

# **22c:145 Artificial Intelligence**

## **First-Order Logic**

**Readings: Chapter 8 of Russell &  
Norvig.**

# Language of FOL: Grammar

Sentence	::=	AtomicS   ComplexS
AtomicS	::=	<b>True</b>   <b>False</b>   RelationSymb(Term, ...)   Term = Term
ComplexS	::=	(Sentence)   Sentence Connective Sentence   $\neg$ Sentence   Quantifier Sentence
Term	::=	FunctionSymb(Term, ...)   ConstantSymb   Variable
Connective	::=	$\wedge$   $\vee$   $\Rightarrow$   $\Leftrightarrow$
Quantifier	::=	$\forall$ Variable   $\exists$ Variable
Variable	::=	$a$   $b$   ...   $x$   $y$   ...
ConstantSymb	::=	$A$   $B$   ...   <i>John</i>   0   1   ...   $\pi$   ...
FunctionSymb	::=	$F$   $G$   ...   <i>Cosine</i>   <i>Height</i>   <i>FatherOf</i>   +   ...
RelationSymb	::=	$P$   $Q$   ...   <i>Red</i>   <i>Brother</i>   <i>Apple</i>   >   ...

# Universal quantification

$\forall \langle \text{variables} \rangle \langle \text{sentence} \rangle$

Everyone at Ulowa is smart:

$\forall x \text{ At}(x, UIowa) \implies \text{Smart}(x)$

$\forall x P(x)$  is true in an interpretation  $m$  iff  $P(x)$  is true with  $x$  being each possible object in the domain

Roughly speaking, equivalent to the conjunction of instantiations of  $P$

$\text{At}(\text{KingJohn}, UIowa) \implies \text{Smart}(\text{KingJohn})$

$\wedge \text{At}(\text{Richard}, UIowa) \implies \text{Smart}(\text{Richard})$

$\wedge \text{At}(UIowa, UIowa) \implies \text{Smart}(UIowa)$

$\wedge \dots$

# A common mistake to avoid

Typically,  $\implies$  is the main connective with  $\forall$

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

$$\forall x \text{ At}(x, UIowa) \wedge \text{Smart}(x)$$

means “Everyone is at Ulowa and everyone is smart”

# Existential quantification

$\exists \langle \text{variable} \rangle \langle \text{sentence} \rangle$

Someone at Stanford is smart:

$\exists x \text{ At}(x, \text{Stanford}) \wedge \text{Smart}(x)$

$\exists x P$  is true in an interpretation  $I$  iff  $P$  is true with  $x$  being *some* possible object in the domain

Roughly speaking, equivalent to the disjunction of instantiations of  $P$

- $\text{At}(\text{KingJohn}, \text{Stanford}) \wedge \text{Smart}(\text{KingJohn})$
- $\vee \text{ At}(\text{Richard}, \text{Stanford}) \wedge \text{Smart}(\text{Richard})$
- $\vee \text{ At}(\text{Stanford}, \text{Stanford}) \wedge \text{Smart}(\text{Stanford})$
- $\vee \dots$

# Another common mistake to avoid

Typically,  $\wedge$  is the main connective with  $\exists$

Common mistake: using  $\implies$  as the main connective with  $\exists$ :

$$\exists x \text{ At}(x, \text{Stanford}) \implies \text{Smart}(x)$$

is true if there is anyone who is not at Stanford!

# Properties of quantifiers

$\forall x \forall y$  is the same as  $\forall y \forall x$  (why??)

$\exists x \exists y$  is the same as  $\exists y \exists x$  (why??)

$\exists x \forall y$  is not the same as  $\forall y \exists x$

$\exists x \forall y \text{ Loves}(x, y)$

“There is a person who loves everyone in the world”

$\forall y \exists x \text{ Loves}(x, y)$

“Everyone in the world is loved by at least one person”

Quantifier duality: each can be expressed using the other

$\forall x \text{ Likes}(x, \text{IceCream}) \quad \neg \exists x \neg \text{Likes}(x, \text{IceCream})$

$\exists x \text{ Likes}(x, \text{Broccoli}) \quad \neg \forall x \neg \text{Likes}(x, \text{Broccoli})$

