First-Order Logic

CPSC 470 – Artificial Intelligence

Brian Scassellati

Where we left off…

Propositional Logic Syntax

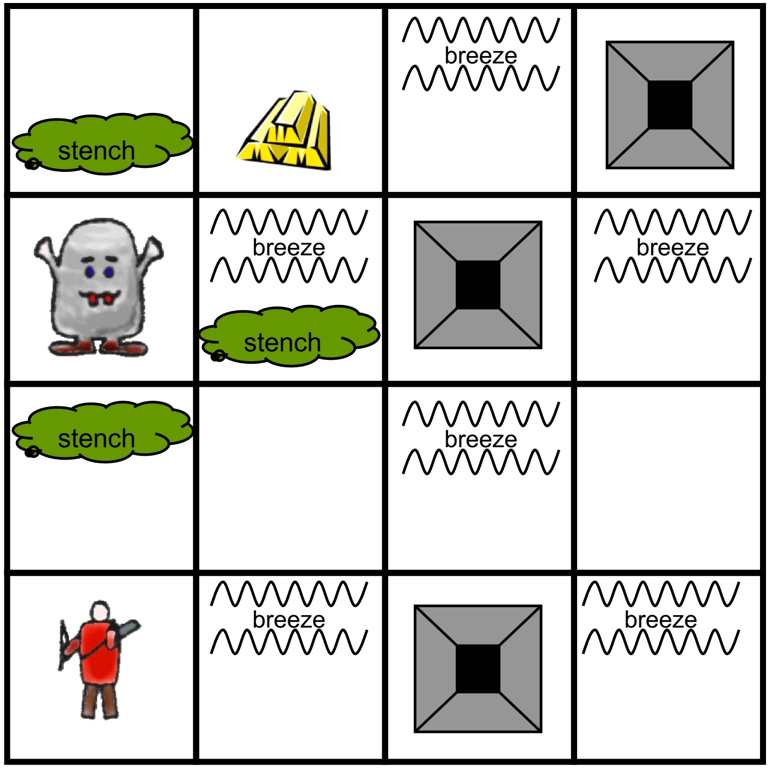
$$Sentence \rightarrow AtomicSentence \mid ComplexSentence$$

$$AtomicSentence \rightarrow True \mid False \mid P \mid Q \mid \ldots$$

$$\begin{aligned} ComplexSentence \rightarrow\ & (Sentence) \mid \\ & Sentence\ Connective\ Sentence \mid \\ & \neg Sentence \end{aligned}$$

$$Connective \rightarrow \wedge \mid \vee \mid \Rightarrow \mid \Leftrightarrow$$

Wumpus World



An Agent for the Wumpus World

- Convert perceptions into sentences:

  "In square [1,1], there is no breeze and no stench" … becomes…

  $\neg B_{11} \wedge \neg S_{11}$

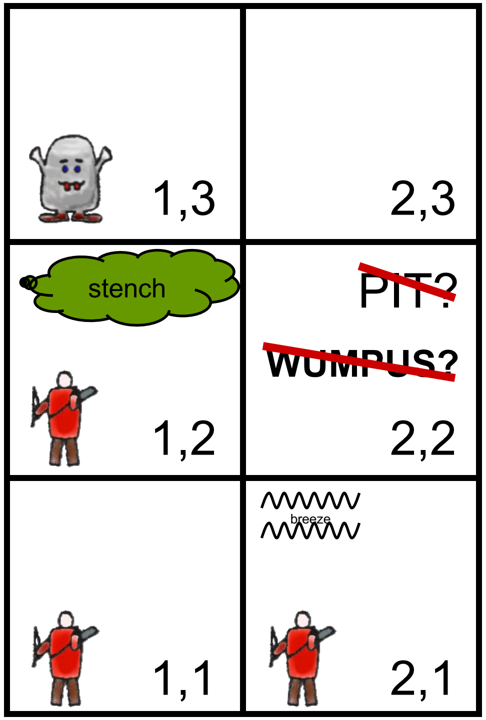
- Start with some knowledge of the world (in the form of rules)

  $R1 : \neg S_{11} \Rightarrow \neg W_{11} \wedge \neg W_{12} \wedge \neg W_{21}$

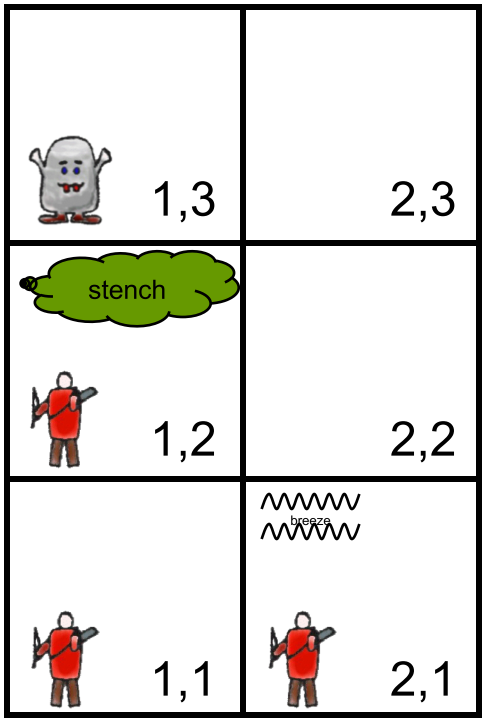
  $R2 : \neg S_{21} \Rightarrow \neg W_{11} \wedge \neg W_{21} \wedge \neg W_{22}$

  ….

