First-Order Logic

CPSC 470 – Artificial Intelligence
Brian Scassellati
Propositional Logic Syntax

Sentence $\rightarrow$ AtomicSentence $|$ ComplexSentence

AtomicSentence $\rightarrow$ True $|$ False $|$ $P$ $|$ $Q$ $|$ ...

ComplexSentence $\rightarrow$ (Sentence) $|$ Sentence Connective Sentence $|$ $\neg$Sentence

Connective $\rightarrow$ $\wedge$ $|$ $\lor$ $|$ $\Rightarrow$ $|$ $\Leftrightarrow$
An Agent for the Wumpus World

- Convert perceptions into sentences:
  
  “In square [1,1], there is no breeze and no stench” … becomes…

  \[ \neg B_{11} \land \neg S_{11} \]

- Start with some knowledge of the world (in the form of rules)

  R1 : \( \neg S_{11} \Rightarrow \neg W_{11} \land \neg W_{12} \land \neg W_{21} \)

  R2 : \( \neg S_{21} \Rightarrow \neg W_{11} \land \neg W_{21} \land \neg W_{22} \)

  ....

  R4 : \( S_{12} \Rightarrow W_{13} \lor W_{12} \lor W_{22} \lor W_{11} \)
Finding the Wumpus

1. Apply modus ponens and and-elimination to \( \neg S_{11} \Rightarrow \neg W_{11} \land \neg W_{12} \land \neg W_{21} \) to get:
   \[
   \neg W_{11}, \ \neg W_{12}, \ \neg W_{21}
   \]

2. Apply modus ponens and and-elimination to \( \neg S_{21} \Rightarrow \neg W_{11} \land \neg W_{21} \land \neg W_{22} \) to get:
   \[
   \neg W_{22}, \ \neg W_{21}, \ \neg W_{31}
   \]

3. Apply modus ponens to
   \[
   S_{12} \Rightarrow W_{13} \lor W_{12} \lor W_{22} \lor W_{11}
   \]
   to get:
   \[
   W_{13} \lor W_{12} \lor W_{22} \lor W_{11}
   \]

4. Apply unit resolution to #3 and #1
   \[
   W_{13} \lor W_{22}
   \]

5. Apply unit resolution to #4 and #2
   \[
   W_{13}
   \]

The wumpus is in square [1,3]!!!
Problems with Propositional Logic

• Too many propositions!
  – How can you encode a rule such as “don’t go forward if the wumpus is in front of you”?
  – In propositional logic, this takes (16 squares * 4 orientations) = 64 rules!

• Truth tables become unwieldy quickly
  – Size of the truth table is $2^n$ where $n$ is the number of propositional symbols
More Problems with Propositional Logic

• No good way to represent changes in the world
  – How do you encode the location of the agent?
• What kinds of practical applications is this good for?
  – Relatively little
First-Order Logic

• Also known as First-Order Predicate Calculus (FOPC)
• Most studied form of knowledge representation
• Ontological commitments
  – World is composed of objects and properties
• Expressions will include both
  – Sentences: represent facts
  – Terms: represent objects
Syntax and Semantics of First-Order Logic

- **Constant Symbols** \((A, B, John, \ldots)\)
  - A symbol names exactly one object
  - But each object might have multiple names (and some objects might not have a name)

- **Predicate Symbols** \((Round, Brother, \ldots)\)
  - Defined by a set of tuples of objects that satisfy the predicate

- **Function Symbols** \((Cosine, FatherOf, \ldots)\)
  - A relation in which any given object is related to exactly one other object by this relation
  - Uniquely determines an object (without giving it a name)
Syntax and Semantics of First-Order Logic

- **Variables** \((a, b, c, \ldots)\)
  - Stand for an object (without naming it)

- **Terms** \((\text{John}, \text{FatherOf(John)}, a, \ldots)\)
  - A logical expression that refers to an object
  - Can be a constant or a variable
  - Can be a function of a list of terms
Syntax and Semantics of First-Order Logic

Sentence → AtomicSentence
    | Sentence Connective Sentence
    | Quantifier Variable,...Sentence
    | ¬Sentence
    | (Sentence)

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

Term → Function(Term,...)
    | Constant
    | Variable

Connective → ⇒ | ∧ | ∨ | ⇔
Quantifier → ∀ | ∃
Variable → a | b | c |...
Function → Mother | LeftLegOf |...
Predicate → Before | HasColor | Raining |...
Constant → A | X₁ | John |...