# Fun with sentences

Brothers are siblings

$\forall x, y \text{ Brother}(x, y) \implies \text{Sibling}(x, y).$

“Sibling” is symmetric

# Fun with sentences

Brothers are siblings

$$\forall x, y \text{ Brother}(x, y) \implies \text{Sibling}(x, y).$$

“Sibling” is symmetric

$$\forall x, y \text{ Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x).$$

One's mother is one's female parent

# Fun with sentences

Brothers are siblings

$$\forall x, y \text{ Brother}(x, y) \implies \text{Sibling}(x, y).$$

“Sibling” is symmetric

$$\forall x, y \text{ Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x).$$

One’s mother is one’s female parent

$$\forall x, y \text{ Mother}(x, y) \Leftrightarrow (\text{Female}(x) \wedge \text{Parent}(x, y)).$$

A first cousin is a child of a parent’s sibling

# Fun with sentences

**Brothers are siblings**

$$\forall x, y \text{ Brother}(x, y) \implies \text{Sibling}(x, y).$$

**“Sibling” is symmetric**

$$\forall x, y \text{ Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x).$$

**One’s mother is one’s female parent**

$$\forall x, y \text{ Mother}(x, y) \Leftrightarrow (\text{Female}(x) \wedge \text{Parent}(x, y)).$$

**A first cousin is a child of a parent’s sibling**

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

# 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

E.g.,  $1 = 2$  and  $\forall x \times(Sqrt(x), Sqrt(x)) = x$  are satisfiable  
 $2 = 2$  is valid

E.g., definition of (full) *Sibling* in terms of *Parent*:

$\forall x, y \text{ Sibling}(x, y) \Leftrightarrow [\neg(x = y) \wedge \exists m, f \neg(m = f) \wedge$

$\text{Parent}(m, x) \wedge \text{Parent}(f, x) \wedge \text{Parent}(m, y) \wedge \text{Parent}(f, y)]$

# Semantics of First-Order Logic

(A little) more formally:

An **interpretation** is a pair  $(\mathcal{D}, \sigma)$  where

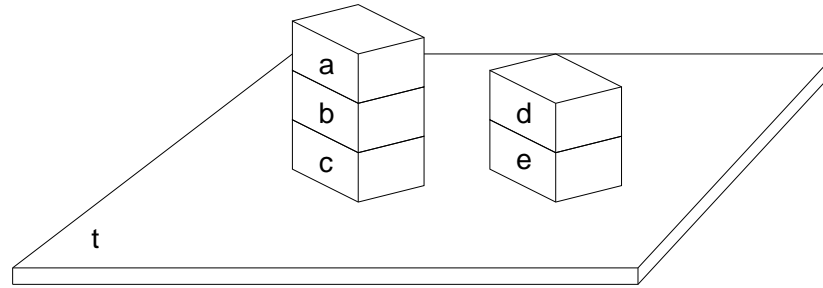
- $\mathcal{D}$  is a set of objects, the universe (or *domain*);
- $\sigma$  is mapping from variables to objects in  $\mathcal{D}$ ;
- $C^{\mathcal{D}}$  is an object in  $\mathcal{D}$  for every constant symbol  $C$ ;
- $F^{\mathcal{D}}$  is a function from  $\mathcal{D}^n$  to  $\mathcal{D}$  for every function symbol  $F$  of arity  $n$ ;
- $R^{\mathcal{D}}$  is a relation over  $\mathcal{D}^n$  for every relation symbol  $R$  of arity  $n$ ;

# An Interpretation $\mathcal{A}$ in the Blocks World

Constant Symbols:  $A, B, C, D, E, T$

Function Symbols:  $Support$

Relation Symbols:  $On, Above, Clear$



$$A^{\mathcal{A}} = a, B^{\mathcal{A}} = b, C^{\mathcal{A}} = c, D^{\mathcal{A}} = d, E^{\mathcal{A}} = e, F^{\mathcal{A}} = f, T^{\mathcal{A}} = t$$

$$Support^{\mathcal{A}} = \{\langle a, b \rangle, \langle b, c \rangle, \langle c, t \rangle, \langle d, e \rangle, \langle e, t \rangle, \langle t, t \rangle\}$$

$$On^{\mathcal{A}} = \{\langle a, b \rangle, \langle b, c \rangle, \langle d, e \rangle\}$$

$$Above^{\mathcal{A}} = \{\langle a, b \rangle, \langle a, c \rangle, \langle b, c \rangle, \langle d, e \rangle\}$$