  $R4 : S_{12} \Rightarrow W_{13} \vee W_{12} \vee W_{22} \vee W_{11}$



Finding the Wumpus



Percepts:

$\neg S_{11}$

$\neg S_{21}$

S_{12}

1. Apply modus ponens and and-elimination to $\neg S_{11} \Rightarrow \neg W_{11} \wedge \neg W_{12} \wedge \neg W_{21}$ to get

   $\neg \mathbf{W_{11}} \quad \neg \mathbf{W_{12}} \quad \neg \mathbf{W_{21}}$

2. Apply modus ponens and and-elimination to $\neg S_{21} \Rightarrow \neg W_{11} \wedge \neg W_{21} \wedge \neg W_{22}$ to get

   $\neg \mathbf{W_{22}} \quad \neg \mathbf{W_{21}} \quad \neg \mathbf{W_{31}}$

3. Apply modus ponens to

   $S_{12} \Rightarrow W_{13} \vee W_{12} \vee W_{22} \vee W_{11}$ to get

   $\mathbf{W_{13}} \vee \mathbf{W_{12}} \vee \mathbf{W_{22}} \vee \mathbf{W_{11}}$

4. Apply unit resolution to #3 and #1

   $\mathbf{W_{13}} \vee \mathbf{W_{22}}$

5. Apply unit resolution to #4 and #2

   $\mathbf{W_{13}}$

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

$$
\begin{aligned}
\textit{Sentence} &\rightarrow \textit{AtomicSentence} \\
&\quad \mid \textit{Sentence Connective Sentence} \\
&\quad \mid \textit{Quantifier Variable},\ldots \textit{Sentence} \\
&\quad \mid \neg \textit{Sentence} \\
&\quad \mid (\textit{Sentence}) \\
\textit{AtomicSentence} &\rightarrow \textit{Predicate}(\textit{Term},\ldots) \\
&\quad \mid \textit{Term} = \textit{Term} \\
\textit{Term} &\rightarrow \textit{Function}(\textit{Term},\ldots) \\
&\quad \mid \textit{Constant} \\
&\quad \mid \textit{Variable} \\
\textit{Connective} &\rightarrow \Rightarrow \mid \wedge \mid \vee \mid \Leftrightarrow \\
\textit{Quantifier} &\rightarrow \forall \mid \exists \\
\textit{Variable} &\rightarrow a \mid b \mid c \mid \ldots \\
\textit{Function} &\rightarrow \textit{Mother} \mid \textit{LeftLegOf} \mid \ldots \\
\textit{Predicate} &\rightarrow \textit{Before} \mid \textit{HasColor} \mid \textit{Raining} \mid \ldots \\
\textit{Constant} &\rightarrow A \mid X_1 \mid \textit{John} \mid \ldots
\end{aligned}
$$

- Constant Symbols *(A, B, John, …)*
  - 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,…)*
  - Defined by a set of tuples of objects that satisfy the predicate
- Function Symbols *(Cosine, FatherOf, …)*
  - 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

$$
\begin{aligned}
Sentence &\rightarrow AtomicSentence \\
&\quad \mid Sentence\ Connective\ Sentence \\
&\quad \mid Quantifier\ Variable, \ldots Sentence \\
&\quad \mid \neg Sentence \\
&\quad \mid (Sentence) \\
AtomicSentence &\rightarrow Predicate(Term, \ldots) \\
&\quad \mid Term = Term \\
Term &\rightarrow Function(Term, \ldots) \\
&\quad \mid Constant \\
&\quad \mid Variable \\
Connective &\rightarrow \Rightarrow \mid \wedge \mid \vee \mid \Leftrightarrow \\
Quantifier &\rightarrow \forall \mid \exists \\
Variable &\rightarrow a \mid b \mid c \mid \ldots \\
Function &\rightarrow Mother \mid LeftLegOf \mid \ldots \\
Predicate &\rightarrow Before \mid HasColor \mid Raining \mid \ldots \\
Constant &\rightarrow A \mid X_1 \mid John \mid \ldots
\end{aligned}
$$

- Variables *(a, b, c, …)*
  - Stand for an object (without naming it)
- Terms *(John, FatherOf(John), a,…)*
  - 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

$$
\begin{aligned}
&Sentence \rightarrow AtomicSentence\\
&\qquad\qquad \mid Sentence\ Connective\ Sentence\\
&\qquad\qquad \mid Quantifier\ Variable, \ldots Sentence\\
&\qquad\qquad \mid \neg Sentence\\
&\qquad\qquad \mid (Sentence)\\
&AtomicSentence \rightarrow \quad Predicate(Term, \ldots)\\
&\qquad\qquad\qquad\qquad \mid Term = Term\\
&Term \rightarrow Function(Term, \ldots)\\
&\qquad\qquad \mid Constant\\
&\qquad\qquad \mid Variable\\
&Connective \rightarrow \Rightarrow \mid \wedge \mid \vee \mid \Leftrightarrow\\
&Quantifier \rightarrow \forall \mid \exists\\
&Variable \rightarrow a \mid b \mid c \mid \ldots\\
&Function \rightarrow Mother \mid LeftLegOf \mid \ldots\\
&Predicate \rightarrow Before \mid HasColor \mid Raining \mid \ldots\\
&Constant \rightarrow A \mid X_1 \mid John \mid \ldots
\end{aligned}
$$

