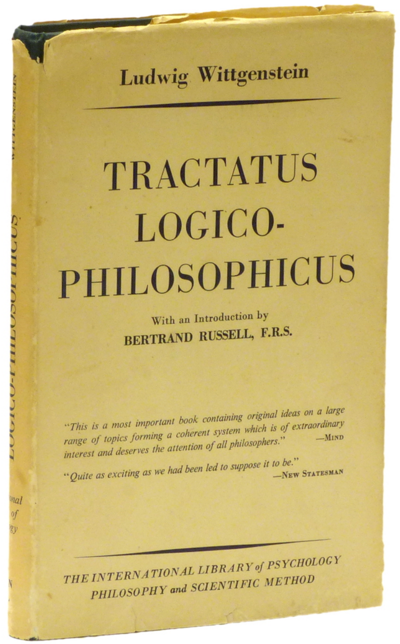
First Order Logic

Philipp Koehn

27 February 2025

The World is Defined by Facts



1. The world is everything that is the case.

2. What is the case (a fact) is the existence of states of affairs.

3. A logical picture of facts is a thought.

4. A thought is a proposition with a sense.

5. A proposition is a truth-function of elementary propositions.
   (An elementary proposition is a truth-function of itself.)

6. The general form of a proposition is the general form of a truth function, which is: $[\bar{p}, \bar{\xi}, N(\bar{\xi})]$. This is the general form of a proposition.

7. Whereof one cannot speak, thereof one must be silent.

Outline

- Why first order logic?
- Syntax and semantics of first order logic
- Fun with sentences
- Wumpus world in first order logic

why?

Pros and Cons of Propositional Logic

- **PRO:** Propositional logic is **declarative**: pieces of syntax correspond to facts
- **PRO:** Propositional logic allows partial/disjunctive/negated information
  (unlike most data structures and databases)
- **PRO:** Propositional logic is **compositional**:
  meaning of $B_{1,1} \wedge P_{1,2}$ is derived from meaning of $B_{1,1}$ and of $P_{1,2}$
- **PRO:** Meaning in propositional logic is **context-independent**
  (unlike natural language, where meaning depends on context)
- **CON:** Propositional logic has very limited expressive power
  (unlike natural language)
  E.g., cannot say "pits cause breezes in adjacent squares"
  except by writing one sentence for each square

First-Order Logic

- Propositional logic: world contains **facts**

- First-order logic: the world contains **objects**, **relations**, and **functions**

- **Objects**: people, houses, numbers, theories, Ronald McDonald, colors, baseball games, wars, centuries . . .

- **Relations**: red, round, bogus, prime, multistoried . . .,
  brother of, bigger than, inside, part of, has color, occurred after, owns, comes between, . . .

- **Functions**: father of, best friend, third inning of, one more than, end of . . .

More Logics

Language	Ontological Commitment	Epistemological Commitment
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
Fuzzy logic	facts + degree of truth	known interval value

Higher-order logic:

relations and functions operate not only on objects,
but also on relations and functions

syntax and semantics

Syntax of FOL: Basic Elements

- Constants: $KingJohn,\ 2,\ UCB, \ldots$
- Predicates: $Brother,\ >, \ldots$
- Functions: $Sqrt,\ LeftLegOf, \ldots$
- Variables: $x,\ y,\ a,\ b, \ldots$
- Connectives: $\wedge \ \vee \ \neg \ \implies \ \Leftrightarrow$
- Equality: $=$
- Quantifiers: $\forall \ \exists$

Atomic Sentences

- Atomic sentence = $predicate(term_1, \ldots, term_n)$
  or $term_1 = term_2$
- Term = $function(term_1, \ldots, term_n)$
  or $constant$ or $variable$
- E.g., $Brother(KingJohn, RichardTheLionheart)$
  $>(Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))$