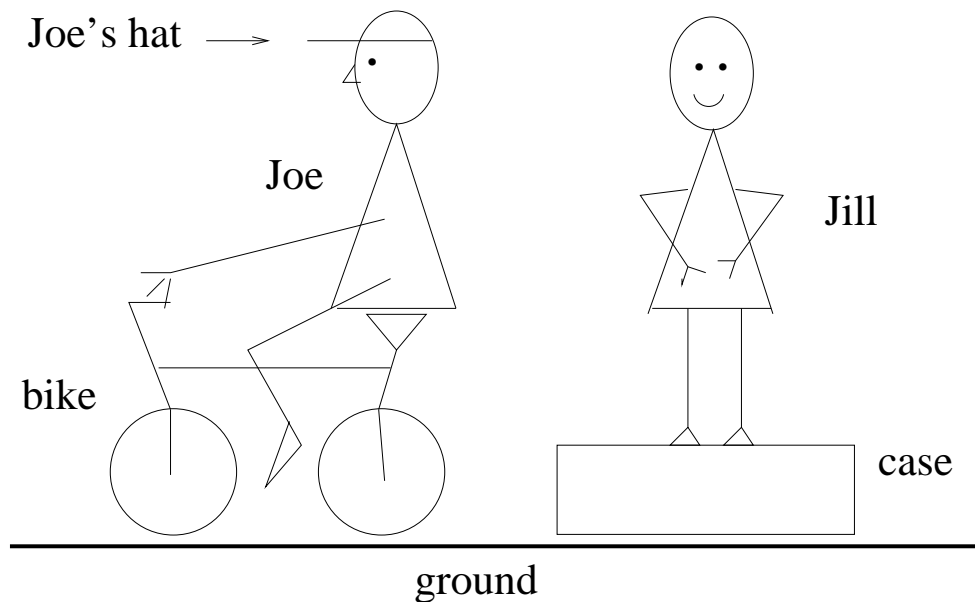
$$Clear^{\mathcal{A}} = \{\langle a \rangle, \langle d \rangle\}$$

# A different interpretation: $\mathcal{B}$

Constant Symbols:  $A, B, C, D, E, T$

Function Symbols:  $Support$

Relation Symbols:  $On, Above, Clear$



$A^{\mathcal{B}} = Joe'sHat, B^{\mathcal{B}} = Joe, C^{\mathcal{B}} = bike, D^{\mathcal{B}} = Jill, E^{\mathcal{B}} = case, T^{\mathcal{B}} = ground$

$Support^{\mathcal{B}} = \{\langle Joe's\ hat, Joe \rangle, \langle Joe, bike \rangle, \langle bike, ground \rangle, \langle Jill, case \rangle, \langle case, ground \rangle, \langle ground, ground \rangle\}$

$On^{\mathcal{B}} = \{\langle Joe's\ hat, Joe \rangle, \langle Joe, bike \rangle, \langle Jill, case \rangle, \}$

$Above^{\mathcal{B}} = \{\langle Joe's\ hat, Joe \rangle, \langle Joe's\ hat, bike \rangle, \dots\}$

$Clear^{\mathcal{B}} = \{\langle Jill \rangle\ (no\ hat), Joe's\ hat\}$

# Models for FOL: Lots!

We *can* enumerate the models for a given FOL sentence:

For each number of universe elements  $n$  from 1 to  $\infty$

For each  $k$ -ary predicate  $P_k$  in the sentence

For each possible  $k$ -ary relation on  $n$  objects

For each constant symbol  $C$  in the sentence

For each one of  $n$  objects mapped to  $C$

...

Enumerating models is not going to be easy!

# Semantics of First-Order Logic

- Let  $(\mathcal{D}, \sigma)$  be an interpretation and  $E$  an expression of FOL. We write  $\llbracket E \rrbracket_{\sigma}^{\mathcal{D}}$  to denote the *meaning of  $E$  in the domain  $\mathcal{D}$  under the variable assignment  $\sigma$* .
- The meaning  $\llbracket t \rrbracket_{\sigma}^{\mathcal{D}}$  of a term  $t$  is an object of  $\mathcal{D}$ . It is inductively defined as follows.