- 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

$$
\begin{aligned}
Sentence \rightarrow\ & AtomicSentence \\
& \mid Sentence\ Connective\ Sentence \\
& \mid Quantifier\ Variable, \ldots Sentence \\
& \mid \neg Sentence \\
& \mid (Sentence) \\
AtomicSentence \rightarrow\ & Predicate(Term, \ldots) \\
& \mid Term = Term \\
Term \rightarrow\ & Function(Term, \ldots) \\
& \mid Constant \\
& \mid Variable \\
Connective \rightarrow\ & \Rightarrow \mid \wedge \mid \vee \mid \Leftrightarrow \\
Quantifier \rightarrow\ & \forall \mid \exists \\
Variable \rightarrow\ & a \mid b \mid c \mid \ldots \\
Function \rightarrow\ & Mother \mid LeftLegOf \mid \ldots \\
Predicate \rightarrow\ & Before \mid HasColor \mid Raining \mid \ldots \\
Constant \rightarrow\ & A \mid X_1 \mid John \mid \ldots
\end{aligned}
$$

- 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

$$
\begin{aligned}
Sentence &\rightarrow AtomicSentence \\
&\quad \mid Sentence\ Connective\ Sentence \\
&\quad \mid Quantifier\ Variable, \ldots Sentence \\
&\quad \mid \neg Sentence \\
&\quad \mid (Sentence) \\
AtomicSentence &\rightarrow Predicate(Term, \ldots) \\
&\quad \mid Term = Term \\
Term &\rightarrow Function(Term, \ldots) \\
&\quad \mid Constant \\
&\quad \mid Variable \\
Connective &\rightarrow \Rightarrow \mid \wedge \mid \vee \mid \Leftrightarrow \\
Quantifier &\rightarrow \forall \mid \exists \\
Variable &\rightarrow a \mid b \mid c \mid \ldots \\
Function &\rightarrow Mother \mid LeftLegOf \mid \ldots \\
Predicate &\rightarrow Before \mid HasColor \mid Raining \mid \ldots \\
Constant &\rightarrow A \mid X_1 \mid John \mid \ldots
\end{aligned}
$$

- Connectives ($\Rightarrow \wedge \vee \Leftrightarrow$)
  - Work the same way as in predicate calculus
- Complex Sentence
  - Atomic Sentence
  - Connectives
  - Negated Sentence
  - Parentheses

Syntax and Semantics of First-Order Logic

$$
\begin{aligned}
Sentence &\rightarrow AtomicSentence \\
&\quad \mid Sentence\ Connective\ Sentence \\
&\quad \mid Quantifier\ Variable, \ldots Sentence \\
&\quad \mid \neg Sentence \\
&\quad \mid (Sentence) \\
AtomicSentence &\rightarrow Predicate(Term, \ldots) \\
&\quad \mid Term = Term \\
Term &\rightarrow Function(Term, \ldots) \\
&\quad \mid Constant \\
&\quad \mid Variable \\
Connective &\rightarrow \Rightarrow \mid \wedge \mid \vee \mid \Leftrightarrow \\
Quantifier &\rightarrow \forall \mid \exists \\
Variable &\rightarrow a \mid b \mid c \mid \ldots \\
Function &\rightarrow Mother \mid LeftLegOf \mid \ldots \\
Predicate &\rightarrow Before \mid HasColor \mid Raining \mid \ldots \\
Constant &\rightarrow A \mid X_1 \mid John \mid \ldots
\end{aligned}
$$

- Quantifiers ($\exists$, $\forall$)
  - 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 ($\forall$)
    - "all cats are mammals"
      $\forall x\ Cat(x) \Rightarrow Mammal(x)$
  - Existential Quantifier ($\exists$)
    - "there exists a fish that can fly"
      $\exists x\ \ Fish(x) \wedge CanFly(x)$

Universal Quantification ($\forall$)

- Makes a statement about all objects in the universe
  - $\forall x\ Cat(x) \Rightarrow Mammal(x)$ expands using conjunction:

    $Cat(Felix) \Rightarrow Mammal(Felix) \wedge Cat(Fluffy) \Rightarrow Mammal(Fluffy) \wedge$
    $Cat(Spot) \Rightarrow Mammal(Spot) \wedge$
    $Cat(Sylvester) \Rightarrow Mammal(Sylvester) \wedge \ldots$
  - What if the universe includes non-cats?

    $Cat(Scaz) \Rightarrow Mammal(Scaz) \wedge Cat(Tree) \Rightarrow Mammal(Tree) \wedge \ldots$
    - Still OK… because if $Cat(Scaz)$ is false, then $Cat(Scaz) \Rightarrow Mammal(Scaz)$ is true
  - Can we express "all cats are mammals" as

    $\forall x\ Cat(x) \wedge Mammal(x)$
    - 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
    $\exists x \;\; Sister(x,Spot) \wedge Cat(x)$
  - Expands using disjunction:
    $(Sister(Fluffy, Spot) \wedge Cat(Fluffy)) \vee$
    $(Sister(Richard, Spot) \wedge Cat(Richard)) \vee$
    $(Sister(BigRock, Spot) \wedge Cat(BigRock)) \vee \ldots$
  - What if multiple objects fulfill the requirements?
    - Still ok… True or True is still True
  - Can you express this with an implication?
    $\exists x \;\; Sister(x,Spot) \Rightarrow Cat(x)$
    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"

  $\forall x\ \forall y\ \ \text{Parent}(x,y) \Rightarrow \text{Child}(y,x)$

  Syntactic sugar:

  $\forall x,y\ \ \text{Parent}(x,y) \Rightarrow \text{Child}(y,x)$
- "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":

  $\forall x \neg Likes(x,Parsnips)$ is equivalent to

  $\neg \exists x \; Likes(x,Parsnips)$

- Similarly, "Everyone likes Scheme" is equivalent to "there is no one who does not like Scheme"

  $\forall x \; Likes(x,Scheme)$ is equivalent to

  $\neg \exists x \neg Likes(x,Scheme)$

Extensions and Variations of First-Order Logic

Higher-Order Logics

- "First-Order" Logic implies that you can quantify over ***objects***, but not over ***relations***
- Higher order logics allow quantification over relations and functions
  - Define equality as objects that have the same properties