Complex Sentences

- Complex sentences are made from atomic sentences using connectives

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

- For instance

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

Truth in First-Order Logic

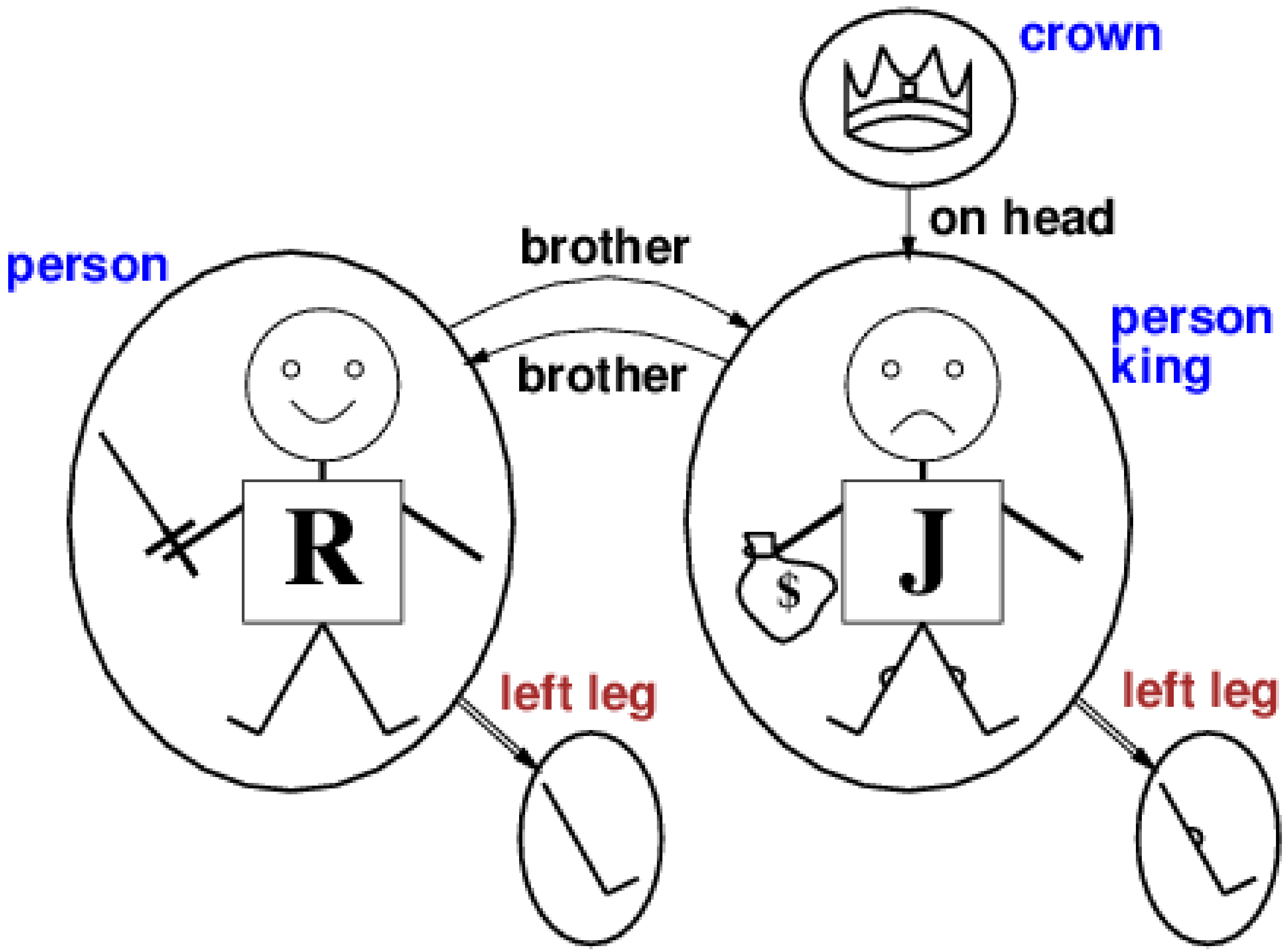
- Sentences are true with respect to a **model** and an **interpretation**

- **Model** contains 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: Example



Truth Example

- Object symbols
  - $Richard$ → **Richard the Lionheart**
  - $John$ → **the evil King John**

- Predicat symbol
  - $Brother$ → **the brotherhood relation**

- Atomic sentence
  - $Brother(Richard, John)$

  true iff **Richard the Lionheart** and **the evil King John**
  are in **the brotherhood relation** in the model

Models for FOL: Lots!