$\llbracket x \rrbracket_{\sigma}^{\mathcal{D}} \quad := \quad \sigma(x) \quad \text{for all variables } x$

$\llbracket C \rrbracket_{\sigma}^{\mathcal{D}} \quad := \quad C^{\mathcal{D}} \quad \text{for all constant symbols } C$

$\llbracket F(t_1, \dots, t_n) \rrbracket_{\sigma}^{\mathcal{D}} \quad := \quad F^{\mathcal{D}}(\llbracket t_1 \rrbracket_{\sigma}^{\mathcal{D}}, \dots, \llbracket t_n \rrbracket_{\sigma}^{\mathcal{D}}) \quad \text{for all function symbols } F$   
of arity  $n$

# Example

- Consider the symbols *MotherOf*, *SchoolOf*, *Bill* and the interpretation  $(\mathcal{D}, \sigma)$  where

*MotherOf* <sup>$\mathcal{D}$</sup>  is a unary fn mapping people to their mother

*SchoolOf* <sup>$\mathcal{D}$</sup>  is a unary fn mapping people to their school

*FchildOf* <sup>$\mathcal{D}$</sup>  is a binary fn mapping a couple to their first child

*Bill* <sup>$\mathcal{D}$</sup>  is Bill Clinton

$$\sigma := \{x \mapsto \text{Chelsea Clinton}, y \mapsto \text{Hillary Clinton}\}$$

- What is the meaning of  $\boxed{\text{MotherOf}(x)}$  according to  $(\mathcal{D}, \sigma)$ ?

$$\llbracket \text{MotherOf}(x) \rrbracket_{\sigma}^{\mathcal{D}} = \llbracket \text{MotherOf} \rrbracket_{\sigma}^{\mathcal{D}} (\llbracket x \rrbracket_{\sigma}^{\mathcal{D}}) = \text{MotherOf}^{\mathcal{D}} (\sigma(x)) = \text{Hillary Clinton}$$

- What is the meaning of  $\boxed{\text{SchoolOf}(\text{FchildOf}(y, \text{Bill}))}$ ?

$$\llbracket \text{SchoolOf}(\text{FchildOf}(y, \text{Bill})) \rrbracket_{\sigma}^{\mathcal{D}} = \text{SchoolOf}^{\mathcal{D}} (\text{FchildOf}^{\mathcal{D}} (\sigma(y), \text{Bill}^{\mathcal{D}})) = \text{S}$$

# Semantics of First-Order Logic

- The meaning  $\llbracket \varphi \rrbracket_{\sigma}^{\mathcal{D}}$  of a formula  $\varphi$  is either *True* or *False*.
- It is inductively defined as follows.

$$\begin{aligned} \llbracket t_1 = t_2 \rrbracket_{\sigma}^{\mathcal{D}} &:= \textit{True} && \text{iff } \llbracket t_1 \rrbracket_{\sigma}^{\mathcal{D}} \text{ is the same as } \llbracket t_2 \rrbracket_{\sigma}^{\mathcal{D}} \\ \llbracket R(t_1, \dots, t_n) \rrbracket_{\sigma}^{\mathcal{D}} &:= \textit{True} && \text{iff } \langle \llbracket t_1 \rrbracket_{\sigma}^{\mathcal{D}}, \dots, \llbracket t_n \rrbracket_{\sigma}^{\mathcal{D}} \rangle \in R^{\mathcal{D}} \\ \llbracket \neg \varphi \rrbracket_{\sigma}^{\mathcal{D}} &:= \textit{True/False} && \text{iff } \llbracket \varphi \rrbracket_{\sigma}^{\mathcal{D}} = \textit{False/True} \\ \llbracket \varphi_1 \vee \varphi_2 \rrbracket_{\sigma}^{\mathcal{D}} &:= \textit{True} && \text{iff } \llbracket \varphi_1 \rrbracket_{\sigma}^{\mathcal{D}} = \textit{True} \text{ or } \llbracket \varphi_2 \rrbracket_{\sigma}^{\mathcal{D}} = \textit{True} \\ \llbracket \exists x \varphi \rrbracket_{\sigma}^{\mathcal{D}} &:= \textit{True} && \text{iff } \llbracket \varphi \rrbracket_{\sigma'}^{\mathcal{D}} = \textit{True} \text{ for some } \sigma' \text{ coinciding} \\ &&& \text{with } \sigma \text{ except maybe for } x \end{aligned}$$

