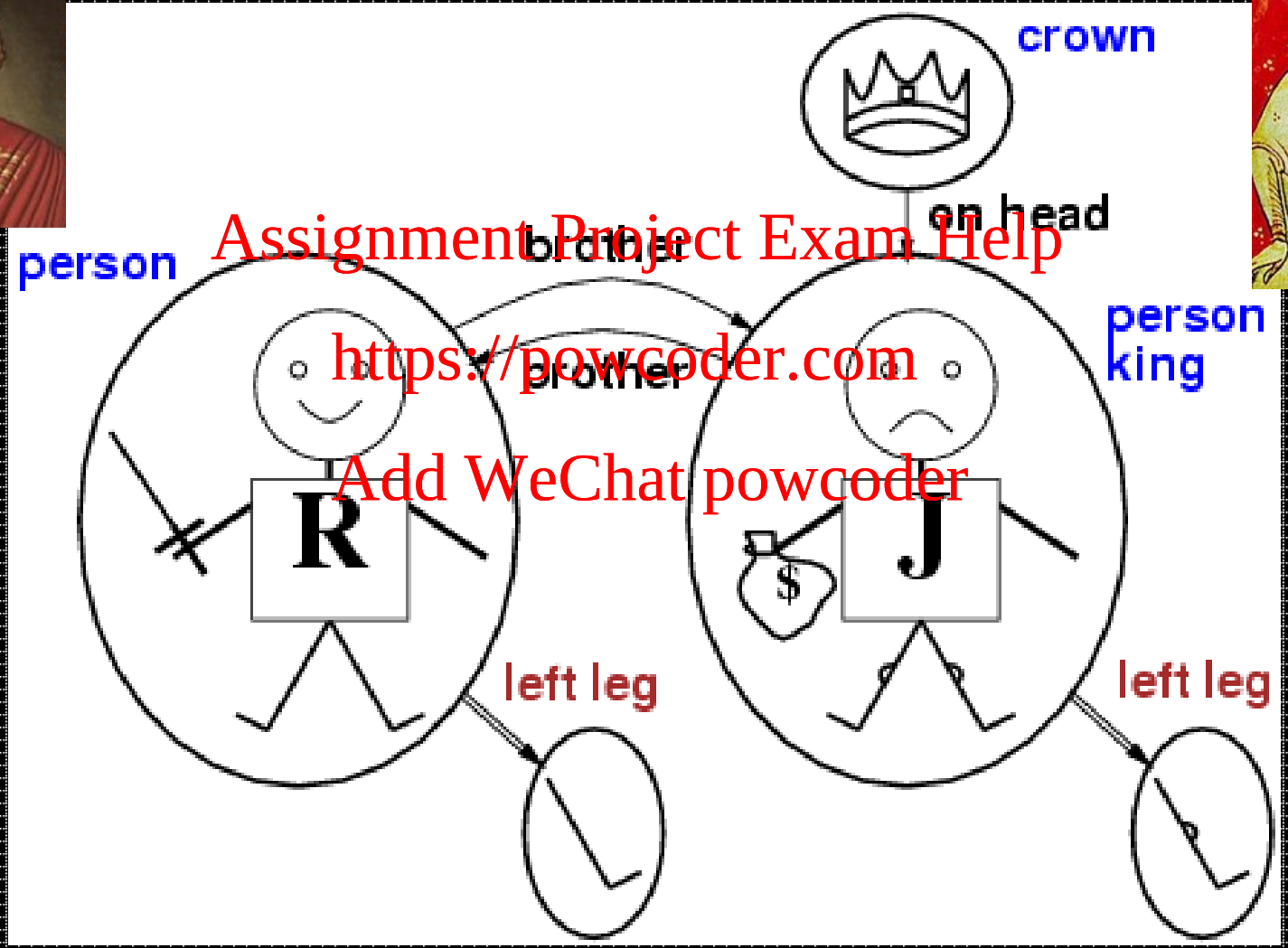
functional relations

objects

relation

Models for Examples





Universal quantification

∀ x at(x, Fordham) ⇒ smart(x)

Everyone at Fordham is smart

∀ x at(x, Fordham) ⇒ smart(x)

conjunction

instantiations

At(KingJohn,Fordham) ⇒ Smart(KingJ

∧ At(Richard, Fordham) ⇒ Smart(Richa

∧ At(Fordham,Fordham) ⇒ Smart(Fordh

∧ ...

Existential instantiation

∨ instantiations

disjunction

At(KingJohn,Fordham) ∧ Smart(KingJohn)
∨ At(Richard, Fordham) ∧ Smart(Richard)
∨ At(Fordham,Fordham) ∧ Smart(Fordham)
∨ ...

Properties of quantifiers

- ∀ ∀x ∀y is the same as ∀y ∀x
- ∀ ∃x ∃y is the same as ∃y ∃x

- ∀ ∃x ∀y is not the same as ∀y ∃x
- ∀ ∃x ∀y Loves(x,y)
  - "There is a person who loves everyone in the world"

- ∀ ∀y ∃x Loves(x,y)
  - "Everyone in the world is loved by at least one person"

- Quantifier duality

∀x Likes(x,IceCream)	¬∃x ¬Likes(x,IceCream)
∃x Likes(x,Broccoli)	¬∀x ¬Likes(x,Broccoli)

Inferring with rules

Tell(KB, Percept([Smell,Breeze,None],5))
Ask(KB, ∃a BestAction(a,5))

substitution

S = Smarter(x,y)
σ = {x/Hillary,y/Bill}
*S*σ = Smarter(Hillary,Bill)

Knowledge in the computer

- Perception
- Reflex

- Diagnostic rule---infer cause from effect
  $\forall s\ Breezy(s) \Rightarrow \exists r\ Adjacent(r,s) \wedge Pit(r)$
- Causal rule---infer effect from cause
  $\forall r\ Pit(r) \Rightarrow [\forall s\ Adjacent(r,s) \Rightarrow Breezy(s)]$

The electronic circuits domain

One-bit full adder