• Atomic Sentences via Predicates
  – Examples:
    • Round(Coconut)
    • Brother(Cain, Abel)
    • Older(John, 35)
    • Square(Baseball)
  – Assertions that represent a fact about the world
  – Again, can be true or false given the state of the world
Syntax and Semantics of First-Order Logic

Sentence $\rightarrow$ AtomicSentence
  $|$ Sentence Connective Sentence
  $|$ Quantifier Variable,...Sentence
  $|$ $\neg$Sentence
  $|$ $(Sentence)$

AtomicSentence $\rightarrow$ Predicate(Term,...)
  $|$ Term $=$ Term

Term $\rightarrow$ Function(Term,...)
  $|$ Constant
  $|$ Variable

Connective $\rightarrow$ $\Rightarrow$ $|$ $\land$ $|$ $\lor$ $|$ $\Leftrightarrow$

Quantifier $\rightarrow$ $\forall$ $|$ $\exists$

Variable $\rightarrow$ $a$ $|$ $b$ $|$ $c$ $|$ ...

Function $\rightarrow$ Mother $|$ LeftLegOf $|$ ...

Predicate $\rightarrow$ Before $|$ HasColor $|$ Raining $|$ ...

Constant $\rightarrow$ $A$ $|$ $X_1$ $|$ John $|$ ...

• Atomic Sentences via Equality
  – Examples
    • Father(John)=Henry
    • 1=Cosine(pi)
    • Three=Two

  – Asserts that the two terms refer to the same real-world object
Syntax and Semantics of First-Order Logic

- **Connectives** ($\Rightarrow \land \lor \equiv$)
  - Work the same way as in predicate calculus

- **Complex Sentence**
  - Atomic Sentence
  - Connectives
  - Negated Sentence
  - Parentheses
Syntax and Semantics of First-Order Logic

Sentence → AtomicSentence
  | Sentence Connective Sentence
  | Quantifier Variable,...Sentence
  | ¬Sentence
  | (Sentence)
AtomicSentence → Predicate(Term,...)
  | Term = Term
Term → Function(Term,...)
  | Constant
  | Variable
Connective → ⇒ | ∧ | ∨ | ⇔
Quantifier → ∀ | ∃
Variable → a | b | c | ...
Function → Mother | LeftLegOf | ...
Predicate → Before | HasColor | Raining | ...
Constant → A | X₁ | John | ...

• Quantifiers (∃, ∀)
  - The real power of first-order logic
  - Express properties of entire collections of objects rather than having to enumerate all the objects by name
  - Universal Quantifier (∀)
    • “all cats are mammals”
      ∀x Cat(x)⇒Mammal(x)
  - Existential Quantifier (∃)
    • “there exists a fish that can fly”
      ∃x Fish(x)∧CanFly(x)
Universal Quantification (∀)

- Makes a statement about all objects in the universe
  - ∀x Cat(x)⇒Mammal(x) expands using conjunction:
    Cat(Felix)⇒Mammal(Felix) ∧ Cat(Fluffy)⇒Mammal(Fluffy) ∧ Cat(Spot)⇒Mammal(Spot) ∧ Cat(Sylvester)⇒Mammal(Sylvester) ∧ ...
  - What if the universe includes non-cats?
    Cat(Scaz)⇒Mammal(Scaz) ∧ Cat(Tree)⇒Mammal(Tree) ∧ ...
    • Still OK… because if Cat(Scaz) is false, then Cat(Scaz)⇒Mammal(Scaz) is true
  - Can we express “all cats are mammals” as
    ∀x Cat(x) ∧ Mammal(x)
    • No… requires that all objects are both cats and mammals
Existential Quantification ($\exists$)

- Makes a statement about some object in the universe
  - “Spot has a sister that is a cat” is expressed as
    \[ \exists x \; \text{Sister}(x, \text{Spot}) \land \text{Cat}(x) \]
  - Expands using disjunction:
    \[ (\text{Sister}(\text{Fluffy}, \text{Spot}) \land \text{Cat}(\text{Fluffy})) \lor \]
    \[ (\text{Sister}(\text{Richard}, \text{Spot}) \land \text{Cat}(\text{Richard})) \lor \]
    \[ (\text{Sister}(\text{BigRock}, \text{Spot}) \land \text{Cat}(\text{BigRock})) \lor \ldots \]
  - What if multiple objects fulfill the requirements?
    - Still ok… True or True is still True
  - Can you express this with an implication?
    \[ \exists x \; \text{Sister}(x, \text{Spot}) \Rightarrow \text{Cat}(x) \]
    Results in nonsense… if any object is not Spot’s sister, then this relation is true
Nested Quantifiers