# Semantics of First-Order Logic

- The meaning of formulas built with the other logical symbols can be defined by reduction to the previous symbols.

$$\llbracket \varphi_1 \wedge \varphi_2 \rrbracket_{\sigma}^{\mathcal{D}} := \llbracket \neg(\neg\varphi_1 \vee \neg\varphi_2) \rrbracket_{\sigma}^{\mathcal{D}}$$

$$\llbracket \varphi_1 \Rightarrow \varphi_2 \rrbracket_{\sigma}^{\mathcal{D}} := \llbracket \neg\varphi_1 \vee \varphi_2 \rrbracket_{\sigma}^{\mathcal{D}}$$

$$\llbracket \varphi_1 \Leftrightarrow \varphi_2 \rrbracket_{\sigma}^{\mathcal{D}} := \llbracket (\varphi_1 \Rightarrow \varphi_2) \wedge (\varphi_2 \Rightarrow \varphi_1) \rrbracket_{\sigma}^{\mathcal{D}}$$

$$\llbracket \forall x \varphi \rrbracket_{\sigma}^{\mathcal{D}} := \llbracket \neg \exists x \neg \varphi \rrbracket_{\sigma}^{\mathcal{D}}$$

- If a sentence is closed (no free variables), its meaning *does not depend* on the the variable assignment (although it may depend on the domain):

$$\llbracket \forall x \exists y R(x, y) \rrbracket_{\sigma}^{\mathcal{D}} = \llbracket \forall x \exists y R(x, y) \rrbracket_{\sigma'}^{\mathcal{D}}, \quad \text{for any } \sigma, \sigma'$$

# Models, Validity, etc. for Sentences

- An interpretation  $(\mathcal{D}, \sigma)$  **satisfies** a sentence  $\varphi$ , or is a **model** for  $\varphi$ , if  $\llbracket \varphi \rrbracket_{\sigma}^{\mathcal{D}} = \text{True}$ .
- A sentence is **satisfiable** if it has at least one model.  
*Examples:*  $\forall x x \geq y, P(x)$
- A sentence is **unsatisfiable** if it has no models.  
*Examples:*  $P(x) \wedge \neg P(x), \neg(x = x)$
- A sentence  $\varphi$  is **valid** if every interpretation is a model for  $\varphi$ .  
*Examples:*  $P(x) \Rightarrow P(x), x = x$
- $\varphi$  is valid/unsatisfiable iff  $\neg\varphi$  is unsatisfiable/valid.
- Valid sentences do not tell us anything about the world.  
(They are always true!)

# Models, Validity, etc. for Sets of Sentences

- An interpretation  $(\mathcal{D}, \sigma)$  **satisfies** a set  $\Gamma$  of sentences, or is a **model** for  $\Gamma$ , if it is a model for *every* sentence in  $\Gamma$ .
- A set  $\Gamma$  of sentences is **satisfiable** if it has at least one model.

$$\text{Ex: } \{\forall x x \geq 0, \forall x x + 1 > x\}$$

- $\Gamma$  is **unsatisfiable**, or **inconsistent**, if it has no models.

$$\text{Ex: } \{P(x), \neg P(x)\}$$

- As in Propositional Logic,  $\Gamma$  **entails** a sentence  $\varphi$  ( $\Gamma \models \varphi$ ), if every model of  $\Gamma$  is also a model of  $\varphi$ .