    $\forall x,y\ (x=y) \Leftrightarrow (\forall p\ \ p(x) \Leftrightarrow p(y))$
  - or the equality of functions that give the same value for all arguments

    $\forall f,g\ (f=g) \Leftrightarrow (\forall x\ \ f(x)=g(x))$

Functional and Predicate Expressions using λ

- 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"

    $\lambda\, x,y\ \mathrm{Gender}(x) \neq \mathrm{Gender}(y) \wedge \mathrm{Age}(x) = \mathrm{Age}(y)$
- Should look familiar from Scheme/Lisp

Other Notations

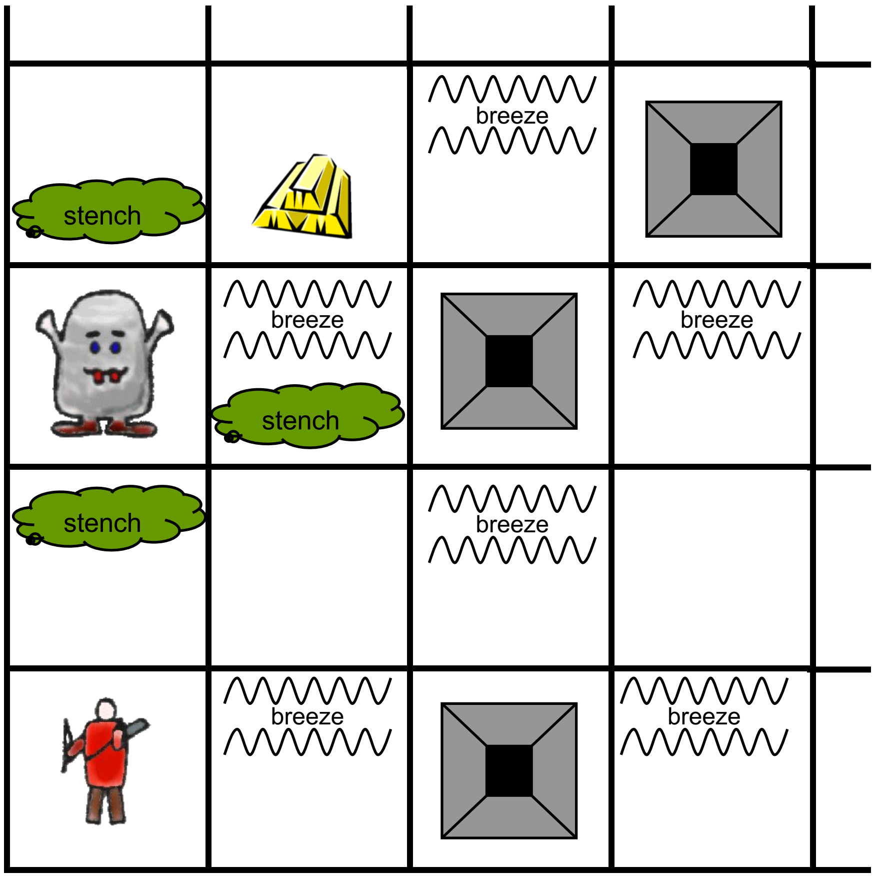
- Notational variations exist (especially within other fields that use logic)
- Some other operators are also useful
  - Uniqueness quantifier
    - “Every student has exactly one advisor”

      $\forall x\ \ \text{Student}(x) \Rightarrow \exists!\ y\ \text{Advisor}(y, x)$
  - Uniqueness operator
    - “The y that is the advisor to Jessica is on sabbatical”

      $\text{Sabbatical}(\iota\ y\ \text{Advisor}(y, \text{Jessica}))$

Greek letter iota

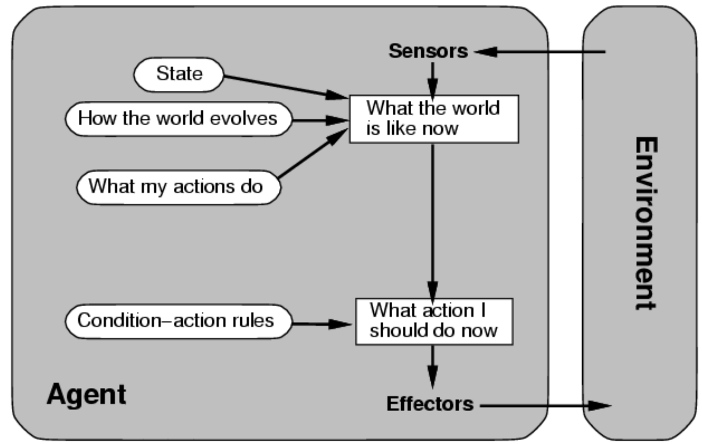
Advantages of Using First-Order Logic: Wumpus Example



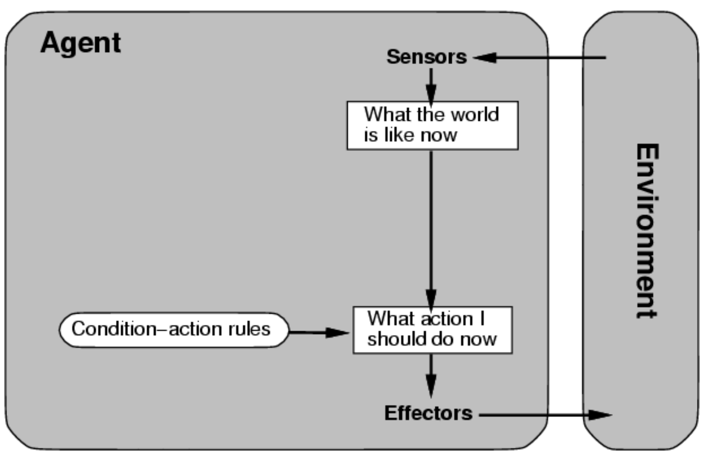
- Consider an infinite or unknown board configuration
- How many propositional rules are required?
- First-order logic can handle this

