Chapter 8

First-Order Logic

CS361 Artificial Intelligence

Dr. Khaled Wassif

Spring 2017

(This is the instructor's notes and student has to read the textbook for complete material.)

Chapter Outline

- More about Representation Languages
- Syntax and Semantics of First-Order Logic
  - Constants and Variables
  - Functions and Predicates
  - Equality and Quantifiers
  - Nested Quantifiers
  - Interpretation
- Translate English Sentences
- Using First-Order Logic

Pros and Cons of Propositional Logic

- **Pros:**
  - A declarative language that allow to represent partial, conjunctive, disjunctive, and negated knowledge.
  - Meaning is context-independent – unlike natural language.
  - Has a sound, complete inference procedures.
- **Cons:**
  - Very limited expressive power – unlike natural language.
  - Lack of variables prevents stating more general rules.
  - Changing of the knowledge base over time is difficult to represent.

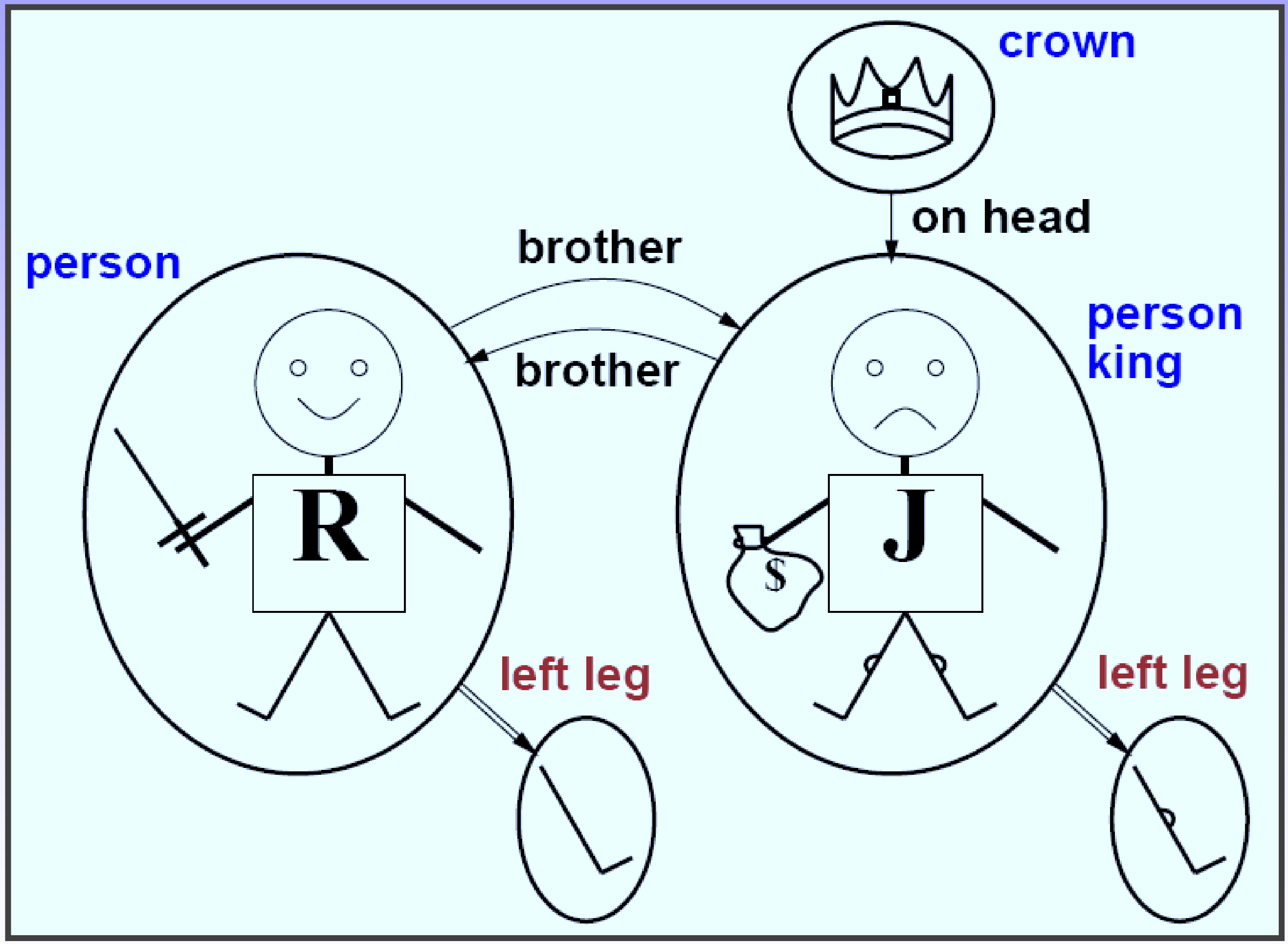
First-Order Logic (FOL)