- 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!

Universal Quantification

- Syntax: $\forall \langle variables \rangle \ \langle sentence \rangle$
- Everyone at JHU is smart:
  $\forall \, x \ \ At(x, JHU) \implies Smart(x)$
- $\forall \, x \ \ P$ is true in a model m iff P is true with x being **each** possible object
- **Roughly** speaking, equivalent to the **conjunction** of **instantiations** of P

$$
\begin{aligned}
& (At(KingJohn, JHU) \implies Smart(KingJohn)) \\
\wedge \ & (At(Richard, JHU) \implies Smart(Richard)) \\
\wedge \ & (At(Jane, JHU) \implies Smart(Jane)) \\
\wedge \ & \ldots
\end{aligned}
$$

A Common Mistake to Avoid

- Typically " $\implies$ " is the main connective with $\forall$

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

$$\forall\, x \;\; At(x, JHU) \wedge Smart(x)$$

means "Everyone is at JHU and everyone is smart"

- Correct

$$\forall\, x \;\; At(x, JHU) \implies Smart(x)$$

means "For everyone, if she is at JHU, then she is smart"

Existential Quantification

- Syntax: $\exists \langle variables \rangle \; \langle sentence \rangle$

- Someone at JHU is smart:

  $\exists x \;\; At(x, JHU) \wedge Smart(x)$

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

- **Roughly** speaking, equivalent to the **disjunction** of **instantiations** of P

$$
\begin{array}{ll}
 & (At(KingJohn, JHU) \wedge Smart(KingJohn)) \\
\vee & (At(Richard, JHU) \wedge Smart(Richard)) \\
\vee & (At(JHU, JHU) \wedge Smart(JHU)) \\
\vee & \ldots
\end{array}
$$

Another Common Mistake to Avoid

- Typically "$\wedge$" is the main connective with $\exists$
- Common mistake: using "$\implies$" as the main connective with $\exists$:

$$\exists\, x \;\; At(x, JHU) \implies Smart(x)$$

is true if there is anyone who is **not** at JHU

- Correct

$$\exists\, x \;\; At(x, JHU) \wedge Smart(x)$$

is true if there is someone who is at JHU and smart

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

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

- For instance
  - $1 = 2$ and $\forall\, x \;\; \times(Sqrt(x), Sqrt(x)) = x$ are satisfiable
  - $2 = 2$ is true

  (note: syntax does not imply anything about the semantics of 1, 2, $Sqrt(x)$, etc.)

- Definition of (full) $Sibling$ in terms of $Parent$

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

fun with sentences

Fun with Sentences

- Brothers are siblings

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

- "Sibling" is symmetric

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

- One's mother is one's female parent

  $\forall\, x, y\;\; Mother(x, y) \Leftrightarrow (Female(x) \wedge Parent(x, y))$

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

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

Lincoln Quote

You can fool all the people some of the time,
and some of the people all the time,
but you cannot fool all the people all the time.

$$\forall\, p\ \exists\, t\ \ Fool(p,t)$$
$$\wedge$$
$$\exists\, p\ \forall\, t\ \ Fool(p,t)$$
$$\wedge$$
$$\neg\ \forall\, p\ \forall\, t\ \ Fool(p,t)$$