Logical Agents for the Wumpus World

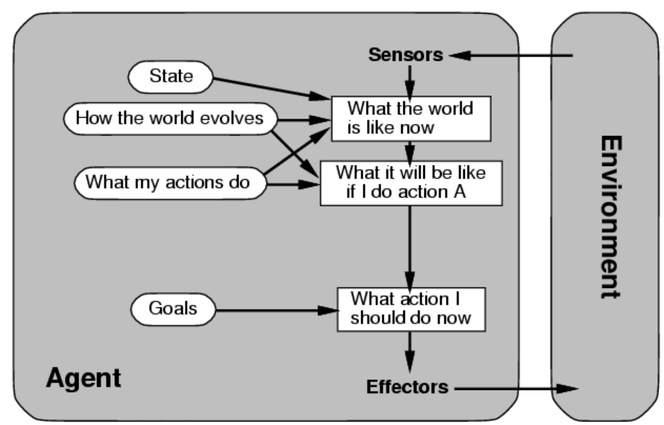
- We will consider three types of agent:
  - Reflex Agent
  - Model-Based Agent
  - Goal-Based Agent



Model-Based Agent



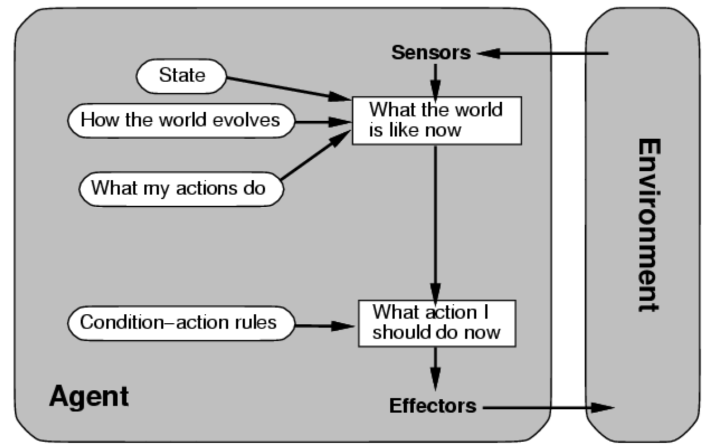
Reflex Agent



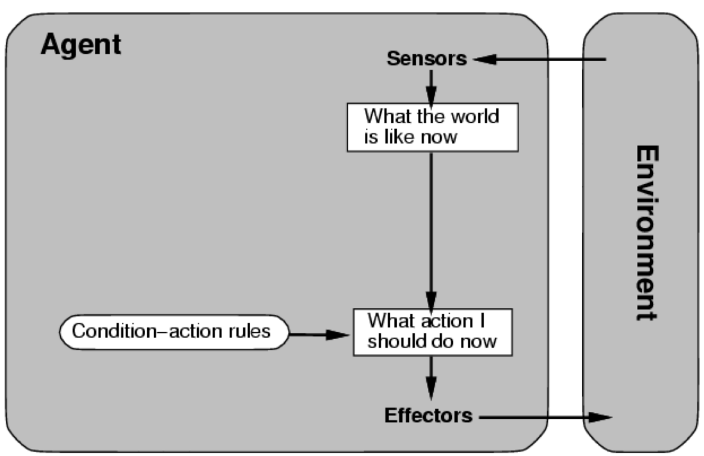
Goal-Based Agent

Logical Agents for the Wumpus World

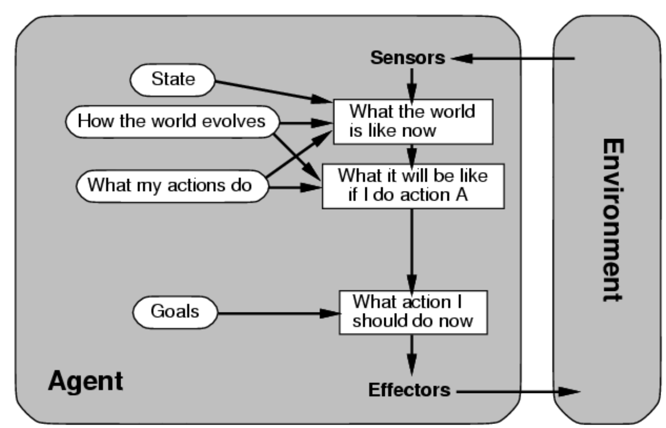
- We will consider three types of agent:
  - Reflex Agent
  - Model-Based Agent
  - Goal-Based Agent