• “If x is the parent of y, then y is the child of x”
  \[ \forall x \forall y \text{ Parent}(x,y) \implies \text{Child}(y,x) \]

  Syntactic sugar:
  \[ \forall x,y \text{ Parent}(x,y) \implies \text{Child}(y,x) \]

• “Everybody loves somebody”
  \[ \forall x \exists y \text{ Loves}(x,y) \]

• Is this the same as \[ \exists y \forall x \text{ Loves}(x,y) \] ?
  – No… this sentence states that “there exists someone who is loved by everyone”
Connections between $\forall$ and $\exists$

• “Everyone dislikes parsnips” is equivalent to “there does not exist someone who likes parsnips”:
  \[
  \forall x \neg \text{Likes}(x, \text{Parsnips}) \quad \text{is equivalent to} \quad \neg \exists x \text{ Likes}(x, \text{Parsnips})
  \]

• Similarly, “Everyone likes Scheme” is equivalent to “there is no one who does not like Scheme”
  \[
  \forall x \text{ Likes}(x, \text{Scheme}) \quad \text{is equivalent to} \quad \neg \exists x \neg \text{ Likes}(x, \text{Scheme})
  \]
Extensions and Variations of First-Order Logic
Higher-Order Logics

• “First-Order” Logic implies that you can quantify over \textit{objects}, but not over \textit{relations}

• Higher order logics allow quantification over relations and functions
  – Define equality as objects that have the same properties
    \[
    \forall x,y \ (x=y) \iff (\forall p \ p(x) \iff p(y))
    \]
  – or the equality of functions that give the same value for all arguments
    \[
    \forall f,g \ (f=g) \iff (\forall x \ f(x)=g(x))
    \]
Functional and Predicate Expressions using $\lambda$

- Allows for the construction of complex predicates
- Examples
  - A predicate for “difference of squares”
    \[ \lambda x,y \ x^2-y^2 \]
    \[ (\lambda x,y \ x^2-y^2)(2, 1) = 3 \]
  - A predicate for “are of differing gender and of the same age”
    \[ \lambda x,y \ \text{Gender}(x) \neq \text{Gender}(y) \land \text{Age}(x) = \text{Age}(y) \]
- Should look familiar from Scheme/Lisp
Other Notations

• Notational variations exist (especially within other fields that use logic)

• Some other operators are also useful
  – Uniqueness quantifier
    • “Every student has exactly one advisor”
      $\forall x \ Student(x) \Rightarrow \exists! y \ Advisor(y, x)$
  – Uniqueness operator
    • “The y that is the advisor to Jessica is on sabbatical”
      $\text{Sabbatical}(\iota y \ Advisor(y, \text{Jessica}))$

Greek letter iota
Advantages of Using First-Order Logic: Wumpus Example

- Consider an infinite or unknown board configuration
- How many propositional rules are required?
- First-order logic can handle this
Logical Agents for the Wumpus World

- We will consider three types of agent:
  - Reflex Agent
  - Model-Based Agent
  - Goal-Based Agent
Logical Agents for the Wumpus World

- We will consider three types of agent:
  - Reflex Agent
  - Model-Based Agent
  - Goal-Based Agent
Defining the Interface

- Percept as a statement:
  - Percept([Stench, Breeze, Glitter, Bump, Scream], time)

- Agent’s actions
  - Turn(left)
  - Turn(right)
  - Forward
  - Shoot
  - Grab
  - Drop

- Determine the best action for a particular time
  \( \exists a \text{ Action}(a,t) \)
A Simple Reflex Wumpus-Hunter

• Determine a set of action rules
  – Anytime you see gold, grab it
    \[ \forall s,b,u,c,t \text{ Percept}([s,b,\text{Glitter},u,c],t) \Rightarrow \text{Action}(\text{Grab},t) \]

• Make some simplifications for perception
  – Declare AtGold(t) anytime you detect glitter
    \[ \forall s,b,u,c,t \text{ Percept}([s,b,\text{Glitter},u,c],t) \Rightarrow \text{AtGold}(t) \]
  – Simplified action rule
    \[ \forall t \text{ AtGold}(t) \Rightarrow \text{Action}(\text{Grab},t) \]

• How many rules would you need to do this in propositional logic?
Limitations of the Simple Reflex Wumpus-Hunter