Donkey Sentences

- *Every farmer owns a donkey.*
  - $\forall\, f\ \ (Farmer(f) \implies \exists\, d\ \ (Donkey(d) \wedge Own(f,d)))$
  - $\exists\, d\ \ (Donkey(d) \wedge \forall\, f\ \ (Farmer(f) \wedge Own(f,d)))$

- *Every human lives on a planet.*
  - $\exists\, p\ \ (Planet(p) \wedge \forall\, h\ \ (Human(h) \wedge LivesOn(h,p)))$

- *Every farmer who owns a donkey feeds it.*
  - $\forall\, f\ \ Farmer(f) \wedge \exists\, d\ \ (Donkey(d) \wedge Own(f,d) \implies Feeds(f,d))$
    but what if a farmer has a donkey d_1 and a pig d_2 and he feeds neither $Donkey(d_2) \wedge Own(f,d_2) \implies Feeds(f,d_2)$ is true ($false \wedge true \implies false$)
  - $\forall\, f\ \ \forall\, d\ \ (Farmer(f) \wedge Donkey(d) \wedge Own(f,d) \implies Feeds(f,d))$
    but this means "Every farmer feeds every donkey he owns."

Natural Language

- First order logic is close to the semantics of natural language
- But there are limitations
  - *"There is at least one thing John has in common with Peter."*
    Requires a quantifier over predicates.
  - *"The cake is very good."*
    $\exists\, c\;\; Cake(c) \land Good(c)$ but not $Very(c)$
    Functions and relations cannot be qualified.
- Natural language sentences are often intentionally vague and ambiguous

wampus world

Knowledge Base for the Wumpus World

- **"Perception"**: at current square, three perceptions expressed as variables
  - s either $Smell$ or $\neg Smell$
  - b either $Breeze$ or $\neg Breeze$
  - g either $Glitter$ or $\neg Glitter$

  $Percept([s, b, g], t)$ at time t

- Shorthands
  - $\forall\, b, g, t \;\; Percept([Smell, b, g], t) \implies Smelt(t)$
  - $\forall\, s, b, t \;\; Percept([s, b, Glitter], t) \implies AtGold(t)$

- **Reflex**: $\forall\, t \;\; AtGold(t) \implies Action(Grab, t)$

- **Reflex with internal state**: do we have the gold already?
  $\forall\, t \;\; AtGold(t) \wedge \neg Holding(Gold, t) \implies Action(Grab, t)$

- $Holding(Gold, t)$ cannot be observed
  $\Rightarrow$ keeping track of change is essential

Deducing Hidden Properties

- Properties of locations:
  $\forall\, x,t \;\; At(Agent,x,t) \land Smelt(t) \implies Smelly(x)$
  $\forall\, x,t \;\; At(Agent,x,t) \land Breeze(t) \implies Breezy(x)$

- Squares are breezy near a pit

- **Diagnostic** rule—infer cause from effect
  $$\forall\, y \;\; Breezy(y) \implies \exists\, x \;\; Pit(x) \land Adjacent(x,y)$$

- **Causal** rule—infer effect from cause
  $$\forall\, x,y \;\; Pit(x) \land Adjacent(x,y) \implies Breezy(y)$$

- Neither of these is complete—e.g., the causal rule doesn't say whether squares far away from pits can be breezy

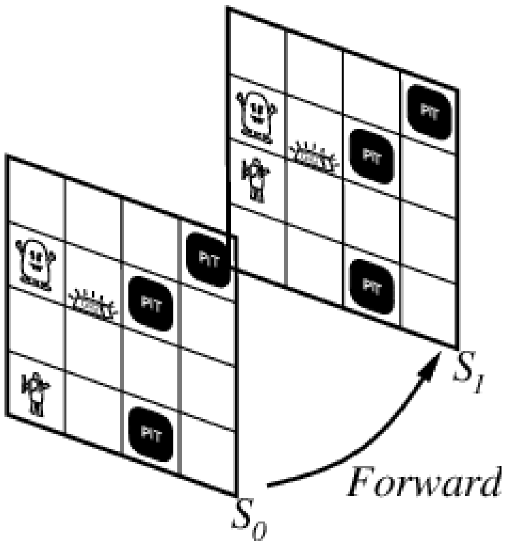
- **Definition** for the $Breezy$ predicate:
  $$\forall\, y \;\; Breezy(y) \Leftrightarrow [\exists\, x \;\; Pit(x) \land Adjacent(x,y)]$$