- Also known as Predicate Logic or Predicate Calculus.
- Like the propositional logic:
  - A declarative language and its semantics is based on a truth relation between sentences and possible worlds.
  - A sentence represents a fact and the agent either believes it to be true, believes it to be false, or has no opinion.
- Much more powerful
  - Greater expressive power than propositional logic:
    - Can represent general laws or rules – no longer need a separate rule for each square to say which squares are breezy/pits.
    - Can also express facts about *some* or *all* of the objects in the universe.
  - Allows objects with certain relations among them:
    - In programming terms, allows classes, functions and variables.

First-Order Logic (FOL)

- Like natural language assumes the world contains:
  - **Objects**
    - » Generally correspond to English nouns
    - » E.g. people, houses, numbers, theories, colors, wars, centuries, …
  - **Relations** or **Predicates**
    - » Generally correspond to English verbs
    - » Associate or connect between objects.
    - » E.g. like, study, brother of, bigger than, inside, part of, has color, occurred after, owns, comes between, …
    - » Can be unary relations or **properties**, E.g. red, round, prime, …
  - **Functions**
    - » Arguments of each are objects; the return of each is only one object.
    - » E.g. father of, best friend, max of, square of, third quarter of, one more than, beginning of, …

First-Order Logic Example



Object
Relation
Function

Syntax of First-Order Logic

$$
\begin{array}{rcl}
Sentence & \rightarrow & AtomicSentence \mid ComplexSentence \\
AtomicSentence & \rightarrow & Predicate \mid Predicate(Term, \ldots) \mid Term = Term \\
ComplexSentence & \rightarrow & (\ Sentence\ ) \mid [\ Sentence\ ] \\
& \mid & \neg\ Sentence \\
& \mid & Sentence \land Sentence \\
& \mid & Sentence \lor Sentence \\
& \mid & Sentence \ \Rightarrow\ Sentence \\
& \mid & Sentence \ \Leftrightarrow\ Sentence \\
& \mid & Quantifier\ Variable, \ldots\ Sentence
\end{array}
$$

Syntax of First-Order Logic

$$
\begin{array}{rcl}
Term & \rightarrow & Function(Term, \ldots) \\
 & | & Constant \\
 & | & Variable \\
 & & \\
Quantifier & \rightarrow & \forall \mid \exists \\
Constant & \rightarrow & A \mid X_1 \mid John \mid \cdots \\
Variable & \rightarrow & a \mid x \mid s \mid \cdots \\
Predicate & \rightarrow & True \mid False \mid After \mid Loves \mid Raining \mid \cdots \\
Function & \rightarrow & Mother \mid LeftLeg \mid \cdots
\end{array}
$$

Constants, Functions and Predicates

- Basic syntax elements are the three kinds of symbols:
  - **Constant** symbols
    - » Represent objects in the world.
    - » E.g., Ahmed, KingJohn, Earth, AI, Computer, Green, 2, 3.4, …
  - **Functions** symbols
    - » Stand for functions (maps a tuple of objects to an object)
    - » E.g., Sqrt(3), LeftArmOf(KingJohn), Max(12, 25), Add(3, 6, 2), …
  - **Predicate** symbols
    - » Stand for relations (maps a tuple of objects to a truth-value)
    - » E.g., Brother(Richard, John), Greater_than(3, 2), Study(Ahmed, AI), Grade(Mona, DB, B+), ...
- All these symbols will begin with uppercase letters.
- Each function and predicate symbol comes with an **arity** that fixes the number of its arguments.