- Unable to maintain state
  - How do you know when you’ve grabbed the gold, or that the wumpus is already dead?
- Unable to avoid infinite loops
  - If you have the gold and are tracing back through your steps, the states look the same and thus the actions must be the same
Logical Agents for the Wumpus World

• We will consider three types of agent:
  – Reflex Agent
  – Model-Based Agent
  – Goal-Based Agent
Representing Change in the World

• Maintaining an internal model of the world
• Many ways to accomplish this
  – Continually change the knowledge base (erase some sentences and add others)
    • Erase Location(Agent)=Square(1,1)
    • Add Location(Agent)=Square(1,2)
  – Maintain past knowledge as part of the world state (and perhaps future possible actions)
• Representing situations and actions is no different than representing objects and relations
Situation Calculus

- Simplest (and oldest) solution to internal modeling
- World consists of a sequence of *situations* or snapshots
- New situations are generated by taking an action
Situation Calculus

- Situations are indexed
  \[ \text{At}(\text{Agent}, [1, 1], S_0) \land \text{At}(\text{Agent}, [1, 2], S_1) \]

- Changes from one situation to the next
  \[ \text{Result}(\text{Forward}, S_0) \Rightarrow S_1 \]
  \[ \text{Result}(\text{Turn(R)}, S_1) \Rightarrow S_2 \]
Situation Calculus Axioms

• **Effect axioms** tell how the world changes between situations
  – After you drop an object, you are no longer holding it
    \[ \forall x, s \, \neg \text{Holding}(x, \text{Result}(\text{Drop}, s)) \]

• **Frame axioms** tell how the world stays the same between situations
  – If you are holding an object and you do not drop it, you are still holding it
    \[ \forall a, x, s \, \text{Holding}(x, s) \land (a \neq \text{Drop}) \Rightarrow \text{Holding}(x, \text{Result}(a, s)) \]
Situation Calculus Axioms

- **Successor State Axioms** combine a frame axiom with an effect axiom to tell how modifiable predicates change over time.

  true afterwards ↔ [an action made it true
  ∨ true already and no action made it false]

\[ \forall a,x,s \; \text{Holding}(x, \text{Result}(a,s)) \iff \\
[(a = \text{Grab} \land \text{Present}(x,s) \land \text{Portable}(x))
\lor (\text{Holding}(x,s) \land a \neq \text{Release})] \]
Two ways to represent world knowledge

- **Diagnostic rules** infer the presence of hidden properties directly from percepts
  \[
  \forall l,s \ \text{At}(\text{Agent},l,s) \land \text{Stench}(s) \Rightarrow \text{Smelly}(l)
  \]
- **Causal rules** reflect the assumed direction of causality in the world
  \[
  \forall l_1,l_2,s \ \text{At}(\text{Wumpus},l_1,s) \land \text{Adjacent}(l_1,l_2) \\ \Rightarrow \text{Smelly}(l_2)
  \]
- Systems that use causal rules are called model-based reasoning systems
  - These differences will come up again in a few weeks…
Finding the Wumpus

- A **diagnostic rule** can be used to determine the location of the wumpus

\[ \forall l_1, s \ Smelly(l_1) \Rightarrow [ \exists l_2 \ At(Wumpus,l_2,s) \land (l_1 = l_2 \lor \text{Adjacent}(l_1,l_2)) ] \]
Finding the Safe Squares

- A **diagnostic rule** can only draw a weak conclusion about safe squares
  \[ \forall x, y, g, u, c, s \ \text{Percept}([\text{None}, \text{None}, g, u, c], t) \land \text{At(Agent,} x, s) \land \text{Adjacent}(x, y) \Rightarrow \text{OK}(y) \]
- But sometimes a square can be safe when smells and breezes abound.
- A **causal rule** gives a better representation
  \[ \forall x, t \ (\neg \text{At(Wumpus,} x, t) \land \neg \text{Pit}(x)) \Leftrightarrow \text{OK}(x) \]
Logical Agents for the Wumpus World

- We will consider three types of agent:
  - Reflex Agent
  - Model-Based Agent
  - Goal-Based Agent
Toward a Goal-Based Agent

• How do you turn around once you have the gold?
  – Add a new state that represents the goal action
    \( \forall s \; \text{Holding(Gold, s)} \implies \text{Goal(GoHome, s)} \)

• How do you find the sequence of actions?
  – Search
  – Inference
  – Planning (coming up in a few weeks…)
Coming Up…

• How to use first-order logic to solve these problems
  – Forward chaining
  – Backward chaining
• Things that first-order logic can’t do