Model-Based Agent

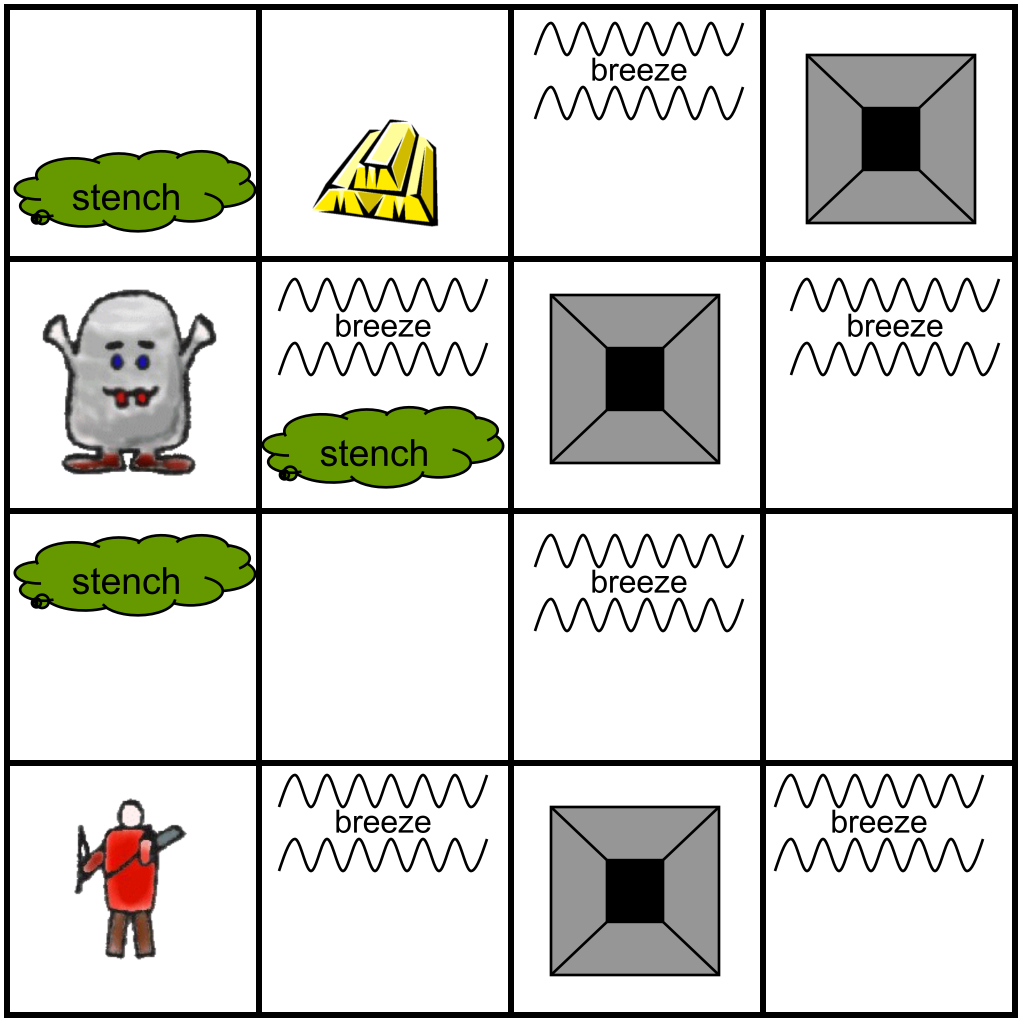


Reflex Agent



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

  $\exists a$ 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 Percept([s,b,Glitter,u,c],t) $\Rightarrow$ Action(Grab,t)
- Make some simplifications for perception
  - Declare AtGold(t) anytime you detect glitter

    $\forall$s,b,u,c,t Percept([s,b,Glitter,u,c],t) $\Rightarrow$ AtGold(t)
  - Simplified action rule

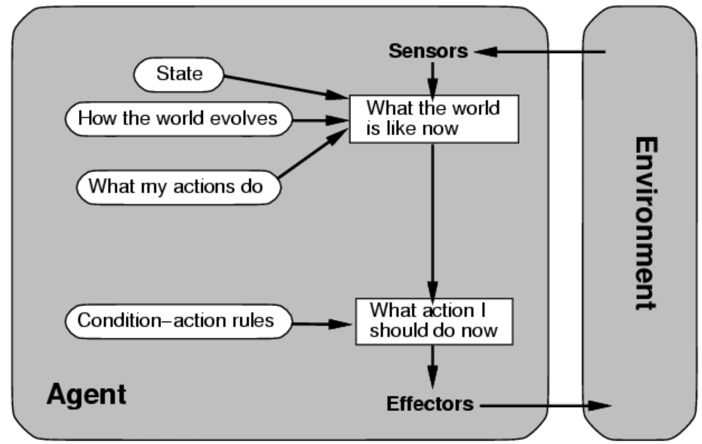
    $\forall$t AtGold(t) $\Rightarrow$ Action(Grab,t)
- How many rules would you need to do this in propositional logic?

Limitations of the Simple Reflex Wumpus-Hunter

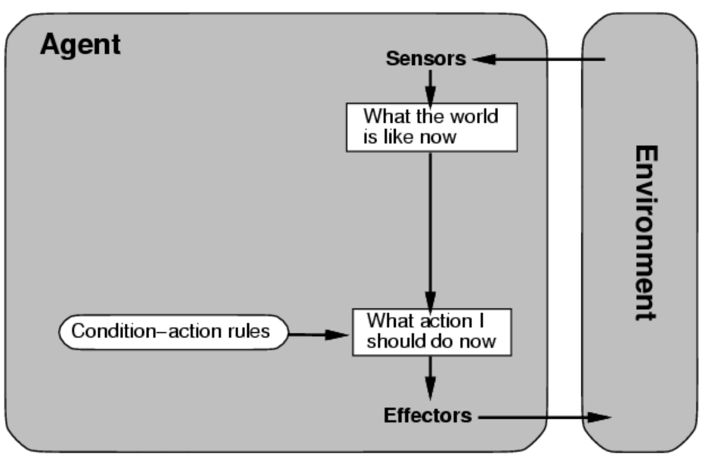
- Unable to maintain state
  - How do you know when you’ve grabbed the gold, or that the wumpus is already dead?
- Unable to avoid infinite loops
  - If you have the gold and are tracing back through your steps, the states look the same and thus the actions must be the same