Constants, Functions and Predicates

- Every model consists of a set of objects and an **interpretation** to determine if any given sentence is true or false..
  - Each interpretation maps constant symbols to objects, predicate symbols to relations on those objects, and function symbols to functions on those objects.
- Entailment and validity; like propositional logic, are defined in terms of ***all possible models***.
  - Different models vary in how many objects they contain (from one up to infinity) and in the way the constant symbols map to objects.
- Because number of possible models is unbounded, checking entailment by enumeration of all possible models is not feasible – unlike propositional logic.

Components of First-Order Logic

- **Term**
  - A logical expression that refers to an object (real individual).
  - Can be a constant symbol, a variable symbol, or an *n*-place function of *n* terms:
    - » Constant, e.g. Red – Variable, e.g. person – Function of constant, e.g. Color(Block1) – Function of variables, e.g. Fun(x_1, x_2, x_3)
  - A term with no variables is called a **ground term**.
- **Atomic Sentence** – state a fact
  - An *n*-place predicate of *n* ground terms – without variables.
    - » Brother (John, Richard)
    - » Married (Mother(John), Father(John))
- **Complex Sentence**
  - Atomic sentences + logical connectives ($\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$)
    - » Brother (John, Richard) $\wedge$ $\neg$Brother (John, Father(John))

Components of First-Order Logic

- **Quantifiers**
  - Each quantifier defines a variable for the duration of the followed expression to indicate the truth of that expression.
  - **Universal quantifier** (∀) "for all"
    - » The expression is true for every possible value of the variable.
    - » $\forall x$ P(x) means that P is true for **all** values of x in the domain associated with that variable.
    - » $\forall x$ King(x) ⇒ Person(x) "For all x, if x is a king, then x is a person"
    - » The ⇒ with ∀ are perfect for writing general rules.
  - **Existential quantifier** (∃) "there exists"
    - » The expression is true for at least one value of the variable.
    - » $\exists x$ P(x) means that P is true for **some** value of x in the domain associated with that variable.
    - » $\exists x$ Crown(x) ∧ OnHead(x, John) "King John has a crown on his head"
    - » Allow to make a sentence about some object without naming it.

Components of First-Order Logic

- **Nested quantifiers**
  - Switching the order of the same quantifiers *does not* change the meaning:
    - » $\forall x\ \forall y\ P(x, y) \equiv \forall y\ \forall x\ P(x, y)$ and also $\exists x\ \exists y\ P(x, y) \equiv \exists y\ \exists x\ P(x, y)$
    - » $\forall x\ \forall y\ \text{Brother}(x, y) \Rightarrow \text{Sibling}(x, y)$
    - » $\forall x, y\ \text{Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x)$
  - Switching the order of different quantifiers *does* change meaning:
    - » Everyone likes some kind of food: $\forall x\ \exists y\ \text{food}(\text{x}) \wedge \text{likes}(x, y)$
    - » There is a kind of food that everyone likes: $\exists y\ \forall x\ \text{food}(\text{x}) \wedge \text{likes}(x, y)$
  - Always use different variable names with nested quantifiers:
    - » $\forall x\ (\text{Crown}(x) \vee (\exists x\ \text{Brother}(\text{Richard}, x)))$ make confusion.

Components of First-Order Logic

- **Connections between quantifiers**
  - The two quantifiers are actually closely connected with each other, through negation. :
    - » $\forall x\ \neg Likes(x, Sadness)$ is equivalent to $\neg\exists x\ Likes(x, Sadness)$
    - » $\forall x\ Likes(x, IceCream)$ is equivalent to $\neg\exists x\ \neg Likes(x, IceCream)$
    - » Because $\forall$ is really a conjunction over the universe of objects and $\exists$ is a disjunction.
  - Therefore, the De Morgan rules can be applied over the two quantifiers as follow:
    - » $\forall x\ \neg P \equiv \neg\exists x\ P$
    - » $\neg\forall x\ P \equiv \exists x\ \neg P$
    - » $\forall x\ P \equiv \neg\exists x\ \neg P$
    - » $\neg\forall x\ \neg P \equiv \exists x\ P$

Syntactic “sugar”: we really need one quantifier only

Components of First-Order Logic

- **Equality**
  - Another way to make atomic sentences, by using the **equality symbol** to denote that two terms that refer to the same object.
  - Can be used to state facts about a given function:
    - » Father(John) = Henry
  - Can also be used with negation to insist that two terms are not the same object:
    - » $\exists x, y$ Brother(x, Richard) $\land$ Brother(y, Richard) $\land \neg(x = y)$
    - » $\exists x, y$ Brother(x, Richard) $\land$ Brother(y, Richard) – does not have the intended meaning.
  - The notation $x \neq y$ is sometimes used as an abbreviation for $\neg(x = y)$.