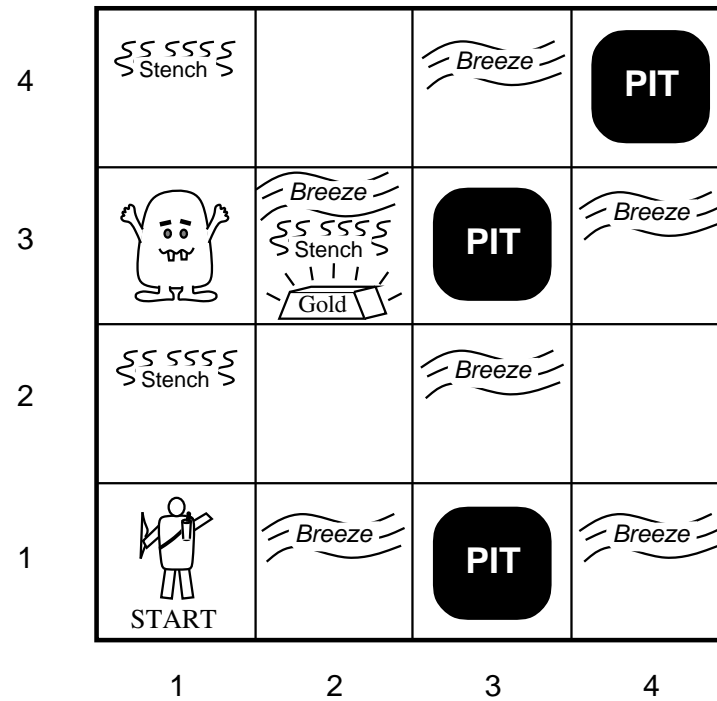
$$\text{Ex: } \{\forall x P(x) \Rightarrow Q(x), P(A_{10})\} \models Q(A_{10})$$

- **Note:** Again,  $\Gamma \models \varphi$  iff  $\Gamma \wedge \neg\varphi$  is unsatisfiable.

# Possible Interpretations Semantics

- Sentences can be seen as *constraints* on the set  $S$  of all possible interpretations.
- A sentence *denotes* all the possible interpretations that satisfy it (the models of  $\varphi$ ).  
If  $\varphi_1$  denotes a set of interpretations  $S_1$  and  $\varphi_2$  denotes a set  $S_2$ , then
  - $\varphi_1 \vee \varphi_2$  denotes  $S_1 \cup S_2$ ,
  - $\varphi_1 \wedge \varphi_2$  denotes  $S_1 \cap S_2$ ,
  - $\neg\varphi_1$  denotes  $S \setminus S_1$ ,
  - $\varphi_1 \models \varphi_2$  iff  $S_1 \subseteq S_2$ .
- A sentence denotes either no interpretations or an infinite number of them!

# The Wumpus World in FOL



# Interacting with FOL KBs

Suppose a wumpus-world agent is using an FOL KB and perceives a smell and a breeze (but no glitter) at  $t = 5$ :

$Tell(KB, Percept([Smell, Breeze, None], 5))$

$Ask(KB, \exists a \text{ Action}(a, 5))$

I.e., does the KB entail any particular actions at  $t = 5$ ?

Answer: *Yes*,  $\{a/Shoot\}$  ← substitution (binding list)

Given a sentence  $S$  and a substitution  $\sigma$ ,

$S\sigma$  denotes the result of plugging  $\sigma$  into  $S$ ; e.g.,

$S = Smarter(x, y)$

$\sigma = \{x/Hillary, y/Bill\}$

$S\sigma = Smarter(Hillary, Bill)$

$Ask(KB, S)$  returns some/all  $\sigma$  such that  $KB \models S\sigma$

# Knowledge base for the wumpus world

“Perception”

$\forall b, g, t \text{ Percept}([Smell, b, g], t) \implies Smelt(t)$

$\forall s, b, t \text{ Percept}([s, b, Glitter], t) \implies AtGold(t)$

**Reflex:**  $\forall t \text{ AtGold}(t) \implies \text{Action}(Grab, t)$

**Reflex with internal state: do we have the gold already?**

$\forall t \text{ AtGold}(t) \wedge \neg \text{Holding}(Gold, t) \implies \text{Action}(Grab, t)$

*Holding*(Gold, *t*) cannot be observed

$\implies$  keeping track of change is essential

# Deducing Hidden Properties

- Properties of locations:

- $\forall x, t \text{ At}(\text{Agent}, x, t) \wedge \text{Smelt}(t) \implies \text{Smelly}(x)$

- $\forall x, t \text{ At}(\text{Agent}, x, t) \wedge \text{Breeze}(t) \implies \text{Breezy}(x)$

- Squares are breezy near a pit:

- Diagnostic rule—infer cause from effect

- $\forall y \text{ Breezy}(y) \implies \exists x \text{ Pit}(x) \wedge \text{Adjacent}(x, y)$

- Causal rule—infer effect from cause

- $\forall x, y \text{ Pit}(x) \wedge \text{Adjacent}(x, y) \implies \text{Breezy}(y)$

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

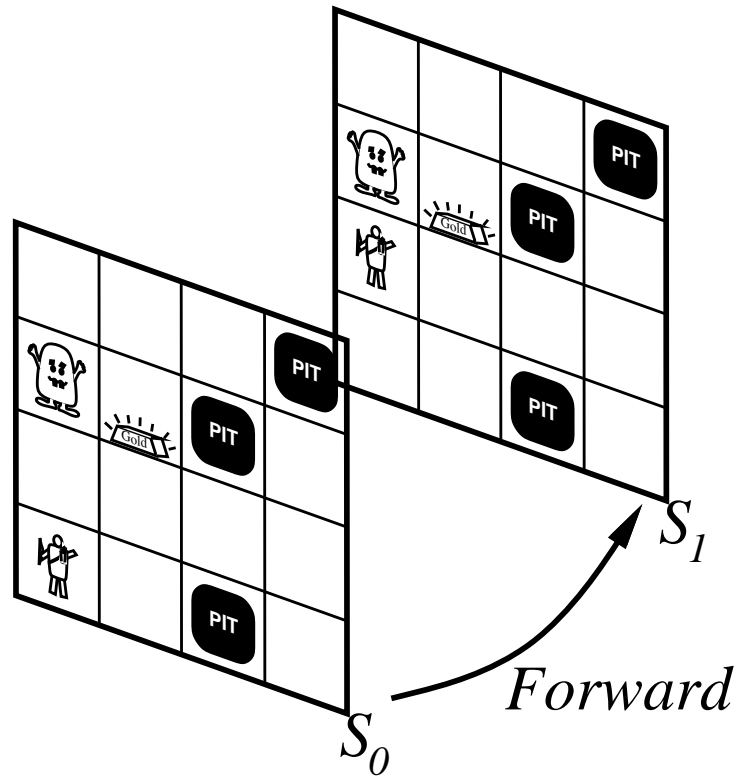
- Definition for the *Breezy* predicate:

- $\forall y \text{ Breezy}(y) \Leftrightarrow [\exists x \text{ Pit}(x) \wedge \text{Adjacent}(x, y)]$

# Keeping Track of Change

Facts hold in situations, rather than eternally

E.g.,  *Holding(Gold, Now)* rather than just  *Holding(Gold)*



# Situation Calculus

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 (or a *time stamp*).

Situations are connected by the *Result* function  
*Result(a, s)* is the situation that results from doing *a* in *s*

# Describing Actions

- “Effect” axiom—describe changes due to action  
 $\forall s \text{ AtGold}(s) \implies \text{Holding}(\text{Gold}, \text{Result}(\text{Grab}, s))$
- “Frame” axiom—describe non-changes due to action  
 $\forall s \text{ HaveArrow}(s) \implies \text{HaveArrow}(\text{Result}(\text{Grab}, s))$

# Frame, Qualification, and Ramification

- Frame problem: find an elegant way to handle non-change
  - representation—avoid frame axioms
  - inference—avoid 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):

$P$  true afterwards  $\Leftrightarrow$  [an action made  $P$  true  $\vee$   
 $P$  true already and no action made

- Example: For holding the gold:

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

# Making Plans

- Initial condition in KB:

$At(Agent, [1, 1], S_0)$

$At(Gold, [1, 2], S_0)$

- Query:  $Ask(KB, \exists s \text{ Holding}(Gold, s))$

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  $[a_1, a_2, \dots, a_n]$
- $PlanResult(p, s)$  is the result of executing  $p$  in  $s$
- Then the query  
 $Ask(KB, \exists p \text{ Holding}(Gold, PlanResult(p, S_0)))$   
has the solution  $\{p/[Forward, Grab]\}$
- Definition of  $PlanResult$  in terms of  $Result$ :  
 $\forall s \text{ PlanResult}([], s) = s$   
 $\forall a, p, s \text{ PlanResult}([a|p], s) = PlanResult(p, Result(a, s))$
- 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