Logical Agents for the Wumpus World

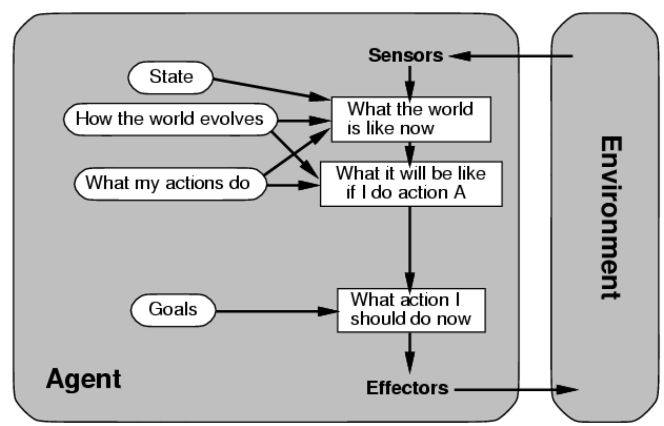
- We will consider three types of agent:
  – Reflex Agent
  – Model-Based Agent
  – Goal-Based Agent



Model-Based Agent



Reflex Agent

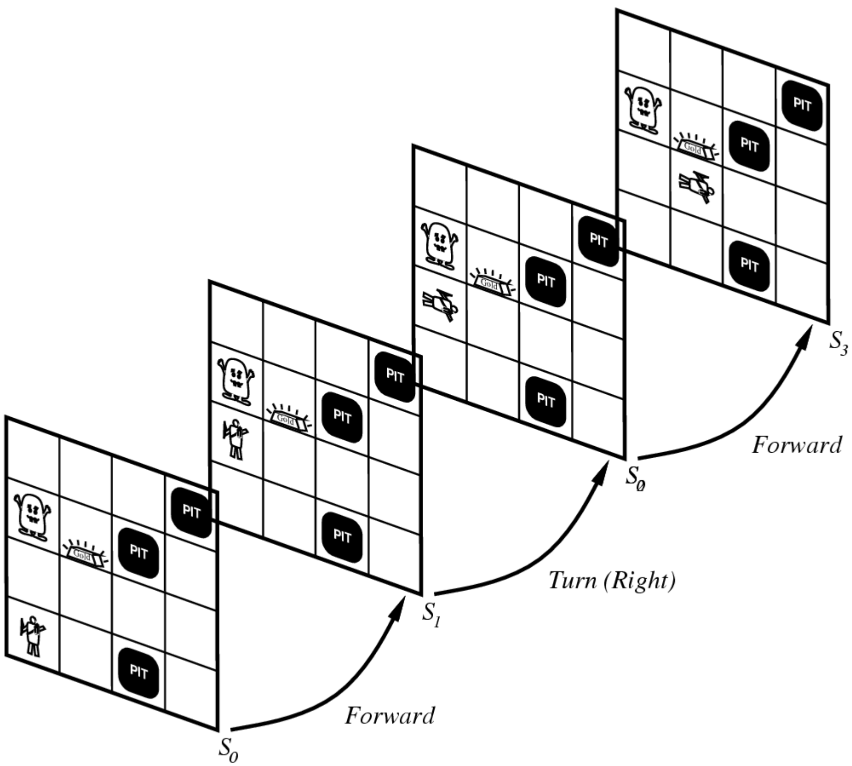


Goal-Based Agent

Representing Change in the World

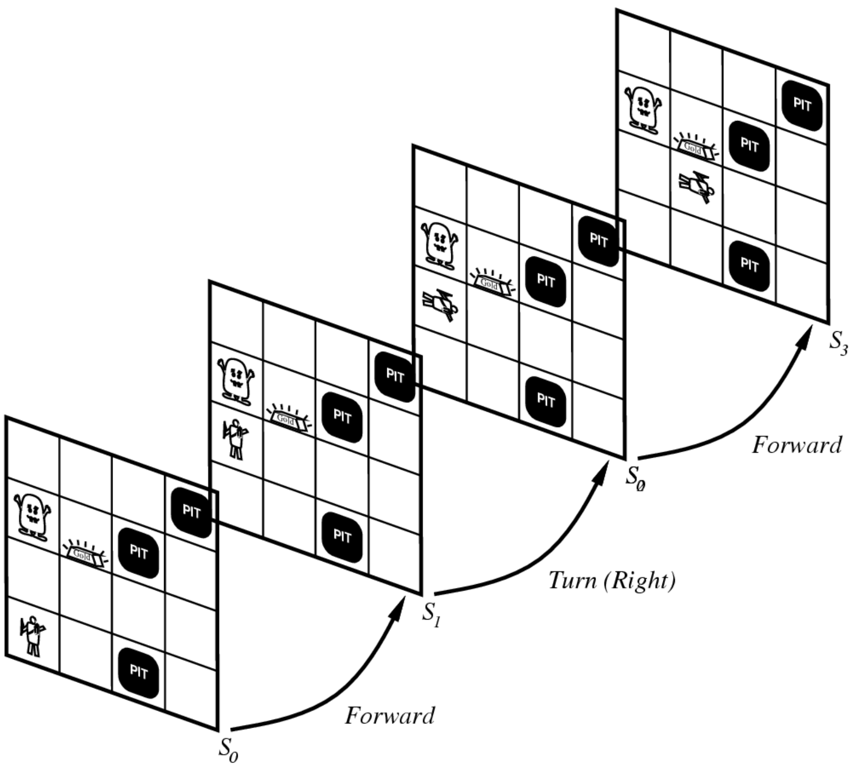
- Maintaining an internal model of the world
- Many ways to accomplish this
  - Continually change the knowledge base (erase some sentences and add others)
    - Erase Location(Agent)=Square(1,1)
    - Add Location(Agent)=Square(1,2)
  - Maintain past knowledge as part of the world state (and perhaps 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