First-Order Logic: Examples

- Farah is either a surgeon or a lawyer.

  Surgeon(Farah) $\vee$ Lower(Farah)

- All surgeons are doctors.

  $\forall x$ Surgeon(x) $\Rightarrow$ Doctor(x)

- Some doctors are also lawyers

  $\exists x$ Doctor(x) $\wedge$ Lawyer(x)

- Every person who loves one of his brothers is happy.

  $\forall x$ Person(x) $\wedge$ ($\exists y$ Brother(x, y) $\wedge$ Loves(x, y)) $\Rightarrow$ Happy(x)

- One's mother is one's female parent.

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

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

  $\forall x$, y Cousin(x, y) $\Leftrightarrow$ ($\exists p$, s Parent(p, x) ^ Sibling(s, p) ^ Parent(s, y))

Translate English Sentences to FOL

- **“All persons are mortal.”**

  [Use: Person(x), Mortal (x)]

  – $\forall x\ Person(x) \Rightarrow Mortal(x)$
  – $\forall x\ \neg Person(x) \lor Mortal(x)$

- **Common Mistakes:**
  – $\forall x\ Person(x) \land Mortal(x)$

Translate English Sentences to FOL

- **"Fifi has a sister who is a cat."**

  [Use: Sister(Fifi, x), Cat(x)]

  – $\exists x$ Sister(Fifi, x) $\wedge$ Cat(x)
- **Common Mistakes:**
  – $\exists x$ Sister(Fifi, x) $\Rightarrow$ Cat(x)

Translate English Sentences to FOL

- **"For every food, there is a person who eats that food."**

  [Use: Food(x), Person(y), Eats(y, x)]

  - $\forall x\ \exists y\ Food(x) \Rightarrow [\ Person(y) \wedge Eats(y, x)\ ]$
  - $\forall x\ Food(x) \Rightarrow \exists y\ [\ Person(y) \wedge Eats(y, x)\ ]$
  - $\forall x\ \exists y\ \neg Food(x) \vee [\ Person(y) \wedge Eats(y, x)\ ]$
  - $\forall x\ \exists y\ [\ \neg Food(x) \vee Person(y)\ ] \wedge [\neg Food(x) \vee Eats(y, x)\ ]$
  - $\forall x\ \exists y\ [\ Food(x) \Rightarrow Person(y)\ ] \wedge [\ Food(x) \Rightarrow Eats(y, x)\ ]$

- **Common Mistakes:**
  - $\forall x\ \exists y\ [\ Food(x) \wedge Person(y)\ ] \Rightarrow Eats(y, x)$
  - $\forall x\ \exists y\ Food(x) \wedge Person(y) \wedge Eats(y, x)$

Translate English Sentences to FOL

- **"Every person eats every food."**

  [Use: Person (x), Food (y), Eats(x, y)]

  – $\forall x\ \forall y\ [\ Person(x) \wedge Food(y)\ ] \Rightarrow Eats(x, y)$
  – $\forall x\ \forall y\ \neg Person(x) \vee \neg Food(y) \vee Eats(x, y)$
  – $\forall x\ \forall y\ Person(x) \Rightarrow [\ Food(y) \Rightarrow Eats(x, y)\ ]$
  – $\forall x\ \forall y\ Person(x) \Rightarrow [\ \neg Food(y) \vee Eats(x, y)\ ]$
  – $\forall x\ \forall y\ \neg Person(x) \vee [\ Food(y) \Rightarrow Eats(x, y)\ ]$

- **Common Mistakes:**
  – $\forall x\ \forall y\ Person(x) \Rightarrow [Food(y) \wedge Eats(x, y)\ ]$
  – $\forall x\ \forall y\ Person(x) \wedge Food(y) \wedge Eats(x, y)$

Translate English Sentences to FOL

- **"All greedy kings are evil."**

  [Use: King(x), Greedy(x), Evil(x)]