States and Fluents

- By acting, the agent moves through a sequence of situations s
- Fluents: aspects of the world that may change
  - current position
  - having an arrow
  - holding the gold
- Taking actions requires updates to the fluents

Keeping Track of Change

- Facts hold in **situations**, rather than eternally
  E.g., $Holding(Gold, Now)$ rather than just $Holding(Gold)$
- **Situation calculus** is one way to represent change in FOL:
  Adds a situation argument to each non-eternal predicate
  E.g., Now in $Holding(Gold, Now)$ denotes a situation
- Situations are connected by the $Result$ function
  $s' = Result(a, s)$ is the situation that results from doing a in s



Describing Actions

- "Effect" axiom—describe changes due to action
  $\forall\, s \;\; AtGold(s) \implies Holding(Gold, Result(Grab, s))$

- "Frame" axiom—describe **non-changes** due to action
  $\forall\, s \;\; HaveArrow(s) \implies HaveArrow(Result(Grab, s))$

- **Frame problem**: find an elegant way to handle non-change
  (a) **representation**: too many frame axioms
  (b) **inference**: too many repeated "copy-overs" to keep track of state

- **Qualification problem**: true descriptions of real actions require endless caveats—what if gold is slippery or nailed down or ...

- **Ramification problem**: real actions have many secondary consequences—what about the dust on the gold, wear and tear on gloves, ...

Describing Actions

- **Successor-state axioms** solve the representational frame problem

- Each axiom is "about" a **predicate** (not an action per se):

$$\text{P true afterwards} \quad \Leftrightarrow \quad [\text{an action made P true}$$
$$\vee \quad \text{P true already and no action made P false}]$$

- For holding the gold:

$$\forall\, a, s \;\; Holding(Gold, Result(a, s)) \;\Leftrightarrow$$
$$[(a = Grab \wedge AtGold(s))$$
$$\vee\, (Holding(Gold, s) \wedge a \neq Release)]$$

Making Plans

- Initial condition in KB:
  $At(Agent, [1,1], S_0)$
  $At(Gold, [1,2], S_0)$

- Query: $Ask(KB, \exists\, s\;\; Holding(Gold, s))$
  i.e., in what situation will I be holding the gold?

- Answer: $\{s/Result(Grab, Result(Forward, S_0))\}$
  i.e., go forward and then grab the gold

- This assumes that the agent is interested in plans starting at S_0 and that S_0 is the only situation described in the KB

Making Plans: A Better Way

- Represent **plans** as action sequences $[a_1, a_2, \ldots, a_n]$
- $PlanResult(p, s)$ is the result of executing p in s
- Then the query $Ask(KB, \exists\, p \;\; Holding(Gold, PlanResult(p, S_0)))$
  has the solution $\{p/[Forward, Grab]\}$
- Definition of $PlanResult$ in terms of $Result$:
  $\forall\, s \;\; PlanResult([\,], s) = s$
  $\forall\, a, p, s \;\; PlanResult([a|p], s) = PlanResult(p, Result(a, s))$
- **Planning systems** are special-purpose reasoners designed to do this type of inference more efficiently than a general-purpose reasoner

Summary

- First-order logic
  - objects and relations are semantic primitives
  - syntax: constants, functions, predicates, equality, quantifiers
- Increased expressive power: sufficient to define wumpus world
- Situation calculus
  - conventions for describing actions and change in FOL
  - can formulate planning as inference on a situation calculus KB