  $At(Agent,[1,1], S_0) \wedge$
  $At(Agent,[1\ 2], S_1)$

- Changes from one situation to the next

  $Result(Forward, S_0) \Rightarrow S_1$

  $Result(Turn(R), S_1) \Rightarrow S_2$

Situation Calculus Axioms

- Effect axioms tell how the world changes between situations
  - After you drop an object, you are no longer holding it

    $\forall x,s \ \neg Holding(x, Result(Drop,s))$
- 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

    $\forall a,x,s \ Holding(x,s) \wedge (a \neq Drop) \Rightarrow$
    $Holding(x, Result(a,s))$

Situation Calculus Axioms

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

true afterwards $\Leftrightarrow$ [an action made it true
$\vee$ true already and no action made it false]

$\forall a,x,s\ \ Holding(x,Result(a,s)) \Leftrightarrow$
$[(a = Grab \wedge Present(x,s) \wedge Portable(x))$
$\vee\ (Holding(x,s) \wedge a \neq Release)]$

Two ways to represent world knowledge

- Diagnostic rules infer the presence of hidden properties directly from percepts

  $\forall l,s \;\; At(Agent,l,s) \wedge Stench(s) \Rightarrow Smelly(l)$

- Causal rules reflect the assumed direction of causality in the world

  $\forall l_1,l_2,s \;\; At(Wumpus,l_1,s) \wedge Adjacent(l_1,l_2)$
  $\Rightarrow Smelly(l_2)$

- 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

$$\forall l_1,s\ Smelly(l_1) \Rightarrow [$$
$$\exists l_2\ \ At(Wumpus,l_2,s) \wedge$$
$$(l_1{=}l_2 \vee Adjacent(l_1,l_2))\ \ ]$$

Finding the Safe Squares

- A diagnostic rule can only draw a weak conclusion about safe squares

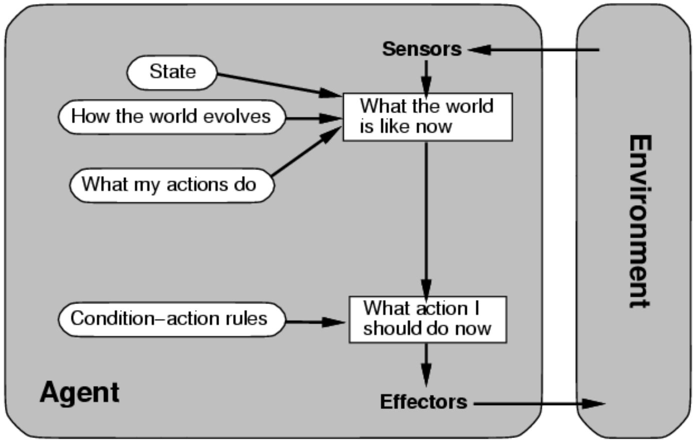
  $\forall x,y,g,u,c,s \;\; Percept([None,None,g,u,c],t) \wedge At(Agent,x,s) \wedge Adjacent(x,y) \Rightarrow OK(y)$

- But sometimes a square can be safe when smells and breezes abound.
- A causal rule gives a better representation

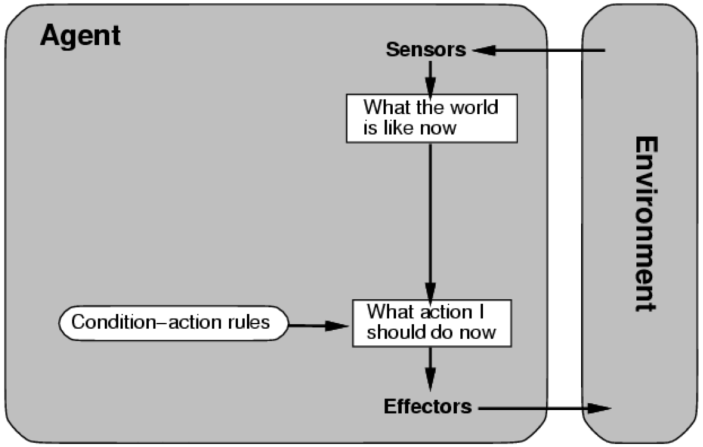
  $\forall x,t \;\; (\neg At(Wumpus,x,t) \wedge \neg Pit(x)) \Leftrightarrow OK(x)$

Logical Agents for the Wumpus World

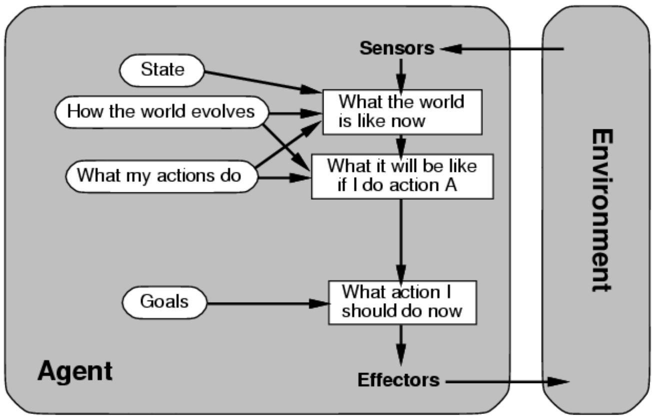
- We will consider three types of agent:
  - Reflex Agent
  - Model-Based Agent
  - Goal-Based Agent



Model-Based Agent



Reflex 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

    $\forall s \;\; Holding(Gold, s) \Rightarrow Goal(GoHome, s)$
- 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