  - $\forall x\ [\ Greedy(x) \wedge King(x)\ ] \Rightarrow Evil(x)$
  - $\forall x\ \neg Greedy(x) \vee \neg King(x) \vee Evil(x)$
  - $\forall x\ Greedy(x) \Rightarrow [\ King(x) \Rightarrow Evil(x)\ ]$
- **Common Mistakes:**
  - $\forall x\ Greedy(x) \wedge King(x) \wedge Evil(x)$

Translate English Sentences to FOL

- **"Everyone has a favorite food."**

  [Use: Person(x), Food(y), Favorite(y, x)]

  – $\forall x\ \exists y\ Person(x) \Rightarrow [\ Food(y) \wedge Favorite(y, x)\ ]$

  – $\forall x\ Person(x) \Rightarrow \exists y\ [\ Food(y) \wedge Favorite(y, x)\ ]$

  – $\forall x\ \exists y\ \neg Person(x) \vee [\ Food(y) \wedge Favorite(y, x)\ ]$

  – $\forall x\ \exists y\ [\neg Person(x) \vee Food(y)] \wedge [\neg Person(x) \vee Favorite(y, x)]$

  – $\forall x\ \exists y\ [Person(x) \Rightarrow Food(y)\ ] \wedge [Person(x) \Rightarrow Favorite(y, x)]$

- **Common Mistakes:**

  – $\forall x\ \exists y\ [\ Person(x) \wedge Food(y)\ ] \Rightarrow Favorite(y, x)$

  – $\forall x\ \exists y\ Person(x) \wedge Food(y) \wedge Favorite(y, x)$

Translate English Sentences to FOL

- **"There is someone at FCI who is smart."**

  [Use: Person(x), At(x, FCI), Smart(x)]

  – $\exists x\ Person(x) \wedge At(x, FCI) \wedge Smart(x)$
- **Common Mistakes:**
  – $\exists x\ [\ Person(x) \wedge At(x, FCI)\ ] \Rightarrow Smart(x)$

Translate English Sentences to FOL

- **"Everyone at FCI is smart."**

  [Use: Person(x), At(x, FCI), Smart(x)]

  - $\forall x$ [Person(x) $\wedge$ At(x, FCI)] $\Rightarrow$ Smart(x)
  - $\forall x$ $\neg$[Person(x) $\wedge$ At(x, FCI)] $\vee$ Smart(x)
  - $\forall x$ $\neg$Person(x) $\vee$ $\neg$At(x, FCI) $\vee$ Smart(x)
- **Common Mistakes:**
  - $\forall x$ Person(x) $\wedge$ At(x, FCI) $\wedge$ Smart(x)
  - $\forall x$ Person(x) $\Rightarrow$ [At(x, FCI) $\wedge$ Smart(x)]

Translate English Sentences to FOL

- **"Every person eats some food."**

  [Use: Person (x), Food (y), Eats(x, y)]

  – $\forall x\ \exists y\ Person(x) \Rightarrow [\ Food(y) \wedge Eats(x, y)\ ]$

  – $\forall x\ Person(x) \Rightarrow \exists y\ [\ Food(y) \wedge Eats(x, y)\ ]$

  – $\forall x\ \exists y\ \neg Person(x) \vee [\ Food(y) \wedge Eats(x, y)\ ]$

  – $\forall x\ \exists y\ [\neg Person(x) \vee Food(y)] \wedge [\neg Person(x) \vee Eats(x, y)]$

- **Common Mistakes:**

  – $\forall x\ \exists y\ [\ Person(x) \wedge Food(y)\ ] \Rightarrow Eats(x, y)$

  – $\forall x\ \exists y\ Person(x) \wedge Food(y) \wedge Eats(x, y)$

Translate English Sentences to FOL

- **"Some person eats some food."**

  [Use: Person (x), Food (y), Eats(x, y)]

  – $\exists x\ \exists y\ Person(x) \wedge Food(y) \wedge Eats(x, y)$
- **Common Mistakes:**
  – $\exists x\ \exists y\ [\ Person(x) \wedge Food(y)\ ] \Rightarrow Eats(x, y)$

Using of First-Order Logic

- Interacting with first-order knowledge bases:
  - Sentences (called **assertions**) are added to a knowledge base using **TELL**:
    - » Facts:
      - Tell(*KB*, King(John))
      - Tell(*KB*, Person(Richard))
    - » Rules:
      - Tell(*KB*, $\forall x$ King(x) $\Rightarrow$ Person(x))
      - Tell(*KB*, $\forall x$ Person(x) $\Rightarrow$ Likes(x, McDonalds))
  - Can ask questions (queries) of the knowledge base using **ASK**:
    - » Ask(*KB*, King(John))
    - » Ask(*KB*, Person(John))
    - » Ask(*KB*, Likes(John, McDonalds))
    - » Ask(*KB*, $\exists x$ Likes(x, McDonalds))

Using of First-Order Logic

- Types of Answers:
  - Fact is in the *KB*
    - » Ask(*KB*, King(John))
    - » Yes.
  - Fact is not in the *KB*
    - » Ask(*KB*, Person(John))
    - » Yes (if it can be proven by a rule from the *KB*)
    - » No (otherwise)
  - Fact contains variables
    - » Ask(*KB*, ∃*x* Likes(*x*, McDonalds))
    - » **Substitution** or **binding list** for which the fact can be proven, e.g. {*x*/John}{*x*/Richard} …

Using of First-Order Logic

- **Example domains**
  - Kinship (family relationships) domain
    - » Include facts such as mother and father, and rules such parent.
  - Numbers, sets, and lists
    - » Describe the natural numbers and mathematical operations.
    - » Able to represent sets and order sets (lists) with their operations.
  - The Wumpus World
    - » Unlike propositional logic; the first-order description for the wumpus world is much more concise, capturing in a natural way.
- Whatever your domain:
  - If **axioms** correctly and completely describe how world works.
  - Any complete logical **inference procedure** will infer strongest possible description of the world, given available **percepts**.

Knowledge Base for Wumpus World

- The sentences stored in the knowledge base must include both the percept and time at which it occurred.
- The wumpus agent receives a percept vector with five elements and an integer for time steps:

  Percept([Stench, Breeze, Glitter, Bump, Scream], *t*)

- Suppose the agent perceive a smell and breeze, but no glitter at $t = 5$:
  - Tell(*KB*, Percept([Stench, Breeze, None , None , None], 5))
  - Ask(*KB*, ∃*a* BestAction(*a*, 5))
    - » i.e. does the *KB* entail any particular action at $t = 5$?
  - Answer: Yes, {*a*/Shoot} – substitution (binding list)

Knowledge Base for Wumpus World

- Represent perception (input)
  - Raw percept data implies certain facts about the current state.
  - For example:

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

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

……

- Represent simple reflex (output)

$$\forall t\ \text{Glitter}(t) \Rightarrow \text{BestAction}(\text{Grab}, t)$$

- Represent the environment itself
  - Adjacent squares

$$\forall x, y, a, b\ \text{Adjacent}([x, y],[a, b]) \Leftrightarrow$$
$$(x = a \wedge (y = b - 1 \vee y = b + 1)) \vee (y = b \wedge (x = a - 1 \vee x = a + 1))$$

Knowledge Base for Wumpus World

- The agent's location changes over time:
  - Use At(Agent, *s*, *t*) to mean that the agent is at square *s* at time *t*.
- Properties of locations:

$$\forall s, t\ \mathrm{At}(\mathrm{Agent}, s, t) \wedge \mathrm{Smelt}(t) \Rightarrow \mathrm{Smelly}(s)$$

$$\forall s, t\ \mathrm{At}(\mathrm{Agent}, s, t) \wedge \mathrm{Breeze}(t) \Rightarrow \mathrm{Breezy}(s)$$

- Squares are breezy near a pit:

$$\forall s\ \mathrm{Breezy}(s) \Rightarrow \exists r\ \mathrm{Adjacent}(r, s) \wedge \mathrm{Pit}(r)$$

- From these example sentences, we can see that the first-order logic formulation is no less concise than the original English description of the Wumpus world.

SUMMARY

- First-order logic:
  - Much more expressive than propositional logic
  - Allows objects and relations as semantic primitives
  - Universal and existential quantifiers
- Syntax: constants, variables, functions, predicates, equality, quantifiers
- Nested quantifiers
  - Order of unlike quantifiers matters (the outer scopes the inner)
    - » Like nested ANDs and ORs
  - Order of like quantifiers does not matter
    - » like nested ANDS and ANDs
- Translate simple English sentences to FOL and back